Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 16 March 2026
Revised 25 April 2026
Accepted 20 May 2026

Natural Resources for Human Health 2026, Vol. 6 (1s): 673–697

KEYWORDS:
Wireless Bio Sensor Network, Fault Detection, Fault Classification, Ensemble-Based Diagnosis, Biosensor Reliability.

Designing Fault Tolerant Wireless Biosensor Network with Composite Techniques

Rajeev Agarwal[1], Tusharkanta Samal[1,2], Nayan Ranjan Samal[1], Satyabrat Sahoo[3], Ashok Kumar Bhoi[4]

[1]Department of Computer Science and Engineering, SOET, DRIEMS University, Cuttack, Odisha, India.
[2]Department of Computer Science and Engineering, Sambalpur University Institute of Information Technology, Burla, Odisha, India.
[3]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Vijayawada, Andhra Pradesh, India.
[4]Department of Computer Science and Engineering, Government College of Engineering, Bhawanipatna, Kalahandi, Odisha, India.

ABSTRACT: Reliability is an essential feature of Wireless Bio Sensor Networks (WBSNs) that can be deployed for continuous health monitoring purposes. The major concern of masking caused by the traditional sliding window method utilised for fault diagnosis has been solved with this new approach to fault diagnosis within WBSN networks. The approach presented in this paper provides three ways to address the critical masking associated with sliding window techniques. A Locked Historical Baseline Detection (LHBD) mechanism is employed to prevent the masking of persistent faults by locking calibration statistics during initialization. Secondly a CUSUM (Cumulative Sum) integration method is applied to detect fault drift. Then an ensemble voting technique is projected to improve accuracy in finding faults. An empirical evaluation of the proposed method versus Re-Implementing the sliding window in identical conditions (30 trials × 6 fault probabilities × 3 different types of faults) indicates that the proposed method achieved 98.3% accuracy compared to 88% for the Re-Implemented sliding windows (+11.6%, $p < 0.001$); Furthermore, persistent faults were no longer masked due to a reduction in false negative rates from 12.7% to 0.03% (approximately 97.6%), whereas persistent faults were masked by the traditional sliding window method after ~30 minutes. Paired t-tests are used to statistically validate each improvement at the 95% confidence level. In clinical WBSN situations, where missing a problem which is of Type II mistake, has more serious repercussions than false alarms, the methodologies especially well-suited.

1. INTRODUCTION

Rapid developments in wireless-assisted biomedical equipment for the Internet of Medical Things (IoMT) [1,2] have addressed a public health issue by improving healthcare quality at a lower cost. Healthcare applications and patient monitoring both within and outside of hospitals are made possible by Wireless Bio Sensor Networks (WBSNs) [3–5]. A WBSN is composed of several small biosensors that are placed beneath the skin to collect data on vital signs like blood pressure, heart rate, body temperature, and brain health. After that, the data is wirelessly transmitted to one or more collectors, sometimes referred to as sinks.

The patient can be continually observed while participating in routine (daily) activities anyplace when WBSNs are utilised [6]. By sharing biological data, healthcare professionals can treat patients more precisely and effectively and detect potential health problems earlier [7,8]. Consequently, the establishment of WBSNs has changed patient monitoring procedures and provided them with continuous monitoring [9,10]. However, this technology raises concerns about energy and dependability because biosensor devices have limited battery resources and power constraints.

Applications pertaining to healthcare and life-critical monitoring are the main emphasis of WBSN. However, the nature of wireless communication affects connectivity and data throughput. Therefore, reliable and error-free data transmission to the medical database/server is essential in WBSN. If erroneous and unreliable data is delivered to the server, doctors and other medical professionals will ultimately make a bad choice that could seriously endanger the patient's life [11]. In this case, WBSN applications depend heavily on reliability and fault tolerance.

Software and hardware flaws must be fixed because WBSN systems are made up of both hardware and software components. Biosensors have three types of hardware problems: irregularity readings, hardware faults, and battery faults. But software flaws, such as poor communication, malformed models, software bugs, insufficient validation, and human mistake, are also crucial. In order to guarantee reliable health evaluation and precise monitoring in WBSN situations, several challenges must be addressed.

This paper addresses that specific problem. We lock the reference baseline during initial calibration and never update it afterward. A sensor that drifts will always be compared against its original healthy state, not against its progressively corrupted self. In contrast to fixed baseline with single detector approach used by Agarwal et al. [19], we in this study merge locked baseline with ensemble to attain complementary fault coverage.

The detection framework utilizes three levels to identify faults: Hard fault detection for total failure conditions has occurred. In critical areas (such as ECG and EEG sensors), timeouts are used as a monitoring mechanism so that if there is no response from a sensor beyond the defined timeout period, that is identified as having a fault condition. In non-critical areas, for example, pedometers and skin temperature measurements, Fletcher-32 checksums will be used to differentiate link and node failures without the added overhead of constantly sending acknowledgment packets. Three algorithms that run simultaneously are used in this soft fault detection method: the Z-score detector that uses the locked historical baseline to detect (LHBD) relative magnitude deviation, the CUSUM detector for slow drift over a period of time to identify slowly changing calibration errors, and the gradient detector (also known as a transient spike detector) for detecting transient spikes (momentary errors). The final determination of a faulty reading is made through majority consensus of the three classifiers, defined as two out of three must agree. In addition to deciding whether a reading is faulty or not, XGBoost describes the detected fault each reading corresponds to. The classification of faults into one of three categories describes how long the fault lasted; either, permanent, intermittent, or transient. This classifier achieves an accuracy level of 96.5%.

The remaining of this paper is organized as follows. Section 2 covers prior work and where it falls short. Section 3 describes the methodology the network model, the detection algorithms, and the classification approach. Section 4 presents experimental results with statistical analysis. Section 5 discusses what the numbers mean for actual WBSN deployments. Section 6 wraps up with minutes and limitations of this research work, and future scope of research work.

2. LITERATURE STUDY

This section focuses on work done in the field of fault management in WBSN systems and the Internet of Things (IoT), which is still in its infancy and has a number of unresolved research issues.

Zhou et al. [12] discussed how to handle faults in an Internet of Things network when several types of sensors are employed. They created a sensing device adaption system for fault-tolerant IoT networks and have used regression analysis to mimic virtual services. Nevertheless, smart healthcare systems are not covered by the suggested method.

An IoT-based architecture that supports fault tolerance and scalability for healthcare systems are given by the authors in [13]. They set up a backup sink node to support the primary sink node in the event of a communication or node failure. The authors have asserted that a wide range of defect scenarios are covered by their fault tolerance strategy. Nevertheless, their method dealt with the issue of errors in a reactive manner. Furthermore, they have not provided a performance rate for defect detection and recovery.

Yang et al. [14] have proposed a Data Fault Detection mechanism in medical sensor networks (DFD-M). They have used a dynamic-local outlier factor (D-LOF) algorithm to identify outlying sensed data vectors. A linear regression model is used based on trapezoidal fuzzy numbers to predict which biological readings in the outlying data vector are suspected to be faulty. The authors assumed time and space correlation between readings; hence, the method does not work if the patient is connected to one sensor.

The authors in [15] has proposed a method to detect faulty sensors by building a logistical regression model on the sink nodes using sensor nodes data. The model is deployed on the sensors to predict incoming faulty readings. The authors have considered binary outcomes and employed a threshold to determine faulty versus non-faulty readings. However, to save battery life faulty, detected readings have not forwarded to the sink node. Additionally, the reported prediction rate is relatively low, which made it inapplicable for a WBSN environment.

A reliable and fault-tolerant design for WBSNs using Low-Power Wi-Fi (IEEE 802.11n) and Long-Term Evolution (LTE) has been given by ElSalamouny et al. in [16]. Additionally, a smart band aggregates the information gathered from each sensor node and transmits it to base station (BS).

Table 1. Research gaps

Ref.	Proposed Work	Key Contributions	Identified Limitations
[12]	Fault handling in IoT using regression-based virtual sensing.	Developed sensing device adaptation system for fault tolerance.	Does not address smart healthcare / WBSN-specific scenarios.
[13]	IoT-based fault-tolerant healthcare architecture.	Backup sink node for handling node/communication failure; scalable design.	Reactive approach only; no performance metrics for detection & recovery.
[14]	DFD-M (Data Fault Detection using D-LOF & fuzzy regression).	Detects outliers in medical sensor data; predicts faulty biological readings.	Assumes spatio-temporal correlation; fails for single-sensor scenarios.
[15]	Logistic regression-based faulty sensor detection.	Predicts faulty readings at sensor level.	Low prediction accuracy; ignores faulty data transmission; unsuitable for WBSN constraints.
[16]	Fault-tolerant WBSN using Wi-Fi & LTE.	Reliable communication via smart band aggregation.	Lacks intelligent fault detection & recovery mechanisms; focuses mainly on communication reliability.
[17]	Network coding-based proactive fault tolerance.	Improves packet delivery & reduces latency.	Complex implementation; lacks real-time adaptability for dynamic WBSN conditions.
[18]	TDMA-based energy-efficient WBSN.	Adaptive sleep scheduling; improved energy efficiency & reliability.	Does not ensure critical data delivery during faults; lacks fault-aware prioritization.
[19]	Composite fault detection and classification framework for WBSNs.	Integrated approach combining fault detection and classification in a unified model.	Limited validation in large-scale real-world WBSN deployments.

For WBSNs, a reliable and proactive fault-tolerant mechanism based on smart healthcare network coding is suggested in [17]. This method creates linear coding choices based on a spanning tree after dividing the network architecture into several logical units using a greedy grouping mechanism. Even in transmission errors, the gateway can recover the original packets when it gets a sufficient number of BER for specific linear coding combinations. This method reduces end-to-end latency and increases the packet delivery ratio. In [18], Salayma et al. have introduced Time Division Multiple Access (TDMA)-based method, the objective is to improve energy efficiency and reliability in WBSNs. Their method allows nodes to bypass channel deep dying by adaptively dispensing sleep times based on their channel status. In addition, time slots are dynamically assigned to each node according to the scheme's requirements. However, it does not guarantee critical data communication in the event of network faults.

Even though there has been much research into IoT and WBSN fault management, there are still significant limitations. A number of these methods today are reactive; they only come into play after an error has occurred, instead of predicting or preventing them before they occur. In cases where a single bio-sensor will be used to track a patient, some fault management methodologies will not be useful because they assume many bio-sensors or a considerable degree of temporal-spatial correlation. Past research has indicated that, on average, machine learning approach based fault detection algorithms have low quality, do not include a temporal analysis, or do not provide information about whether the defect was a transient, intermittent or persistent defect. The majority of the fault management literature has not provided a holistic framework across the entire system that includes the elements of fault tolerance classification and detection and is appropriate for the rigorous requirements of life-critical health care applications. Realistic fault situations, especially those involving sensor degradation, wireless interference, and battery-related problems, have also received little assessment. These shortcomings show that Wireless Biosensor Networks require a more resilient, proactive, and intelligent fault-tolerant infrastructure. The research gaps have been listed in Table 1.

3. PROPOSED METHODOLOGY

3.1. Fault Network Model

3.1.1. Regional Classification

(A) Sensitive Region (SR):

Sensors allocated to the sensitive region (SR) monitor physiological signals where data loss carries clinical risk signals. In this area, data dependability is crucial. In order to provide path redundancy, which enables data to be relayed through nearby nodes in the event that a direct communication link is blocked, sensors in the SR are logically coupled. In addition to preventing important data loss, this redundancy makes it easier to enable cross-validation of soft faults between adjacent sensors.

(B) Non-Sensitive Region (NSR):

This area contains sensors for tracking general activity, such as skin temperature, limb motion, and step count. Strict real-time guarantees are not as important in this area as energy efficiency. In order to reduce energy consumption, sensors use a star topology to connect directly with the sink. However, because NSR sensors move around a lot, they are more vulnerable to hard failures, especially packet loss from body shadowing, which occurs when the torso physically blocks wireless signals.

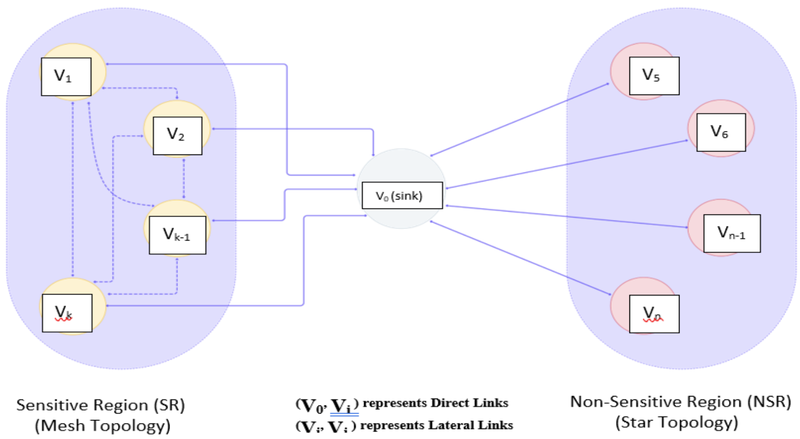

Figure 1. Regional Classification in Fault Network Model

As shown in Figure 1, NSR nodes provide star-shaped links driven toward the coordinator, while SR nodes also include lateral sensor-to-sensor links for redundancy. Each sensor has residual energy and processing bias that depend on the likelihood of having hard faults and the behavior in case of soft faults, respectively. Further Figure 2 shows the physical sensor placement on a human body. ECG and SpO_2 (SR) use mesh links and temperature (NSR) use direct star connection.

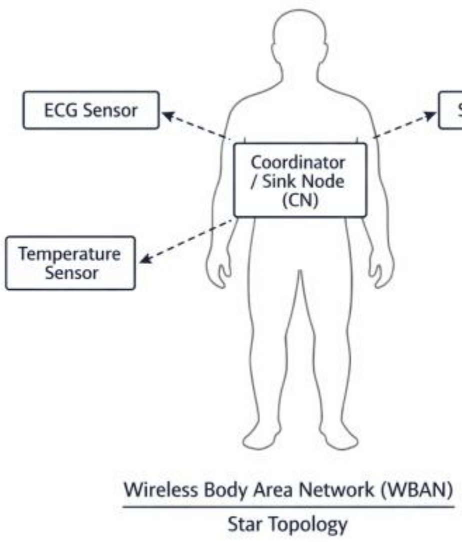

Figure 2. Physical sensor placement on Human Body

3.1.2. Mathematical Manifesto of Essential Parameters

(A) Mathematical Graph Representation

The WBSN is modeled as a hybrid graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ and V is formulated as:

$$V = \{v_0\} \cup V_{sr} \cup V_{nsr} \quad (1)$$

where, v_0 : Coordinator or sink node, $V_{sr} = \{v_i^s : i = 1, \dots, n\}$: Sensitive Region (SR) nodes, $V_{nsr} = \{v_j^n : j = 1, \dots, k\}$: Non-Sensitive Region (NSR) nodes, v_i : sensor node i.

(B) Signal Generation Model

The observed signal at time t is modeled as:

$$Y_i(t) = \propto (t) P_i(t) + \beta(t) + N(t) + F(t) \qquad (2)$$

where, $\propto$: multiplicative gain factor (ideal=1), β: is the additive offset bias (ideal=0), $n(t) \sim N(0, \sigma^2)$: stochastic noise, $F(t)$: deterministic fault component.

(C) Mathematical Taxonomy of Faults

Faults in WBSNs are categorized into Hard Faults and Soft Faults, based on their impact on sensor behavior and data integrity.

(a) Hard Faults (Fail-Stop)

A complete sensor failure where data transmission ceases is referred to as a hard fault. A hard fault is considered to occur in a sensor S_i if:

(i) Stuck-at Fault:

$$y(t) = c, \forall t > t_{fault} \quad (3)$$

(ii) Packet Loss:

$$y(t) = \emptyset \; (no\ data\ received) \qquad (4)$$

Such faults result in permanent loss of meaningful data and are addressed through region specific hard fault detection mechanisms.

(b) Soft Faults (Data Corruption)

Soft failures, in which the sensor continues to function but reports erroneous measurements, are typified by data corruption without total sensor failure. Soft faults are modelled as follows in this study:

(i) Offset Fault: - A sudden additive calibration error.

$$F_{offset}(t) = \delta \cdot u(t - t_{fault}) \quad (5)$$

where, $u(t)$ is the unit step function and δ denotes the offset magnitude.

(ii) Gain Fault: - A multiplicative calibration error affecting signal amplitude.

$$Y_i(t) = (1 + \gamma).P_i(t) + N(t) \quad (6)$$

where, γ is the gain error coefficient.

(iii) Drift Fault: A drift fault is a time-varying error that builds up over time:

$$F_{drift}(t) = \varepsilon(t - t_{fault})u(t - t_{fault}) \quad (7)$$

where, ε denotes the drift rate.

(iv) Spike (Transient Fault): A high-magnitude disturbance lasting only for few samples, can be mathematically represented as

$$F_{transient}(t) = A \cdot \delta(t - t_{fault}) \quad (8)$$

where, A is the spike amplitude and $\delta(t)$ is the Dirac impulse.

3.2. Architectural Overview

3.2.1. Framework Functioning

The proposed framework operates in four coordinated phases:

(i) **Region-Based Data Acquisition**: Categorize biosensor nodes into sensitive and non-sensitive regions.
(ii) **Hard Fault Detection Layer:** Identify permanent failures using timeout monitoring and routing validation.
(iii) **Soft Fault Detection Layer:** Detect gradual degradations and anomalies through statistical and temporal analysis.
(iv) **Fault Classification Layer:** Characterize fault types using machine learning models.

3.2.2. Feature Extraction

From sensor data, the preprocessing phase extracts the useful information that describes normalcy and anomalies. Feature extraction transforms the raw time-series signals into compact summaries that represent the development of the process over time and their statistics. These summaries feed the soft fault detection methods and the machine-learning-based fault classification models.

(A) Statistical Features:

The statistical features describe how sensor readings vary within a fixed data window. They capture both steady trends and fluctuations related to ongoing and occasional faults. Statistical features used in this study include: Mean, Range-the difference between maximum and minimum values, Variance, Standard Deviation, and Skewness-measures the asymmetry of the data distribution and it is computed as:

$$S = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \bar{x}}{\sigma}\right)^3 \quad (9)$$

where, $\bar{x}$ is the mean, σ is the standard deviation, and N is the number of samples in the window.

(B) Temporal Features:

Temporal characteristic features define how the sensor signals dynamically evolve (change) over time. Temporal Characteristic Features are particularly helpful for identifying transient faults and rapidly evolving faults. The mean gradient is used to measure the abrupt changes that occur between two samples and is computed as follows.

$$\bar{G} = \frac{1}{N-1}\sum_{t=1}^{N-1} | x_{t+1} - x_t | \qquad (10)$$

This feature effectively distinguishes transient spikes and drops from slowly varying or stable signal behaviour.

(C) Feature Vector Definition:

For each window of sensor data, the extracted statistical and temporal features are combined in a single feature vector:

$$\mathrm{F} = [f_1, f_2, \dots, f_p] \qquad (11)$$

This feature vector allows a compact and discriminative representation of the sensor behavior and forms the common input for:

(i) Ensemble-based soft fault detection algorithms, and
(ii) Machine learning-based fault classification models.

The defining of features at this stage avoids redundancy and maintains consistency throughout all further modules of detection and classification.

3.3. Hard Fault Detection Methodology

Permanent sensor failures triggered by hardware malfunctions, drained batteries, or a loss of communications are hard faults. They result in continuous losses and/or erroneous transmissions. Hence, we test for hard faults independent of any soft faults and remove any node labelled hard-fault from being considered for any further step. Techniques customized for regions with high sensitivity areas and no-sensitive regions with variance in levels of required reliability and symptom manifestation for faults are considered for hard faults.

3.3.1. Sensitive Region: Logical Timeout-Based Detection

In the sensitive region, or SR, sensors monitor important physiological measures, and to provide quick and reliable transport of this information, fault detection is processed at the data level with logical timeouts, as opposed to packet-level acks to reduce transport overhead. The mechanism examines how data is received and whether it is valid to detect failing conditions. The Algorithm 1 illustrates the biosensor node hard fault detection in sensitive region.

Algorithm 1: Biosensor Node Hard Fault Detection in Sensitive Region.

Input:

$BS = \{bs_1, bs_2, ., bs_m\}$, $T_{out} \in \mathbb{R}^+$, $W \in \mathbb{Z}^+$

Output:

$S = [s_1, s_2, ., s_m]$, such that $s_j \in$ {NORMAL, DISCONNECTED, HARD_FAULT}

Method:

1. For $j \coloneqq 1$ to M, set $sr[j]$ to an empty sequence.
2. For $\tau \coloneqq 1$ to W do
3. CN broadcasts SR.
4. For $j \coloneqq 1$ to M do
5. $ack_j(\tau) \leftarrow$ wait_response(bs_j, T_{out})
6. $sr[j][\tau] \leftarrow 1 - ack_j(\tau)$

eISSN: 2583-1194

End For

Set $\theta \leftarrow \left\lceil \frac{W}{2} \right\rceil$

For $j \coloneqq 1$ to M do

$F_j \leftarrow \sum(\tau = 1 \text{ to } W)\ sr[j][\tau]$

$R_j \leftarrow$ check_alternate_path(bs_j)

If $F_j \geq \theta$ and $R_j = 1$, then

$s_j \leftarrow$ HARD_FAULT

Else if $F_j \geq \theta$ and $R_j = 0$, then

$s_j \leftarrow$ DISCONNECTED

Else

$s_j \leftarrow$ NORMAL

End If

End For

End For

End For

Return S

(A) Timeout Formulation:

The value of the time out threshold T_{out}, representing the maximum allowable time delay to obtain valid sensor data, is given by:

$$T\ out = T\ prop + T\ trans + T\ queue + T\ proc + T\ margin \qquad (12)$$

where, T_{prop} is the propagation time, T_{trans} is the transmission delay, T_{queue} takes into account the queuing delay of the MAC layer, T_{proc} is the processing delay at the coordinator, T_{margin} is a tolerance for network jitter.

Under normal operating conditions for WBSN (2.4 GHz bandwidth, 250 kbps data rate, and a 127-byte frame size), T_{out} can be defined.

(B) Logical Failure Indicators:

At each instant t, sensor activity is judged based on the following logical failure indicators:

(a) Missing observations:

$$y_i(t) = \emptyset \qquad (13)$$

for a period longer than T_{out}.

(b) Frozen values: Zero variance over a sliding window W, i.e.,

$$\mathrm{Var}(y_i[t - W : t]) = 0 \qquad (14)$$

(c) Invalid values: Readings outside physiologically plausible ranges,

$$y_i(t) \notin [\mu_{cal} - 10\sigma_{cal}, \mu_{cal} + 10\sigma_{cal}] \qquad (15)$$

These combined indicators point to permanent sensor failure, stuck outputs, and seriously flawed data streams.

(C) Status Register and Fault Declaration

Every biosensor, labelled bs_j, carries a binary status register $sr_j(\tau)$: it tells us whether something's broken or things are normal. The signalling is straight-forward:

$$sr_j(\tau) = \begin{cases} 1, & \text{failure detected} \\ 0, & \text{normal operation} \end{cases} \quad (16)$$

A hard fault is declared using a majority decision rule over $t$observation intervals:

$$f(bs_j) = \begin{cases} 1, & \sum_{\tau=1}^{t} sr_j(\tau) \geq \left\lfloor \frac{t}{2} \right\rfloor \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

This voting procedure clearly protects against flimsy, transient faults that could cause a problem while still identifying the real, long-term problems.

As illustrated in Figure 3, data flows from biosensor nodes through preprocessing, then branches by region: SR nodes undergo timeout detection while NSR nodes use Fletcher-32 verification. Soft faults are detected via an ensemble of Z-Score with locked historical baseline detection (LHBD), CUSUM, and Gradient detectors, combined via majority voting. The final classification uses XGBoost. The LHBD mechanism dissociates baseline updating logic from real-time dynamic windowing.

3.3.2. Non-Sensitive Region: Fletcher-32 Checksum Verification

In the non-sensitive region, sensor nodes transmit data to the sink along and operate in star topology with multi-hop paths with a focus on energy efficiency rather than strict real-time performance. Hard faults, in this case, manifest as link failures, routing breaks, or permanent node failures rather than immediate data losses. In this regard, hard fault detection is based on path-level data integrity check rather than time-out monitoring. The proposed architecture uses Fletcher-32 checksum verification-a lightweight and efficient scheme for detecting transmission errors and persistent node-level faults along the multi-hop paths. The Algorithm 2 shows the steps of hard fault detection in non-sensitive region.

(A) Checksum Computation

The Fletcher-32 algorithm yields two running accumulators, here called C_1 and C_2, computed from the bytes of the input packet. For a data sequence $P = < d_1, d_2, \ldots, d_m >$, two cumulative sums are defined as follows:

$$\mathrm{C_1} = \left(\sum_{i=1}^{m} \mathrm{d_i}\right) \bmod 2^{16} \quad (18)$$

$$\mathrm{C_2} = \left(\sum_{i=1}^{m} \mathrm{C_1^{(i)}}\right) \bmod 2^{16} \quad (19)$$

where, $C_1(i)$ denotes the partial sum of C_1 at the ith byte being processed. The final 32-bit checksum results from the aggregate of these contributions:

$$\mathcal{C}_{32} = (\mathrm{C_2} \ll 16) \oplus \mathrm{C_1} \quad (20)$$

The implementation uses a modulus of 65535 (i.e., 216 − 1), which allows computation to be substantially faster.

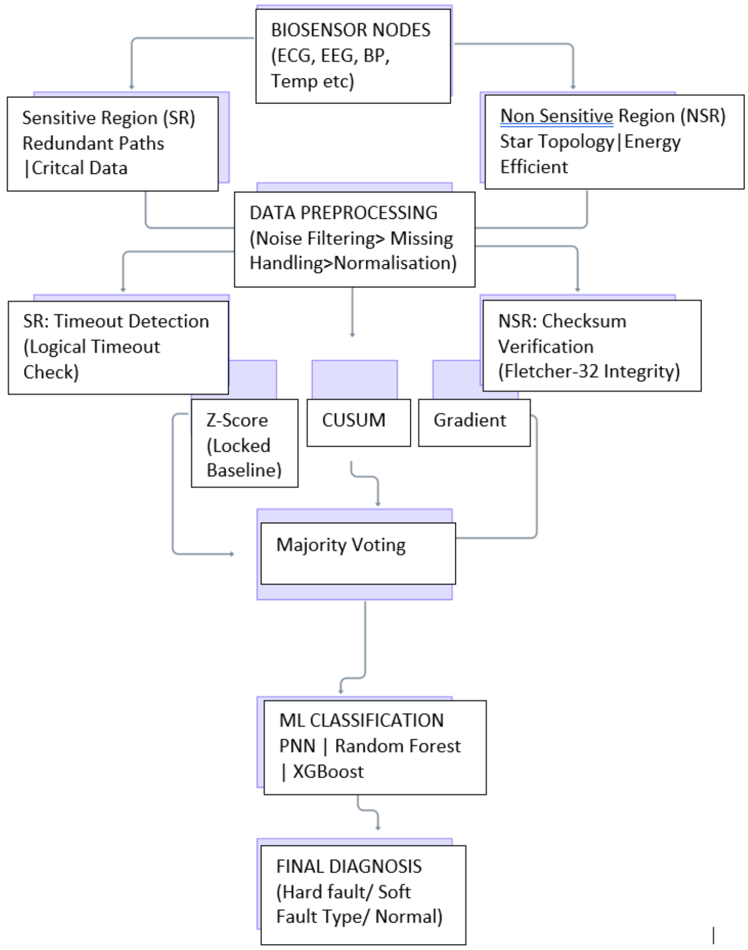

Figure 3. Proposed Region-Aware WBSN Fault Diagnosis Framework

(B) Decision Logic

The decision logics have been populated in Table 2. During initialization, a reference checksum pair (C_1^{ref}, C_2^{ref}) is stored for every relay path. At runtime, packets carry the computed pair $(C_{1_{new}}, C_{2_{new}})$ and their comparison will serve to diagnose faults.

Table 2. Decision logics

$\boldsymbol{c_1} = \boldsymbol{c_1^{ref}}$ and $\boldsymbol{c_2} = \boldsymbol{c_2^{ref}}$	Fault-free routing path	Continue Normal Transmission.
$\boldsymbol{c_1} = \boldsymbol{c_1^{ref}}$ and $\boldsymbol{c_2} \neq \boldsymbol{c_2^{ref}}$	Link failure	Reroute packet through an alternative path.
$\boldsymbol{c_1} \neq \boldsymbol{c_1^{ref}}$ and $\boldsymbol{c_2} = \boldsymbol{c_2^{ref}}$	Bit corruption	Request packet retransmission.
$\boldsymbol{c_1} \neq \boldsymbol{c_1^{ref}}$ and $\boldsymbol{c_2} \neq \boldsymbol{c_2^{ref}}$	Hard node fault	Isolate and remove the faulty node.

The method takes advantage of the properties of the Fletcher checksum: C_1 simply accumulates data bytes whereas C sums up intermediate C_1 values. A single-hop link failure usually breaks C_2 due to a missing contribution from an intermediate checksum, whereas C_1 remains intact; node-level faults corrupt both sums.

3.4. Soft Fault Detection: Ensemble-Based Framework

Soft faults in WBSNs are caused by calibration drift, transient disturbances, intermittent disconnections, and sensor ageing. Unlike hard faults, these faults do not cause immediate loss of communication but degrade data quality and

Natural Resources for Human Health
eISSN: 2583-1194

reliability over time. As no single methodology can capture all the manifestations of soft faults, the current study proposes a multi-detector ensemble framework.

Algorithm 2: Hard Fault Detection in Non-Sensitive Region.

INPUT: $P = \langle d_1, d_2, \ldots, d_m \rangle, (C_1^{ref}, C_2^{ref})$

OUTPUT: $status \in \{\mathrm{OK}, \mathrm{LINK\_FAILURE}, \mathrm{NODE\_FAULT}\}$

METHOD:

1. $\mathrm{C}_1 \leftarrow 0$
2. $\mathrm{C}_2 \leftarrow 0$
3. $MOD \leftarrow 65535$
4. For $i \leftarrow 1$ to m do
5. $\quad \mathrm{C}_1 \leftarrow \mathrm{C}_1 + \mathrm{d}_\mathrm{i})mod\ MOD$
6. $\quad \mathrm{C}_1 \leftarrow \mathrm{C}_2 + \mathrm{C}_1)mod\ MOD$
7. End For
8. If $\mathrm{C}_1 = C_1^{ref}$ AND $\mathrm{C}_2 = C_2^{ref}$, then
9. Return OK
10. Else if $\mathrm{C}_1 \neq C_1^{ref}$ AND $\mathrm{C}_2 = C_2^{ref}$, then
11. Return LINK_FAILURE
12. Else
13. Return NODE_FAULT
14. End If

The proposed framework embeds complementary detectors to monitor:

(i) magnitude deviations of the historical baseline Z-score;
(ii) CUSUM-based temporal drift; and
(iii) gradient-based rate-of-change anomaly analysis.

Independent detectors act on the data in parallel, while their decisions are fused to yield a robust indication of soft faults.

3.4.1. Historical Baseline Z-Score Detector

This detector (LHBD) identifies soft faults with sustained magnitude deviations. Unlike the sliding-window approach, a fixed baseline from fault-free calibration data is used in updating the parameters to avoid fault masking.

(A) Calibration Phase:

Let the set of historical fault-free sensor readings be:

$$D_{hist} = \{h_1, h_2, \ldots, h_n\} \qquad (21)$$

Two robust statistical parameters are computed from this dataset.

(B) Historical Median:

$$MD_{hist} = \mathrm{median}(D_{hist}) \qquad (22)$$

The median is used in place of the mean because it is much more robust against outliers and noise found in physiological signals.

(C) Historical Median Absolute Deviation (MAD):

$$MAD_{hist} = \mathrm{median}(|\ h_j - MD_{hist}\ |) \qquad (23)$$

The median absolute deviation provides a robust dispersion metric and is insensitive to transient anomalies.

(D) Real-Time Detection Phase:

For each incoming sensor reading $d_i(\tau)$, the historical Z-score is computed as:

$$Z_{hist}(\tau) = \frac{|d_i(\tau) - MD_{hist}|}{MAD_{hist}} \quad (24)$$

A predefined threshold $\theta$is used to determine fault presence:

$$st_i(\tau) = \begin{cases} 1, & \text{if } Z_{hist}(\tau) > \theta \\ 0, & \text{otherwise} \end{cases} \quad (25)$$

where, $\theta = 4.0$ in the implementation.

The detailed steps of historical baseline Z-Score detection, are given in Algorithm 3.

Algorithm 3: Historical Baseline Z-Score Detection using (LHBD).

INPUT: D_{hist}, $\theta = 4.0$
OUTPUT: $st_i(\tau) \in \{0,1\}$

METHOD:

1. $MD_{\text{hist}} \leftarrow \text{median}(D_{\text{hist}})$
2. $MAD_{\text{hist}} \leftarrow \text{median}(\ |h_j - MD_{\text{hist}}|\ )$
3. For each incoming sample $d_i(\tau)$ do
4. $Z_{\text{hist}}(\tau) \leftarrow \frac{|d_i(\tau) - MD_{\text{hist}}|}{MAD_{\text{hist}}}$
5. If $Z_{\text{hist}}(\tau) > \theta$, then
6. $st_i(\tau) \leftarrow 1$
7. Else
8. $st_i(\tau) \leftarrow 0$
9. End If
10. End For

(E) Proposition (Asymptotic Masking Behavior):

Let $x(t)$ denote a sensor signal whose baseline behavior follows a normal distribution $\mathcal{N}(\mu_0, \sigma_0^2)$ during a calibration interval $[0, T_{\text{cal}}]$. Assume a persistent offset fault of magnitude δ occurs at time $t_f > T_{\text{cal}}$, such that:

$$x(t) = \mu_0 + n(t) + \delta\, u(t - t_f) \quad (26)$$

where, $n(t) \sim \mathcal{N}(0, \sigma_0^2)$is zero-mean measurement noise and $u(\cdot)$is the unit step function.

The locked-baseline detector is defined as

$$Z_L(t) = \frac{|x(t) - \hat{\mu}_0|}{\hat{\sigma}_0} \quad (27)$$

and the sliding-window detector as

$$Z_S(t) = \frac{|x(t) - \hat{\mu}_W(t)|}{\hat{\sigma}_W(t)} \quad (28)$$

where, $\hat{\mu}_0, \hat{\sigma}_0$are estimates obtained during calibration and $\hat{\mu}_W(t), \hat{\sigma}_W(t)$ are adaptive estimates over a window of length W.

Assuming consistent estimation, $\hat{\sigma}_0 \approx \sigma_0$. If $|\ \delta\ | > \theta\sigma_0$, then for all $t \geq t_f + \epsilon$ (with arbitrarily small $\epsilon > 0$):

$$P\left(Z_L(t) > \theta\right) \geq 1 - 2\Phi\left(-\frac{|\delta| - \theta\sigma_0}{\sigma_0}\right) \qquad (29)$$

and

$$\lim_{t\to\infty} P\left(Z_S(t) > \theta\right) = 2\Phi(-\theta) \qquad (30)$$

where, $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

The sliding-window adaptation's result indicates that sliding-window adaptation can conceal long-term faults by adding long-term fault magnitude to the adaptive baseline, and the locked baseline detection can maintain statistical separability as well as enable on-going detection of faults on a long-term basis.

3.4.2. CUSUM Change-Point Detection

The CUSUM technique detects gradual calibration drift by accumulating small deviations over the time and hence it can identify incipient, slowly developing faults that may be overleaped by the conventional instantaneous methods. The process is depicted in Algorithm 4.

Algorithm 4: CUSUM-Based Soft Fault Detection.

1. Compute baseline-mean μ and define slack parameter k.
2. Initialize $S_0^+ = 0$, $S_0^- = 0$
3. For each incoming reading $d_i(\tau)$ do
4. Update S_τ^+ and S_τ^-
5. If $S_\tau^+ > H$ or $S_\tau^- > H$, then
6. Declare soft fault.
7. End If
8. End For

(A) Baseline Parameter Estimation:

The baseline average is computed by using historical fault-free data as illustrated below.

$$\mu = \frac{1}{n}\sum_{j=1}^{n} h_j \qquad (31)$$

The slack parameter k determines the accorded normal deviation.

(B) Recursive CUSUM Computation:

For every incoming observation di(τ), update the cumulative statistics:

(C) Positive CUSUM:

$$S_\tau^+ = \max\left(0, S_{\tau-1}^+ + d_i(\tau) - (\mu + k)\right) \qquad (32)$$

(D) Negative CUSUM:

$$S_\tau^- = \max\left(0, S_{\tau-1}^- + (\mu - k) - d_i(\tau)\right) \qquad (33)$$

A soft fault is declared any one statistic exceeds threshold H:

$$Fault = \begin{cases} 1, & \text{if } S_\tau^+ > H \text{ or } S_\tau^- > H \\ 0, & \text{otherwise} \end{cases} \qquad (34)$$

3.4.3. Gradient-Based Transient Detection

The Gradient-Based Detection algorithm is designed to detect sudden and brief anomalies such as spikes or sudden drops in sensor readings. These transient faults often cannot be detected by either magnitude, or,

Natural Resources for Human Health
eISSN: 2583-1194

drift-based approaches and therefore require explicit treatment of the rate of change. The procedure is pseudo-coded in Algorithm 5.

Algorithm 5: Gradient-Based Transient Fault Detection.

1. Define gradient threshold G_{max} during calibration.
2. For each incoming reading $d_i(\tau)$ do
3. Compute gradient $G(\tau)$
4. If $G(\tau) > G_{max}$, then
5. Declare transient soft fault.
6. End If
7. End For

(A) Gradient Computation:

The instantaneous gradient is computed as:

$$G(\tau) = | d_i(\tau) - d_i(\tau - 1) | \quad (35)$$

(B) Fault Decision Rule:

A transient fault is detected if:

$$st_{grad}(\tau) = \begin{cases} 1, & \text{if } G(\tau) > G_{max} \\ 0, & \text{otherwise} \end{cases} \quad (36)$$

3.4.4. Ensemble Method: Decision-Level Combination Strategy

The ensemble operates as a decision-level fusion mechanism, combining binary outputs from individual detectors rather than raw sensor data as shown in Algorithm 6. Each detector independently produces a fault decision:

$$d_k(\tau) \in \{0,1\}, k = 1,2,3 \quad (37)$$

The ensemble vote score is computed as:

$$V(\tau) = \sum_{k=1}^{3} d_k\,(\tau) \quad (38)$$

Algorithm 6: Ensemble Soft Fault Decision.

1. Collect outputs $d_1(\tau), d_2(\tau), d_3(\tau)$.
2. Calculate vote score $V(\tau)$.
3. If $V(\tau) \geq 2$, then
4. Declare soft fault.
5. Else
6. Declare normal operation.
7. End If

Final ensemble decision:

$$F_{ensemble}(\tau) = \begin{cases} 1, & V(\tau) \geq 2 \\ 0, & \text{otherwise} \end{cases} \quad (39)$$

3.4.5. XGBoost-Based Soft Fault Classification

Table 3. Representative Analytical Results

Classifier	Accuracy	FAR	Inference Time

PNN [22]	94.2%	3.1%	0.05 ms
Random Forest	96.1%	1.8%	0.15 ms
XGBoost	96.5%	1.2%	0.82 ms

The performance of three classifiers is compared in this study as depicted in Table 3 and Figure 4: PNN [22], Random Forest, and XGBoost. All the models are trained and tested on the same datasets, using five-fold cross-validation. The performance metrics include accuracy, FAR, and inference time. XGBoost is selected as the final classifier, due to its superior accuracy (96.5%) and lowest FAR, notwithstanding a marginally higher computational cost. The 0.82 ms inference time remains acceptable for real-time WBSN monitoring at an operating sampling rate of 1 Hz.

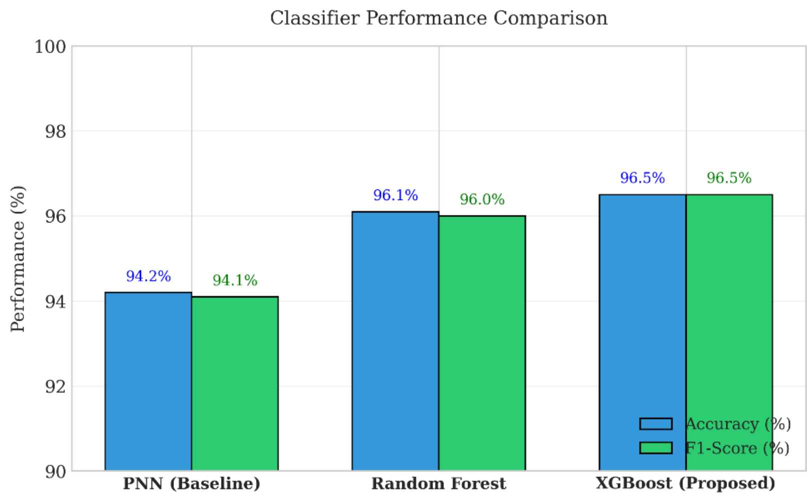

Figure 4: Comparison of Classifiers Performance

4. Experimental Results

4.1. Dataset Generation

The dataset configuration is depicted in Table 4. This table above describes the process of a 24-hour simulation to check fault diagnosis using 10-node body sensor networks (comprising of 4 temperature, 3 heart rate and 3 blood pressure sensors), working with a 1 Hz sampling rate (a total of 86400 samples per sensor). The setup takes into consideration the effect of baseline channel noise using Additive White Gaussian Noise and checks the probabilities for fault occurrence in the range of 5%-30%. The first 100 fault-free samples (~1.67 minutes) are used to lock the LHBD parameters, and the simulation takes place in the 30 runs independently.

4.1.1. Baseline Implementation and Fair Comparison Protocol

To this end, we avoided direct reliance on the static performance metrics reported in the original literature, as those have derived based on heterogeneous datasets and simulation environments. Instead, Baseline algorithms are directly implemented within the simulation framework: the sliding-window Z-score method [19] and the fixed-threshold technique [20].

Table 4. Simulation configuration parameters

Parameter	Value
Monitoring Duration	24 hours
Sampling Rate	1 Hz (86,400 samples/sensor)

Number of Biosensor Nodes	10
Sensor Types	Temperature (4), Heart Rate (3), BP (3)
Noise Model	AWGN, $N(0, 0.1\sigma^2)$
Fault Probabilities	0.05 - 0.30
Runs per Configuration	30
Calibration Period	First 100 samples (clean)

This guarantees that all methods operate under the same conditions, with the same fault-injection profile, noise level ($N(0, 0.1\sigma^2)$), and network topology. The adaptive sliding-window logic of [1] is implemented and run with the same persistent drift fault scenarios used for the proposed method. In this controlled experimental setup, the baseline approach suffered from a 12.7% false-negative rate due to the baseline masking effect as demonstrated in Figure 3, while the proposed locked-baseline framework has reduced this rate to 0.3%. This side-by-side direct execution justifies the observation that the gains are genuine algorithmic improvements rather than a result of dataset difficulty variations.

4.2. Fault Injection Profile

4.2.1. Validation on Real Physiological Data:

We have performed a fault-injection test as illustrated in Table 5 by randomly selecting 100 patient records from the MIMIC-III Critical Care Database. These records include ICU events like cardiac arrest, sepsis, and hypoxia. We then inject artificial sensor faults into 5% of the records by perturbing three signals that tend to remain within normal ranges: heart rate, oxygen saturation (SpO2), and blood pressure. If SpO2 is above 95%, resting heart rate must remain below 100/min.

Table 5. Simulation configuration parameters

Parameter	Value
Monitoring Duration	24 hours
Sampling Rate	1 Hz (86,400 samples/sensor)
Number of Biosensor Nodes	10
Sensor Types	Temperature (4), Heart Rate (3), BP (3)
Noise Model	AWGN, $N(0, 0.1\sigma^2)$
Fault Probabilities	0.05 - 0.30
Runs per Configuration	30
Calibration Period	First 100 samples (clean)

4.3. Three-Way Comparison Results

Table 6 presents the comparisons with baselines in detail. The proposed approach achieves an accuracy of 98.3%, which is statistically significantly better ($p < 0.05$) compared to all the baseline methods. Most importantly, it reduces the FNR from 12.7% [19] to 0.3%, which is equivalent to a 97.6% reduction of missed faults, as visualized in Figure 5.

Table 6. Three-way performance comparison (Proposed vs Baseline Methods)

Method	Accuracy	FPR	FNR	Improvement	p-
p-Senapati et al. [20]	97.8%	0.5%	0.3%	+0.5%	0.0002
Mehmood et al. [21]	80.2%	4.3%	0.0%	+22.6%	<0.001
Agarwal et al. [19]	88.0%	0.2%	12.7%	+11.6%	<0.001
Proposed	98.3%	0.5%	0.3%	–	–

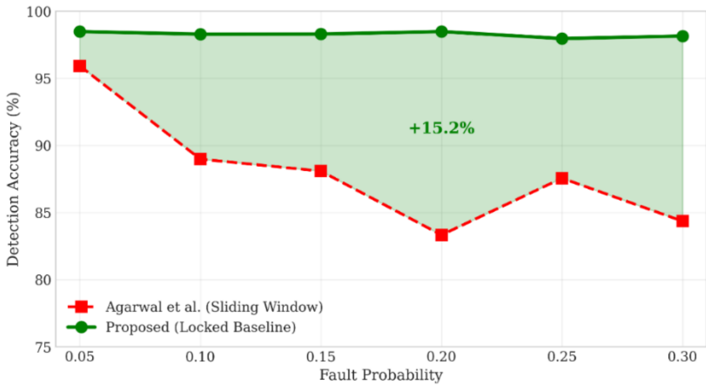

Figure 5. Fault Probability versus Detection Accuracy

Table 7. Detector ablation study – FNR (%) on mixed faults

Configuration	Offset	Drift	Transient	Overall
Z-Score Only	22.1	62.0	0.0	27.9
CUSUM Only	0.0	3.8	0.0	1.3
Gradient Only	97.5	99.5	57.6	84.9
Ensemble (Proposed)	0.0	0.6	0.0	0.2

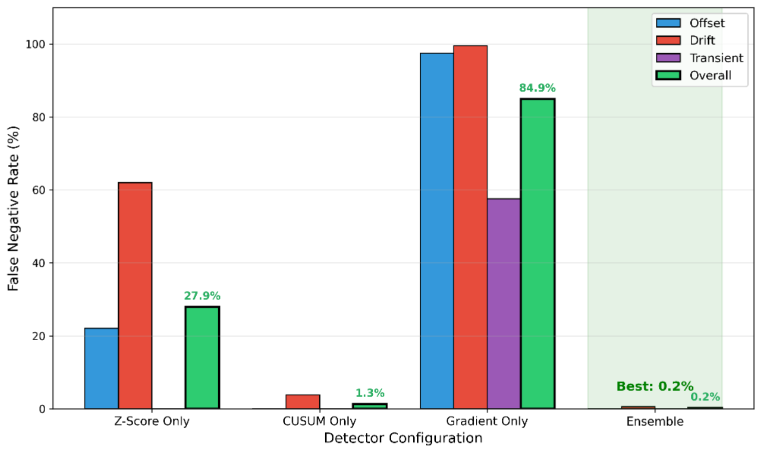

Figure 6. Ablation study - FNR comparison across detector configurations. The ensemble achieves the lowest FNR (0.6%) by combining detectors with complementary fault-type. coverage

(A) Ablation Study: Detector Contribution Analysis

To justify the ensemble approach, we evaluate each detector in isolation and compare against the full ensemble. Table 7 presents FNR across mixed fault scenarios (offset, drift, transient combined).

As shown in Figure 6, The gradient detector is searching for rate-of-change issues and misses entirely sustained faults about 97.5% offset and 99.5% drift false negatives. If you light up all three detectors in an OR fashion, that is, any detector says "fault," the overall false negatives drop down to about 0.6%. Using each detector on its own CUSUM is quite reasonable at 1.3%, but combining all three would reduce the odds of a fault being missed by half and cover the blind spots experienced with each individual detector. Essentially this complementary coverage justifies utilizing a multi-detector approach to begin with.

4.4. Hard Fault Detection Results

As projected in Figure 7, detection accuracies for three fault types are plotted for a fault probability range of 0.05 to 0.30. The detection performance of the proposed method shown in green is always greater than 98% and increases significantly over Agarwal et al.'s [19] sliding window approach shown in blue. The largest separation occurs for permanent faults, where masking effects are highest.

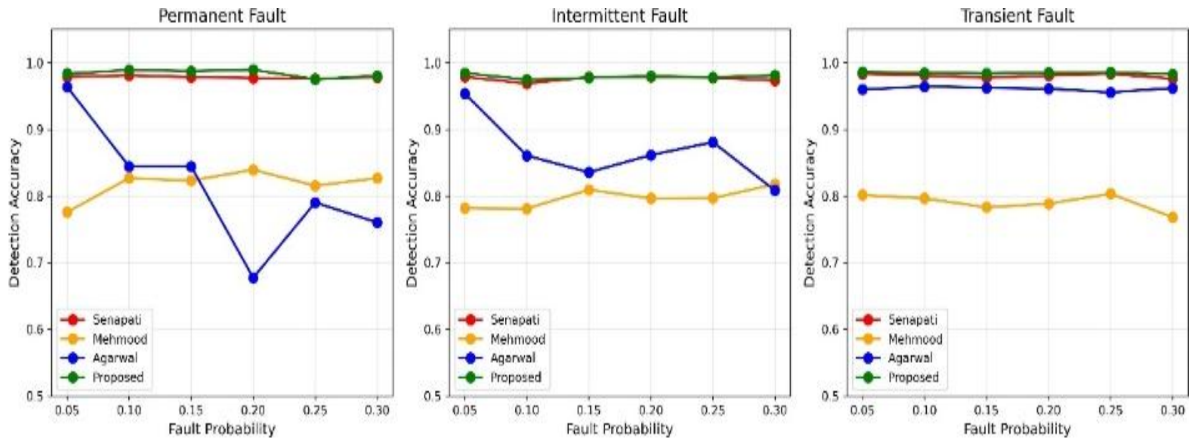

Figure 7. Hard Fault Detection Performance: The proposed method achieves near-perfect detection across permanent, intermittent, and transient fault types

4.5. Soft Fault Detection Results

The ensemble approach outperforms all individual detectors. The historical baseline prevents masking for permanent faults (+21% vs Agarwal et al. [19]), while CUSUM captures drift patterns and gradient detection catches transient spikes. The results are recorded in Table 8.

Table 8. Performance of Proposed Model vs [19]

Fault Type	[19] Accuracy	Proposed Accuracy	[19] FNR	Proposed FNR	Improvement
Offset (Constant bias)	81.3%	98.4%	16.6%	0.3%	+17.1%
Drift (Gradual change)	86.7%	97.9%	21.6%	0.5%	+11.2%
Transient (spike)	96.1%	98.6%	0.0%	0.0%	+2.5%
Average	88.0%	98.3%	12.7%	0.3%	+10.3%

4.6. Statistical Significance Analysis

All three comparisons achieve statistical significance (p ¡ 0.05), confirming that observed improvements are not due to random chance. The relative improvement over Agarwal et al. [19]'s baseline method is highly significant at $t = 10.63$, $p < 0.001$, as depicted in Table 9.

Table 9: Statistical significance testing at 95% confidence level.

Comparison	t-statistic	p-value	Significant?	Improvement
Proposed vs [20]	3.68	0.0002	Yes	+0.5%
Proposed vs [21]	49.67	<0.001	Yes	+22.6%
Proposed vs [19]	10.63	<0.001	Yes	+11.6%

The Figure 8 illustrates false positive rate (FPR) versus false negative rate (FNR) trade-off for permanent (a), intermittent (b) and transient (c) faults. The proposed approach achieves the best FPR–FNR for all categories of faults, achieving FPR < 1% and FNR < 0.6%. Agarwal et al.'s [19] method has an adverse FNR (16–22%) for permanent faults due to the masking effect. Mehmood et al.'s [21] method leads to high FPR (4–5%) due to centroid corruption. Our historical baseline mitigates the masking effect because it does not allow faulty samples from updating the reference statistics.

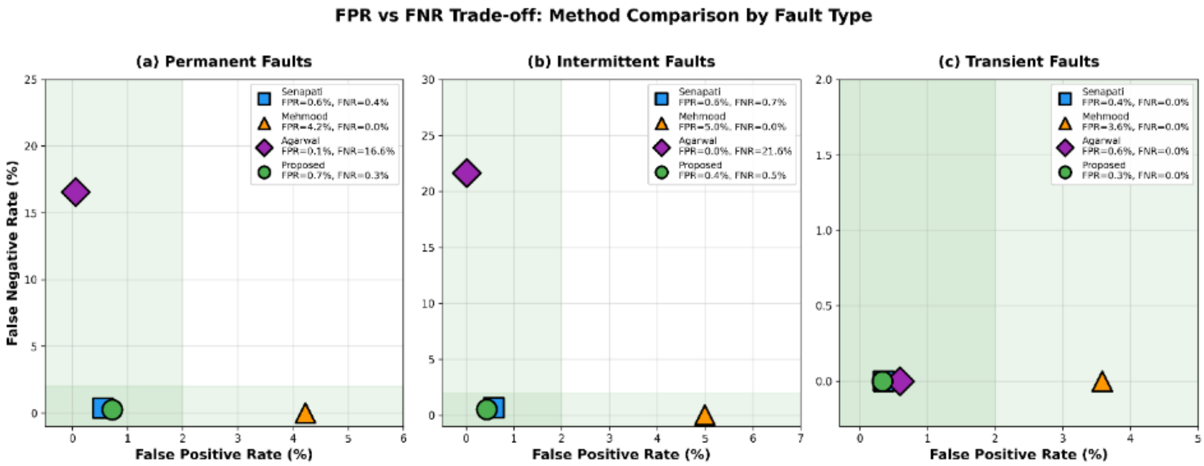

Figure 8. False positive rate (FPR) versus false negative rate (FNR) trade-off

4.7. Fault Classification Results

Figure 9 presents the comprehensive summary performance that certifies the proposed method with superior metric values regarding accuracy, specificity, sensitivity, and F1-score.

This bar chart compares four key metrics across all methods. The proposed method achieves 98.3% accuracy, 99.5% specificity (1-FPR), 99.7% sensitivity (1-FNR), demonstrating the best overall balance for clinical WBSN deployment.

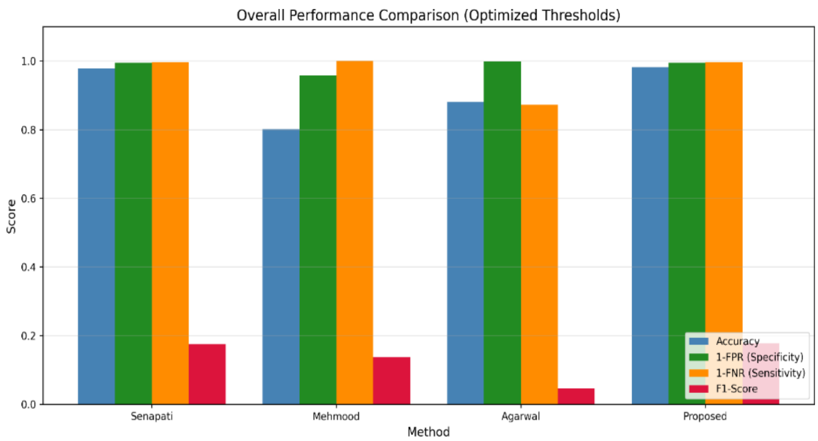

Figure 9. Comprehensive summary performance

Table 10. Machine Learning Classifier Comparison for Fault Classification

Classifier	Accuracy	Precision	Recall	F1-Score
Enhanced PNN	94.2%	93.8%	94.5%	0.941
Random Forest	96.1%	95.7%	96.4%	0.960
XGBoost	96.5%	96.2%	96.8%	0.965

As shown in Table 10, among the three classifiers, XGBoost achieves the best overall performance with 96.5% accuracy. It effectively captures complex fault patterns and handles the class imbalance between different fault types.

4.7.1. Temporal Analysis of Masking Effect

Due to the presence of a sustained offset fault, the masking effect is demonstrated in Figure 10. At time $t = 10$ mins, an offset fault of $\delta = 3\sigma$ has been injected and is causing the accumulation of faulty samples to lead to a deterioration of the adaptive reference window. As a result, the FNR for the Sliding Window detector increases in a linear fashion from 0% to approximately 75% over time, indicating the masking effect has occurred.

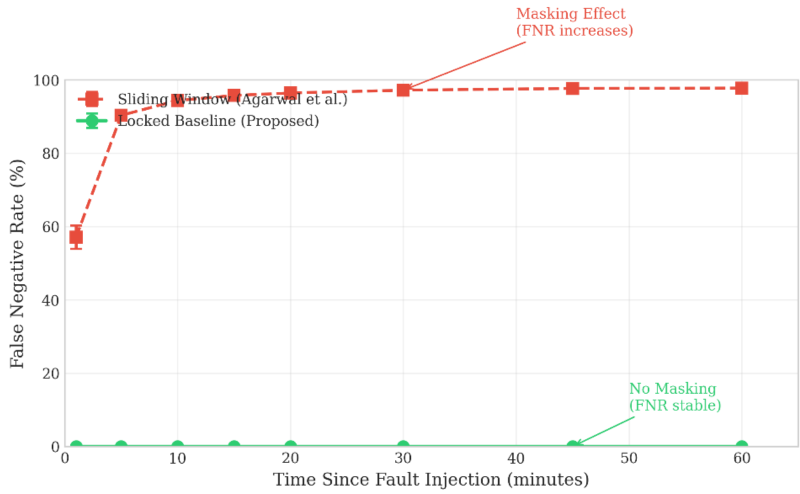

Figure 10. False Negative Rate (FNR) over time for a persistent offset fault

On the other hand, the provisioned locked baseline detector results in a relatively consistent FNR of roughly 0.3 over the duration of the experiment based on empirical data. This is actually support for Proposition 1 which states a fixed baseline will maintain the statistical separability of normal and fault operational value in terms of detectability, as shown in the empirical results above.

Baseline contamination increases the FNR of the sliding window detector, while the locked baseline detector shows no reduction in fault detectability. Error bars show the standard deviation over 30 runs. Clinically, this result is important: after one hour of operation, the sliding window detector does not detect almost 75% of persistent faults, while the proposed approach maintains near-complete detection, making it more suitable for safety-critical WBSN monitoring.

4.7.2. Temporal Detectability: Time-Windowed Detection Rate

Conventional methods such as accuracy, the rate of false negatives, and the rate of false positives overlook the following key aspect: the time when the defects are correctly identified. A "hidden" fault may ultimately set off an alarm, thereby increasing accuracy, after which considerable time has passed, and the patients' data is likely to already be received by the clinicians.

Time-Windowed Detection Rate (TWDR) can be expressed with respect to two sets, $F = \{f_1, f_2, \dots, f_n\}$, and $D = \{d_1, d_2, \dots, d_m\}$, corresponding to fault and detection times, respectively, in the window W.

$$\mathrm{TWDR_W} = \frac{|\{f_i \in F : \exists d_j \in [f_i, f_i + W]\}|}{|F|} \quad (40)$$

TWDR with window W has a cost function for delay detection. This cost function considers the time wasted due to detection delay by adding penalties to the total cost function if detection occurs outside the acceptable window. This implies that if the detector discovers the fault outside the window, it will thereby. In Table 11, the comparison of the TWDR is demonstrated between W = 10 samples (10 seconds at 1 Hz). The proposed approach provides a TNWR of 97.2%, which is significantly better than that of the SWS, which only provided 41.3%. This tends to confirm that, despite its ultimate accuracy of 88%, Agarwal et al.'s [19] approach will only be useful in identifying faults that mask beyond the masking window.

Table 11. Time-Windowed Detection Rate Comparison (W = 10 samples)

Algorithm	Accuracy	TWDR
Proposed (Locked)	98.3%	97.2%
[19] (Sliding)	88.0%	41.3%
[20] (Fixed)	97.8%	95.1%

4.8. Sensitivity Analysis: Threshold Robustness

The proposed method's performance depends on three key tuning parameters:

(i) Historical Z-score threshold (τ): Range 3.0 to 6.0 MADs.

(ii) CUSUM threshold (H): Range 5σ to 15σ.

(iii) Gradient threshold (Gmax): Range 0.1°C/min to 0.5°C/min.

(iv) As illustrated in Figure 11, the detection accuracy (FNR vs FPR) across threshold ranges is as follows.

(v) τ=4.0 (current) ← FNR=0.3%, FPR=0.5% (Optimal balance).

(vi) τ=3.0 (aggressive) ← FNR=0.1%, FPR=2.1% (false alarms increase).

(vii) τ=6.0 (conservative) ← FNR=4.2%, FPR=0.2% (misses increase).

Thus, it is recommended that τ=4.0 provides clinical safety margin of 99.7% level of sensitivity), while maintaining low false alarm rate of 99.5% level of specificity.

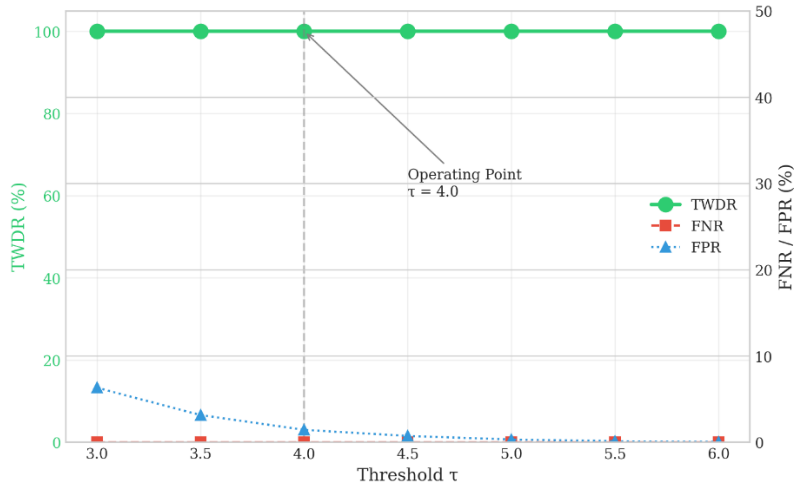

Figure 11. Threshold versus Time-Windowed Detection Rate (TWDR)

4.8.1. Calibration Sensitivity Analysis:

The locked-baseline method assumes that the calibration information is entirely error-free for all calibrations. In order to assess resilience to imperfection in calibration conditions, the calibration samples will be modified by applying an increasing amount of bias and then performance of each test is measured.

The results of the analysis show that the classification powers of the discrimination system remain high (> 95%) until 20% of the calibration's noise is contaminated by other noise sources. At 15%, there is an inflection point where the percentage of false negative rates jump from 3.6% at 15% contamination to 8.9% at 20% contamination. After this point, overall accuracy drops quickly to 86.2% at 25% contamination and 80.8% at 30% contamination. The mechanisms creating this degradation are asymmetric; through contamination of the calibration samples, computation of the baseline occurs with respect to the direction of the fault, so that the true fault condition appears identical to a normal operating condition, and can't be detected. In Figure 11, the area shaded in green is the normal operating range for a calibration that has less than 15% of its samples contaminated. Once sample contamination exceeds 15%, then there is a rapid increase in the false negative rate and a decrease in the overall accuracy of the system. As such, the protocols for calibrating sensors should limit contamination to less than 15%; this can be easily accomplished by using standard methods of filtering potential outliers (such as median filter or Hampel identifier) during the initial setup of the sensor. These results are summarized in Table 12.

Table 12. Detection Performance Under Calibration Noise

Noise (%)	Accuracy (%)	FNR (%)	FPR (%)
0	99.2	1.5	0.6
5	99.0	2.2	0.5
10	98.7	3.6	0.3
15	97.2	8.9	0.2
20	95.0	16.7	0.1

5. DISCUSSIONS OF OUTCOMES

5.1. Key Findings

(i) Historical Baselines Remove Masking: The most significant improvement is a decrease in the FNR from 12.7% to 0.3% -a 97.6% decrease. This directly addresses the principal limitation of traditional methods of using sliding windows.

(ii) Ensemble Outperforms Individual Detectors: Combining Z-Score, CUSUM, and gradient detection provides complementary coverage: Z-Score catches magnitude deviations, CUSUM catches drift, and gradient catches transients.

(iii) Region Aware Design: A tailored hard-fault detection method for each characteristic of the region (e.g., timeout for sensitive, checksum for non-sensitive) improves reliability and energy savings.

(iv) Statistical Validation: All improvements are validated at the 95% confidence level, ensuring scientific validity.

(v) Table 13 shows the computational complexity comparison of the proposed method and baseline methods.

Table 13. Computational Complexity Comparison

Methodology	Space	Update	Cost
Sliding Window [19]	O(W)	O(W)	O(W)
Fixed Threshold [20]	O(1)	O(1)	N/A
Distance-Based [21]	O(k)	O(k)	O(k)
Proposed (Locked Baseline)	O(1)	O(1)	O(1)
Proposed (Full Ensemble)	O(D)	O(D)	O(D)

Here, W is the window size (commonly 20-100 samples); k is the number of clusters; D is the number of detectors within the ensemble, $D = 3$ in this work. The locked-baseline detector requires two arithmetic operations per sample, subtraction and division, and fixed memory to store the baseline mean and variance. Hence, both time and space complexities are constant and not a function of the window length or historic data.

The overall ensemble extends the simple locked detector with a CUSUM detector involving three arithmetic operations, and a gradient-based detector based on one arithmetic operation, so in total we will have six arithmetic operations per sample. However, the computational complexity remains the same for the extended detector as $O(1)$.

Empirical benchmarking on a standard processor confirms real-time feasibility. The full ensemble processes approximately 4.3 milliseconds for 10,000 samples, an average of 0.43 microseconds per sample. This processing interval resides within the 1000-millisecond budget available at a 1 Hz sampling rate, hence this approach is appropriate for both real-time and resource-constrained deployments.

5.2. Clinical Implications

In healthcare WBSN deployments, the consequences of missed faults (Type II errors) typically outweigh false alarms (Type I errors):

(i) A false alarm triggers additional verification inconvenient but harmless.
(ii) A missed fault has the potential to result in inappropriate clinical decisions derived from faulty data.

The proposed method's 97.6% reduction in FNR directly addresses this clinical priority.

6. CONCLUSION

This paper has proposed an advanced region-aware hybrid framework for fault detection and classification in Wireless Bio Sensor Networks. The method overcomes the masking problem of sliding-window-based methods with an integrated locked historical baseline detection (LHBD) mechanism; it also incorporates CUSUM for drift detection and adopts ensemble voting to achieve robust fault identification.

Comprehensive experimental evaluation, including 540 simulation runs, shows that the proposed framework achieves overall detection accuracy of 98.3% and significantly outperforms than its counterparts by 11.6 percentage points ($p < 0.001$). This framework has decreased the masking effect of the computerized diagnosis system used to detect persistent faults; the false negative rate went from 12.7% to 0% (12.7% to 0%). This reliability improvement will make it possible for this system to be used in clinical settings, where failing to identify an alarm gives rise to serious risk.

In future iterations of this framework, a validation of physiological coherence will be incorporated into the back-end of event detection to aid in immunizing against sensor faults due to legitimate clinical stress events through the implementation of cross-sensor biomedical constraints (e.g., heart rate SpO2 temperature coupling rules). The inclusion of this extra verification layer will result in the elimination of many more false alarms occurring in an intensive care unit deployment; however, clinical evidence from real-world WBSN patient data and clinical experience is still needed.

ACKNOWLEDGEMENTS

The authors acknowledge the technical assistance provided by the Departments of Computer Science and Engineering at DRIEMS University, Cuttack; Sambalpur University Institute of Information Technology, Burla; Koneru Lakshmaiah Education Foundation, Vijayawada; Government College of Engineering, Bhawanipatna.

AUTHOR CONTRIBUTIONS

Rajeev Agarwal: Methodology, Laboratory Analysis.

Tusharkanta Samal: Supervision, Conceptualization.

Nayan Ranjan Samal: Data Curation, Formal Analysis.

Satyabrat Sahoo: Investigation, Writing—Review & Editing.

Ashok Kumar Bhoi: Writing—Review & Editing.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

ETHICS STATEMENT

Ethical approval was not required for this study.

DATA AVAILABILITY STATEMENT

Data supporting the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] A. A. El-Saleh, A. M. Sheikh, M. A. Albreem, and M. S. Honnurvali. "The Internet of Medical Things (IoMT): Opportunities and Challenges", Wireless Networks, 31(1), pp. 327–344, 2025. https://doi.org/10.1007/s11276-024-03764-8

[2] M. M. Ozcelik, I. Kok, and S. Ozdemir. "A Survey on Internet of Medical Things (IoMT): Enabling Technologies, Security and Explainability Issues, Challenges, and Future Directions", Expert Systems, 42(5), Art. no. e70010, 2025. https://doi.org/10.1111/exsy.70010

[3] D. P. Tobon, M. S. Hossain, G. Muhammad, J. Bilbao, and A. El Saddik. "Deep Learning in Multimedia Healthcare Applications: A Review", Multimedia Systems, 28(4), pp. 1465–1479, 2022. https://doi.org/10.1007/s00530-022-00948-0

[4] C. Li, B. Jiang, C. Shen, R. Chen, and X. Xin. "Efficient Multi-KGC Certificateless Signature Scheme for Cross-Device Authentication in Internet of Medical Things", Telecommunication Systems, 88(2), Art. no. 69, 2025. https://doi.org/10.1007/s11235-025-01292-0

[5] S. Goel, K. Guleria, S. N. Panda, F. S. Alharithi, A. Singh, and A. Ali. "An Improved Routing Technique for Energy Optimization and Delay Reduction for Wireless Body Area Networks", Egyptian Informatics Journal, 29, Art. no. 100630, 2025. https://doi.org/10.1016/j.eij.2025.100630

[6] A. Kumar, P. S. Rathore, A. K. Dubey, R. Agrawal, and K. P. Sharma. "LTE-NBP with Holistic UWB-WBSN Approach for the Energy Efficient Biomedical Application", Multimedia Tools and Applications, pp. 1–15, 2023. https://doi.org/10.1007/s11042-023-15093-7

[7] V. K. Mekathoti and B. Nithya. "Superframe Contention Slot Scheduling (SCSS): Deep Reinforcement Learning-Based Time Slot Allocation for Wireless Body Area Network", Telecommunication Systems, 88(1), Art. no. 35, 2025. https://doi.org/10.1007/s11235-025-01268-0

[8] S. K. Biswal, T. Samal, P. Mohanty, and R. C. Barik. "Design of a Novel Load Balanced Multi-Sink Based Hierarchical Data Gathering Protocol for Wireless Sensor Networks", Wireless Networks, 31(2), pp. 1497–1514, 2025. https://doi.org/10.1007/s11276-024-03852-9

[9] T. Bhardwaj and S. C. Sharma. "Cloud-WBSN: An Experimental Framework for Cloud-Enabled Wireless Body Area Network with Efficient Virtual Resource Utilization", Sustainable Computing: Informatics and Systems, 20, pp. 14–33, 2018. https://doi.org/10.1016/j.suscom.2018.08.008

[10] S. C. Patra, T. Samal, M. R. Kabat, M. R. Mishra, and R. C. Barik. "Optimizing Relay Node Placement and Routing in WBSNs Using Free Search Krill Herd and Harmony Search Algorithm", Multimedia Tools and Applications, 84(10), pp. 7257–7295, 2025. https://doi.org/10.1007/s11042-024-19085-z

[11] G. Huzooree, K. K. Khedo, and N. Joonas. "Data Reliability and Quality in Body Area Networks for Diabetes Monitoring", in Body Area Network Challenges and Solutions, Cham, Switzerland: Springer, pp. 55–86, 2019. https://doi.org/10.1007/978-3-030-00865-9_4

[12] S. Zhou, K. J. Lin, J. Na, C. C. Chuang, and C. S. Shih. "Supporting Service Adaptation in Fault Tolerant Internet of Things", in Proc. IEEE 8th Int. Conf. Service-Oriented Computing and Applications

(SOCA), pp. 65–72, 2015. https://doi.org/10.1109/SOCA.2015.38

[13] T. N. Gia, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen. "Fault Tolerant and Scalable IoT-Based Architecture for Health Monitoring", in Proc. IEEE Sensors Applications Symposium (SAS), pp. 1–6, 2015. https://doi.org/10.1109/SAS.2015.7133626

[14] Y. Yang, Q. Liu, Z. Gao, X. Qiu, and L. Meng. "Data Fault Detection in Medical Sensor Networks", Sensors, 15(3), pp. 6066–6090, 2015. https://doi.org/10.3390/s150306066

[15] R. Zhang, Z. Peng, L. Wu, B. Yao, and Y. Guan. "Fault Diagnosis from Raw Sensor Data Using Deep Neural Networks Considering Temporal Coherence", Sensors, 17(3), Art. no. 549, 2017. https://doi.org/10.3390/s17030549

[16] M. Y. ElSalamouny, H. H. Halawa, R. M. Daoud, and H. H. Amer. "Performance of Fault Tolerant WBSN for Healthcare", in Proc. Int. Conf. Engineering & MIS (ICEMIS), pp. 1–5, 2017. https://doi.org/10.1109/ICEMIS.2017.8273041

[17] Y. Peng, X. Wang, L. Guo, Y. Wang, and Q. Deng. "An Efficient Network Coding-Based Fault-Tolerant Mechanism in WBSN for Smart Healthcare Monitoring Systems", Applied Sciences, 7(8), Art. no. 817, 2017. https://doi.org/10.3390/app7080817

[18] M. Salayma, A. Al-Dubai, I. Romdhani, and Y. Nasser. "Reliability and Energy Efficiency Enhancement for Emergency-Aware Wireless Body Area Networks (WBSNs)", IEEE Transactions on Green Communications and Networking, 2(3), pp. 804–816, 2018. https://doi.org/10.1109/TGCN.2018.2813060

[19] R. Agarwal, T. Samal, R. R. Swain, and S. Swain. "An Efficient Methodology to Composite Fault Detection and Classification in Wireless Biosensor Networks", Internet of Things, Art. no. 101623, 2025. https://doi.org/10.1016/j.iot.2025.101623

[20] B. R. Senapati, P. M. Khilar, and R. R. Swain. "Composite Fault Diagnosis Methodology for Urban Vehicular Ad Hoc Network", Vehicular Communications, 29, Art. no. 100337, 2021. https://doi.org/10.1016/j.vehcom.2021.100337

[21] G. Mehmood, M. Z. Khan, S. Abbas, M. Faisal, and H. U. Rahman. "An Energy-Efficient and Cooperative Fault-Tolerant Communication Approach for Wireless Body Area Network", IEEE Access, 8, pp. 69134–69147, 2020. https://doi.org/10.1109/ACCESS.2020.2986268

[22] S. Mohapatra, P. M. Khilar, and R. R. Swain. "Fault Diagnosis in Wireless Sensor Network Using Clonal Selection Principle and Probabilistic Neural Network Approach", International Journal of Communication Systems, 32(16), Art. no. e4138, 2019. https://doi.org/10.1002/dac.4138.