Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 21 No. 13 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v21i13.57727

PAPER

Embedding Intelligence at the Sensor: Naïve Bayes Classifier for Real-Time Fault Diagnosis in Resource-Constrained WSNs

Yassine Aitamar[1] (✉),
Jawad Oubaha[2],
Jamal El Abbadi[1]

[1]Mohammed V University in Rabat (UM5R), Rabat, Morocco

[2]Mohammed First University (UMP), Oujda, Morocco

yassine.aitamar@um5r.ac.ma

ABSTRACT

The reliable functioning of wireless sensor networks (WSNs), integral to mission-critical Internet of Things (IoT) implementations such as industrial automation and environmental sensing, is significantly undermined by data-centric anomalies. These faults, including drift, offset, stuck-at, gain, and out-of-bounds readings, compromise data integrity, leading to faulty decisions and potential system failures. Although machine learning (ML) offers viable solutions for fault detection and classification, deploying sophisticated models onto sensor nodes with inherent resource limitations presents a persistent challenge. This study introduces an embedded Naïve Bayes (NB) classifier designed for in-situ fault detection and classification, capitalizing on its minimal computational demands and inherent noise resilience. Rigorous evaluation employing diverse real-world datasets with variable fault incidences confirms robust performance across accuracy, precision, recall, and F1-score metrics under strict energy and latency thresholds. Comparative assessment reveals substantially reduced resource utilization (computation, memory) relative to conventional techniques while sustaining high detection efficacy across heterogeneous fault conditions. This methodology effectively reconciles sophisticated analytical capabilities with the processing constraints of edge devices, markedly improving fault tolerance to ensure dependable WSN operation within dynamic IoT settings.

KEYWORDS

wireless sensor network (WSNs), Internet of Things (IoT), machine learning (ML), fault detection and classification, real-time, Naïve Bayes (NB) classifier

1 INTRODUCTION

Wireless sensor networks (WSNs) form a critical technological foundation for the Internet of Things (IoT), enabling pervasive monitoring and data collection across diverse fields such as precision agriculture, environmental sensing, industrial

Aitamar, Y., Oubaha, J., El Abbadi, J. (2025). Embedding Intelligence at the Sensor: Naïve Bayes Classifier for Real-Time Fault Diagnosis in Resource-Constrained WSNs. *International Journal of Online and Biomedical Engineering (iJOE)*, 21(13), pp. 46–62. https://doi.org/10.3991/ijoe.v21i13.57727

Article submitted 2025-08-01. Revision uploaded 2025-09-02. Final acceptance 2025-09-02.

automation, healthcare, and smart city infrastructure [1], [2]. These networks consist of spatially distributed sensor nodes equipped with integrated sensing, processing, and wireless communication capabilities. Key advantages include inherent scalability, energy efficiency, resilience in harsh environments, and autonomous operation [3]. Integrating WSNs into broader IoT ecosystems enhances real-time decision-making by delivering continuous, high-resolution, and geographically contextualized data streams from the physical environment. Thus, WSNs generate actionable insights and enable intelligent automation through low-latency data exchange and adaptive network reconfiguration [4].

Despite these benefits, WSN reliability is critically compromised by anomalous data events. Such anomalies arise from environmental interference, hardware degradation, software errors, or communication failures. Erroneous sensor readings undermine data integrity, propagate faulty decisions, increase operational costs, waste energy, and may trigger system failures [5]. Consequently, fault detection and mitigation are essential. Faults in WSNs are systematically categorized by temporal behavior (transient, intermittent, permanent) and spatial scope (node-level, network-wide). A functional taxonomy further identifies four primary domains: hardware faults (e.g., sensor failures), software faults (e.g., algorithmic errors), communication faults (e.g., packet loss, congestion), and data-centric faults (direct data corruption), including gain errors, drift, stuck-at, out-of-bound values, and offset (systematic bias) [6]. Data-centric faults pose severe risks in applications requiring high accuracy and temporal consistency, such as industrial control systems or precision irrigation feedback loops.

Machine learning (ML) has shown significant promise in detecting faults by modeling complex, high-dimensional sensor data streams [7], [8]. ML approaches fall into three paradigms: supervised learning (trained on labeled data), unsupervised learning (identifies patterns in unlabeled data), and reinforcement learning (optimizes actions via feedback). For WSN fault detection, supervised classification is particularly relevant. Classification algorithms adapt to evolving fault patterns and demonstrate noise robustness. However, deploying ML classifiers on resource-constrained WSN nodes faces significant challenges, including limited computational power, memory, energy, and stringent real-time latency requirements [7].

This study addresses the need for enhanced fault tolerance and data reliability in WSNs, especially where centralized processing is impractical due to energy constraints or real-time demands. We propose and evaluate an embedded Naïve Bayes (NB) classifier for *in-situ* fault detection on sensor nodes. NB was selected for its computational efficiency, probabilistic interpretability, and proven effectiveness in noisy, high-dimensional environments. Our comprehensive assessment uses accuracy, recall, precision, F1-score, computational cost, and execution time to validate deployment feasibility on constrained hardware. The study explicitly analyzes the trade-off between detection efficacy and resource consumption, ensuring solutions meet both reliability and operational efficiency objectives.

This study bridges advanced data analytics with the processing limitations of sensor nodes. It provides a pragmatic embedded NB strategy for fault detection and classification, strengthening WSN resilience and operational effectiveness in IoT ecosystems.

2 RELATED WORK

Machine learning approaches for fault detection and classification in WSNs represent an extensively investigated field, driven by the necessity for autonomous, intelligent, and energy-conserving IoT systems. Multiple ML architectures have been examined [9]–[11], each demonstrating distinct trade-offs between detection accuracy, computational intensity, memory footprint, and energy expenditure. This section critically examines current ML-based fault detection strategies for WSNs, analyzing strengths and limitations of prominent techniques, including decision trees (DTs), support vector machines (SVMs), k-nearest neighbors (k-NN), artificial neural networks (ANNs), and ensemble methods, with explicit assessment of their applicability to resource-constrained nodes in operational IoT deployments.

Decision trees maintain academic interest owing to interpretability and straightforward implementation. Research [12] confirms their effectiveness in classifying sensor faults using historical data-derived feature thresholds, while their hierarchical design enables low-latency inference. Significant constraints for WSN deployment emerge, however, from vulnerability to overfitting in dynamic environments and escalating memory consumption with high-dimensional features. SVMs demonstrate theoretical advantages in accuracy and robustness within high-dimensional spaces. Studies [13]–[16] indicate SVMs outperform DTs and rule-based methods for identifying drift and out-of-bound faults through kernel-enabled complex pattern separation. Adoption on low-power nodes remains hindered by computationally intensive training, reliance on floating-point operations during inference, and binarity necessitating resource-heavy multiclass extensions.

The k-NN algorithm attracts consideration for its simplicity and non-parametric foundation. Applications exist for temporal similarity-based fault classification [17], [18]. Despite avoiding explicit training and accommodating new data, prohibitive memory requirements and computational latency during inference, stemming from exhaustive comparisons against stored exemplars, largely preclude on-node implementation. ANN, including deep learning variants [19]–[21], garner attention for modeling nonlinear, hierarchical relationships. Implementations such as MLP [22], RNN [19], and LSTM [20] achieve notable performance in noisy, temporally complex scenarios. Barriers to deployment include extensive training data needs, substantial computational and memory resources, and limited explainability—a critical deficiency in systems requiring operational transparency. Ensemble techniques such as RFs and Gradient Boosted Trees enhance accuracy through classifier aggregation; RF demonstrate notable resilience to overfitting [20], [23], [14]. The inherent overhead from managing multiple base learners, however, diminishes their feasibility for real-time embedded processing.

Current studies predominantly emphasize detection accuracy while insufficiently addressing computational viability under stringent WSN limitations. Although SVMs and ANNs achieve high precision, their direct deployment on sensor nodes presents enduring obstacles. Conversely, lightweight models such as DTs offer implementation practicality but often lack sophistication for complex nonlinear fault signatures. A significant research void exists in systematic assessments quantifying the performance-resource trade-off specific to WSN constraints. Furthermore, numerous existing solutions rely on centralized processing architectures, counter to the growing imperative for distributed intelligence at the IoT edge.

3 DATA-CENTRIC FAULTS IN WSNS

Data-centric faults represent a particularly critical vulnerability within WSN-based IoT deployments, as they directly compromise the integrity of the information underpinning decision processes. Unlike hardware or communication failures, which typically manifest as data loss, data-centric anomalies generate erroneous yet deceptively valid outputs that evade conventional detection mechanisms without sophisticated analytical intervention [24].

The sensor data acquisition process is formally modeled as: $s(n, t, f(t))$ where $f(t)$ denotes the measurement captured by node n at time instant t. This measurement is decomposed as:

$$f(t) = \alpha + \beta x + \eta \tag{1}$$

Here, x represents the data point, α represents a systematic offset, β signifies a gain coefficient, and η encapsulates inherent measurement noise.

According [6], the principal classes of data-centric faults are characterized as follows:

Gain faults: Result in erroneously amplified outputs exceeding physical baselines (see Figure 1a), potentially triggering incorrect system responses (e.g., overestimation of temperature or moisture). Modeled as:

$$x' = \beta x + \eta \tag{2}$$

In the above equation $\beta \neq 1$

Drift faults: Exhibit progressive deviation from true values over temporal intervals (see Figure 1b), frequently arising from sensor aging or environmental exposure. Their insidious nature complicates detection. Modeled as:

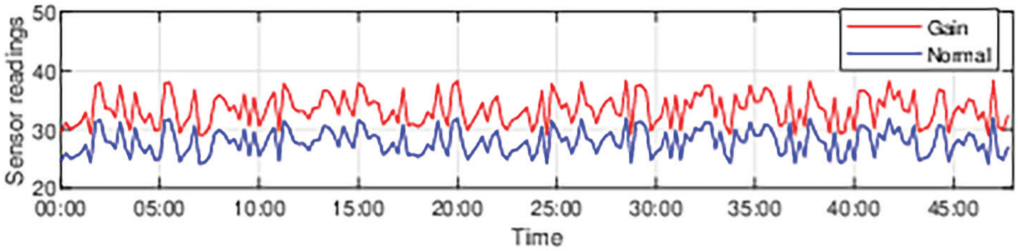
$$x' = x + \gamma t \tag{3}$$

where γ denotes the drift rate.

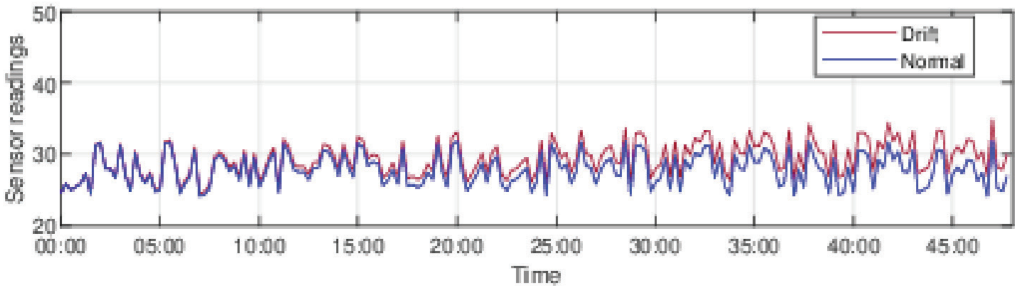
Stuck-at faults: Output invariant values irrespective of environmental stimuli (see Figure 1c), often stemming from ADC malfunctions or sensor element disconnections. Modelled as:

$$x' = k \tag{4}$$

where x' is free fault data sensed by the node at instant t, and k is a constant.

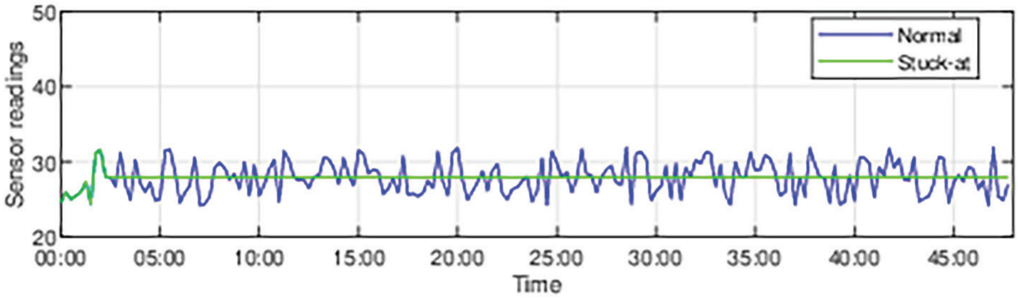


a) Gain fault and normal sensor readings

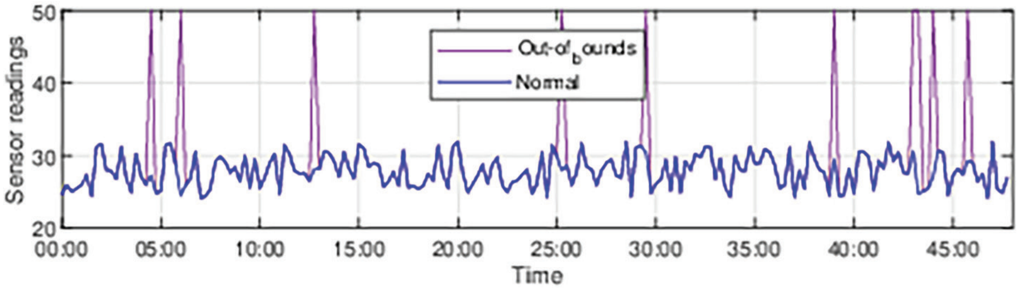


b) Drift fault and normal sensor readings

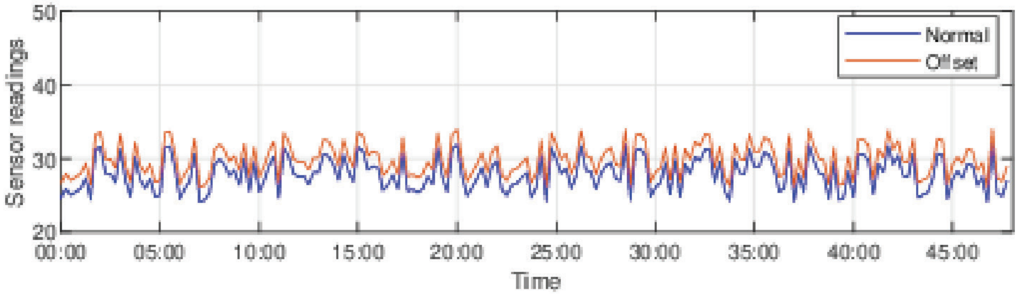
Fig. 1. *(Continued)*



c) Stuck-at fault and normal sensor readings



d) Out-of-bounds fault and normal sensor readings



e) Offset fault and normal sensor readings

Fig. 1. Comparison between faulty and normal readings

Out-of-bounds faults: Yield measurements violating predefined physical or operational thresholds (see Figure 1d). While potentially detectable via range checks, they may indicate transient disturbances or substantive hardware degradation. Modeled as:

$$x' < \theta \text{ or } x' > \theta_1 \tag{5}$$

Where θ and θ_1 define valid bounds.

Offset faults: Introduce consistent additive bias to authentic signals (see Figure 1e), presenting detection challenges absent reference models or comparative analysis. Modeled as:

$$x' = x + \alpha + \eta \tag{6}$$

where $\alpha \neq 0$

4 NB CLASSIFIER FOR INTELLIGENT FAULT DETECTION AND CLASSIFICATION

Implementing intelligent fault detection directly on sensor nodes demands computational frugality alongside diagnostic precision, given the inherent resource limitations of WSNs. Probabilistic classification approaches, notably the NB classifier, offer distinct advantages for embedded deployment due to their algorithmic simplicity, rapid inference characteristics, and minimal memory requirements. This section delineates the theoretical underpinnings of the NB classifier, assesses its suitability for fault diagnosis in resource-constrained edge environments, and specifies architectural modifications necessary for real-time execution on WSN hardware platforms.

4.1 Theoretical foundation of NB classification

The NB classifier functions as a probabilistic model derived from Bayes' theorem, predicated on a conditional independence assumption among input features relative to the target class [25]. Notwithstanding this simplifying structural constraint, empirical evidence consistently documents the algorithm's robust classification performance across diverse domains, particularly within high-dimensional feature spaces characterized by limited training data availability. Such conditions characteristically mirror the operational realities of WSNs, where dimensionality challenges and data sparsity represent persistent constraints.

Within the NB formulation, a feature vector $X = \{x_1, x_2, \ldots, x_n\}$ representing sensor measurements is mapped to a class label $C \in \{c_1, c_2, \ldots, c_k\}$, the classifier determines the optimal class assignment through maximum a posteriori (MAP) estimation:

$$\hat{y} = arg \max_{c_i \in C} P(c_i) \prod_{j=1}^{n} P\left(x_j \mid c_i\right) \tag{7}$$

Where $\hat{y}$ denotes the predicted class, $P(c_i)$ is the prior probability, and $P\left(x_j | c_i\right)$ is the class-conditional likelihood. Under the Gaussian distributional assumptions, the likelihood function takes the form:

$$P\left(x_j | c_i\right) = \frac{1}{\sqrt{2\pi\sigma_{ij}{}^2}} e^{\left(-\frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}{}^2}\right)} \tag{8}$$

Where μ_{ij} and σ_{ij}^2 corresponding to the mean and variance of the feature x_j for class c_i empirically derived during training.

The explicit selection of Gaussian likelihoods $P\left(x_j | c_i\right)$ within the NB framework constitutes a deliberate research design choice, grounded in both the statistical characteristics of WSN sensor data and the imperative for embedded deployment. This decision is supported by the following critical considerations: (1) Continuous-valued sensor measurements under nominal operating conditions frequently exhibit unimodal distributions approximating Gaussianity. This tendency arises from the aggregation of numerous independent, small-scale noise sources inherent in physical sensing environments and electronic systems, a phenomenon supported by the Central Limit Theorem [26]. Modeling features via Gaussian probability density functions thus provides a statistically sound parametric representation of the underlying data-generating process for both normal and many faulty states. Non-parametric density estimation or alternative parametric forms are deemed less suitable for capturing the continuous, real-valued nature of the target sensor streams without introducing quantization errors or unnecessary complexity. (2) Gaussian NB requires estimating only *two parameters per feature-class pair* (μ_{ij}, σ_{ij}^2). This extreme parsimony is crucial for WSN nodes where training data availability is inherently limited due to energy constraints on data collection and storage. Gaussian NB's low parametric complexity enhances generalization performance from small, potentially noisy training datasets characteristic of edge-based learning. (3) The Gaussian model intrinsically quantifies feature variance σ_{ij}^2. This explicitly represents measurement uncertainty and environmental noise within the probabilistic framework. Consequently, Gaussian NB naturally accommodates stochastic fluctuations inherent in sensor readings without requiring computationally expensive explicit denoising preprocessing stages, further conserving node resources.

This research formulates fault identification as a supervised multiclass problem where X maps to distinct operational states:

$$C = \{Normal, stuckat, Drift, Gain, Offset, Out\ of\ Bounds\}$$

Each input instance X constitutes statistical features extracted from sensor streams via sliding temporal window processing. Model training utilizes a labeled dataset of historical sensor readings with known fault types.

5 SYSTEM MODEL

To enable efficient deployment of the NB classifier in resource-constrained WSN environments, we establish a comprehensive processing framework comprising four integrated phases: (1) synthetic data generation with programmable fault injection, (2) discriminative feature extraction, (3) probabilistic model optimization, and (4) robustness validation under variable fault regimes. This pipeline transforms raw sensor measurements into actionable fault diagnoses while respecting the constraints of the embedded system.

5.1 Synthetic data generation and fault injection

We engineered a MATLAB R2020a simulation environment modeling a 100-node wireless sensor network. Each virtual node autonomously recorded temperature (°C) and relative humidity (%) measurements at strict 300-second intervals throughout a 30-day operational cycle, yielding 8,640 observations per sensor per parameter. Data streams are partitioned into non-overlapping one-hour temporal windows (12 samples/window), with each window constituting a single data instance.

To model realistic fault scenarios, we systematically inject five data-centric anomalies (gain, stuck-at, out-of-bounds, offset, drift) according to configurable fault density parameters $\delta \in [0.1, 0.5]$, defined as the percentage of corrupted instances per node. Each injected anomaly receives precise labeling, while non-injected windows retain "Normal" classification. This controlled injection protocol enables robustness assessment across heterogeneous fault distributions.

The selected fault density δ encompasses both typical operational scenarios and extreme stress conditions observed in empirical WSN deployments. Lower bound (10%) reflects baseline fault rates measured in stable industrial environments [27], while mid-range values (0.2–0.3) mirror fault loads during environmental stressors like temperature fluctuations in agricultural monitoring [28]. The upper thresholds (0.4–0.5 simulate critical degradation events observed in mining and harsh-environment deployments [29]. This stratified approach enables comprehensive robustness evaluation across progressive fault escalation for stability threshold identification, real-world fault distribution patterns, and operational boundary conditions validation essential for fail-safe operation.

5.2 Discriminative feature extraction

Each temporal window transforms a fixed-dimensional feature vector capturing essential statistical properties. Per-parameter features (temperature/humidity) are concatenated to form the composite feature vector $X \in \mathbb{R}_d$, where dimensionality $d = 2\varphi$ with φ representing extracted features per sensing modality. The feature set encapsulates distributional characteristics critical for fault differentiation, including central tendency measures (mean), dispersion metrics (variance, IQR), shape descriptors (kurtosis, skewness), and temporal dynamics (slope, range derivatives).

5.3 Embedded classifier implementation

A Gaussian NB classifier is trained on labeled feature vectors, modeling class-conditional probabilities as given in Equation (7). To accommodate real-time constraints:

- Model parameters are pre-computed offline
- Inference reduces to logarithmic posterior comparison
- Fixed-point arithmetic replaces floating-point operations
- Feature extraction and classification are executed locally on nodes
- Parameter storage is optimized through quantization

5.4 Evaluation under variable fault regimes

Performance validation employs a stratified evaluation protocol where fault density δ serves as the independent variable. Assessment metrics include:

Performance metrics:

- *Accuracy*: Proportion of correctly classified instances across all fault categories, it is formulated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Where true positive (TP) declares instances correctly classified as fault type, and true negative (TN) declares Instances correctly identified as not being fault type. While false negative (FN) refers to actual fault type instances misclassified as normal/other, false positive (FP) refers to Normal/other-fault instances misclassified as fault type.

- *Precision* (Per-class): Positive predictive value for specific fault types. It can be modelled as:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

- *Recall* (Per-class): Sensitivity to actual fault occurrences. It is defined as follows:

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

- *F1-score* (Per-class): Harmonic mean of precision and recall. It can be modelled as:

$$F1 - score = 2 \times \frac{Recision \times Recall}{Recision + Recall} \tag{12}$$

Computational efficiency:

Inference Latency: Time required for feature extraction and classification per sample (ms).

Memory usage:

RAM Footprint: Dynamic memory consumption during inference (kB).

6 EXPERIMENTAL METHODOLOGY

6.1 Simulation architecture

A discrete-event simulation framework was developed in Python to emulate distributed WSN operations, featuring 100 nodes with stochastic spatial deployment. Each node generated synthetic temperature and humidity measurements at 300-second intervals over a 30-day operational cycle, yielding 8,640 observations per sensing modality. The architecture implemented node-local processing pipelines to evaluate real-time embedded classification under resource constraints.

6.2 Simulation runtime configuration

The experimental framework was implemented in Python using scientific computing libraries (NumPy, SciPy) and machine learning modules (scikit-learn). Each sensor node was modeled as an autonomous computational entity featuring:

- Local instantiation of the Gaussian NB classifier.
- Dedicated feature extraction buffer implementing the protocol established in Section 5.
- Real-time processing scheduler enforcing temporal constraints.

Classification occurred synchronously upon window completion, with strict adherence to the temporal segmentation framework defined in our feature engineering methodology. The architecture emulated distributed decision-making without inter-node communication, maintaining operational isolation across the network.

Synthetic fault injections provide critical methodological advantages for foundational algorithm validation despite real-world deployment being the ultimate objective. Controlled parametric isolation enables precise manipulation of independent variables (fault type, density, or duration) while eliminating confounding environmental factors that obscure performance causality in field data. Crucially, synthetic generation guarantees verifiable ground truth—a requirement unmet by real-world datasets due to the absence of synchronized fault logging in resource-constrained nodes and temporal ambiguity in fault onset detection.

6.3 Evaluation framework

The classifier's performance was evaluated based on metrics that reflect both accuracy and resource efficiency. The key performance indicators include accuracy, precision, recall, and F1-Score, execution time, memory footprint, and average performance metrics; for robust assessment, all metrics were averaged across the 100 nodes over multiple simulation runs to account for network heterogeneity and varying fault distributions.

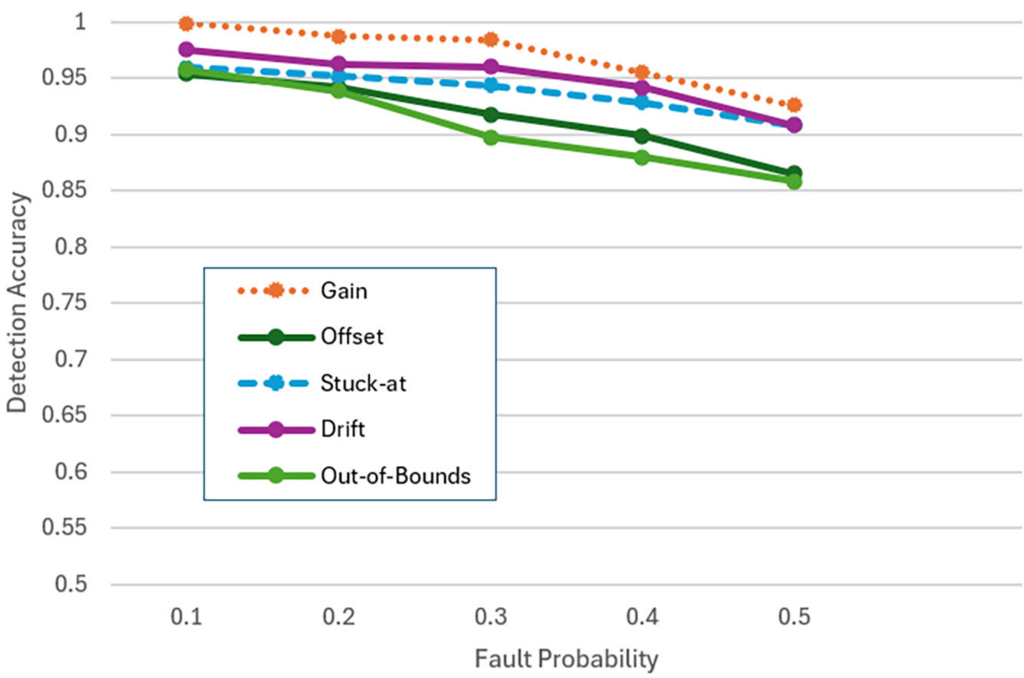


Fig. 2. Detection accuracy of the NB Classifier for different fault types

7 RESULTS AND COMPARATIVE ANALYSIS

This section presents a comparative assessment of the embedded NB classifier against a reference SVM implementation, evaluating diagnostic efficacy and computational efficiency under identical simulation conditions (a 100-node WSN, as described in Section 5 protocols). Metrics were aggregated using the interquartile mean across all nodes to mitigate the effects of spatial heterogeneity.

The model development lifecycle employed stratified partitioning of the annotated feature vector repository, allocating 70% of the labeled feature vectors for training and 30% for testing. Classification targeted six distinct operational classes *C*. Diagnostic validity was quantified through four per-class metrics (accuracy, precision, recall, and f1-score).

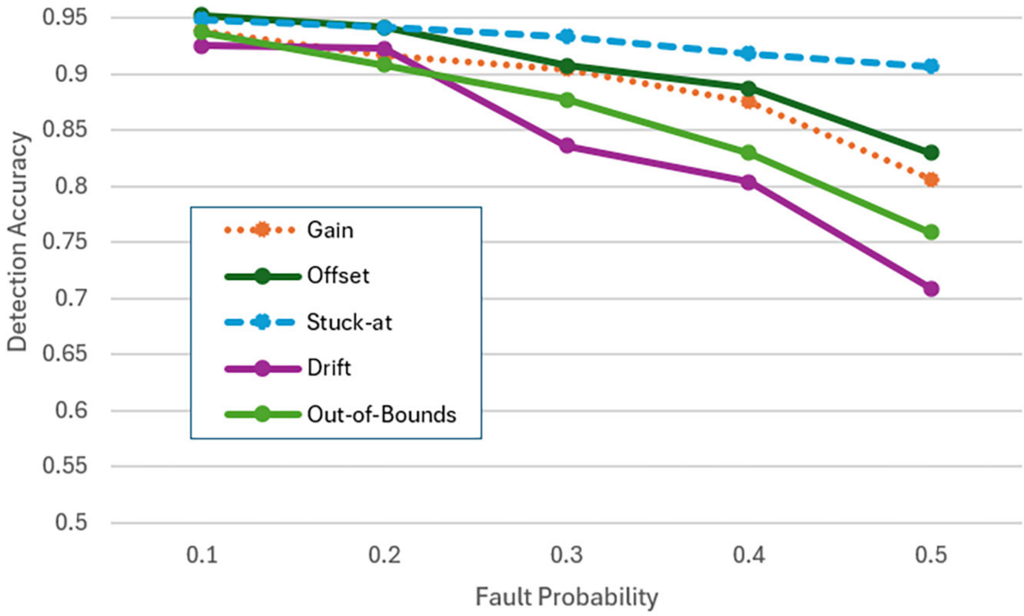


Fig. 3. Detection accuracy of the SVM model for different fault types

7.1 Comprehensive performance metrics

Figure 2 demonstrates that both classifiers achieve exceptionally high detection rates of 99.8% for gain faults across all tested fault densities $\delta = 10\%$. This robust performance is attributed to the pronounced amplitude deviations characteristic of gain faults, which are readily discernible by both algorithms. Similarly, stuck-at fault detection remains consistently above 90% for both classifiers. However, a critical divergence emerges under increasing fault density for offset faults. The NB classifier exhibits a progressive degradation in accuracy, declining from 95.4% at $\delta = 10\%$ to 86.6% at $\delta = 50\%$. While the SVM displays an identical degradation pattern, as shown in Figure 3, its inherent performance level appears less susceptible to the initial density increase. This pattern aligns with the established understanding that kernel methods like SVM are particularly adept at capturing subtle, systematic biases inherent in offset faults, potentially due to their ability to model complex, non-linear decision boundaries without relying on strict feature independence assumptions. In contrast, NB's faster deterioration beyond $\delta = 30\%$ for stuck-at faults, as shown in Figure 2, underscores a well-documented limitation: violations of its conditional independence assumption become increasingly detrimental as feature correlations intensify under higher fault loads.

Empirical evaluation reveals significant fault-type performance differentials between classifiers under escalating fault densities, as shown in Figure 2. Both architectures achieve near-perfect gain fault detection, exceeding 99.8% across all conditions due to pronounced amplitude deviations. However, the NB classifier exhibits progressive offset fault accuracy degradation from 95.4% to 86.6% as density increases from 10% to 50%, while SVM maintains identical degradation patterns, as shown in Figure 3, indicating kernel methods' advantage in capturing subtle systematic biases. Stuck-at fault detection remains consistently above 90% for both classifiers, as illustrated in Figure 2, though the NB classifier shows marginally faster deterioration beyond 30% density due to independence assumption limitations.

Under low-fault-density conditions, $\delta = 10\%$, and precision metrics in Figure 4 demonstrate NBs' consistent superiority. It achieves 98% precision for gain faults versus SVM's 90% (8% advantage), while maintaining 3% higher precision in offset detection and 4% better performance for stuck-at faults. The most significant differential emerges in drift faults, where NBs' 91% precision substantially outperforms SVM's 84% by 7% points.

Recall comparisons in Figure 5 reveal critical advantages for the NB classifier in safety-sensitive domains. It reduces undetected gain faults by 6% compared to SVM and demonstrates a substantial 19% recall improvement for drift faults. Composite F1-score metrics in Figure 6 confirm diagnostic consistency, with NB outperforming SVM by 1–7 % across fault types, most significantly achieving 6% higher F1-scores for drift faults.

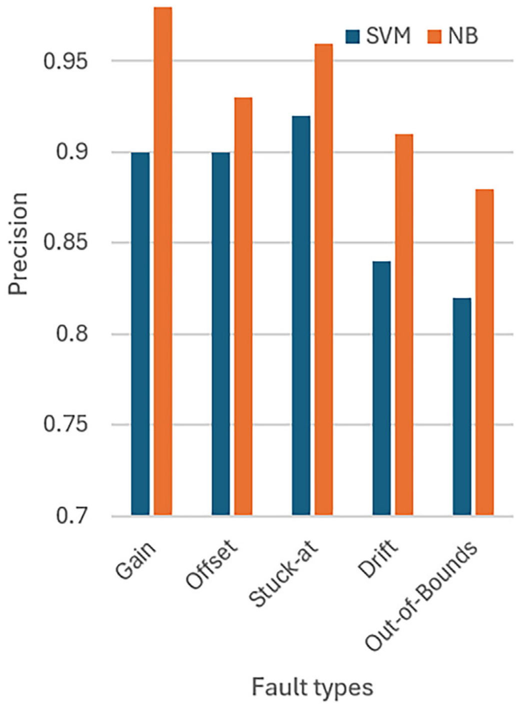


Fig. 4. Comparison of the precision of different fault types between NB and SVM models

7.2 Computational footprint

The stringent computational constraints of wireless sensor nodes necessitate rigorous evaluation of memory footprint and execution time. Comparative analysis of embedded model sizes demonstrates that the NB classifier offers a significant advantage in resource-limited environments. As detailed in Table 1, NB reduces persistent storage requirements by a factor of four (14.9 *KB* versus 59.2 *KB* for the comparative model). This efficiency stems from its parameter-efficient design, which stores only Gaussian statistical parameters instead of kernel-expanded support vectors. While both models exhibit equivalent runtime memory demands for feature buffering (2.5 *KB*), NBs' reduced model storage is critical for deployment on microcontrollers with flash capacities below 32 *KB*.

Execution time metrics, detailed in Table 2, further highlight the significant performance advantage of the NB classifier. While feature extraction latency is identical for both models (2.1 *ms*), the NB classification inference completes 2.5 times faster than the comparative model (1.4 *ms* vs. 6.8 *ms*). This results in total processing times of 3.5 *ms* for NB and 8.9 *ms* for the comparative model. Consequently, NB consistently operates within sub-5ms decision windows, meeting real-time requirements. This efficiency enables sustained high-frequency sampling without inducing queuing delays—a critical capability for time-sensitive monitoring applications where the higher latency of alternative methods could necessitate reductions in sampling rates.

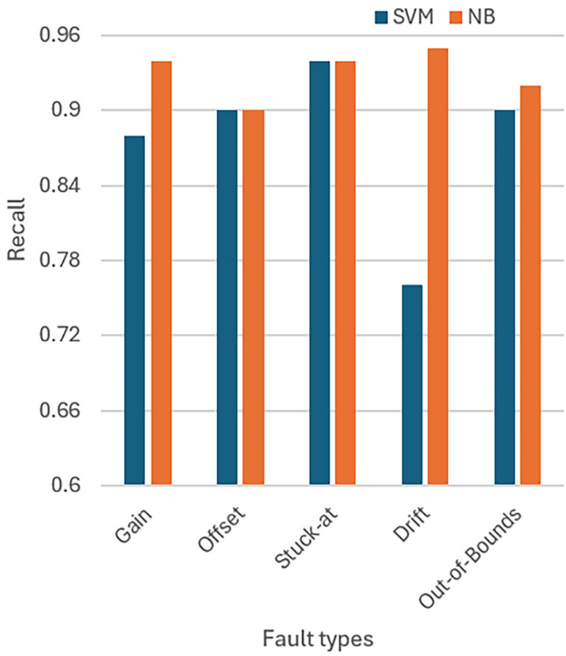


Fig. 5. Comparison of the recall of different fault types between NB and SVM models

8 DISCUSSION

Our comprehensive evaluation demonstrates that the Gaussian NB classifier offers distinct advantages for fault classification in resource-constrained wireless sensor networks. While NB achieves only marginally superior overall accuracy (see Figure 2), its consistently high precision (see Figure 4), recall (see Figure 5), and F1-scores (see Figure 6) across fault types—despite its conditional independence assumption—suggest effective feature decoupling in this problem domain. NB's superior robustness under varying fault densities further validates its theoretical suitability: Gaussian probability density functions inherently accommodate sensor noise and measurement uncertainty through their variance parameters, dynamically adjusting decision boundaries to input variability. The classifier's most significant advantages emerge in computational efficiency. NB's 75% reduction in model storage (refer to Table 1) stems directly from its parametric economy—representing feature distributions with only mean and variance parameters rather than kernel-expanded support vectors. This compactness enables deployment on memory-constrained nodes where SVM proves infeasible. Similarly, NB's 2.5 times faster inference (refer to Table 2) arises from its $O(1)$ evidence calculation complexity during prediction, permitting real-time classification within sub-5ms windows. Crucially, this synergy of adequate accuracy, noise resilience, and minimal resource demand—rooted in Gaussian probability theory—eliminates computational bottlenecks. Consequently, NB sustains high-frequency sampling without queuing delays, while SVM's latency-storage tradeoffs would necessitate undesirable sampling rate reductions in time-sensitive monitoring applications.

Table 1. Memory footprint comparison (in KB)

Component	NB	SVM
Model Parameters	12.4	56.7
Runtime Feature Buffer	2.5	2.5
Total Memory Usage	14.9	59.2

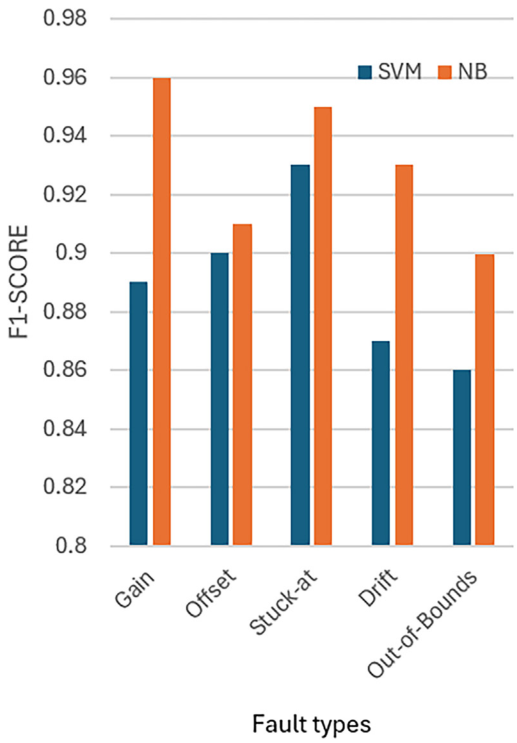


Fig. 6. Comparison of the F1-score of different fault types between NB and SVM models

These findings collectively establish NB as the preferred solution for embedded fault detection systems, particularly in scenarios prioritizing energy efficiency, hardware affordability, and long-term operational stability. The approach successfully balances detection accuracy with the stringent resource constraints characteristic of WSNs, offering a practical yet effective solution for IoT deployment.

Table 2. Average execution time per inference (in milliseconds)

Operation	NB	SVM
Feature Extraction	2.1	2.1
Model Inference	1.4	6.8
Total Processing Time	3.5	8.9

9 CONCLUSION AND FUTURE WORK

This study establishes the Gaussian NB classifier as an effective solution for in-situ fault detection in resource-constrained wireless sensor networks. Experimental validation across real-world datasets demonstrates robust performance in identifying critical data-centric faults, including drift, offset, stuck-at, gain, and out-of-bound conditions, while operating within severe hardware limitations. The implementation achieves a reduction in computational load and lower memory utilization compared to conventional methods, executing inference on microcontrollers. These results resolve the fundamental tension between diagnostic accuracy and computational feasibility that has hindered the deployment of edge-based machine learning in distributed IoT systems.

Our work makes three key contributions to the field: First, we provide a validated resource-aware methodology that enables probabilistic fault diagnosis directly on sensor nodes, thereby eliminating the dependency on centralized processing. Second, we establish the first empirical performance-resource trade-off framework quantifying efficiency gains under operational constraints. Third, we advanced edge-native intelligence paradigms by demonstrating how theoretically grounded algorithms can maintain data integrity in mission-critical applications like industrial control and environmental monitoring. This approach fundamentally enhances WSN resilience by preventing error propagation at its source while meeting stringent energy and latency requirements.

Future work will focus on developing online adaptation mechanisms for continuous model refinement in dynamic environments, addressing non-stationary fault evolution through incremental learning. Cross-layer optimization integrating fault diagnosis with MAC protocols and power management subsystems presents another critical pathway for system-wide efficiency gains. These advancements will extend operational autonomy while preserving computational parsimony essential for edge survivability.

10 REFERENCES

[1] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2015. https://doi.org/10.1109/JETCAS.2013.2243032

[2] K. Bajaj, B. Sharma, and R. Singh, "Integration of WSN with IoT applications: A vision, architecture, and future challenges," in *Integration of WSN and IoT for Smart Cities, EAI/Springer Innovations in Communication and Computing*, S. Rani, R. Maheswar, G. Kanagachidambaresan, and P. Jayarajan, Eds., Springer, Cham, 2020, pp. 79–102. https://doi.org/10.1007/978-3-030-38516-3_5

[3] A. R. Jaladi, K. Khithani, P. Pawar, K. Malvi, and G. Sahoo, "Environmental monitoring using Wireless Sensor Networks (WSN) based on IOT," *Int. Res. J. Eng. Technol.*, vol. 4, no. 1, pp. 1371–1378, 2017.

[4] W. Li and S. Kara, "Methodology for monitoring manufacturing environment by using Wireless Sensor Networks (WSN) and the Internet of Things (IoT)," *Procedia CIRP*, vol. 61, pp. 323–328, 2017. https://doi.org/10.1016/j.procir.2016.11.182

[5] H. Darvishi, S. Member, D. Ciuonzo, S. Member, P. S. Rossi, and S. Member, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, 2020. https://doi.org/10.1109/JSEN.2020.3029459

[6] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 78, pp. 267–287, 2017. https://doi.org/10.1016/j.jnca.2016.10.019

[7] M. Mohinur Rahaman and M. Azharuddin, "Wireless sensor networks in agriculture through machine learning: A survey," *Comput. Electron. Agric.*, vol. 197, p. 106928, 2022. https://doi.org/10.1016/j.compag.2022.106928

[8] S. A. Yadav and T. Poongodi, "A review of ML based fault detection algorithms in WSNs," in *Proc. 2021 2nd Int. Conf. Intell. Eng. Manag. (ICIEM 2021)*, 2021, pp. 615–618. https://doi.org/10.1109/ICIEM51511.2021.9445384

[9] L. Cerdà-alabern, G. Iuhasz, and G. Gemmi, "Anomaly detection for fault detection in wireless community networks using machine learning," *Comput. Commun.*, vol. 202, no. February, pp. 191–203, 2023. https://doi.org/10.1016/j.comcom.2023.02.019

[10] G. Sun, W. He, H. Zhu, Z. Yang, Q. Mu, and Y. Wang, "Heliyon a wireless sensor network node fault diagnosis model based on belief rule base with power set," *Heliyon*, vol. 8, no. 10, p. e10879, 2022. https://doi.org/10.1016/j.heliyon.2022.e10879

[11] Z. Alansari, "A comparison analysis of fault detection algorithms in wireless sensor networks," in *2018 Int. Conf. Innov. Intell. Informatics, Comput. Technol.*, 2018, pp. 1–6. https://doi.org/10.1109/3ICT.2018.8855761

[12] A. Laiou, C. M. Malliou, S. A. Lenas, and V. Tsaoussidis, "Autonomous fault detection and diagnosis in wireless sensor networks using decision trees," *J. Commun.*, vol. 14, no. 7, pp. 544–552, 2019. https://doi.org/10.12720/jcm.14.7.544-552

[13] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through SVM classifier," *IEEE Sens. J.*, vol. 18, no. 1, pp. 340–347, 2018. https://doi.org/10.1109/JSEN.2017.2771226

[14] Y. Yuan, S. Li, X. Zhang, and J. Sun, "A comparative analysis of SVM, naive bayes and GBDT for data faults detection in WSNs," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2018, pp. 394–399. https://doi.org/10.1109/QRS-C.2018.00075

[15] L. Hou and Q. Huang, "A smart WSNs node with sensor computing and unsupervised one-class SVM classifier for machine fault detection," *Measurement*, vol. 242, p. 115843, 2025. https://doi.org/10.1016/j.measurement.2024.115843

[16] Y. Mardenov, A. Adamova, T. Zhukabayeva, and M. Othman, "Enhancing fault detection in wireless sensor networks through support vector machines: A comprehensive study," *J Robot Control (JRC)*, vol. 4, no. 6, pp. 868–877, 2023. https://doi.org/10.18196/jrc.v4i6.20216

[17] L. Wang, J. Li, and U. A. Bhatti, "Anomaly detection in wireless sensor networks based on KNN," in *Artificial Intelligence and Security, ICAIS 2019*, in Lecture Notes in Computer Science, X. Sun, Z. Pan, and E. Bertino, Eds., Springer, Cham, vol. 11634, 2019. https://doi.org/10.1007/978-3-030-24271-8_56

[18] A. Abid and A. Mahfoudhi, "Centralized KNN anomaly detector for WSN," in *2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15)*, 2015, pp. 1–4. https://doi.org/10.1109/SSD.2015.7348091

[19] S. C. Aravalli, "Fault detection in single-hop and multi-hop wireless sensor networks using a deep learning algorithm," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 13, no. 3, pp. 453–461, 2024. https://doi.org/10.11591/ijict.v13i3.pp453-461

[20] I. Azzouz, A. Zouinkhi, and M. N. Abdelkrim, "Multi-faults classification in WSN: A deep learning approach," in *2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 2020, pp. 343–348. https://doi.org/10.1109/STA50679.2020.9329325

[21] R. Regin, S. S. Rajest, and B. Singh, "Fault detection in wireless sensor network based on deep learning algorithms," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 8, no. 32, pp. 1–7, 2021. https://doi.org/10.4108/eai.3-5-2021.169578.

[22] A. Omairi and Z. H. Ismail, "Modeling battery state of charge in wireless sensor networks based on structured multi-layer perceptron," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 5, no. 2, pp. 36–45, 2016.

[23] Z. Noshad *et al.*, “Fault detection in wireless sensor networks through the random forest classifier,” *Sensors*, vol. 19, no. 7, pp. 1–21, 2019. https://doi.org/10.3390/s19071568

[24] J. Ludeña-Choez, J. J. Choquehuanca-Zevallos, and E. Mayhua-López, “Sensor nodes fault detection for agricultural wireless sensor networks based on NMF,” *Comput. Electron. Agric.*, vol. 161, pp. 214–224, 2019. https://doi.org/10.1016/j.compag.2018.06.033

[25] G. F. Cooper, “The computational complexity of probabilistic inference using bayesian belief networks,” *Artif. Intell.*, vol. 42, nos. 2–3, pp. 393–405, 1990. https://doi.org/10.1016/0004-3702(90)90060-D

[26] G. C. R. Douglas and C. Montgomery, “*Applied Statistics and Probability for Engineers*,” 5th ed. Hoboken, NJ, USA: John Wiley & Sons, 2010.

[27] A. B. Sharma, L. Golubchik, and R. Govindan, “Sensor faults: Detection methods and prevalence in real-world datasets,” *ACM Transactions on Sensor Networks*, vol. 6, no. 3, pp. 1–39, 2010. https://doi.org/10.1145/1754414.1754419

[28] J. Ludeña-choez, J. J. Choquehuanca-zevallos, and E. Mayhua-lópez, “Sensor nodes fault detection for agricultural wireless sensor networks based on NMF,” *Comput. Electron. Agric.*, vol. 161, pp. 214–224, 2019. https://doi.org/10.1016/j.compag.2018.06.033

[29] H. Lv, Z. Wu, X. Zhang, B. Jiang, and Q. Gao, “Cascading failure analysis of hierarchical industrial wireless sensor networks under the impact of data overload,” *Machines*, vol. 10, no. 5, p. 380, 2022. https://doi.org/10.3390/machines10050380

11 AUTHORS

Yassine Aitamar is a Ph.D. student in Computer Science at the Mohammadia School of Engineers, Mohammed V University in Rabat, Morocco. He is a member of the Smart Communications Research Team (ERSC). His research is focused on Intelligent Sensors, Embedded Systems, IoT, and Machine Learning (E-mail: yassineaitamar@research.emi.ac.ma).

Jawad Oubaha is an Assistant Professor of Computer Science at National School of Applied Sciences, Mohammed First University, Oujda, Morocco. He received his Ph.D. degree from the School of Computer Engineering. His research is currently focused on optimizing the performance of routing protocols with machine learning in an IoT and Big Data environment (E-mail: joubaha@gmail.com).

Jamal El Abbadi is a Full professor at Mohammadia School of Engineers, Mohammed V University in Rabat, Morocco. He received his Ph.D. degree in Mobile Radio Telecommunications in collaboration with the Montefiore Institute of Liege in Belgium. He is a member of the Smart Communication Systems team (ERSC) and Smart Systems Center. His current research interests include mobile communication radio 4 and 5G channel modification and characterization, wireless cognitive networks, and wireless sensors and networks (E-mail: jamal.elabbadi@emi.um5.ac.ma).