MJI
Media Jurnal Informatika
Vol. 17, No. 2, December 2025
p-issn : 2088-2114, e-issn : 2477-2542

Multi-Class Fault Detection under Class-Imbalance in Wireless Sensor Network Using Random Undersampling and Extra Trees

David Yusup Saputra[a,1], Luh Kesuma Wardhani[a,2,*], Herlino Nanang[a,3]

[a] Department of Informatics, UIN Syarif Hidayatullah Jakarta, Kota Tangerang Selatan 15412, Indonesia
[1] davidsaputra20@mhs.uinjkt.ac.id; [2] luhkesuma@uinjkt.ac.id; [3] herlino.nanang@uinjkt.ac.id
[*] corresponding author

ARTICLE INFO

Article history
Received 2025/12/05
Revised 2025/12/16
Accepted 2025/12/25

Keywords
Class imbalance
Extra Tree
Fault detection
Multi-class classification
Random Undersampling

ABSTRACT

Wireless Sensor Networks (WSN) are widely used in various monitoring applications, including environmental observation, smart infrastructure, and Internet of Things (IoT) systems. Despite their widespread adoption, WSN are highly susceptible to data errors caused by sensor degradation, hardware malfunctions, environmental disturbances, and communication issues. These faults can significantly reduce data reliability and lead to incorrect system decisions if not properly handled. This study proposes a multi-class data-fault detection approach for WSN under imbalanced data conditions by integrating Random Undersampling (RUS) with the Extra Trees classification algorithm. This study aims to address the class imbalance problem commonly found in sensor fault datasets while improving fault detection performance across multiple fault types. Experiments were conducted using a WSN dataset containing temperature and humidity measurements with combined with three artificial fault types: Bias, Drift, and Spike were analyzed alongside normal sensor data. The experimental results demonstrate that RUS leads to a substantial improvement in classification performance. Without RUS, the Extra Trees classifier achieved an accuracy of 48% and failed to detect spike faults. After applying RUS, classification accuracy increased to 91%, accompanied by balanced precision, recall, and F1-score values across all classes. These findings indicate that the combination of RUS and Extra Trees provides an effective and reliable solution for multi-class data fault detection in WSN environments. Regardless, the detection of drift faults remains more challenging than other fault types due to their gradual nature and overlap with normal sensor behavior, indicating the need for further improvement in future studies.

1. Introduction

Wireless Sensor Networks have been used for modern monitoring systems, including environmental monitoring, industrial automation, smart home, and Internet of Things (IoT) applications. They enabled distributed, real-time data acquisition across wide, often hard-to-access areas, supporting continuous observation and decision support in operational settings [1]. However, the reliability of WSN based systems depends on the quality of sensor data, as misclassification or delayed detection of sensor faults may lead to incorrect control actions, increased operational costs, or safety-related risks, underscoring the need for reliable fault detection mechanisms in WSN [2].

In practical deployments, sensor measurements are often affected by data faults from hardware degradation, calibration issues, environmental interference, and communication disruptions. These issues can affect temperature and humidity readings, reducing the reliability of monitoring results. It also may lead to incorrect decisions if faults are not handled appropriately [3], [4]. As a result, fault detection becomes a requirement for maintaining dependable WSN based monitoring.

Early fault-detection methods in WSN mostly relied on threshold based rules or binary anomaly detection to distinguish normal from abnormal measurements. Although simple computations and binary decisions are often sufficient in simple WSN settings, they are often insufficient in complex WSN settings because faults may exhibit distinct signatures and require distinct mitigation strategies [5]. For this reason, multi-class fault detection has gained attention, as it can provide more informative diagnoses by distinguishing specific fault types, such as bias, drift, and spikes, rather than flagging abnormality [6]. Many supervised classification approaches have been implemented for WSN fault detection and have shown competitive performance compared with rule-based techniques [3], [5].

Although several studies have explored multi-class fault detection in WSN, most existing work uses balanced datasets or focuses on general accuracy metrics. As a result, the impact of class imbalance on minority and transient fault classes, such as spike faults, is often less analyzed. In fault detection of practical WSN deployments, normal readings dominate the data stream and faults appear rarely. The imbalance between normal and faulty data can bias a model toward the majority class and reduce detection performance for minority faults that may be operationally critical [7]–[9]. Failure to detect minority fault classes may lead to delayed maintenance actions, increased operational costs, and safety risks. Therefore, addressing class imbalance is not only a matter of classification performance but also a crucial factor in minimizing cost impact and safety consequences. [10], [11].

To address this limitations, this study focuses on analyzing class-wise performance under imbalanced data conditions, analyzes how the data imbalance affects error distribution of confusion matrix level, and examines the sensitivity of an ensemble based classifier to minority and transient fault types. Data-level resampling is a common strategy to mitigate imbalance. In particular, Random Undersampling (RUS) reduces the size of the majority class to create a more balanced class distribution. Although undersampling may discard part of the majority data, it is often effective as a simple preprocessing step for imbalanced classification, especially when the model is otherwise dominated by the majority class [12], [13]. Prior work also highlights that undersampling can be integrated with other selection strategies and can improve learning under class imbalance in practical settings [14], while the behavior of under and oversampling under noise and imbalance remains an important consideration for classification performance [8]. Unlike prior studies that extend fault detection by increasing the number of fault classes, this work introduces additional methodological and analytical depth by systematically examining class-wise performance, confusion-matrix–level error redistribution, and classifier sensitivity under imbalanced data conditions.

Besides data balancing, the choice of classifier is important for handling noisy and variable sensor environments. Tree-ensemble methods have been widely used because of their robustness to noise and their ability to turn nonlinear decision boundaries. Extra Trees (Extremely Randomized Trees) is an ensemble of decision trees that can improve generalization and reduce variance in complex datasets [15]. In WSN fault diagnosis, Extra Trees has been reported as an effective approach for identifying multiple fault types under challenging conditions. Extra Trees has been shown support reliable fault detection with relatively low computational cost and favorable residual energy consumption in network environments [16], [17].

Although previous studies have demonstrated the effectiveness of machine learning and ensemble-based classifiers for WSN fault detection, classification performance and data imbalance are often addressed as separate issues. Consequently, the combined impact of RUS and tree-based ensemble classifiers, particularly Extra Trees, on multi-class WSN fault detection has not been sufficiently investigated through systematic and class-level analysis. This study also extends our previous work reported in [18] which focused on binary-class fault detection in WSN sensor data. It demonstrated the effectiveness of machine learning for distinguishing between normal and faulty data, but it did not address multi-class fault classification. In practical WSN deployments, different fault types may show distinct characteristics and require specific handling strategies. Therefore, for improving diagnostic capability in real-world WSN monitoring systems, we need to extend fault detection from binary to multi-class classification.

Based on these motivations, we analyzed imbalance multi-class WSN data for fault detection using the Extra Trees classifier in this study. The study uses a public WSN dataset consisting of temperature and humidity measurements and analyzes three common fault types: bias, drift, spike, and normal data. The contributions of this study are (1) applying RUS to address class imbalance in multi-class WSN fault detection, (2) evaluating Extra Trees for classifying multi-class sensor fault types in imbalanced and balanced settings, and (3) presenting a comparative analysis of performance before and after undersampling using standard classification metrics to clarify class-wise behavior under imbalance. The remainder of this paper presents the methodology, experimental results, discussion, and conclusions with directions for future work.

2. Method

This study employed a quantitative experimental research approach, based on a data-mining framework, to evaluate multi-class fault-detection performance in WSN data. It adopted the CRISP–DM (Cross-Industry Standard Process for Data Mining) framework, comprising five main stages: business understanding, data understanding, data preparation, modeling, and evaluation. This framework was selected because it provides a structured, systematic approach to data-driven classification problems, including fault detection in sensor-based systems. For example, as reported in [19], the CRISP-DM methodology is effective for developing models which can predict potential failures and prevent them.

Based on the CRISP-DM framework, the research procedure in this study began with business understanding, data understanding, and preprocessing, followed by the preparation of two experimental scenarios: classification without RUS and classification with RUS applied to the training data. The Extra Trees classifier was trained and evaluated under both scenarios with the same dataset. For evaluation, the performance was analyzed using evaluation metrics to assess the impact of data imbalance and undersampling on multi-class fault detection performance.

Each CRISP–DM phase in this study is explicitly implemented through the corresponding methodological subsections (Sections 2.1–2.5), ensuring a clear connection between the framework and the research workflow.

2.1 Business Understanding

The business understanding stage focused on identifying the main problem and defining an appropriate solution strategy. The main problem was class imbalance in multi-class fault detection for WSN data, which could significantly affect classification accuracy and reliability.

In real WSN deployments, normal sensor data usually dominate the dataset, while faulty data occur less frequently. Data imbalance can bias machine learning models toward the majority class, leading to poor performance in detecting minority fault classes. To address this issue, RUS was applied as a data-level preprocessing technique to balance the class distribution before classification with the Extra Trees algorithm.

2.2 Data Understanding

The dataset used in this study was obtained as secondary data from a publicly available WSN repository provided by the University of North Carolina at Greensboro (UNC Greensboro) [20]. The dataset has been widely used in fault detection research and contains multihop WSN measurements collected from an outdoor deployment [6], [18].

The dataset consisted of 4,690 records with five attributes: Reading ID, Mote ID, Humidity, Temperature, and Label.

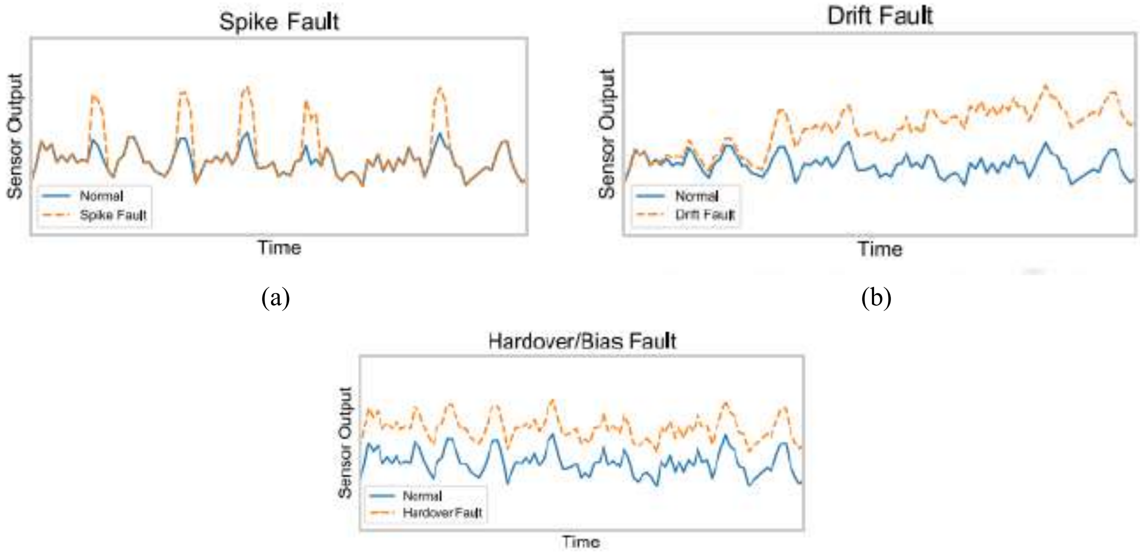
The data were collected from a multihop WSN deployed in an outdoor environme. Among these attributes, only temperature and humidity represent continuous sensor measurements, and the remaining attributes serve as identifiers or metadata. Therefore, this study focused on temperature and humidity readings as the primary sensor variables. They constitute the complete set of physical sensing modalities available in the dataset. Temperature and humidity readings were the primary sensor measurements, while the label attribute indicated the sensor's operational condition.

In its original form, the dataset consists of two categories: normal sensor data and anomalous data. The normal data represent fault-free sensor measurements and were used in this study as baseline data for fault simulation. The anomalous data in the dataset were not used directly for multi-class classification, as the focus of this study was on controlled simulation of specific fault types.

2.3 Data Preparation

The data preparation stage consists of four steps: data cleaning, normalization, fault simulation, and data labeling. In data cleaning the missing values were removed to ensure data consistency and reliability. Irrelevant attributes such as Reading ID and Mote ID were removed. The temperature and humidity were retained as the main features for classification. Feature normalization was applied using the StandardScaler method, which removes the mean and scales to unit variance. This step was necessary to prevent attributes with larger numerical ranges from dominating the classification process.

Fault injection was conducted to generate labeled multi-class fault data that reflects realistic fault conditions in WSN deployments. Since the original dataset only provided binary labels (normal and anomalous), the normal sensor data were used as baseline signals for fault simulation. Three fault types: Bias, Drift, and Spike were injected into the baseline data to represent common sensor faults reported in real-world WSN environments based on formula on [21]. The forms of those three faults are shown in Figure 1.



(a)

(b)

(c)

Figure 1 – Spike, drift and bias fault model [16][21]

A bias fault refers to a constant offset added to sensor measurements over a specific period. This type of fault occurs due to calibration errors or sensor aging. Although the signal pattern remains stable, the measured values deviate consistently from the actual physical conditions, making bias faults easier to detect than other fault types. The formula for generating bias fault is shown in Formula (1).

$$S_n^{bias} = S_n^{normal} + b_n, \qquad b_n = konstan \tag{1}$$

The drift fault is defined by a gradual and continuous change in sensor readings over time. It can be caused by long-term sensor degradation. Unlike bias faults, drift faults do not perform sudden changes; they make it harder to characterize from normal conditions within short observation windows. As a result, drift faults often overlap with normal and bias patterns, increasing the risk of misclassification. The formula for generating drift fault is shown in Formula (2).

$$\begin{gathered} S_n^{drift} = S_n^{normal} + b_n \\ b_n = nb_{0,} \quad b_n = konstanta \end{gathered} \tag{2}$$

The spike fault is defined as a sudden and short-duration deviation from the normal signal. Transient disturbances, such as communication noise or temporary hardware interference, typically cause spike faults. Although spikes have large magnitudes, their brief duration makes them challenging to detect, especially when normal samples dominate the dataset. As a result, spike faults are particularly sensitive to class imbalance issues in multi-class fault detection. The formula for generating spike fault is shown in Formula (3).

$$S_n^{spike} = S_n^{normal} + b_n \qquad (3)$$
$$b_n = \begin{cases} b, & n = v \times \eta, \quad v = (1,2,\dots), \quad \eta = konstanta \\ 0, & otherwise \end{cases}$$

After fault injection, each data instance was labeled with its corresponding fault type, resulting a multi-class dataset consist of normal, bias, drift, and spike classes. This dataset was then used to evaluate the performance of the proposed multi-class fault detection framework under imbalanced and balanced data conditions.

2.4 Modelling

The Extra Trees algorithm was employed as the classifier for multi-class fault detection. Extra Trees was selected for its strong ability to handle noisy and complex data. The algorithm builds an ensemble of decision trees by adding randomness to the split thresholds, so it can reducing overfitting and improving generalization.

Extra Trees was selected over other ensemble-based classifiers, such as Random Forest and Gradient Boosting, for its greater randomization in split selection. This ability improves robustness to noisy sensor data and reduces model variance. Unlike Random Forest, Extra Trees selects split thresholds randomly rather than optimally, which results inbetter generalization in heterogeneous WSN environments. Compared to Gradient Boosting, Extra Trees has lower computational complexity and simpler parameter tuning, making it more suitable for fault detection in resource constrained scenarios.

To examine the effect of class imbalance, the modeling process was conducted under two conditions. First, the Extra Trees classifier was trained using the original dataset with an imbalanced class distribution. Second, RUS was applied to the training data to reduce the dominance of the majority class and to obtain a more balanced class distribution prior to training.

2.5 Evaluation

All experiments were conducted using Python. The dataset was split into training and test sets at an 80:20 ratio. To address class imbalance, RUS was applied to the training data, while the test data remained unchanged to ensure unbiased evaluation. Classification performance was assessed using accuracy, precision, recall, and F1-score, with particular attention to class-wise metrics given the dataset's imbalance. This experimental setup was designed to ensure fair comparison between scenarios with and without RUS, while maintaining reproducibility and transparency in the evaluation process.

3. Results and Discussion

3.1 Experiment without RUS

In the first scenario, the Extra Trees model was trained using the original imbalanced dataset. The dataset exhibited a strong class imbalance, with the Normal class overwhelmingly dominating the fault classes, namely bias, drift, and spike.

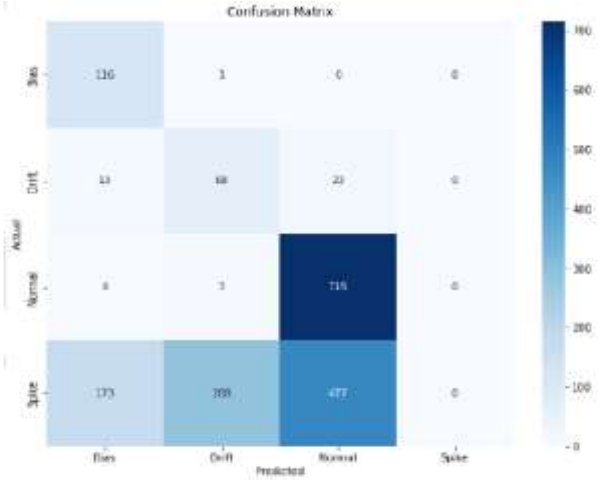


Figure 2 - Confusion matrix as experiment result for Extra Trees fault detection model

Figure 2 presents the confusion matrix of the classification results for four sensor conditions—bias, drift, normal, and spike—using the Extra Trees Classifier without the application of the RUS technique. Overall, the model achieved an accuracy of 47.92%, with 899 correct predictions out of 1,876 test samples. Although this accuracy value appears moderate, a deeper analysis using the confusion matrix and class-wise performance metrics reveals a substantial imbalance across classes driven by the data distribution.

The results show that the normal class achieved the best performance, with a recall of 99.58%, indicating that the model correctly identified almost all normal conditions. A similarly high recall was observed for the bias class, reaching 99.15%. The value indicates that constant-offset sensor deviations could be learned by the Extra Trees classifier. However, the precision for the bias class was only 38.41%, indicate most of bias predictions originated from other classes, particularly spike and drift.

For drift faults, the model performed worse than under bias and normal conditions. The recall of 66.02% indicates that some drift data was successfully detected but precision of 18.89% confirms a high false-positive rate. This result indicates significant overlap in feature characteristics among the drift, bias, and normal classes due to the gradual nature of drift faults, which do not produce extreme changes over short time intervals.

In addition to general accuracy, Macro F1-score is used to evaluate classification performance under class imbalance conditions. Macro F1 is calculated as the unweighted average of F1-scores across all classes, ensuring equal importance for both majority and minority fault classes. Based on the class-wise F1-scores reported in Table 1, the resulting Macro F1-score is 0.395. This result indicates that overall accuracy alone overestimates model performance due to the dominance of the normal class and the poor detection of minority faults, particularly spike faults.

The most critical finding was observed for spike faults. Based on the confusion matrix, none of the spike data were correctly classified (recall of 0%) (Table 1). All spike samples were misclassified as bias, drift, or normal. This condition indicates that the Extra Trees classifier failed to learn the characteristics of spike faults, which are abrupt and very short in duration. Moreover, the value of 0% for the spike class caused the precision value for this class to be undefined.

Table 1 – Extra Trees fault detection model's performance

Class	Precision	Recall	F1-score
Bias	0.38	0.99	0.55
Drift	0.19	0.66	0.29
Normal	0.59	1.00	0.74
Spike	0.00	0.00	0.00

Therefore, these results confirm that global accuracy is insufficient to represent model performance under imbalanced data conditions. The Extra Trees classifier tended to be biased toward the majority class and classes with stable patterns, such as normal and bias, while exhibiting significant failure in detecting transient faults such as spikes. These findings indicate that, without appropriate data imbalance handling strategies the use of Extra Trees for multi-class fault detection may lead to unreliable systems for critical fault detection.

Further examination of the confusion matrix reveals that classification errors in the Without-RUS scenario are primarily concentrated in minority fault classes. Spike faults, which are consistently misclassified as normal, bias, or drift, indicate that the majority class influences the Extra Trees classifier under imbalanced data conditions. This behavior demonstrates that the model fails to learn discriminative patterns associated with transient faults.

High false-positive rates for bias and drift faults indicate feature overlap among fault classes. These findings confirm that global accuracy does not provide a reliable measure in imbalanced multi-class fault detection. Consequently, without appropriate data balancing, the Extra Trees classifier may yield misleading results in global performance and remainineffective for detecting critical minority faults. From these findings, it can be concluded that, under imbalanced data conditions, the Extra Trees classifier tends to favor the majority and stable classes. However, it misleads global accuracy and becomes ineffective at detecting minority and transient faults.

3.2 Experiment with RUS

In the second scenario, RUS was applied before model training. The more balanced dataset allowed the Extra Trees model to learn patterns from each class more effectively. Figure 3 presents the confusion matrix of the classification results for four sensor condition classes: bias, drift, normal, and spike, as well as the performance of the Extra Trees classifier. Overall, the confusion matrix shows that most samples were correctly classified. Of 751 test samples, 680 were correctly predicted, resulting in an overall accuracy of 91%. This value reflects a significant improvement over the scenario without data imbalance handling, particularly for fault classes that are difficult to detect.

The bias class demonstrated strong performance which correctly predicting 200 out of 208 samples. Misclassifications in this class were quite limited and mainly were confused with the drift class. This result indicates that constant offset–type changes in sensor values can be effectively learned by the Extra Trees classifier, even after undersampling.

For the normal class, the model also gives high stability. A total of 178 out of 184 normal samples were correctly classified, with minimal misclassification into other classes. The low false-alarm rate under normal conditions is an important indicator for fault detection systems, as errors in this class can directly affect the reliability of field monitoring systems.

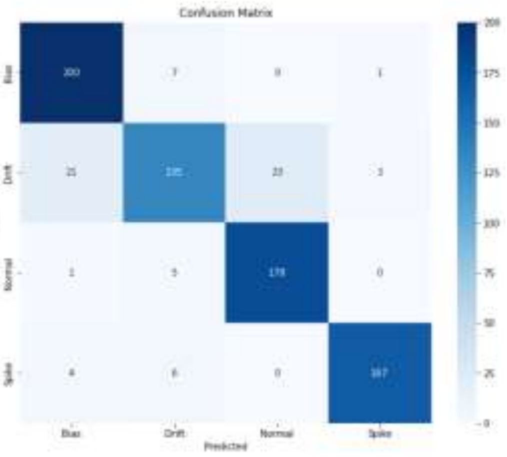


Figure 3 - Confusion matrix as experiment result for using RUS

Table 2 – Fault detection model's performance : Extra Trees and RUS

Class	Precision	Recall	F1-score
Bias	0.88	0.96	0.92
Drift	0.88	0.74	0.81
Normal	0.89	0.97	0.92
Spike	0.98	0.94	0.96

The spike class showed a very significant performance improvement compared to the scenario without undersampling. A total of 167 out of 177 spike samples were successfully detected. This result demonstrates that RUS played an important role in increasing the model's sensitivity to abrupt, short-duration transient faults. With a more balanced class distribution, spike characteristics became more prominent during the decision tree construction process in the Extra Trees algorithm.

In contrast, the drift class remained the most challenging to classify. Although 135 out of 182 drift samples were correctly identified, a portion of the drift data was still misclassified as normal (23 samples) and bias (21 samples). This error pattern reflects the gradual nature of drift faults, in which the signal changes slowly and remains close to normal conditions or stable offset levels in some segments. This phenomenon is consistent with the reported characteristics of drift faults.

As shown at Table 2, The drift class obtained an F1-score of 0.81, which was lower than that of the other classes due to a reduced recall value (0.74). This result confirms that although RUS improved class balance and overall model performance, drift characteristics remain the main challenge due to their close similarity to normal and bias patterns.

RUS leads to a change in the distribution of classification errors among fault classes. Compared to the Without RUS scenario, the number of false negatives for spike faults is mostly reduced, showing improved sensitivity to spike faults. This result indicates that balancing the training data enables the Extra Trees classifier to capture better classes with specific patterns that were previously dominated by the majority class.

Nevertheless, some misclassifications persist between drift and normal conditions, reflecting the gradual nature of drift faults and their similarity to normal sensor behavior over short observation windows. Despite this limitation, the overall reduction in error rates across minority classes confirms that RUS is a key factor for improving the system's robustness and reliability in multi-class fault detection using the Extra Trees classifier. The Extra Trees classifier learn specific patterns for all fault types, particularly minority and transient faults.

3.3 Discussion

This study evaluated the performance of the Extra Trees classifier for multi-class fault detection in WSN data under two experimental configurations: Without-RUS and with RUS. As shown in

Figure **4** the application of RUS significantly improves classification performance, particularly for spike and drift faults, as indicated by increases in recall and F1-score. In the Without-RUS scenario, the classifier achieved an overall accuracy of 48% with a macro-averaged F1-score of 0.40. Although high recall values were gained for the bias and normal classes, the model failed to detect any spike faults. Similar limitations have been reported in recent WSN and sensor fault detection studies, where models trained on imbalanced data resulted dominant operating states and overlook rare but critical faults. These studies emphasize that accuracy alone is an insufficient metric for evaluating fault detection performance in imbalanced sensor data scenarios.

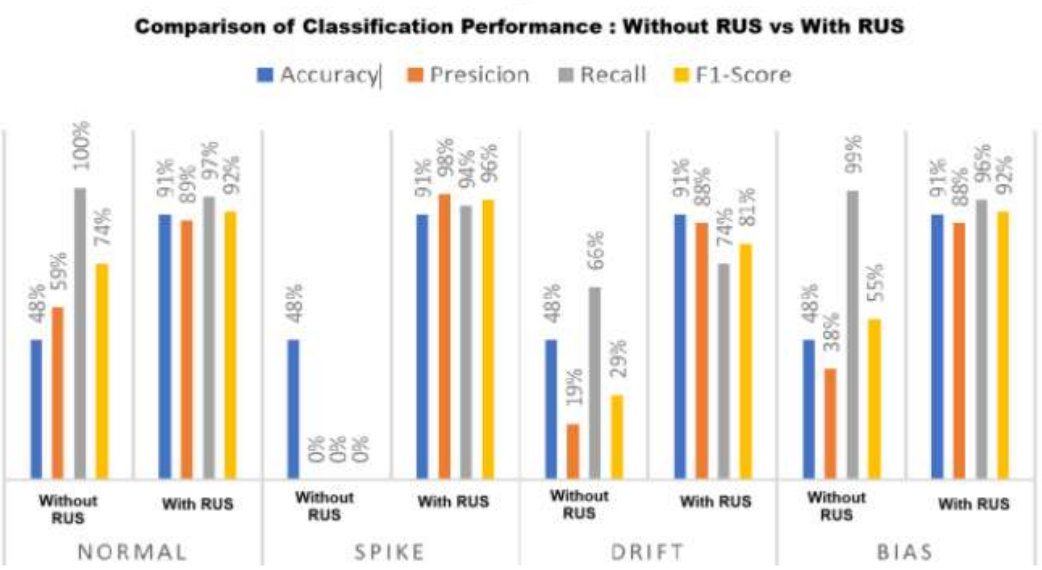


Figure 4 - Comparison of accuracy, precision, recall, and F1-score obtained by the Extra Trees classifier with and without Random Undersampling (RUS) for different fault classes.

When RUS was applied, classification performance improved and achieving 91% accuracy and a weighted F1-Score of 0.90. This result is consistent with recent findings showing that balancing strategies significantly enhance the performance of ensemble-based classifiers, particularly decision-tree ensembles, in imbalanced learning problems [22], [23]. By balancing class representation, the Extra Trees model resulted in more accurate predictions across all fault categories.

From an application perspective, the observed performance degradation on minority and transient fault classes has important operational implications. In fault-critical WSN application domains such as industrial monitoring, automotive sensing, and healthcare IoT systems, missed detection of transient faults (e.g., spike faults) may lead to incorrect control actions, delayed system response, or unstable system behavior. Prior studies have reported that misclassification in safety-related sensing systems can significantly increase operational risks and system failures [22], [23]. These findings indicate that, without appropriate data imbalance handling strategies, the use of ensemble classifiers such as Extra Trees may yield high overall accuracy while failing to reliably detect transient fault events which potentially compromise system reliability in real world deployments.

The class-wise performance observed in this study is quantitatively consistent with trends reported in recent literature. Previous studies have shown that classification models trained on imbalanced WSN datasets tend to achieve high recall for dominant normal classes, lower precision and poor detection of minority fault classes due to class imbalance [24]. This behavior is similar to our results, whereas the application of RUS leads to a more balanced precision–recall trade-off across fault classes, indicating improved discrimination capability.

The significant result is the improvement for transient spike faults. Without class balancing, spike faults are completely missed, whereas applying RUS makes the model to achieve high precision (0.98) and recall (0.94). This outcome is consistent with prior findings that transient or abrupt faults are particularly difficult to detect unless minority classes are adequately represented during training [6]. These results confirm that appropriate imbalance handling is critical for enabling ensemble-based classifiers to learn short-duration fault patterns effectively.

Despite these improvements, drift faults are still the most challenging class, even after undersampling. Although the F1-score increases with RUS, the recall for drift faults stays lower than for other fault types. Similar result has been reported in previous studies, which indicate that gradual faults often overlap with normal operating conditions and are therefore inherently difficult to distinguish using standard classification approaches [25]. These findings show that class balancing is essential for improving overall multi-class fault detection. Still, to further enhance drift-fault detection performance, additional feature-level or temporal modeling strategies may be required.

Overall, the results of this study are consistent with recent studies on WSN fault diagnosis, which show that class imbalance is a significant obstacle to multi-class sensor fault detection. By integrating RUS with the Extra Trees classifier, this study achieves balanced and robust performance across all fault types, particularly for minority and transient faults. It is also suitable for practical WSN deployments,where class imbalance is essential for achieving reliable multi-class fault detection in WSN environments.

3.4 Implications of the Results

This study resulted in several important practical and theoretical implications for the development of fault-detection systems in the WSN. From a practical perspective, the combination of RUS and the Extra Trees classifier provides an effective solution for improving multi-class fault detection performance under imbalanced data conditions. This approach can be applied to WSN-based monitoring systems in domains such as environmental monitoring, smart agriculture, and industrial sensing, where normal sensor readings dominate and fault occurrences are relatively rare but critical.

The findings also demonstrate clear benefits for system operators and stakeholders. Improved detection of transient and spike faults can enhance system reliability, reduce false alarms, and support maintenance decisions.

From a theoretical standpoint, this study contributes to the existing body of knowledge on WSN fault detection by analyzing the impact of class imbalance in a multi-class classification setting. While previous studies have often focused on binary or anomaly-based detection, this work extends previous research by showing that the balancing techniques influence ensemble-based classifiers at the class level. The class-wise performance analysis presented in this study gives deeper insight into classifier behavior under imbalanced conditions. This aspect has received limited attention in earlier WSN fault-detection research.

In terms of future use and potential expansion, this study can be extended by incorporating additional sensor modalities, temporal feature extraction, or spatial correlation among sensor nodes.

3.5 Limitation of Study

Alongside the promising results, several limitations of this study should be acknowledged. First, the dataset used in this work was derived from a publicly available WSN deployment that contained only two classes. Although fault simulation was applied to generate multi-class fault scenarios, the resulting dataset may not fully capture the diversity and complex fault behaviors encountered in real-world WSN deployments.

Second, from a methodological perspective, the use of RUS introduces an inherent trade-off by discarding a portion of the majority-class samples. While undersampling improves the detection of minority faults, it can also remove informative normal samples and, in some cases, limit model generalization. Drift faults remain particularly difficult to detect because of their gradual changes, which often resemble normal sensor behavior, even after data balancing is applied.

4. Conclusion

This study demonstrated that the classification performance of the Extra Trees algorithm for multi-class fault detection in WSN sensor data was strongly influenced by class distribution imbalance. In the scenario without RUS, the model tended to be biased toward the majority class and failed to detect minority faults, particularly spike faults. The implementation of RUS significantly improved classification performance, achieving 91% accuracy and restored spike fault detection capability with high precision and recall. Nevertheless, drift faults remained the most challenging class due to their gradual nature and overlap with normal operating conditions. Overall, the combination of Extra Trees and RUS proved to be effective and suitable for improving the reliability of WSN-based fault detection systems.

For future work, it is recommended to explore feature representations that are more sensitive to gradual changes to improve drift fault detection, and to compare RUS with other data balancing techniques, such as SMOTE or hybrid methods. In addition, evaluation using real-time field sensor data and analysis of computational complexity should be conducted to ensure that the proposed method can be efficiently implemented in edge-based WSN systems.

Declarations

Author contribution. First author was responsible for literature review, developing model and experiments. Second author was responsible for research concept, methodology and manuscript preparation. Third author was responsible for result analysis and manuscript preparation. All authors contributed to the discussion, reviewed the manuscript, and approved the final version.

Funding statement. This research received no external funding.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The data and software supporting the results of this study are available upon request. The datasets used during the study, including those related to testing results, are available from the corresponding author upon reasonable request.

References

[1] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, “Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios,” *IEEE Access*, vol. 8, pp. 23022–23040, 2020, doi: 10.1109/ACCESS.2020.2970118.

[2] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, “A Survey on Fault Diagnosis in Wireless Sensor Networks,” *IEEE Access*, vol. 6, pp. 11349–11364, 2018, doi: 10.1109/access.2018.2794519.

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

[4] R. Ahmad, R. Wazirali, and T. Abu-Ain, “Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues,” *Sensors*, vol. 22, no. 13, 2022, doi: 10.3390/s22134730.

[5] S. Zidi, T. Moulahi, and B. Alaya, “Fault detection in wireless sensor networks through SVM classifier,” *IEEE Sensors Journal*, vol. 18, no. 1, pp. 340–347, 2018, doi: 10.1109/JSEN.2017.2771226.

[6] U. Saeed, S. U. Jan, Y.-D. Lee, and I. Koo, “Fault diagnosis based on extremely randomized trees in wireless sensor networks,” *Reliability Engineering & System Safety*, vol. 205, no. September 2020, p. 107284, Jan. 2021, doi: 10.1016/j.ress.2020.107284.

[7] R. Mohammed, J. Rawashdeh, and M. Abdullah, “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results,” in *2020 11th International Conference on Information and Communication Systems (ICICS)*, 2020, pp. 243–248, doi: 10.1109/ICICS49469.2020.239556.

[8] P. Kaur and A. Gosain, “Comparing the Behavior of Oversampling and Undersampling Approach of Class Imbalance Learning by Combining Class Imbalance Problem with Noise,” in *ICT Based Innovations*, 2018, pp. 23–30.

[9] T. T. Khuat and M. H. Le, “Evaluation of Sampling-Based Ensembles of Classifiers on Imbalanced Data for Software Defect Prediction Problems,” *SN Computer Science*, vol. 1, no. 2, p. 108, 2020, doi: 10.1007/s42979-020-0119-4.

[10] X. Zhong, L. Zhang, and H. Ban, “Deep reinforcement learning for class imbalance fault diagnosis of equipment in nuclear power plants,” *Annals of Nuclear Energy*, 2023, doi: 10.1016/j.anucene.2023.109685.

[11] W. Lu, W. Wang, X. Qin, and Z. Cai, “Enhancing Fault Diagnosis in Mechanical Systems with Graph Neural Networks Addressing Class Imbalance,” *Mathematics*, 2024, doi:

10.3390/math12132064.

[12] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, “Data Level Preprocessing Methods,” in *Learning from Imbalanced Data Sets*, Cham: Springer International Publishing, 2018, pp. 79–121.

[13] S. Bagui and K. Li, “Resampling imbalanced data for network intrusion detection datasets,” *Journal of Big Data*, vol. 8, 2021, doi: 10.1186/s40537-020-00390-x.

[14] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, “Clustering-based undersampling in class-imbalanced data,” *Information Sciences*, vol. 409–410, pp. 17–26, 2017, doi: https://doi.org/10.1016/j.ins.2017.05.008.

[15] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.

[16] U. Saeed, S. Jan, Y.-D. Lee, and I. Koo, “Fault diagnosis based on extremely randomized trees in wireless sensor networks,” *Reliab. Eng. Syst. Saf.*, vol. 205, p. 107284, 2021, doi: 10.1016/j.ress.2020.107284.

[17] R. Clavijo-López *et al.*, “Energy-aware and Context-aware Fault Detection Framework for Wireless Sensor Networks,” *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 14, pp. 1–13, 2023, doi: 10.58346/jowua.2023.i3.001.

[18] L. K. Wardhani, R. A. Febriyanto, and N. Anggraini, “Fault Detection in Wireless Sensor Networks Data Using Random Under Sampling and Extra-Tree Algorithm,” pp. 1–6, 2022, doi: 10.1109/citsm56380.2022.9935888.

[19] S. Maataoui, G. Bencheikh, and G. Bencheikh, “Predictive Maintenance in the Industrial Sector: A CRISP-DM Approach for Developing Accurate Machine Failure Prediction Models,” *2023 Fifth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, pp. 223–227, 2023, doi: 10.1109/actea58025.2023.10193983.

[20] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, “Labelled data collection for anomaly detection in wireless sensor networks,” in *2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2010, pp. 269–274, doi: 10.1109/ISSNIP.2010.5706782.

[21] U. Saeed, Y.-D. Lee, S. U. Jan, and I. Koo, “CAFD: Context-Aware Fault Diagnostic Scheme towards Sensor Faults Utilizing Machine Learning,” *Sensors* , vol. 21, no. 2. 2021, doi: 10.3390/s21020617.

[22] J. Hancock, T. Khoshgoftaar, and J. Johnson, “The Effects of Random Undersampling for Big Data Medicare Fraud Detection,” *2022 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pp. 141–146, 2022, doi: 10.1109/sose55356.2022.00023.

[23] R. Zuech, J. Hancock, and T. Khoshgoftaar, “Detecting web attacks using random undersampling and ensemble learners,” *Journal of Big Data*, vol. 8, 2021, doi: 10.1186/s40537-021-00460-8.

[24] S. Karimi-Bidhendi, J. Guo, and H. Jafarkhani, “Energy-Efficient Deployment in Static and Mobile Heterogeneous Multi-Hop Wireless Sensor Networks,” *Ieee Transactions on Wireless Communications*, vol. 21, no. 7, pp. 4973–4988, 2022, doi: 10.1109/twc.2021.3135385.

[25] X. Liu *et al.*, “ResGRU: A Novel Hybrid Deep Learning Model for Compound Fault Diagnosis in Photovoltaic Arrays Considering Dust Impact,” *Sensors*, vol. 25, no. 4, p. 1035, 2025, doi: 10.3390/s25041035.