A Machine Learning Approach for Fault Detection and Reliability Investigation in Wireless Sensor Networks

Zahraa A. Habeeb, Hanan A.R. Akkar and Jabbar K. Mohammed
Department of Electrical Engineering, University of Technology, 10066 Baghdad, Iraq
eee.24.02@grad.uotechnology.edu.iq, Hanan.a.akkar@uotechnology.edu.iq, jabbar.k.mohammed@uotechnology.edu.iq

Keywords: Wireless Sensor Networks, Fault Detection, Machine Learning, Anomaly Detection.

Abstract: Data loss, reduced network lifespan, and decreased accuracy are common consequences of wireless sensor network (WSN) faults. WSN performance requires fault detection to be both accurate and efficient. This study proposes a hybrid fault detection for WSNs by integrating several machine-learning models to improve anomaly classification. Our method compares the strength of each classifier, including random forest (RF), support vector machine (SVM), k-nearest neighbour (KNN), naïve Bayes (NB), convolutional neural network (CNN), and multilayer perceptron (MLP). This approach is the key novelty because it compares traditional ML and deep learning models with hyperparameter optimization and with better optimized classifier performance measurement for different fault cases like offset, gain, stuck at, and out of bounds faults. To validate our proposed model, we carry out Python-based simulations and analyze the accuracy, precision, recall, and computational efficiency of the proposed model compared to the rest of the classifiers. The results show that KNN, RF, and CNN get 100% accuracy for fault types, with KNN taking the least response time. Given the recognized need for additional optimization of real-world deployment, this work shows the usefulness of multi-ML in selecting the optimal one for creating better fault detection in WSNs.

1 INTRODUCTION

Since cloud computing has become more prevalent, WSNs have emerged as one of the most significant modern technologies. Researchers commonly use them in many fields, such as smart cities, military services, and smart agriculture, as well as in healthcare systems, environmental monitoring, improving system efficiency, and industrial automation systems. A WSN typically consists of several lightweight, independent sensors that are wirelessly linked to one another and dispersed around a given space to monitor environmental or physical factors like humidity, temperature, and noise levels [1], [2], [3], [4], [5]. Three key features are typically needed to implement a wireless sensor network in whatever field of interest. The wireless sensor network's sensor unit must first detect specific environmental elements. Second, a particular method of processing and storing the gathered data is required. Sensor nodes must connect to the sink and neighbouring nodes [6]. The sensing, processing, transmission, and power units are the four primary components of a sensor node in a wireless sensor network. The sensing unit's main components are the sensor and an Analog-to-Digital Converter (ADC), which transforms analogue data to digital signals. Second, a processing unit includes a microprocessor or microcontroller with a small memory unit for intelligence sensor node control. Thirdly, a transmission component employs a transceiver to send and receive data throughout the network, utilizing the short-range radio frequency spectrum. For each of the components mentioned above to perform its specific functions, the power unit provides power to each of them [7], [8], [9]. Many wireless sensor networks serve in challenging environments like unreachable places.

Sensor data must be reliable and accurate to make informed decisions [10]. However, both internal and external factors have a significant impact on WSN data collection. Internal factors encompass the characteristics of sensor nodes, including resource constraints such as memory, cost, communication bandwidth, battery life, and data generation characterized by inconsistencies, noise, errors, and missing values. The external factor is affected by the number of sensor nodes in WSN and their vulnerability to several types of attacks, including denial-of-service and replay attacks. It assumes that the sensor nodes' data in wireless sensor networks is imperfect and aberrant due to various internal and external factors, which subsequently impact the overall outcomes. Any readings from a sensor node that don't match normal are called outliers or anomalies. In the above scenario, the ability to identify abnormalities in installed sensor nodes within WSNs is crucial since they send functional data [11], [12], [13]. For the discovering faults system to obtain the data required without mistake or delay, it must be efficient and quick enough [14], [15], [16], [17], [18], [19], [20] WSNs can significantly improve their ability to operate and performance by utilizing machine learning techniques. These enhancements might involve various issues, such as boosting data transmission efficiency, minimizing energy consumption, facilitating sophisticated network management, and predicting node failures [21].

2 WORKS ON FAULT DETECTION

Reference [22] employed SVM classification to identify five types of faults, four of which derive from a dataset published in 2010 by scholars at the University of North Carolina at Greensboro, while also introducing a new type of fault referred to as an arbitrary fault. The proposed method consists of two phases: Phase one, expected duration, utilizes SVM for data learning and implements the outcome in the cluster-head for classification purposes. In Phase 2, real-time data collection involves taking two readings of humidity, H1 and H2, along with two readings of temperature, T1 and T2, to create a new vector, Vt. Subsequently, the last three vectors, Vt, Vt-1, and Vt-2, were employed to develop an updated remark vector, followed by the application of SVM for classification purposes. The findings indicate that the detection accuracy surpasses Bayes by 2.25% and exceeds cloud by 19.86% compared to HMM, SOSEN, and Bayes. The false alarm rate is 72% higher for cloud systems and 95% higher for HMM.

Reference [23] introduced an innovative approach for anomaly detection that combines density-based spatial clustering of applications with noise (DBSCAN) and SVM, referred to as HSE. The DBSCAN and SVM methods utilized three columns from the Intel Berkeley Research Lab (IBRL) data sets: temperature, humidity, and voltage. The DBSCAN algorithm was applied to cluster the data, labelling clusters as usual for high density and anomalous for low density. Subsequently, SVM was employed to distinguish anomalous data from normal data. The results were compared with a segment-based approach, showing that the HSE demonstrates superior performance, getting an accuracy of 97.1 and a false positive rate of 0.04.

Reference [24] performed an analysis of three ML techniques: The SVM, NB Approach, and Gradient Boosting Decision Tree. They utilize three types of faults: short-term faults, noise faults, and fixed faults to detect issues from the real datasets at IBRL. The findings indicate that the Gradient Boosting Decision Tree algorithm outperforms SVM and NB, achieving an 88% detection accuracy and an 8% FPR.

Reference [25] utilized six different machine learning classifiers: SVM, Stochastic Gradient Descent (SGD), RF, CNN, MLP, and PNN to identify six failure types: gain, offset, out of bounds, stuck-at faults, spike, along with data loss, the latter two of which scientists incorporate into the dataset released in 2010 by scientists at the University of North Carolina at Greensboro. They introduced faults at various levels (10%, 20%, 30%, 40%, and 50%). The suggested model adhered to the following procedures: Initially, two measurements of Humidity H1 and H2, together with two measurements of Temperature T1 and T2, were used to construct a new vector Vt. Three vectors, Vt, Vt-1, and Vt-2, are used sequentially to create a new vector with 12 measurements. Secondly, prepare the dataset as previously said. Third, an applied machine learning classifier was used on the cluster head of each cluster in the wireless sensor network to establish a decision function and categorize the data into two groups. RF is used to achieve the best possible result, yielding 98% and 100% at fault probabilities of 0.1 and 0.2, respectively, with a TPR of 0.177.

Reference [26] assessed five classifier kinds using machine- learning methodologies, including Long Short-Term Memory (LSTM), SVM, MLP, RF, and PNN. Assessed accuracy, Matthews Correlation Coefficient (MCC), FAR, and TPR for binary and multi-fault classification. The detection accuracy of LSTM was superior, and the FBR was minimal.

Reference [27] Six classifiers have been employed, including CNN, MLP, RF, SVM, Stochastic Gradient Descent, and PNN, in conjunction with WSN to identify six types of faults: Gain fault, Offset fault, Stuck-at fault, Spike fault, Out of Bounds, and Data loss. The findings indicate that the RF classifier achieves the highest accuracy at 97%. The researchers are not calculating the FBR in this article.

Reference [2] introduced an innovative technique for malfunction identification in wireless sensor networks, the innovative anomaly detection model (SADM). The authors' model went through numerous vital stages described below: They started using the Intel MIT Lab dataset, which included measurements from 54 MICA2 sensors gathered over several months. Researchers created two subsets of the data: 30% for testing and 70% for training. Four columns taken from the dataset—temperature, light, humidity, and voltage—are included in each subset. The authors added a 10% error to the original data containing 2.3 million samples. Second, six machine learning classifiers researchers use: SVM, Gaussian Naïve Bayes (GNB), KNN, RF, DT, and Linear Discriminant Analysis (LDA). The research showed that the DT is the best classifier with an FBR of 0.72% and an accuracy of 99.34% in the training subset and 99.25% in the testing subset.

Reference [1] introduced an innovative approach for anomaly detection through an ML scheme that employs supervised training principles. This method utilizes a hybrid model, SVM-RFNet, which integrates SVM, RF, and neural networks (NN). The proposed model attained a peak accuracy rating of 96% and a rate of false positives of 1.5%. Outperforming SVM, RF, and NN. However, it exhibits significantly higher complexity and longer training times than other models. Please remember that all the papers must be in English without orthographic errors.

3 DIFFICULTIES AND PROBLEM DEFINITIONS

The challenges and difficulties of WSNs in defect detection concentrate on the following aspects:

1) Wireless Sensor Networks contain minimal capabilities for each sensor node. Classification algorithms identify faults since they derive from a straightforward computation.
2) Researchers find that the placement of sensor nodes in hazardous environments is crucial.
3) The method of identifying faults is efficient and instantly dismisses any losses. This method of identifying faults requires acknowledging the erroneous data concerning the accurate data, after which the sensor nodes substitute.

Because wireless sensor networks operate in hazardous environments, such as monitoring flood levels in rivers, detecting forest fires, and assessing rainfall, sensor node failures are more likely to happen because of the nature of these situations. The restrictions on sensors for everyday use will not lead to significant consequences. Under demanding situations like catastrophes, wildfires, tsunamis, and similar occurrences, there is a regular occurrence of harm to people's lives, adverse environmental conditions, and financial problems resulting from malfunctioning essential components. It is vital to reduce faults to safeguard against potential losses in challenging situations. Faults are categorized into different classifications through various algorithms based on the mechanisms used for identifying them. The faults are instantly categorized. Actions have been implemented to address the issues by offering suitable solutions [28].

3.1 The Model of Failures in Wireless Sensor Networks

In [25], consider data gathered via the node's sensor as a time series, *d (n, t, f(t))*, where *n* denotes the node identification, *t* clearly shows the exact moment when the value was detected, and *f(t)* Shows the value that node *n* collected at that moment *t*. The function *f(t)* may be mentioned in $\alpha + \beta x + \eta$, where α represents an additive constant known as the offset, a multiplicative constant called gain is defined by β, *x* represents the value of the sensor that is not malfunctioning at time *t*, and η shows the amount of outside noise that is in the results. In a perfect scenario, *f(t)* would equal *x*; however, in practical situations, a fault-free node will exhibit in (1).

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

1) Offset fault. An offset fault indicates a difference in the sensed data, characterized by an additive constant that diverges from the expected data. This may happen as a result of sensor calibration being insufficient. An offset fault can be represented by (2):

$$x' = \alpha + x + \eta, \tag{2}$$

where x' belongs to f(t), and α denotes the constant value added to the standard reading.

2) Gain fault. A failing is designated as a gain fault when the rate of shift in data collected repeatedly diverges from what was expected over an extended period. In gain fault, a constant value is multiplied by the data from the standard device. This could result from faulty tuning of the sensors. A gain defect may be expressed using (3).

$$x' = \beta x + \eta, \tag{3}$$

where x' Belongs to f(t), and β denotes the constant factor applied to the standard reading.

3) Stuck-at fault. In an electronic device, the information on a specific point that requires adaptation depends on the input. Nevertheless, the circuit remains fixed indefinitely in either a zero or a one state without any variation. A stuck-at failing can be represented as in (4).

$$x' = \alpha, \quad (4)$$

where ver x' is an element of f(t), and α denotes a detected fixed amount.

4) Out-of-bounds fault. A failure is classified as an out-of-bounds failure when the collected information exceeds the thresholds established by the problem requirements. A node experiences an out-of-bounds error when x' exceeds θ or falls below θ_1, where x' is an element of f(t), and θ and θ_1 represent the application thresholds.

3.2 Justification for Classifier Selection

This work aimed to choose SVM, RF, NB, KNN, CNN, and MLP for fault detection in WSN due to their ability to handle noisy data, scalability, and computational efficiency. It is for this reason that these models were chosen [29]:

- As SVM can find an optimal decision boundary between faulty and non-faulty states across a maximal margin hyperplane [30], it is easy to see that SVM is suitable for fault detection. High dimensions can be detected well by it, and it operates well in small to medium data sets, making it a good candidate for sensor network anomaly detection [22].
- We use RF as it is robust and good at overfitting and handling missing or noisy data. In contrast with single Decision Trees, which are characterized by high variance, RF uses an ensemble of trees and thus improves the classification accuracy [31]. RF, DT, and NB have been used in the previous WSN fault detection research. RF outperformed DT and NB for detecting multiple types of faults [27].
- NB was included as it is straightforward, has low computational complexity, and can handle independent feature distributions. Though NB requires feature independence that is violated in real-world WSN data, its fast-training time provides an advantage for rapid anomaly detection in large-scale sensor networks [32].
- KNN is a non-parametric method that works well for small datasets with no overlapping classes. Since faults in WSNs tend to have a particular pattern, it is beneficial for fault detection [24]. Moreover, minimal model training is needed in KNN, which makes this a good option for real-time WSN monitoring when computational efficiency is essential.
- CNN are generally used to classify images, but in that respect, they are very well suited to sensor time series data because they learn spatial patterns. Features can be automatically extracted in CNNs, making replacing the need for manual feature engineering easier. As mentioned in previous studies [26], CNN-based models outperform traditional ML on fault detection in complex sensors.
- MLP was used as it can model nonlinear relations and can also detect anomalous sensor behaviour. As opposed to decision boundaries, MLP is capable of capturing fault patterns that are more complex than those of tree-based models on WSN datasets [27].

For some reasons below, the following models were not applied in the study: DT, GBDT, and LSTM networks:

- DT. They are simple modelling approaches that can easily be interpreted; however, they have high variance that causes them to overfit a small amount of data and a low signal-to-noise ratio of sensor data [21]. RF was chosen instead because it fixes the overfitting problem of DT with the help of multiple trees.
- GBDT. Although GBDT achieves greater accuracy than individual DTs, this DT learning model is time-consuming and requires the fine-tuning of its hyperparameters, which is challenging for real-time WSN monitoring, as pointed out by [24]. However, the performance of the presented RF is similar and has a lower computational complexity.
- LSTM Networks. For different types of anomaly detection on time series, LSTMs are appropriate; however, this approach involves labelled training data. However, as mentioned before, in real-world WSN deployments, data labelling is scarce; therefore, excessive use of LSTMs to perform fault detection requires more

training data and effort [26]. Moreover, CNNs have the same feature extraction capability while having lower requirements on the complexity of the computation [33]. Therefore, the selected models will be reasonably accurate, efficient in computational time, and effective in real-time.

4 PROPOSED FAULT DETECTION SYSTEM MODEL

The planned structure comprises dual sensor elements linked to a laptop for record management through wireless communication. Each sensor node independently collects data and transmits it to the computer for storage in databases and additional analysis. Displayed in Figure 1, the suggested structure has three distinct groups:

1) Data collection phase. The current phase of the procedure requires data sense. The detected data is used to generate a measurement vector V_t. The resulting vector contains two functions derived from humidity evaluations, H_1 and H_2, and two temperature readings, T_1 and T_2. V_t, V_{t-1}, and V_{t-2} are the data collections that were measured and aggregated in a sequential order. The contributions have been made to the new observation vector.
2) Fault injection phase. Faults are introduced into the real recognized datasets during this phase. Gain failing, offset failing, out-of-bounds failing, and stuck-at failing are infused as the source indicates.
3) Fault estimation and classification. In this step, the classifiers that have been previously discussed are employed to classify defects based on the estimated outcome. SVM, RF, NB, KNN, CNN, and MLP are among the classification techniques implemented on the data.

The collected data are categorized into two categories following analysis of the result function for each sensor node. If the results function positively, the data is classified as usual; otherwise, they are labelled as failing. At the collection level, two sensor nodes detect data. In the injection phase, these detected data are injected with predefined faults. In this step, we run each classification algorithm on the faulted input to find the added errors. The next step is to use various degrees of categorization metrics to find the problems at varying rates. Based on these observations,

Figure 1 is designed to help with understanding.

Time-series data is processed using the CNN architecture in this system using two 1D convolutional layers with tanh activation. The dimensionality of the features is then decreased by max pooling. The output is flattened and sent to a dense layer with a sigmoid activation function to carry out binary classification.

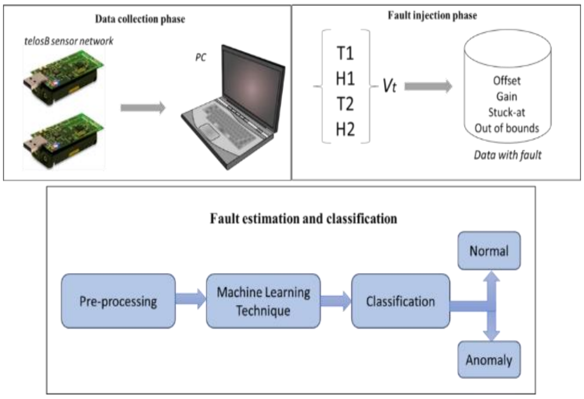

Figure 1: System model.

4.1 Dataset Description

Our dataset was obtained from a multi-hop wireless sensor network. It consists of a set of observations with a dimension of 12. Each vector contains measurements at three successive instances: t_0, t_1, and t_2. Each instance involves two temperature and humidity measurements (T_1, T_2, and H_1, H_2). We have prepared 4,688 observations (data example or vector). A set of faults has been introduced randomly. We prepared 60 datasets, each containing 4688 observations, characterized by varying fault rates (50%, 40%, 30%, 20%, and 10%) and types of data faults (Offset, Gain, Stuck-at, and out of bounds). Various values of beta have been utilized. Each dataset contains two Excel files: one for observations and another for the label y. y equals 1 for an everyday observation and y equals −1 for a fault observation.

5 SIMULATIONS AND RESULTS

The synthetic dataset was based on real-world conditions of WSNs, consisting of temperature and humidity readings. Artificially injected faults were used to simulate sensor failures. The proposed machine learning models were validated, and the known fault types are referenced using the University of North Carolina at Greensboro datasets.

The SVM classification technique detects the errors contained in the sensed units, as illustrated in Figure 2. It successfully identifies all failures in 100% of each sensor's overall number of problems. Except for offset faults, it accurately identifies a smaller number. Figure 3 shows how the RF classifier finds sensing unit failures. It accurately finds all categories of failures in 100% of the faults present within the sensor network. The NB classifier identifies sensing unit errors, as seen in Figure 4. It efficiently identifies stuck-at defects in 76% and offset defects in 68% of the overall errors happening inside the sensor system, with a failure probability of 0.5. The average fault identification rate is lower than the classifiers stated before. The KNN classification technique identifies problems in the sensing unit, shown in Figure 5. With a perfect score of 100%, every defect is found to be within a similar limit. The CNN classifier identifies problems in the sensing unit, shown in Figure 6. It identifies all defects in a similar limit with a 100% accuracy rate, though it requires more time than the previous classifiers.

The MLP classifier similarly identifies errors, as shown in Figure 7. It efficiently identifies all faults, covering 100% of the total faults inside the sensor network. Contrary to the offset error, the detection accuracy was 51% at a 0.5 opportunity of defect. As previously mentioned, the MLP classifier identifies fewer flaws than the CNN classifier. With CNN and KNN recording 0% false positive rates across all fault types and RF exhibiting very slight sensitivity to offset faults, the results showed that CNN, KNN, and RF classifiers performed the best in reducing false alarms. On the other hand, SVM exhibited moderate reliability with few mistakes, but NB and MLP showed increased false positive rates under particular circumstances. According to these results, the best choices for applications requiring high dependability and low false detection are CNN, KNN, and RF.

5.1 Comparing with Existing Studies and Validation

Table 1 compares the proposed method against conventional approaches used in fault detection for WSNs. It compares the proposed work with previous research, which shows the advantages and disadvantages, such as higher accuracy for some classifiers, and real-world validation is needed to confirm the validity and bring a scientific contribution in place. Table 5 shows that our method has fairly comparable accuracy to those existing studies. Our approach outperforms [22], which achieved 97.75% accuracy for a given fault set using an SVM-based approach, as it achieved 100% accuracy for some fault types using KNN, RF, and CNN. Moreover, our method performs better than [23], who study only three fault types per machine, for fault detection in multi-fault environments. We also compared with deep learning models as in [26], with LSTM, and found that CNN performs similarly with lower computational complexity. Next, the strengths of the proposed approach are a low false positive rate compared to previous SVM-based methods and its adaptability to various fault conditions.

We surveyed the literature on previous methods of WSN fault detection to [22], [23], [24]. While they have shown effective classification techniques, no comparison of multiple machine learning models was done. Our study then extends this work to systematically compare six classifiers and the computational performance to understand ML techniques for fault detection.

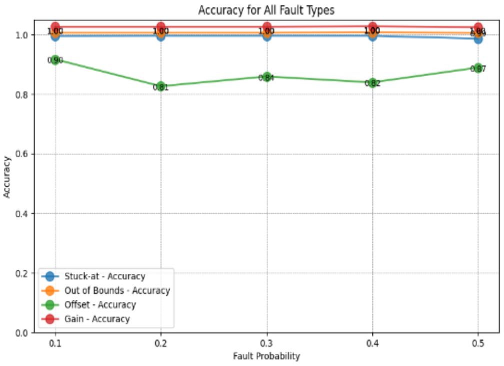

Figure 2: Accuracy of SVM.

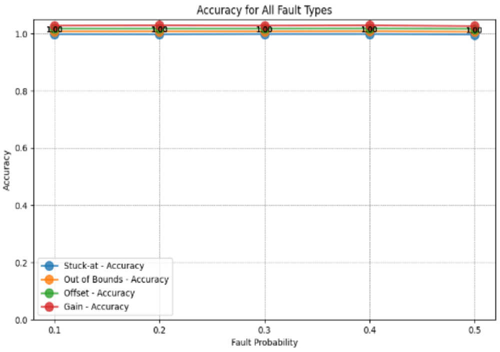

Figure 3: Accuracy of RF.

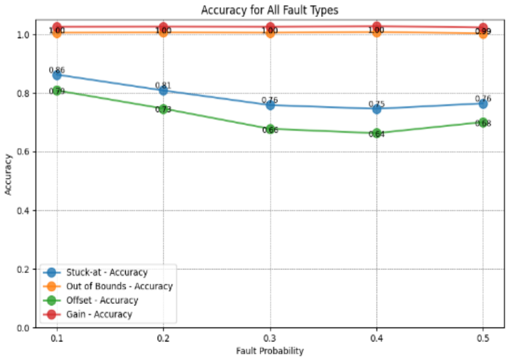

Figure 4: Accuracy of NB.

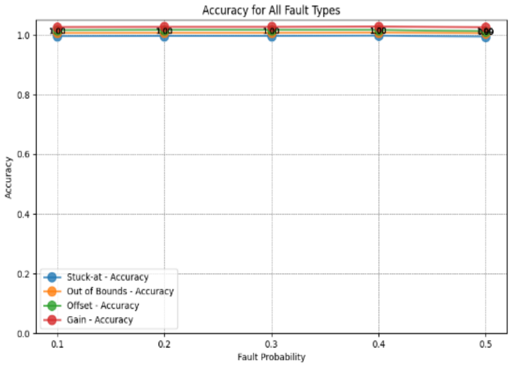

Figure 5: Accuracy of KNN.

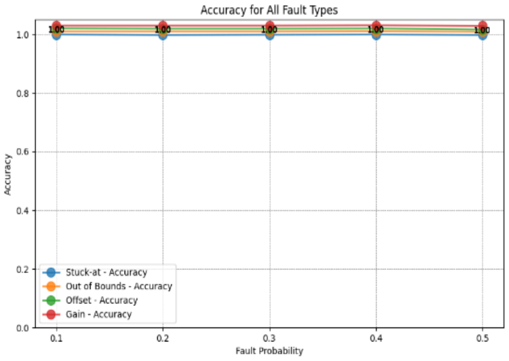

Figure 6: Accuracy of CNN.

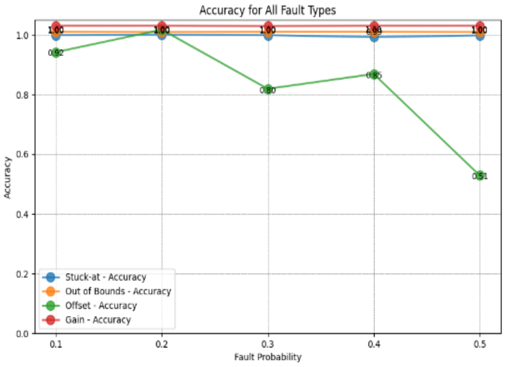

Figure 7: Accuracy of MLP.

Table 1: Proposed method against conventional approaches.

Study	Method	Dataset Used	Fault Types Considered	Best accuracy (%)	Limitations
Proposed Study	RF, SVM, KNN, CNN, NB, MLP	UNC Greensboro Dataset (2010)	Offset, Gain, Stuck-at, Out-of-Bounds	100% (KNN, RF, CNN)	Requires further real-world validation
Zidi et al. (2018)[22]	SVM-based Fault Detection	UNC Greensboro Dataset (2010)	5 Fault Types	97.75%	Higher false positive rate
Saeedi & Mazinani (2018)[23]	DBSCAN + SVM Hybrid Model	Intel Berkeley Lab (IBRL) Dataset	3 Fault Types (Temp, Humidity, Voltage)	97.1%	Not suitable for large datasets
Ye et al. (2018)[24]	Naïve Bayes, SVM, GBDT	Intel Berkeley Lab (IBRL) Dataset	Noise Faults, Fixed Faults	88%	GBDT outperforms traditional ML methods
Noshad et al. (2019)[25]	RF, SGD, CNN, MLP, PNN	UNC Greensboro Dataset (2010)	6 Fault Types	98-100%	Not optimized for time complexity
Azzouz et al. (2020)[26]	LSTM, SVM, MLP, RF, PNN	Simulated Dataset	Multi-Fault Detection	95%	LSTM is computationally expensive

5.2 Confidence Intervals and Standard Deviations

Of the six assessed models, RF, KNN, and CNN attained flawless accuracy (100%) with no fluctuation, signifying strong and consistent performance. SVM and MLP achieved 96% and 95% accuracy, respectively, but with marginally more variability. With the most significant variability and the lowest performance (87%), NB demonstrated inadequate resilience. CNN, KNN, and RF performed better and more consistently, as shown in Table 2.

Table 2: Accuracy with confidence intervals.

Classifier	Accuracy (%)	95% CI (±)	SD
SVM	96	0.03	0.07
RF	100	0	0
NB	87	0.06	0.13
KNN	100	0	0
MLP	95	0.05	0.12
CNN	100	0	0

5.3 k-Fold Cross-Validation Results

The k-fold cross-validation results show significant differences in the model's performance, as shown in Table 3, especially regarding mean accuracy and stability. With a mean accuracy of 99% and no fluctuation over folds, the K-Nearest Neighbours (KNN) classifier proved the best performer, demonstrating its excellent accuracy and resilience. The Naïve Bayes model, on the other hand, had decreased stability as seen by its increased variability (0.14%) and poorer accuracy (87%). Different models with low variability and good performance (96%-99% accuracy) were SVM, RF, and MLP. On the other hand, CNN and other deep learning models did poorly, maybe due to their sensitivity to data properties. Given that KNN is the most dependable model for defect detection in WSN data, our findings highlight the significance of assessing accuracy and stability.

Table 3: Cross-Validation accuracy results.

Classifier	Mean Accuracy (%), K=5	Standard Deviation (%), K=5	Mean Accuracy (%), K=50	Standard Deviation (%), K=50
SVM	93	0.12	96	0.05
RF	80	0.11	99	0.00
NB	79	0.13	87	0.14
KNN	92	0.12	99	0.00
MLP	94	0.14	96	0.10
CNN	70	0.14	66	0.19

6 CONCLUSIONS

The work proposed in this paper employs a multi-classifier machine-learning strategy to identify defects in wireless sensor networks. It contrasts the methods in terms of accuracy and detection time to decide which is the most effective. A systematic evaluation of the performance of 6 different machine learning algorithms (SVM, RF, NB, KNN, CNN, and MLP) on the fault detection of WSN sensor nodes, i.e., offset, gain, stuck at, and out of bounds faults, is proposed. Six algorithms were implemented and evaluated utilizing WSN. The RF, KNN, and CNN classifiers identify Offset faults, Gain faults, Out-of-Bounds faults, and Spike faults with an accuracy of 100%.

In contrast, KNN demonstrated shorter response times with the best accuracy. In summary, ML techniques hold significant promise for identifying faulty nodes in WSN by selecting the appropriate method based on the particular needs and limitations of the WSN implementation. Additional investigation is required to find an efficient ML method for detecting faulty nodes in WSN and to create innovative techniques to enhance the discovery process's precision and speed. RF is a way to aggregate multiple decision trees to reduce overfitting and increase robustness in fault detection with higher accuracy. KNN is theoretically optimal for small datasets with well-separated classes because it is better suited to detect faults with faster computation. CNN can efficiently extract hidden patterns in sensor data and thus detect complex anomalies that traditional statistical methods can no longer detect.

Empirical Validation:

We demonstrate that RF, KNN, and CNN classifiers achieve 100% accuracy in fault detection of some fault categories in our simulation results and surpass prior statistical models and ML approaches [25]. KNN also had the shortest response time, and thus, is suitable for real-time WSN fault monitoring.

KNN, RF, and CNN reach 100% accuracy for gain, out-of-bounds, and stuck at faults. Among all the other methods, KNN offered the least detection time (0.12*s*–0.21*s*) and is suitable for a real-time task. We outperformed previous works [22] using SVM, for which they obtained 97.75% accuracy across multiple fault types.

The RF model built in this work attained 100% accuracy for fault classification, which is a better result than what is presented in [25]. using RF with 98% accuracy. In contrast to previous studies using single classifiers, we compared six classifiers on a systematic basis. We determined the merits and demerits of each classifier under different fault types (fault vessel fusion). Our results show that KNN has the best speed and CNN and RF have the highest accuracy, which are valuable in real-time applications. The accuracy of our model was 2 to 3 % better than the benchmark studies on WSN fault detection.

However, as our proposed method has not been tested in a real WSN deployment, the accuracy of the obtained method is high. Future research should:

- The approach is to determine if it is robust against environmental noise in real-world WSN environments.
- Decrease computation overhead by improving feature selection and the classifier architecture.
- To prove the model's effectiveness in real-life deployment instances, deploy the model for real-world deployment cases.
- On further optimization of fault detection using more advanced deep learning architectures (i.e., Transformers, LSTMs).
- Reduction of computational overhead for the real-time WSN applications.

Deploying machine learning-based fault detection models in large-scale WSNs presents several challenges. These include sensor failures and environmental noise, which can misclassify faults caused by random noise instead of hardware failures. Mitigation strategies include hybrid detection models combining statistical anomaly detection with machine learning to reduce false positives and adaptive thresholding techniques to adjust for environmental variations dynamically. Another challenge is energy constraints and battery life, as WSN nodes are typically battery-powered with limited energy capacity. Mitigation strategies include using lightweight models like RF or KNN, implementing edge computing to process fault detection locally, and reducing data transmission. Network congestion and data latency are other challenges, as large WSNs often experience network congestion due to limited bandwidth, leading to delayed or missing sensor readings. Mitigation strategies include data compression techniques and distributed fault detection models that operate locally on sensor clusters. Scalability in large-scale deployments is another challenge, as most ML models are tested on small-scale datasets. Mitigation strategies include federal learning, incremental learning, and security concerns. Finally, WSNs are vulnerable to cyberattacks, where adversaries intentionally inject false sensor readings to mimic faults. Mitigation strategies include using anomaly detection techniques to detect unusual patterns of sensor failures and implementing blockchain-based security mechanisms to ensure the integrity of WSN data.

REFERENCES

[1] S. A. Shifani, A. A. Mary, M. I. M. Metilda, G. V. Rajkumar, M. S. Sutha, and S. Maheshwari, “A novel machine learning strategy for anomaly identification scheme in wireless sensor networks using supervised training principles,” in Proc. 5th Int. Conf. Electronics and Sustainable Communication Systems (ICESC), 2024, pp. 1754–1761, doi: 10.1109/ICESC60852.2024.10689838.

[2] A. Haque, H. Soliman, and M. N. U. R. Chowdhury, “Wireless sensor networks data anomaly detection: A smart approach,” in Proc. 3rd Int. Conf. Intelligent Technologies (CONIT), 2023, doi: 10.1109/CONIT59222.2023.10205933.

[3] B. Ahmad, W. Jian, Z. A. Ali, S. Tanvir, and M. S. A. Khan, “Hybrid anomaly detection by using clustering for wireless sensor network,” Wireless Personal Communications, vol. 106, no. 4, pp. 1841–1853, Jun. 2019, doi: 10.1007/s11277-018-5721-6.

[4] A. Bader Saeed, A. K. Al-Samarrie, and H. A. R. Akkar, “Design and implementation of an interface unit communicated by a laser system within wireless sensor network,” Engineering and Technology Journal, vol. 34, no. 13, pp. 2507–2517, Dec. 2016, doi: 10.30684/etj.34.13a.13.

[5] S. H. S. Sabah and M. S. Croock, “Software engineering-based fault tolerance method for wireless sensor network,” Iraqi Journal of Computer, Communication, Control and System Engineering, pp. 21–28, Oct. 2020, doi: 10.33103/uot.ijccce.20.4.3.

[6] B. Chander and G. Kumaravelan, “Outlier detection strategies for WSNs: A survey,” Journal of King Saud University – Computer and Information Sciences, Sep. 2022, doi: 10.1016/j.jksuci.2021.02.012.

[7] H. Ayadi, A. Zouinkhi, B. Boussaid, and M. N. Abdelkrim, “A machine learning method: Outlier detection in WSN,” in Proc. 16th Int. Conf. Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2015, pp. 722–727, doi: 10.1109/STA.2015.7505190.

[8] M. Xie, S. Han, B. Tian, and S. Parvin, “Anomaly detection in wireless sensor networks: A survey,” Journal of Network and Computer Applications, vol. 34, no. 4, pp. 1302–1325, Jul. 2011, doi: 10.1016/j.jnca.2011.03.004.

[9] D. Hendrycks, M. Mazeika, and T. Dietterich, “Deep anomaly detection with outlier exposure,” Dec. 2018. [Online]. Available: http://arxiv.org/abs/1812.04606.

[10] F. Fusco, V. U. Castrillo, H. M. R. Giannetta, M. Albano, and E. Cavallini, “Methods, standards and components for wireless communications and power transfer aimed at intra-vehicular applications of launchers,” Aerospace, Feb. 2024, doi: 10.3390/aerospace11020132.

[11] Kurniabudi et al., “Network anomaly detection research: A survey,” Indonesian Journal of Electrical

Engineering and Informatics, vol. 7, no. 1, pp. 36–49, Mar. 2019, doi: 10.11591/ijeei.v7i1.773.

[12] B. Chander and Kumaravelan, "One class SVMs outlier detection for wireless sensor networks in harsh environments: Analysis." [Online]. Available: www.ijrte.org.

[13] R. Ul Islam, M. S. Hossain, and K. Andersson, "A novel anomaly detection algorithm for sensor data under uncertainty," Soft Computing, vol. 22, no. 5, pp. 1623–1639, Mar. 2018, doi: 10.1007/s00500-016-2425-2.

[14] M. M. Gharamaleki and S. Babaie, "A new distributed fault detection method for wireless sensor networks," IEEE Systems Journal, vol. 14, no. 4, pp. 4883–4890, Dec. 2020, doi: 10.1109/JSYST.2020.2976827.

[15] Y. Gao, F. Xiao, J. Liu, and R. Wang, "Distributed soft fault detection for interval type-2 fuzzy-model-based stochastic systems with wireless sensor networks," IEEE Transactions on Industrial Informatics, vol. 15, no. 1, pp. 334–347, Jan. 2019, doi: 10.1109/TII.2018.2812771.

[16] X. Miao, Y. Liu, H. Zhao, and C. Li, "Distributed online one-class support vector machine for anomaly detection over networks," IEEE Transactions on Cybernetics, vol. 49, no. 4, pp. 1475–1488, Apr. 2019, doi: 10.1109/TCYB.2018.2804940.

[17] J. S. Saini and R. Kait, "Exploring machine learning strategies for intrusion detection in wireless sensor networks," in Proc. IEEE 9th Int. Conf. Convergence in Technology (I2CT), 2024, doi: 10.1109/I2CT61223.2024.10543320.

[18] J. M. Aguiar-Pérez et al., "Understanding machine learning concepts," in Encyclopedia of Data Science and Machine Learning, IGI Global, 2022, pp. 1007–1022, doi: 10.4018/978-1-7998-9220-5.ch058.

[19] A. A. Radhi, H. N. Abdullah, and H. A. R. Akkar, "Denoised Jarque-Bera features-based K-means algorithm for intelligent cooperative spectrum sensing," Digital Signal Processing, vol. 129, Sep. 2022, doi: 10.1016/j.dsp.2022.103659.

[20] A. A. Radhi, H. A. R. Akkar, and H. N. Abdullah, "Skewness and access kurtosis as denoised mixed features-based K-medoids for cooperative spectrum sensing," Physical Communication, vol. 54, Oct. 2022, doi: 10.1016/j.phycom.2022.101831.

[21] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative analysis of intrusion detection systems and machine-learning-based model analysis through decision tree," IEEE Access, vol. 11, pp. 80348–80391, 2023, doi: 10.1109/ACCESS.2023.3296444.

[22] 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, Jan. 2018, doi: 10.1109/JSEN.2017.2771226.

[23] H. S. Emadi and S. M. Mazinani, "A novel anomaly detection algorithm using DBSCAN and SVM in wireless sensor networks," Wireless Personal Communications, vol. 98, no. 2, pp. 2025–2035, Jan. 2018, doi: 10.1007/s11277-017-4961-1.

[24] Y. Yuan, S. Li, X. Zhang, and J. Sun, "A comparative analysis of SVM, naïve Bayes and GBDT for data faults detection in WSNs," in Proc. IEEE 18th Int. Conf. Software Quality, Reliability, and Security Companion (QRS-C), 2018, pp. 394–399, doi: 10.1109/QRS-C.2018.00075.

[25] Z. Noshad et al., "Fault detection in wireless sensor networks through the random forest classifier," Sensors, vol. 19, no. 7, Apr. 2019, doi: 10.3390/s19071568.

[26] I. Azzouz, B. Boussaid, A. Zouinkhi, and M. N. Abdelkrim, "Multi-faults classification in WSN: A deep learning approach," in Proc. STA 2020 – 20th Int. Conf. Sciences and Techniques of Automatic Control and Computer Engineering, 2020, pp. 343–348, doi: 10.1109/STA50679.2020.9329325.

[27] S. Gnanavel et al., "Analysis of fault classifiers to detect faults and node failures in a wireless sensor network," Electronics, vol. 11, no. 10, May 2022, doi: 10.3390/electronics11101609.

[28] R. R. Swain, P. M. Khilar, and S. K. Bhoi, "Heterogeneous fault diagnosis for wireless sensor networks," Ad Hoc Networks, vol. 69, pp. 15–37, Feb. 2018, doi: 10.1016/j.adhoc.2017.10.012.

[29] A. Adamova, T. Zhukabayeva, and Y. Mardenov, "Machine learning in action: An analysis of its application for fault detection in wireless sensor networks," in Proc. IEEE Int. Conf. Smart Information Systems and Technologies (SIST), 2023, pp. 506–511, doi: 10.1109/SIST58284.2023.10223548.

[30] A. Ayadi, O. Ghorbel, A. M. Obeid, and M. Abid, "Outlier detection approaches for wireless sensor networks: A survey," Computer Networks, vol. 129, pp. 319–333, Dec. 2017, doi: 10.1016/j.comnet.2017.10.007.

[31] X. Zhou et al., "Fault diagnosis of silage harvester based on a modified random forest," Information Processing in Agriculture, vol. 10, no. 3, pp. 301–311, Sep. 2023, doi: 10.1016/j.inpa.2022.02.005.

[32] Z. Hashim, H. Mohsin, and A. Alkhayyat, "Handwritten signature identification based on hybrid features and machine learning algorithms," Iraqi Journal of Computer, Communication, Control and System Engineering, pp. 43–56, Mar. 2024, doi: 10.33103/uot.ijccce.24.1.4.

[33] A. S. Dawood, "A comparative study using deep learning models and transfer learning for detection and classification of Alzheimer's disease," Iraqi Journal of Computer, Communication, Control and System Engineering, pp. 57–70, Mar. 2024, doi: 10.33103/uot.ijccce.24.1.5.