European Journal of Information Technologies and Computer Science
Vol 4 | Issue 5 | December 2024
ISSN 2736-5492

RESEARCH ARTICLE

A New Fault Method Detection for Wireless Sensor Networks using "Autoencoder and LS-SVM"

Falah Hasan Hani*

ABSTRACT

This study focuses on the issue of fault detection in WSNs while not disturbing the flow of data, and it presents a comprehensive and new approach to dealing with the problem. The first steps in the context of the developed methodology for application to the data of stock exchanges include scaling of samples by the method of min-max, transformation of windows of samples as part of data preparation, as well as preliminary data cleaning and accurate division of data into sections. These steps are important for dataset preparation for further analysis. The proposed method relies on the integration of Autoencoders put alongside Least Squares Support Vector Machines (LSSVM). An Autoencoder network was developed, and the size of the hidden nodes was later adjusted to identify internal parameters in the dataset. It was helpful for the subsequent reconstructions of the data scene and allowed us to obtain high-level features required for fault detection. With the help of these extracted features, the LSSVM model was developed for classifying normal and anomalous conditions in WSNs; the training outcome exhibited high effectiveness where anticipated indexes of the training data set were 99.77% and for the test data set were 99%. The above outcomes support the feasibility and accuracy of the applied approach in fault recognition. The thesis greatly helps in the progression of the field by providing a methodical way of addressing the important problem of fault detection in WSNs and providing experimental evidence and analysis for the stated problem.

Keywords: Autoencoder, Fault detection, LS-SVM, wireless sensor networks.

Submitted: November 01, 2024
Published: December 05, 2024

10.24018/ejcompute.2024.4.5.145

College of Information Technology Engineering, Qom University, Iran.

* ***Corresponding Author:***
e-mail: mustafafadil837@gmail.com

1. Introduction

Wireless Sensor Networks (WSNs) have attracted interest over the years because of their versatility in areas that include environmental and health and intelligent city implementations [1]. They include a group of low-cost sensor nodes that are connected through wireless links to relay information to a central station in real time [2]. However, they are faulty because the systems and networks they manage are dynamic and resource-constrained environments most of the time. Hence, security incidents result from exchange failures with node or software faults; therefore, there is a need to develop adequate protective measures to ensure continuous availability and functionality of the system; for this reason, there are several reasons that make fault detection important in wireless sensor networks [3], [4]. First, these networks are used in intended environments where the sensor nodes are vulnerable to failures due to environmental degradation. Secondly, because the sensor nodes generally contain low power and computational capability, any performance problem should be detected as early as possible. Thirdly, it is important to note that the data collected is largely used in life-determining decisions, and therefore, it is very important to distinguish between liberal data and fake news. Last, WSNs may contain many nodes, potentially in the hundreds or thousands, which will make physical repairs impractical [5].

A method proposed for fault detection is called the enhanced minimum redundancy maximum relevance (MRMR), which pays much attention to the relevance determination for reliable fault detection systems [6], [7]. Literature studies in IoT and cyber-physical systems propose machine-learning approaches for developing sensor-fault detection systems that enable real-time fault diagnosis [8]. Another approach of supervised machine learning comprises decision trees (DTs), which are effective in real-time fault detections; however, the authors did not explicate their improvements [9].

In an attempt to overcome the challenge of class imbalance in fault detection the RUS method has been used together with the Extra-Trees classifier algorithm. This also improves the performance of classifiers where the use of a balanced dataset for training is encouraged [10]. However, the Intelligent Negative Selection Algorithm (INSA) also relies on negative selection in combination with the Support Vector Machines (SVM) in the aspect of fault classification; nevertheless, information about the researchers themselves is not discussed [11].

The portability aspect of WSNs, compounded with SDN and realistic AI tools, have been used to enhance fault detection, and the researchers involved in this study are unknown [12]. Moreover, the Sailfish Optimized Inception with Residual Network (SOIR) model has been presented for fault diagnosis, incorporating advanced optimization techniques as well as deep learning architectures have been integrated into it, although the related researchers have not mentioned it here [13]. This paper aims to identify the major approaches and development plans for fault detection, and proves the importance of the different machine learning methods towards improving the availability of the system.

WSNs are prone to numerous failures due to their environment and the constrained resources they possess within the physical and system realms. Hence it is significant to construct a dependable and efficient fault detection system; nonetheless, in the current era of technological evolution, Machine learning, and Deep learning–Autoencoders, along with Least Square Support Vector Machines (LS-SVM), are quite famous techniques for fault detection. Autoencoders are types of neural networks that automate reconstruction and anomalous data identification, making them efficient for use in WSNs' complex data nature.

2. Methodology

The interactions between intelligent methods and Least Squares SVM for the proposed new feature are illustrated in (Fig. 1). The data collection involves field data acquisition, data pre-processing, feature extraction by autoencoders, and fault classification by LSH-kernel based SVM. To do so, this chapter section gives an overview of the main phases of feature selection, and the succeeding sections provide further descriptions of each of the phases. The methodologies that go along with various algorithms and numerous attributes in selecting the parameters for each phase are also elaborated. The overarching goal is to bring a clear and detailed description and demonstration of this novel perspective and reveal the beneficial results of applying this strategy in the framework of increasing confidence in WSNs in different fields of utilization.

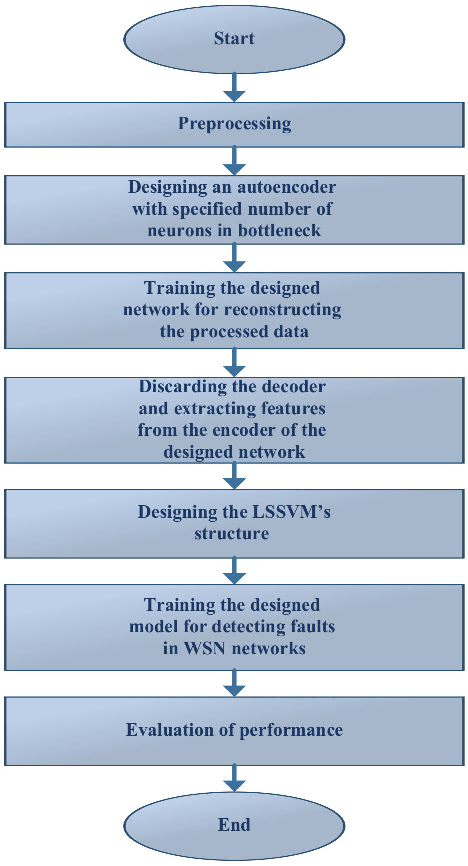

Fig. 1. The steps of our proposed method.

2.1. *Data Preprocessing*

Data preprocessing is one of the most important early stages when performing fault detection in WSNs using Autoencoders and LSSVM. This stage ensures that the data is complete and ready for the subsequent features to be extracted, as well as for looking for potential faults. Here's a detailed overview of the preprocessing steps: Here's a detailed overview of the preprocessing steps:

1. Normalization (Min-Max Method)

Normalization of data is done using the Min-Max method as illustrated in the following (3)–(1), which scales the features data from the range of [0,1]. This step helps to normalize the features so that we do not have favored sensors because of their range.

$$\left(X_{\text{normalized}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \right)$$

2. Reordering Sample Windows

Sample windows are relocated to improve temporal information contents. This procedure is as follows: increasing the size of a sample data matrix from three times.

3. Data Cleaning

Points to are three (t0, t1, t2) to six (t0, t1, t2, t3, t4, t5), the temporal patterns that are vital for fault detection become refined and efficient.

Specifically, data quality encompasses covering the aspects that lead to data quality and the concept of noise, outliers, and missing values. It is imperative to pay special emphasis on this step to reflect real-world networks' conditions.

4. Partitioning

The given dataset is divided into a training dataset and a testing dataset (most often 80/20). This partitioning strategy enables the training of the model on most of the data while evaluating its ability in new and unknown data, which is very important in real-time fault detection.

The above mentioned preprocessing steps separately provide a firm base to our fault detection methodology; thus the subsequent phases, Autoencoders based feature extraction and LSSVM based fault classification is performed on the clean data.

2.2. Architecture of Proposed Approach

Our proposed method involves fault analysis in WSNs by using Autoencoders and LSSVM, while data preprocessing forms the initial phase that enables the creation of a clean dataset.

- *Autoencoder* is set up to include a hidden layer with fewer nodes to achieve the abstract features of WSN data. It encodes the input data and quantizes them into a latent space representation where the required information for fault detection can be learned without supervision.
- *LSSVM* is then used in classifying features and differentiating between normal network operations and faulty ones. This is in light of high performing parameters enabling the determination of faults accurately and reliability in numerous applications.

2.2.1. Autoechoer Yields Machine's Features for it

As one of the core components in our fault detection system for WSNs, the AE is responsible for the feature extraction step of the overall system. AE leverages its unique capabilities to enhance data representation and facilitate fault detection efficiency: AE leverages its unique capabilities to enhance data representation and facilitate fault detection efficiency:

Supervised and unsupervised learning: autoencoder is very effective in extracting underlying trends and patterns from the WSN data and makes it possible to identify efficient features in the detection of faults. This autonomy to search for information on their own makes them flexible to handle any type of faults in the network or its conditions.

It's also worth noting, though, that AE plays a critical role in dimension reduction by eliminating unnecessary data while maintaining the vital data. This feature improves the effectiveness of faulting detection systems by making the processing of high-dimensionality WSN data comparatively simpler. Autoencoder Architecture:

1. Encoder

The encoder, which is positioned tactically to the left of the AE design, functions as a data compressor as it compresses the input data into the latent or pre-bottleneck area. At this point, the important details and desirable patterns from the input data are extracted. Usually, the encoder includes several layers of neurons; the goal of this structure is to downsample while enhancing important features necessary for further analysis.

2. Decoder

The "Autoencoder" has an encoder and a decoding part, where the decoding part replicates the compressed data. The objective is to accurately reconstruct inputs and outputs to reduce the error when achieving the purpose of extracting inherent features of wireless sensor network data. *Its Role in Fault Detection:* In the case of Autoencoder, the main goal is to come up with a summarized representation of a set of data to improve defect detection as well as to eliminate noise. As the architecture of the model, it has a defined number of neurons in the Bottleneck layer to adjust the data compression. The extracted features assist in making wireless networks and the application of reliable information and communication technology.

3. Classification using LSSVM

In this project, LSSVM is chosen as the foundation for the development of the fault detection framework in WSNs. This decision is driven by several distinct advantages that LSSVM offers, particularly well-suited for the complexities of WSN data:

- *Handling High-Dimensional Data:* As is common with WSNs, high-dimensional data is processed using the LSSVM approach. Its optimization approach is highly capable of handling big data sets.
- *Nonlinear Relationship Recognition:* Compared with the original form of SVMs, LSSVM is capable of learning nonlinear relationships with WSNs data contain. This capability is mandatory, especially given the fact that WSN operations are characterized by high dynamism and complexity.
- *Noise Robustness:* Compared to SVM approach, LSSVM introduces least squares to make the classification variant more robust to noise in data and thus increase the reliability of the fault detection results.

3. Results and Discussion

3.1. Analysis and Review of Results

In this research, a fault detection method combines an AE for feature learning and an LSSVM classifier. The action of data cleaning included deleting of Dump, sw,<|reserved_special_token_261|>,Maj5W,,,0,, and Rej55W, whereas data normalization is, where the number of features is reduced to 39 from 41 because of constant features. The HoldOut method was applied for splitting the dataset into the training and test sets in the proportion of 70:30. The autoencoder with the layer of 20 neurons used unsupervised training for epochs of 500 and with L2 regularization of 0.05. Stochastic gradient descent training was used and the specific optimizer employed was the scaled conjugate gradient descent. In the training convergence and sample regeneration by the autoencoder, the convergence is depicted in (Figs. 2–4) which show the regeneration. Therefore, this methodological framework seeks to improve fault detection reliability within WSNs through the use of advanced feature extraction and reliable classification algorithms.

Then, using the characteristics that the autoencoder collected, an LSSVM is created to categorize faults. The

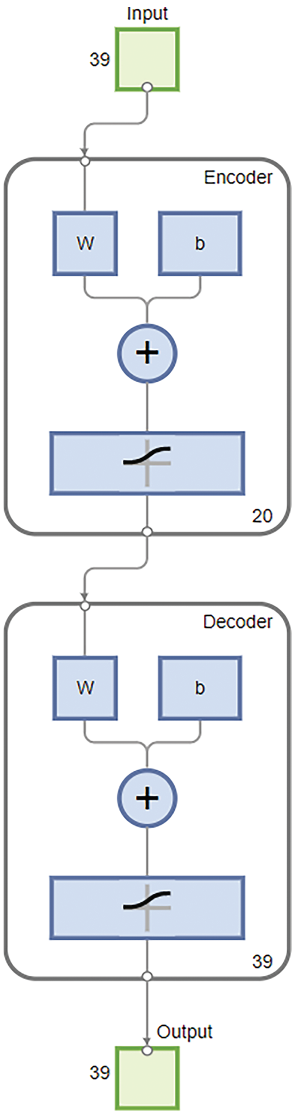

Fig. 2. The structure of the designed autoencoder.

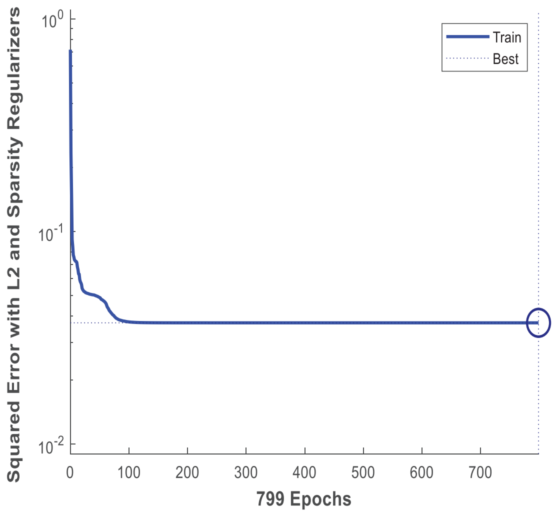

Fig. 3. The training process of the designed autoencoder.

LSSVM classifier's parameters need to be adjusted correctly in order to classify the chosen features appropriately. The categorization parameters are then listed in (Table I).

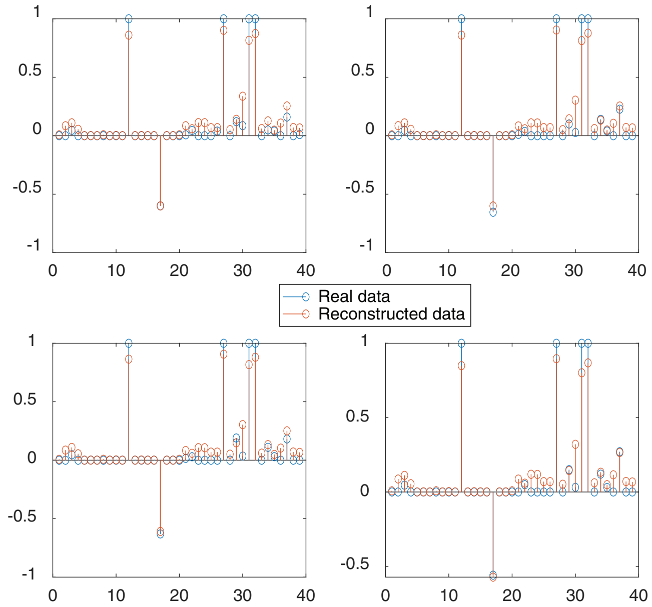

Fig. 4. Reconstruction performance of the trained autoencoder using four different samples.

TABLE I: The Hyperparameters of LSSVM

Parameter	Value
Kernel type	RBF kernel
Gamma	83.249
The first kernel parameter	4.2849
The second kernel parameter	0.7580

The LS-SVM classifier was trained to detect issues in the wireless sensor network once the hyperparameters were established. The performance of the trained model is thoroughly evaluated using a variety of evaluation criteria and visualization approaches in the sections that follow in this chapter. These analyses, which include measures like accuracy, precision, recall, F1 score, and ROC curves, provide insightful information on the effectiveness of the unique fault detection technique.

3.2. Examining the Results of the Test and Training Dataset Based on the Confusion Matrix

A crucial tool for assessing the effectiveness of our suggested problem-detection technique without resorting to additional criteria, such as accuracy, is the confusion matrix. The primary output of the classification results produced by the Autoencoder-LSSVM model is a confusion matrix, which is a tabular form. It divides the forecasts produced by our model into four main groups: It divides the forecasts produced by our model into four main groups:

1. *True Positives (TP):* These kinds of occurrences, in which our model accurately identifies errors in the WP data, are instances of good recognition that we can share with our clients. In our scenario, the pattern has been triggered by the model for real fault occurrence, which is known as true positive.
2. *Real Negatives (TN):* When our model performs well in identifying real positive network behavior, it suggests that the model is functioning correctly.

In other words, it avoids assigning blame that isn't actually there.

3. *False Positives (FP):* This occurs when the model misinterprets an anomaly (another fault) as a healthy sign (a fault). Therefore, a false positive suggests that the network could be functioning well or that peaking and flow changes are making it falsely worrisome.
4. *False Negatives (FN):* These instances indicate that our model is unable to detect irregularities in the system; as a result, just a specific component is subjected to deterministic examination. The model's performance on both an overall and class-wise level is clearly illustrated by the visualization of the overall confusion matrix (Fig. 5) and the class-wise confusion matrices for the training and test datasets (Fig. 6).

The combination of TP, FP, and FN shows how well the model detects defects and how likely it is to overclassify them as such. Additionally, the model's capacity to identify all real anomalies is demonstrated by the fact that the TN should always equal 1. We may score the defect detection model without actually seeing specific performance data to evaluate the overall performance in our thesis setting by evaluating these matrices, which give us insight into the model's strengths and flaws.

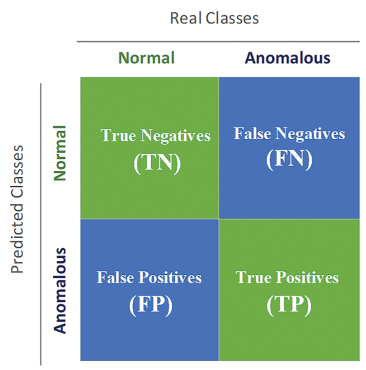

Fig. 5. Fault detection in WSN confusion matrix.

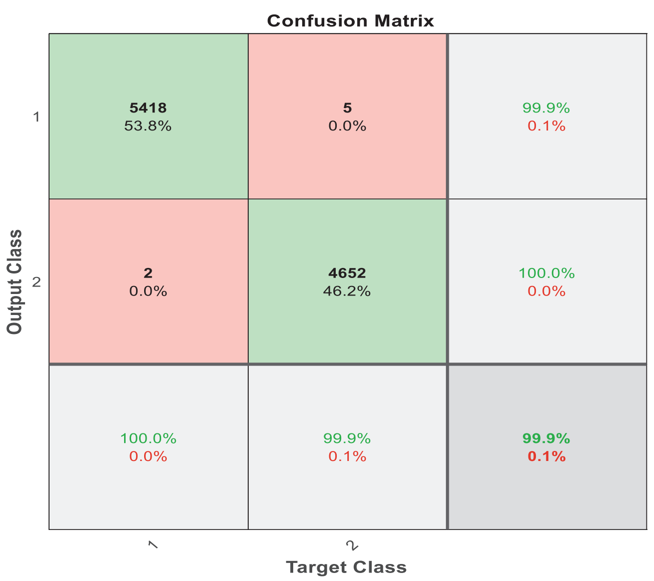

Fig. 6. Fault detection in WSN confusion matrix for training dataset.

3.3. Evaluation Metrics

In evaluating our Fault Detection Methodology using the Autoencoder-LSSVM approach, several key metrics from the confusion matrix guide our assessment:In evaluating our Fault Detection Methodology using the Autoencoder-LSSVM approach, several key metrics from the confusion matrix guide our assessment:

1. *Accuracy:* Measures the overall ability of this model to predict accurately, and is given by the formula: ((TP + TN)/(TP + FP + TN + FN)) It shows the percentage of accurately classified normal and abnormal behavior in the WSN data.
2. *Precision:* Concentrates on the efficiency of the positive forecasts and is measured with TP/(TP + FP). Precision tells how many of the instances that were classified as an anomaly by the model are actually genuine anomalies, therefore guaranteeing efficient accuracy in identifying such cases.
3. *Recall (Sensitivity):* Measures the model's performance in pinpointing all true anamolies, set as TP: (TP + FN). Concerning the model, it highlights how it is capable of capturing all true fault occasions.
4. *F1-Score:* Gives the measure of accuracy in the form of the harmonic mean of precision and recall WHERE Accuracy = 2 (Precision ∗ Recall)/(Precision + Recall). It reports both false positives that the model produces and the events that the model fails to detect as anomalies.

These metrics combine to give a direction on how effective the Autoencoder-LSSVM model is in diagnosing faults in the WSN, and at the same time; measures are put in place to avoid frequent false alarms while at the same time ensuring that actual faults are detected.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{4}$$

3.4. Examining the Results of the Test and Training Dataset Based on the Introduced Evaluation Criteria

In this section, we proceed with evaluating the efficacy of our proposed methodology, utilizing the performance metrics we previously discussed. We visually represent these metrics in (Fig. 7), providing clear graphical insights into the computed values for precision, accuracy, recall, and the F1 score. Noteworthy is the impressive accuracy achieved in both datasets: 99.93% for the training dataset and 99.25% for the test dataset. These results underscore the remarkable capability of our approach. One more thing is they the model performs well in both the training and the test datasets by achieving the same accuracy levels can indicate the model's ability to capture an inherent data pattern and generalize a new data points, which is a basic step in fault detection in the WSNs.

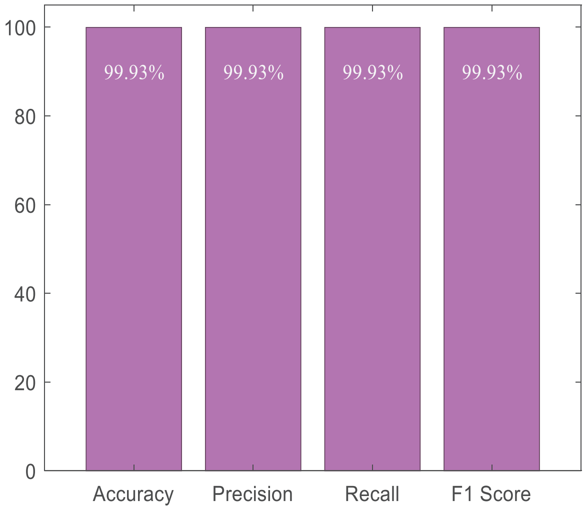

Fig. 7. Fault detection in WSN performance chart for the training dataset.

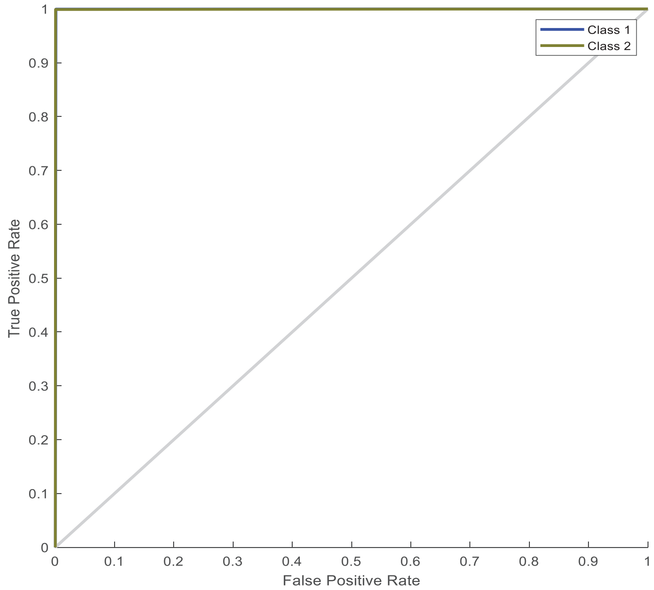

Fig. 8. Fault detection in WSN ROC curve for training datasets.

3.5. Examining the Results of the Test and Training Set Based on the Receiver Operating Characteristic Curve

The ROC curve is a very helpful indicator that shows classifier effectiveness in relation to various scenarios where the model should be able to differentiate between typical network traffic and unusual activity. Plotting sensitivity and specificity on its axis at different classification thresholds allows the ROC curve to visually represent the relationship between the two. The instance that shows how successfully the fault detection model we are suggesting separates real fault occurrences from false alarms within the WSN data is called the ROC curve.ROC curve indicates us which model is performing better than the other by saying it is positioned in the upper left corner.

This shows that the false positive rate is at its lowest outside of the model's ability to get a good true positive rate. The performance of our methodology is illustrated graphically in (Fig. 8) which pertains to the training and test datasets, respectively. This effectively illustrates how our method distinguishes between abnormal and typical actions, and it also suggests that the model may provide unmatched defect detection.

3.6. Comparison of the Proposed Method with Previous Studies

In this work, we have put together a comparative analysis of fault detection techniques for WSNs with references to the autoencoder- LSSVM approach that we recommended and its comparison to other methods. The specificity of the developed method, increasing its robustness, flexibility, and accuracy in fault identification, makes it a large step forward in tackling WSN issues.

In the IoT age, data collected from sensors are valuable for input in decision-making processes; however, environments in which operations take place introduce difficulties and enhance the likelihood of sensor faults. Zidi *et al*. [14] present a novel methodology based on digital twins and GANs as well as GAF encoding with the accuracy of fault detection at 98% and 7%. Since WSNs are established in unstable environments, sensors are exposed to different risks, and hence, effective fault detection techniques must be employed. The machine learning classifiers discussed in Suthaharan *et al*. [15] are instrumental in classifying the data into faulty and non-faulty data categories to tackle the different faults' different scenarios with optimum accuracy.

In Hasan *et al*. [16], applying supervised learning with RUS and ET classification to screenshot sensors, the authors notice that GPU Memory anomalies are evident, and the performance of the proposed method is comparable to SVM and RF algorithms at the same indicators. The summary comparison is provided in (Table II), based on which it could be observed that the present study made several improvements in the methodology of WSN fault detection.

TABLE II: Comparison of Our Proposed Approach with Previous Studies

Reference	Method	Accuracy
[14]	Generative Adversarial Network (GAN) with Gramian Angular Field (GAF)	98.7%
[15]	Multi-layer perceptron classifier	92%
[16]	Random Under Sampling (RUS) and the Extra-Tree (ET) algorithm	96%
Our proposed approach	LSSVM and autoencoder	

4. Conclusion

This work focuses on fault detection of Wireless Sensor Networks (WSNs) by extending the ability to detect faults in a better way. The methodology includes three main preprocessing steps: feature normalization, where the range is minimized and maximized; sample clustering in windows; and data cleansing.

Autoencoders are integrated with Least Square Support Vector Machines (LSSVM) in the proposed strategy. The

Autoencoder learns the main features of data and produces feature maps necessary to detect defects; the LSSVM identifies normal and different behaviors.

The performance estimates prove its high effectiveness; the accuracy rates are equal to 0.9977 on the training set and 0.9925 on the test set. This affirms the effectiveness of identifying fault nodes of the method. However, through Autoencoders and LSSVM integration, the approach also enables early fault detection and more performance of WSN applications. Specifically testing the bot-iot dataset, the model had a 98% precision in classifying the attack and legal transmission, which made it useful precisely.

Conflict of Interest

The authors declare that they do not have any conflict of interest.

References

[1] Panda M, Gouda BS, Panigrahi T. Fault diagnosis in wireless sensor networks using a neural network constructed by deep learning technique. *Spr Trac Nat-Inspi Comp*. Jan 1, 2020:77–101.
[2] De D, Mukherjee A, Das SK, Dey N. Nature inspired computing for wireless sensor networks. *Springer*. 2020. doi: 10.1007/978-981-15-2125-6.
[3] Lee MH, Choi YH. Fault detection of wireless sensor networks. *Comp Communica*. 2008;31(14):3469–75.
[4] Chen J, Kher S, Somani A. Distributed fault detection of wireless sensor networks. *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks DIWANS '06*, Published online 2006.
[5] Jiang P. A new method for node fault detection in wireless sensor networks. *Sensors*. 2009;9(2):1282–94.
[6] Zidi S, Moulahi T, Alaya B. Fault detection in wireless sensor networks through SVM classifier. *IEEE Sen J*. 2018;18(1):340–7.
[7] Priya PI, Muthurajkumar S, Daisy SS. Data fault detection in wireless sensor networks using machine learning techniques. *Wirel Pers Communica*. 2021;122:2441–62.
[8] Jan SU, Lee YD, Koo IS. A distributed sensor-fault detection and diagnosis framework using machine learning. *Inform Sci*. 2021;547(2021):777–96.
[9] Saeed U, Jan SU, Lee YD, Koo I. Fault diagnosis based on extremely randomized trees in wireless sensor networks. *Relia Eng Sys Saf*. 2021;205:1–11.
[10] Vamsi P, Chahuan A. Machine learning based hybrid model for fault detection in wireless sensors data. *ICST Trans on Scal Inform Sys*. 2018;0(0):1–8.
[11] Marino R, Wisultschew C, Otero A, Lanza-Gutierrez JM, Portilla J, de la Torre E. A machine learning-based distributed system for fault diagnosis with scalable detection quality in industrial IoT. *IEEE Inte Thin J*. 2020;8(6):4339–52.
[12] Darvishi H, Ciuonzo D, Rossi PS. A machine-learning architecture for sensor fault detection, isolation, and accommodation in digital twins. *IEEE Sens J*. 2023;23(3):2522–38.
[13] Wardhani LK, Febriyanto RA, Anggraini N. Fault detection in wireless sensor networks data using random under sampling and extra-tree algorithm. *2022 10th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–6, Published online Sep 20, 2022.
[14] Zidi S, Moulahi T, Alaya B. Fault detection in wireless sensor networks through SVM classifier. *IEEE Sens J*. 2018;18(1):340–7.
[15] Suthaharan S, Alzahrani M, Rajasegarar S, Leckie C, Palaniswami M. Labelled data collection for anomaly detection in wireless sensor networks. *Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 269–74, 2011.
[16] MdN H, Jan SU, Koo I. Wasserstein GAN-based digital twin-inspired model for early drift fault detection in wireless sensor networks. *IEEE Sens J*. 2023;23(12):13327–39.