J. Electrical Systems 20-11s (2024): 3291-3309

[1] Ridha Mohammed Alfoudi,

Mohsen Nickray.

Fault Detection in Wireless Sensor Networks Using Horse Herd Algorithm and Convolutional Neural Network with Attention Layer

Journal of Electrical Systems

Abstract: - Reliable and real-time detection of faults in Wireless Sensor Networks (WSNs) is significant for the ongoing flow of critical data despite being a strenuous task. In this article, we present a comprehensive WSN-suitable fault detection system as a solution to this challenge. The primary step involves a thorough pre-processing of data such as partitioning, data cleaning, reordering of sample windows, and normalization through the min-max technique. These steps are fundamental to preparing the dataset, while also assisting systematize the proposed solution. A Horse Optimization Algorithm (HOA) integrated with a Convolutional Neural Network is at the heart of the method. As such, the CNN is able to capture highly advanced spatial and temporal features embedded in the processed data, as its convolutional layers are proficient in pattern extraction. Further, hyperparameter optimization of CNN learning rates, batch sizes, and the total number of convolution layers is performed using HOA, to improve the CNN performance. WSNs can greatly benefit from this, with the proposed model identifying faults with 99.47% accuracy on the test dataset, and 99.63% accuracy on the training dataset. These results show the proposed method's effectiveness and accuracy in addressing fault detection issues in WSNs towards enabling better stability and performance within the network.

Keywords: Wireless Sensor Networks, Convolutional Neural Networks, Horse Optimization Algorithm

1. Introduction

As Wireless Sensor Networks (WSNs) evolve, ensuring the system's reliability and efficiency has become more challenging. With the development of monitoring systems for the environment, healthcare, or industrial processes, WSNs are now confronted with problems such as hardware failures or even communication and environmental changes that disrupt effective transmission of data [1]. It is not uncommon for standard methods of fault detection to lag behind. The complex structure of these networks, combined with traditional approaches, often leads to inaccurate or late fault detection. One of the most critical problems is the absence of strong systems capable of error-free classification of normal and faulty behaviors within noisy and complex data environments [2,3]. An example in point are the most common methods employed today which refer to pattern recognition where no learning is applied. This paper introduces a restructuring approach that implements them together with CNN and HOA. The CNN interprets complex patterns by transforming or reshaping the input data in such a way that the signal is easy to classify, capture, and analyze temporally and spatially.It is expected that the identified HOA will serve the purpose of optimization of some of the hyperparameters within CNNs such as learning rate, batch size, and number of convolutional layers. Such enhancement helps to reach high accuracy and reliability in fault detection in WSNs.

Multiple studies have proposed methods to enhance fault detection performance in Wireless Sensor Networks (WSNs).

Mazibuco et al. in [4] explore fault detection in Wireless Sensor Networks (WSNs) using deep neural networks, particularly Recurrent Neural Networks (RNNs). Addressing limitations of traditional machine learning, the study focuses on temperature and humidity data, emphasizing model selection and tuning for improved accuracy and robustness. Their findings highlight the potential of deep learning to enhance fault detection in WSNs, contributing to improved reliability in applications like environmental monitoring and healthcare.

Singh et al. in [5] propose an autoencoder-based classification model for fault detection in Wireless Sensor Networks (WSNs). The algorithm compares input and output values to identify faults, focusing on parameters like detection accuracy, false alarm rate, and false positive rate. Simulations reveal its effectiveness in detecting

[1] Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran.

Corresponding Author: ridha.alfoudi@gmail.com

faults such as spikes, fixed bias, gain, and out-of-bounds errors, outperforming traditional methods. This approach enhances the reliability of WSNs, crucial for applications in harsh environments.

Mohapatra et al., in [6] a novel approach for fault diagnosis in wireless sensor networks using a combination of artificial immune system principles and probabilistic neural networks . The method employs clonal selection to detect faulty sensor nodes and then classifies faults as permanent, intermittent, or transient using probabilistic neural networks. The algorithm isolates faulty nodes after detection and classification. Performance evaluation metrics include detection accuracy, false alarm rate, fault classification accuracy, diagnosis latency, and energy consumption. Simulation results demonstrate that the proposed algorithm outperforms existing methods in terms of these metrics, particularly in fault classification accuracy and energy efficiency. The authors compare their approach to previous algorithms by Mohapatra, Panda, and Elhadef, showing improvements in diagnosis latency and energy consumption. This research contributes to the important field of fault diagnosis in wireless sensor networks, offering a potentially more effective solution for maintaining network reliability and efficiency.

Bista & Chaudhary, in [7]a novel fault detection approach for Wireless Sensor Networks (WSNs) to address the critical issue of maintaining data precision in harsh environments where sensor nodes are prone to faults and failures . The authors present a method combining Spearman's correlation coefficient and K-nearest neighbor classification algorithm. The correlation coefficient is utilized to assess the internal status of sensor nodes, while the K-nearest neighbor algorithm classifies nodes as normal or abnormal . Through simulation, the researchers compared their proposed algorithm with the existing MCDFD (Multi-Criteria Decision-based Fault Detection) algorithm. The results demonstrated that the new approach outperforms MCDFD in terms of detection accuracy and false positive rate . This innovative fault detection method aims to improve the reliability and efficiency of WSNs by identifying and addressing faulty nodes that may generate erroneous data and lead to misinterpretation or false alarms.

Prasad & Baghel in [8] a novel fault diagnosis technique for wireless sensor networks (WSNs) using feedforward neural networks . Unlike existing distributed, hybrid, and centralized approaches that rely on data transmission between nodes, this method allows each sensor to detect its own fault status using only its own sensed data. The authors argue that current techniques are energy-inefficient and suffer from high communication overhead and delays, which reduce network lifetime. Additionally, existing methods depend on spatial-temporal correlations between nodes to identify faults. The proposed approach aims to address these limitations by enabling individual sensor fault detection. Experimental results demonstrate the effectiveness of this new technique compared to existing fault diagnosis methods across various performance metrics. This research contributes to improving the efficiency and longevity of WSNs deployed in challenging environments such as volcanoes, forests, and highways.

Zainib Noshad et al in [9] investigates fault detection in Wireless Sensor Networks (WSNs) using various machine learning classifiers. WSNs are susceptible to failures due to their deployment in unpredictable environments, making fault detection challenging. The researchers compared six classifiers: Support Vector Machine, Convolutional Neural Network, Stochastic Gradient Descent, Multilayer Perceptron, Random Forest (RF), and Probabilistic Neural Network. These classifiers were used to detect six types of faults: gain, offset, spike, data loss, out of bounds, and stuck-at faults. The study utilized real-world datasets, with spike and data loss faults artificially induced. Performance metrics included Detection Accuracy, True Positive Rate, Matthews Correlation Coefficients, and F1-score. The results of the comparative analysis revealed that the Random Forest algorithm outperformed the other classifiers in fault detection for WSNs .

Laiou et al., in [10] autonomous fault detection and diagnosis in wireless sensor networks using machine learning techniques . The researchers focus on faults caused by external events that disrupt network services, such as connectivity loss, obstacles, and packet loss due to noise or congestion. To address these issues, they employ a decision tree algorithm to train a fault detection model. The resulting model demonstrates high performance, achieving an overall accuracy of 96.46% in identifying faults on test data. Additionally, the model exhibits high precision and recall values for individual fault cases, indicating its effectiveness in fault identification. The authors emphasize the importance of reliable communication in wireless sensor networks and the need for autonomous fault detection to minimize network failures, particularly in open environments. This research contributes to the ongoing efforts in improving the reliability and resilience of wireless sensor networks through advanced fault detection methods.

Panda et al., in [11] the application of deep learning techniques for fault diagnosis in wireless sensor networks (WSNs). The authors highlight that sensor nodes in WSNs are susceptible to various types of faults due to environmental conditions, low battery, calibration issues, and sensor aging . While traditional fault detection methods rely on statistical approaches and neighboring sensor data, these techniques can be problematic when faulty sensor readings mimic non-faulty data. To address this challenge, the researchers propose using neural networks constructed through deep learning algorithms to identify and classify different fault types in WSNs, including hard, soft, intermittent, and transient faults . The study emphasizes the growing interest in deep learning methods for fault diagnosis in both industry and academia, suggesting that these techniques may offer improved accuracy and reliability compared to conventional approaches in detecting and isolating faulty sensors in WSNs.

The study by Gupta et al. in [12] addresses the critical issue of fault detection in Wireless Sensor Networks (WSNs), a key component of the Internet of Things (IoT). The authors propose a deep Bidirectional Long Short-Term Memory (Bi-LSTM) model to detect various types of faults in sensor nodes, including soft permanent, intermittent, and transient faults. This approach aims to overcome the limitations of existing fault detection methods, which often struggle with accurately diagnosing different fault types and have low detection accuracy. The researchers report that their Bi-LSTM-based algorithm outperforms other state-of-the-art fault detection approaches in terms of fault detection accuracy, false alarm rate, and false-positive rate. This improvement in fault detection capabilities is crucial for maintaining the performance and reliability of WSNs, especially given their deployment in challenging environments where sensor nodes are prone to failures.

Adamova et al. in [13] the application of machine learning techniques for fault detection in wireless sensor networks (WSNs). The authors discuss various types of faults in WSNs and present a taxonomy of failures . They outline a methodology for detecting faulty nodes, which involves data collection, feature extraction, model training, and performance evaluation. The study examines several machine learning algorithms, including convolutional neural networks, probabilistic neural networks, multilayer perceptrons, decision trees, support vector machines, random forests, Bayesian belief networks, and gradient boosting techniques. The researchers emphasize the importance of fault detection in WSNs to prevent issues such as data loss, reduced network lifespan, and decreased accuracy of collected data. The paper concludes by highlighting the need for further research to enhance the performance of existing methods and explore new algorithms for fault detection in WSNs.

Azzouz et al., in [14] multi-fault classification in Wireless Sensor Networks (WSNs) using deep learning and machine learning techniques . The researchers employed a Long Short-Term Memory (LSTM) classifier as their deep learning approach and compared its performance to several machine learning methods, including Support Vector Machine (SVM), Random Forest (RF), Multilayer Perceptron (MLP), and Probabilistic Neural Network (PNN). The comparison was based on four key metrics: Detection Accuracy (DA), True Positive Rate (TPR), Matthews Correlation Coefficients (MCC), and False Alarm (FA). The study highlights the importance of fault detection and diagnosis in WSNs, which are often deployed in challenging environments. By leveraging artificial intelligence techniques, particularly machine learning, the researchers aim to improve data accuracy and address the issue of failures in WSNs. This research contributes to the ongoing efforts to enhance the reliability and performance of WSNs in various applications.

Biswas et al in [15]a hybrid approach for fault detection in wireless sensor networks (WSNs) using a combination of Kalman filter (KF) and Extreme Learning Machine (ELM) (. The authors address the limitations of WSNs, such as low-cost sensor nodes and limited battery life, which make them susceptible to faults. The proposed method employs data fusion with KF to train the sink node on faulty data patterns, reducing the need for large training datasets. ELM is then used as a predictive classifier, offering high prediction accuracy with low communication overhead. The researchers evaluated their approach using standard WSN data with inserted random anomalies, measuring performance in terms of detection accuracy and computational time. This hybrid KF-ELM technique aims to improve fault detection in WSNs while minimizing energy consumption and processing requirements, addressing key challenges in WSN applications .

Karmarkar et al. in [16] an optimized Support Vector Machine (SVM) based fault diagnosis scheme for wireless sensor networks (WSNs) to address the critical issue of fault detection in these systems . The authors highlight that sensor nodes in WSNs are prone to failures due to hardware issues, software problems, and energy depletion, which can lead to erroneous measurements and reduced network performance. The proposed method utilizes a Grey Wolf Optimization (GWO) based SVM classifier to detect the fault status of deployed sensor nodes. Additionally, the researchers introduce a cluster-based topology to conserve energy. The effectiveness of the proposed fault detection method is evaluated through extensive simulations under various network settings and compared with existing schemes. The study aims to improve the reliability and efficiency of WSNs in applications such as environmental monitoring, home automation, and medical monitoring .

The fault detection techniques pertaining to wireless sensor networks (WSNs) reviewed in this Articles have their own great challenges. One, and arguably the most important one, is the restricted nature of placing different types of faults. These approaches have insufficient abilities to detect temporary, permanent and intermittent faults, and their detection rates drop along with changes in environmental conditions. In addition, high energy consumption is a very important problem. Nod to node communication, or communication algorithms, may use a large amount of energy which may lead to short lifetime of the network. High computational cost, as a consequence of employing deep networks or more than one algorithm, always need more time and computational power because heavy models are used. One of the barriers is indeed the degree of false alarms, because many approaches are not able to effectively separate healthy data from faulty ones, resulting in a lot of false positives and erroneous warnings. In addition, some algorithms are indeed manual where configuration of optimal parameters is a human affair which makes the process of execution challenging.

The integration of Convolutional Neural Networks, or CNNs, with the Horse Optimization Algorithm, or HOA, permits the simultaneous detection of various challenges and the automation of processes such as optimizing neural network parameters like the number of neurons or even the learning rate. In doing this, HOA abandons performing all the tasks manually and adjusts itself to autonomously improve the performance of CNNs in search for fault patterns from sensor data.

Using CNNs for enhanced fault detection is highly useful due to the ability to construct and learn complex patterns and features from the data set, or in this case, the acquired data. This approach comes with benefits such as being able to decrease energy consumption by relying on local data processing to mitigate constant node communication requirements. With regards to environmental factors, the stability of the network is not compromised thanks to HOA and its adaptive features. These parameters allow HOA optimization to considerably minimize false alarm rates. Additionally, the parameter tuning phase is conducted without the help of any humans to maximize automation.

The usage of CNNs in tandem with HOA renders this technique applicable for scenarios which demand reliability and accuracy; alongside amplifying the capabilities of wireless sensor networks, it allows for advanced capabilities in monitoring of the environment and medical systems.

The structure of this paper is organized as follows: Section 2 provides the basic concepts needed for understanding the proposed method. Section 3 elaborates on the methodology, including the design and integration of the Convolutional Neural Network (CNN) for fault pattern recognition and the Horse Optimization Algorithm (HOA) for optimizing the CNN parameters. Sections 4 and 5 describe the dataset and evaluation metrics used to assess the performance of the proposed approach, respectively. Section 6 presents the simulation results, highlighting the effectiveness of the method under various fault scenarios, including performance evaluations for different parameter configurations. A comparative analysis between the proposed method and other state-of-the-art techniques is provided in Section 7. Finally, Section 8 concludes the paper with a discussion of the results' implications and potential future research directions in fault detection for wireless sensor networks.

2. Basic concepts

In this section, a detailed overview of the fundamental principles and key concepts required to understand the proposed method is presented.

2-1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a state-of-the-art deep learning model built for a specific purpose; to process picture data. Owing to its specific design structure, CNNs can easily extract intricate attributes and characteristics from images [17].

The architecture of a convolutional neural net is made in such a manner that it is able perform extraction of salient features contained in images at different levels. These networks have several layers, each of which has a particular purpose [18].

The layers include:

- Convolutional Layer

- Pooling Layer

- Fully Connected Layer

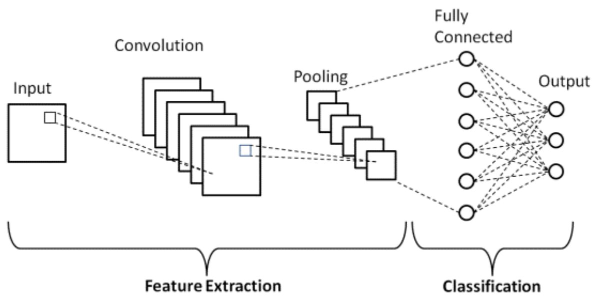

Figure 1 Convolutional neural network architecture [18]

Convolutional Layer

This layer is crucial for the calculation processes of the network. The Convolutional Layer has filters that scan the input data for certain shapes in order to pull specific features from images. Each filter is a small weight matrix that slides over the whole image while calculating the dot product of the weights with the image pixels that they cover. This result of this calculation is a feature map. The image below illustrates the mechanics of the convolution operation [19].

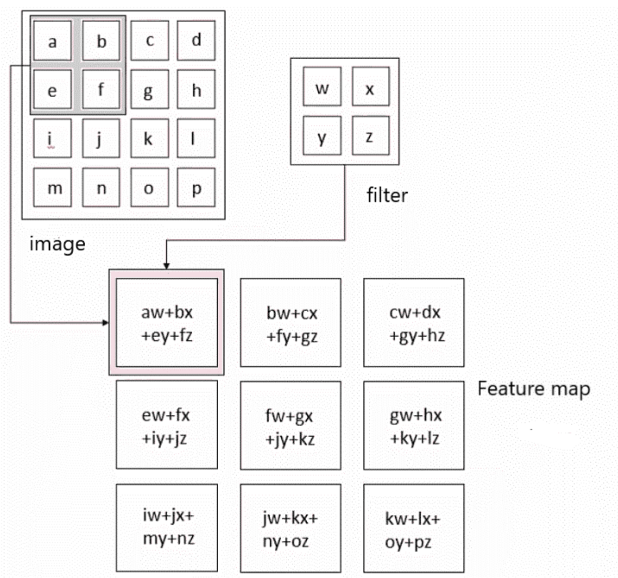

Figure 2 How to operate convolution [18]

As illustrated by the image, the first tensor (the pixel of the image) is multiplied with the second tensor (the kernel). Subsequently, all of the values are summed to achieve the result. This operation is repeated by moving the filter over the entire image. The image below depicts how the filter moves on top of every pixel of the image and presents the corresponding feature map of the convolution results [19].

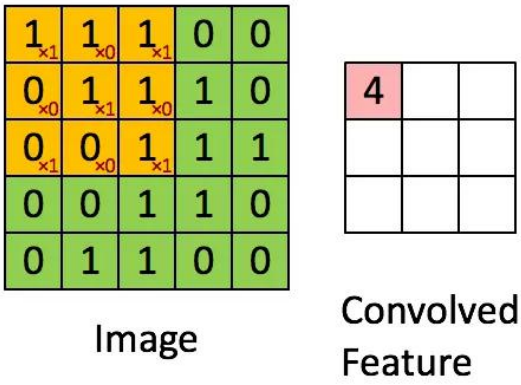

Figure 3 How the filter slides in all the pixels of the images and the result of convolution calculations in the feature map [18]

The Convolutional Layer works based on the following hyperparameters: [19]

Kernel Size: This hyperparameter defines the size of the filter. Filters are usually better when they have smaller and odd dimensions, for instance: 1,3,5, or 7. Most commonly, a kernel size of 3x3 is employed.

Stride: This hyperparameter defines how much the kernel window moves at each step. This is usually set to 1 so that every pixel in the image is processed but it can also be increased in order to decrease the amount of data that needs to be processed.

Padding: This term refers to the technique of placing pixels with a value of 1 on all borders of the image. This is done so that when the filter is applied during the convolution operation, the input and output map maintains its size.

Number of Filters: This reflects the number of different patterns or features the layer is expecting to identify.

There are several factors to consider when considering the output size of the Convolutional Layer, and they will be presented in the next sections.

- Input size

- Filter size

- Padding

For the given parameters, the output size can be calculated using the equation 1 provided .

$$H_{out} = 1 + \frac{H_{in} + (2.pad) - K_{height}}{s} \quad (1)$$

$$W_{out} = 1 + \frac{W_{in} + (2.pad) - K_{width}}{s} \quad (2)$$

In the first equation, ‘H_{out} and H_{in}represent the input and output feature maps, respectively. There are some parameters specified in the term "pad". Pad means the additional zero or some constant values that surround the input feature map. The components $K_{height}$and s correspond to kernel height and stride length. In the second equation, these components refers to their corresponding width values. So, by means of these equations, one can assess the height and width of the output feature map in the convolutional layer of a neural network [20].

In Convoluatioanl neural networks (CNNs) that deal with images having more than a single channel, it is a systematic procedure that each filter shall correspond to the number of input channels. In this case, for RGB format images, there are three channels which means there are three filtros, each one is assigned the weights for each channel's color namely: red green and blue. Each filter has three dimensions, at this stage, it is crucial that the output of the three channels is combined and the required averaged is outputed. Each channel's filter is critical in preserving the textures and information for every region of the image [21].

A non-linear activation function, for instance ReLU, is implemented after every convolution to the output of the new feature map. This aids the model in understanding intricate, non-linear shapes within the data. When an image or any data is fed into a convolutional network, it is first piped into a series of convolutional layers that work in a pipeline mechanism. The first convolutional layer is responsible for extracting features of lower levels like edges. The output of this layer is then passed to the next layer to extract even more advanced features, like corners or combined edges. As the layers move towards the end of the network, more advanced features of the image are obtained. These features are more sophisticated and include things like objects, faces and so on. In the following image, we see the steps in which an input image goes through a neural network. The network in question first pulls initial features from the image and then moves on to more sophisticated features. Lastly, the features of the head layer are used to classify the images [22]

2.2 Horse Optimization Algorithm (HOA)

To solve problems with high dimensionalities, the Horse Herd Optimization Algorithm (HOA) is a nature-inspired algorithm. The HOA algorithm is based on the social and other natural behaviors exhibited by horses and has six key behavioral parameters that it models which are grazing, hierarchies, sociability, imitation, defense mechanisms, and roaming. Following this trait balance, the algorithm is able to balance the search for the optimal solutions (diversification) with the concentration around the most promising areas (intensification). In this manner, each horse's movement is controlled by the age group to which it belongs, determining the horse's behavioral patterns in HOA. Contribution of different age groups: young (δ), adolescent (γ), adult (β), and old (α) is examined differently in the optimization process [23,24].

Key Principles

Herd Structure and Leadership

Horses in the wild exhibit a structured social hierarchy, often led by a dominant leader. The HOA models this behavior by assigning roles such as leaders and followers within the optimization process.

Exploration and Exploitation

Horses balance between exploration (searching new areas) and exploitation (refining solutions in known areas) by alternating movement strategies such as galloping, trotting, and grazing.

Communication and Cooperation

Horses in a herd communicate to avoid predators, find food, and navigate terrain efficiently. HOA mimics this cooperative behavior to improve search efficiency.

The position of each horse in the search space is updated iteratively based on the velocity vector, computed as:

$$\mathrm{X}_{\mathrm{m}}^{\mathrm{Iter,AGE}} = \mathrm{V}_{\mathrm{m}}^{\mathrm{Iter,AGE}} + \mathrm{X}_{\mathrm{m}}^{\mathrm{Iter-1,AGE}} \quad (3)$$

where $\mathrm{X}_{\mathrm{m}}^{\mathrm{Iter,AGE}}$ is the position, $\mathrm{V}_{\mathrm{m}}^{\mathrm{Iter,AGE}}$, AGE is the velocity, and AGE denotes the age group.

a) Calculation of Velocties in Various Age Categories: For every horse, the horse velocity is computed as a combination of the psychological components. The velocity of juvenile horses, for example, is denoted as –

$$V_m^{Iter,\gamma} = G_m^{Iter,\gamma} + H_m^{Iter,\gamma} + S_m^{Iter,\gamma} + I_m^{Iter,\gamma} + D_m^{Iter,\gamma} + R_m^{Iter,\gamma} \ (4)$$

b) Grazing Behavior: The way a horse grazes reflects the horse's movement patterns to search for food and is determined mathematically as follows

$$G_m^{Iter,\mathrm{AGE}} = g^{Iter,AGE}.(u + Pl)[X_m^{Iter-1}] \quad (5)$$

where u and l are bounds, P is a random factor, and $g^{Iter,AGE}$ decays over iterations.

c) Defense Mechanism: Derived from their defense mechanism, horses tend to avoid negative areas like this one, where defenders are vulnerable.

$$D_m^{Iter,\text{AGE}} = -d^{Iter,AGE}[\frac{1}{qN}\sum_{j=1}^{qN} X_j^{Iter-1} - X_m^{Iter-1}] \quad (6)$$

where qN represents the subset of poorly performing solutions, and $d^{Iter,AGE}$ decays over time.

d) Roaming Behavior: The movement of young horses is seen to be random and exploratory in nature.

$$R_m^{Iter,\text{AGE}} = r^{Iter,AGE} P\, X_m^{Iter-1} \quad (7)$$

where $r^{Iter,AGE}$ is a decay factor.

3. Methodology

Our proposed methodology is a brand new and effective approach for performing fault detection in Wireless Sensor Networks (WSN) With this methodology, the Horse Optimization Algorithm (HOA) is integrated for feature selection and Convolutional Neural Networks (CNN) fused with Attention Mechanism to perform the classification task. This method combines the benefits of HOA and CNN equipped with attention mechanism in order to overcome the challenges faced by conventional detection systems.

Because horses exhibit herd behavior, the HOA algorithm exhibits this same behavior. Its major advantages are the effective feature selection, data dimensionality reduction, and system computational improvements. This algorithm is able to focus on the features that matter more, which lays the groundwork for an effective detection system. In contrast, Attention Mechanism CNNs greatly aid in the classification of the optimized feature set by focusing on more important attributes. With this combination, feature selection optimization is possible and detection accuracy is higher with less false alarms.

These two algorithms not only enhance the detection process, but also improve efficiency and scalability which makes this method suitable for dynamic and large-scale wireless sensor networks.

In the initial stage, the HOA is put into consideration, which draws inspiration from the herding behaviour of horses, to feature selection and extraction for the system to learn. This is important as it lowers the volume of information required to process the data and the amount of calculations as well, which in turn allows the system to concentrate its efforts on attributes that are more relevant when attempting to find faults. In the second stage, the subset of features is classified through the use of convolution neural networks with an integrated attention mechanism. This focus on the particular feature improves the model’s classification accuracy. These two approaches work together to mitigate the shortcomings of traditional automated fault detection schemes such as high false detection rates and poor performance. The remaining parts of this document outline the details of the HOA and how it is employed for feature selection, and the CNN design which incorporates the attention mechanism deep within the structure. Furthermore, how these components are combined to achieve higher fault detection rates is addressed. End of course are the details regarding implementation and evaluation of the proposed technique in terms of performance and complexity detection. This step has reduced the workload of activities to be computed and greatly increased the accuracy of the detection of anomalies. This is a major step in the development of security and operational effectiveness of wireless sensor networks.

Machine learning and data analysis proceeds in several steps, the first one being preprocessing. Preprocessing involves steps such as cleaning, transforming, and organizing the data to make it ready for further processing. The goal of data preprocessing is to enhance the quality of data so that it is appropriate for utilization in subsequent analysis. This procedure largely addresses the effects of missing values, inconsistencies, and differences in value scales, thereby increasing the effectiveness and accuracy of machine learning models. For this research, three steps are taken during preprocessing to make the data ready for feature selection and classification. First, to address the concern of missing values, imputation is performed using the average of neighboring values. This method fills the gaps in the dataset by replacing missing values with the average of nearby values, which decreases the bias that could be introduced when estimating missing data. After that, Min-Max Normalization is performed to prevent scale differences from affecting the performance of the model. This method changes the range of data points to a common range, usually between 0 and 1, which ensures every feature is treated equally and improves the learning process of the model.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (8)$$

At this point, the data set is shuffled to mitigate the presence of data instances with relative sequential order in order to eliminate any existing patterns that could lead to a certain level of bias in the learning algorithm. In this way, the model is only dependent on the learned relationships between features and targets, and not the possible relationships contained in the order of the original data.

With these preprocessing techniques, the dataset is enriched and made ready for what is planned to be done with it, that is, to enable the smooth and painless performance of the following feature selection and classification tasks.

Equally important is the feature selection phase that follows data preprocessing. This is done by utilizing the Horse Herd Optimization Algorithm (HOA) which is an optimization algorithm developed from meta-heuristics for solving systems with many dimensions. The algorithm captures the social and natural characteristics of horses through six behavioral models which are grazing, hierarchy, sociability, imitation, defense, and roaming. These behavioral models ensure that the algorithm performs both exploration, searching for optimal solutions, and exploitation, focusing on the solutions already discovered. The movement of each horse in HOA is modeled with respect to his age group and each age category influences horse behavior. There are four age categories: young (δ), adolescent (γ), adult (β), and old (α). Each group has their own optimization contribution.

For feature selection, it is adequate that the herd members are encoded based on the attribute that correlates with the objective function, and that function is created around the Convolutional Neural Network (CNN) error rate. The objective function is formulated in Equation 7.

$$MSE = \frac{1}{n}\sum_{i-1}^{n}(Y_i - \hat{Y}_i)^2 \quad (9)$$

The algorithm's termination condition - which is the maximum number of iterations stipulated - is reached when the appropriate subset of features is achieved. Following the selection of optimal features, the dataset is then split into training and testing subsets in an 80:20 ratio.

Lastly, the data that has been selected and processed is now going to be classified through a Convolutional Neural Network, or CNN, with the Attention Mechanism Architecture. The purpose of this network is to identify deep informative features from the data through attention ‘focusing’ mechanisms. CNNs excel at identifying patterns and local features within data structured in a matrix or image format and for that reason, they are chosen. Here, the CNN serves as a feature extraction methodology that transforms the selected features into higher-level representations.

Adding the Attention Mechanism enables the network to classify data whilst focusing on more critical data features within the sample as it is more informative. The information in the data to be classified is not uniform and thus, the Attention Mechanism comes in handy by focusing on features with positional significance that have greater impact on the final results. In this Reaserch, the Attention Mechanism creates an Attention Map to determine the most important areas within the data and ensure that the network focuses on those regions.

In this work, the CNN encompasses hidden convolutional layers responsible for the initial feature extraction from the incoming data and the processing of such local information with trainable filters. The outputs of the layers are lternatively hyperbolic tangent activations that enhance nonlinear relationships while providing normalization to improve stability and efficiency during learning. This enables the model to mitigate less relevant data and focus more on the important features by amplifying the more critical information and lowering the Attention Mechanism. After feature extraction, the features are added to the final categories using fully connected layers. The output layer applies classification of the data into target classes using Softmax or any other appropriate activation function.

The CNN is being trained by making use of the training dataset created in the prior steps. At this point, a loss function, like Cross-Entropy, is optimization criterion of the network, and the weights of the network are updated through different methods such as Adam during Repeated Forward Passes to adjust the optimization parameter, reduce loss, and increase accuracy of predictions. After this learning process, the model is tested against the test dataset and evaluated according to metrics such as Accuracy, False Alarm Rate, and Detection

Rate. This evaluation confirms that the model performs and generalize well sufficient enough to be effective in Faults detection. With the Attention Mechanism inclusion, the model's ability to prioritize critical features of the data has also increased with classification accuracy attained. Classification errors due to irrelevant data or noise have decreased. In conjunction with the attention mechanism used in the wireless sensor networks where data is complex and scattered, the performance of the fault detection systems is enhanced significantly. The final output of this step is a classification model for fault detection with high accuracy, which is essential to ensure security of wireless sensor networks.

Our proposed method is shown in Figure 4. This figure provides an overview of the entire methodology, highlighting the key phases of data preprocessing, feature extraction using CNN and HOA.

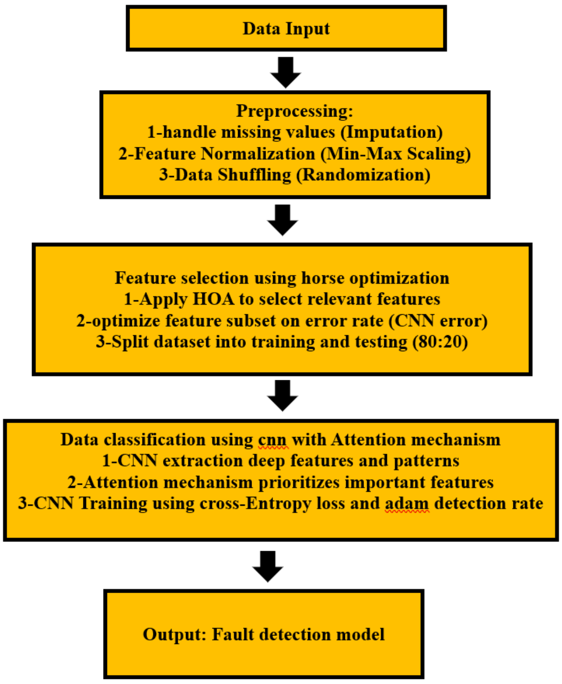

Figure 4 The steps of our proposed method

4. Dataset

In this Article, we worked with the data set mentioned in [25] With the purpose of fault diagnosis in WSNs. This data set was part of another data set obtained in 2010 by researchers from the University of North Carolina in Greensboro [26]. Using TelosB Mote devices that have sensors processors and radio devices, they obtained data from a single and multi-hop Wireless Sensor Network (WSN) configuration. This data set included temperature and humidity measurements every five seconds over a period of six hours. Also, specific events were triggered in the network to exhibit different behaviors like the pouring of warm water into the network to increase humidity and temperature. Their study aimed at classifying the data in two categories, normal data and data which was flagged as an anomaly. The data set that they utilized was a set of open space data which was captured by multi hop wireless sensor networks. This data set has a large volume of records, with each one having 12 feature dimensions. Each feature vector represents measurements of three time samples (t0, t1 and t2). For every time sample, they gathered two temperature readings and two humidity readings. In total, they created 21 datasets which contained 4688 samples altogether.

These datasets encompassed various fault types that incorporate offset, gain, out-of-range, and stuck-at errors, in addition to diverse β values and different error rates such as 50%, 40%, 30%, 20%, and 10%. In summary, the errors used in this dataset are defined as follows:

- Offset Errors: If the output from a certain sensor differs from the standard or true value to a certain extent when a reading is taken, offset errors are present. Changes in temperature, aging of the sensor, or improper calibration can be some of the contributing factors of offset errors. A constant value can be subtracted from the sensor's measured reading to rectify offset errors.

- Gain Errors: Gain errors are present when a constant scaling factor is applied when taking measurements through a sensor when there is a constant Gain errors can also manifest as a result of improper settings on the amplifier, flaws in wiring which lead to voltage drops, or improper calibration. Using a multiplying factor on the sensor's measured reading will correct the gain error.

- Out-of-Range Errors: When a sensor's reading strays away from the accepted range, this type of error occurs. For instance, if a temperature reading is taken but the output from the temperature sensor is beyond the assigned range, it indicates that either the sensor or the whole system has a fault. Blunders in this category can be minimized or completely avoided by setting upper and lower bounds for readings and taking appropriate measures when such boundaries are crossed.

- Stuck-at Errors: If a sensor is frozen at a value and does not reflect any changes to the physical parameters, it will give rise to a stuck-at error. This error type can result from broken sensor parts or bad programming. An alarm can reduce stuck-at errors and replacing the sensor is another course of action.

5. Evaluation Metrics

In this specific Article, we determine the fault detection methodology based on a few specific metrics, which enable us to evaluate the functioning of the Autoencoder-LSSVM model. These metrics are obtained from the confusion matrix, which we have discussed previously and comprises of True Negatives, True Positives, False Positives and False Negatives.

1. Accuracy: Accuracy measures the number of errors made by our model, in this instance fault detection model. It is computed as TN + TP divided by the total number of instances. In this case, accuracy measures how many times the normal and abnormal behaviors of the WSN data are correctly determined.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

2. Precision: Precision concentrates on the accurate hypotheses made by our model (i.e., TP and FP), which means, statistics which were regards as positive outcomes when predicted as anomalies. In our study, precision assesses how well our model minimizes false alarms, which means, ensuring that the predictions about operational fault (anomalous behavior) are correct.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

3. Recall (Sensitivity): Recall evaluates the model's ability to identify all actual anomalies (TP) out of all instances that are truly anomalous (TP + FN). In our context, recall emphasizes the model's effectiveness in capturing all instances of true fault occurrences, minimizing the likelihood of missing any.

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

4. F1-Score: F1-Score is defined as the weighted average of precision and recall measures. It provides a more nuanced assessment as it incorporates the flaws of both false positives and undetected instances of events. In the thesis, F1 Score is given as a combined measure of precision and recall with the goal of quantifying the effectiveness of the model from the perspective of fault detection.

$$F_1 Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (13)$$

These evaluation metrics collectively help us gauge the effectiveness of our Autoencoder-LSSVM model in identifying and classifying normal and anomalous behavior within WSNs. By considering these metrics alongside the confusion matrix metrics (TN, TP, FP, FN), we gain a holistic view of our model's performance, ensuring that it maintains a balance between minimizing false alarms and missing actual fault occurrences in WSN.

6. Simulation results

This section presents simulation results for the proposed fault detection system in Wireless Sensor Networks (WSNs), focusing on the effectiveness of the CNN-based feature extraction and the impact of the Horse Optimization Algorithm (HOA)-optimized parameters on detection accuracy. Evaluations include analyzing the system's performance under various fault scenarios, assessing its robustness against noise and environmental variations, and measuring its capability to distinguish between normal and faulty states. The results also explore the influence of key hyperparameters on the system's overall detection accuracy and reliability.The simulations were conducted using MATLAB 2022 on a system with an Intel Core i7 (4710HQ) @ 2.5 GHz CPU, 12GB RAM,

Data preprocessing is the first step upon receiving the dataset. At this stage, missing data values were filled with the average value of the nearest three neighbors in order to maintain the integrity of the dataset. Afterwards, Min-Max normalization was applied in order to scale the features and remove the effects of differing feature scales. It is done so that all features are treated with equal importance in the model's learning stage, thus mitigating any possible detrimental differences in scale. Next, complexity was reduced and key features selected by employing the Horse Herd Optimization Algorithm (HOA) with the parameters shown on table 1. The HOA algorithm which was developed based on a horse’s natural and social behavior, automatically extracted eight features from the dataset that were most useful using multi-dimensional optimization. An overview of the features selected base of Horse Herd Optimization Algorithm (HOA) is illustrated in figure 5. In this figure, the horizontal axis shows the feature index, while vertical axis displays the frequnecy of the selection of each identified feature in the selection process of the features. From the chart one can see that HOA determined and selected the eight features with greatest importance. These features are the foundation of the classification phase of the proposed method.

The chosen features were picked due to their influence on the drop in the neural network classification error and were then incorporated for advanced classification levels. These selected features helped to effectively enhance the performance and reduce the complexity of the proposed method.

Table 1:Horse Herd Optimization Algorithm Specifications for Feature Selection

Parameter	Description
Number of Selected Features (δ)	30 features (defined by num_features_to_select).
Maximum Iterations (Iter_max)	50 iterations (defined by max_iterations).
Horse Age Groups	All horses participate in optimization; age groups are not explicitly modeled in the implementation.
Objective Function	Minimize the classification error of the Convolutional Neural Network (CNN).
Scoring Metric	Random scoring using uniformly distributed values in the range [0, 1] for each feature in each iteration.
Feature Selection Criterion	Features with the highest cumulative scores over all iterations are selected.
Feature Reduction Method	Filter training (X_train) and testing (X_test) datasets based on the indices of selected features.
Input to the Algorithm	Training dataset (X_train) and corresponding labels (y_train).
Output of the Algorithm	Indices of the top 30 features with the highest importance in reducing classification error.

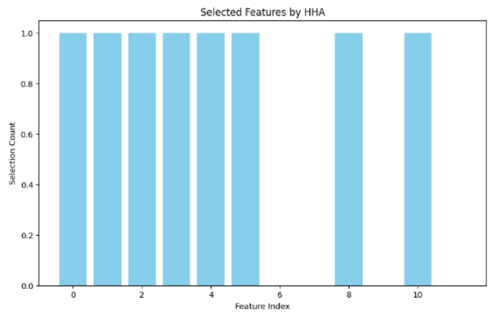

Figure 5 :features selected by the Horse Herd Optimization Algorithm

The last step of the process is done by integrating the features selected through Horse Herd Optimization Algorithm (HOA) into a Convolution Neural Network (CNN) with a newly added attention layer, which is capable of doing data classification. The network is established to process the chosen features and channel the attention towards the features that matter the most. The architecture of the attention network have two convolution layers for local feature extractions in the input data. The first on has 64 filters, 3 x 3 kernel size with padding 1, and the second one has 128 filters, which are exactly the same as the first. They are used in particular to learn the important structures of the data. A subsequent step taken after feature extraction is the implementation of the attention layer. In this step, attention weights are calculated for every feature. This layer mechanism puts more focus on important features and diminishes the effect of unimportant ones, which ameliorates the value of the important features during the classification stage. The output from the attention layer is passed to two fully connected layers. First of these layers is the encoder which consists of 512 neurons to learn high-level features, while the second one is the class predictor and has the same amount of neurons as the class numbers. The softmax activation function is employed on the output layer to make the prediction probabilities.

In an effort to lower the complexity of the data and eliminate the possibility of overfitting, MaxPooling layers were incorporated. In addition, the model was improved in terms of feature selection owing to the non-linearity that is provided by the ReLU activation function. The network was trained using CrossEntropyLoss on the loss Function, and the Adam optimizer on a learning rate of 0.001. Such architecture, with convolutional and attention layers, achieves efficient feature extraction and improves classification performance. One such architecture is the neural network designed in this work which is shown in Figure 6 which is a model he constructed**.**

Figure 6: Convolutional network architecture with attention mechanism

The neural network’s training activity is displayed in figure 7 depicting the improvement of training accuracy (blue) and training loss (red) over several epochs. The epochs are measured on the horizontal axis and the Accuracy and Loss are given on vertical axes. The accuracy of the model is about 50% as is shown in the start of the training process: this is because the network is just beginning to learn and therefore not able to distinguish important features from the data. The model’s accuracy significantly improves along the training phase with each new iteration and training loss goes down which indicates that the model is performing well. In the initial phases of training, when the model’s accuracy improves from 50% to above 99%, the training loss decreases substantially. This proves that the network is capable of identifying primary features of the data. Whenever the training process continues and the epochs increase, the gain in accuracy progressively declines until it levels off. Such stagnation conveys that the model has refined its performance to a stationary point, any further improvement in accuracy would not be pronounced. Likewise, the training loss starts at a high level but goes down slowly as the model gets trained more: near the end, the loss is always very low.

This reduction in loss and the simultaneous increase in accuracy implies that the model has been appropriately learning and effectively tuning its parameters. These metrics, over 99% accuracy and close to zero training loss, prove that the model was able to learn important features from the information and store them permanently. In addition, last phases of training having plateaued accuracy values accompanied by continuously declining loss, confirms there was no overfitting and learning was adequately achieved.

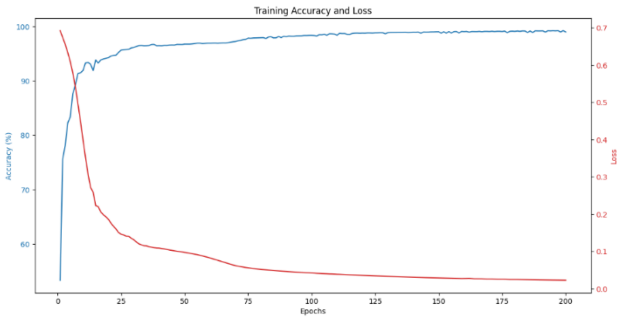

Figure 7: Neural network training process

This section gives a presentation of the analysis for the results gotten from the training and test datasets with the use of confusion matrices and Receiver Operating Characteristic (ROC) evaluations. Indeed these two metrics are sufficient to measure the model’s ability in clsifying faulty and non faulty instances in the wireless sensor network within the parameters set. In figure 8, confusion matrices for both the training and testing datasets are given. The left side displays the accurate results of the classifier with respect to the training dataset. The confusion matrix confirms that this model was able to capture the target concept in the training data set. The number of predicted non-faulty samples was 1135 and 1112 for predicted faulty samples, which were all accurately classified. There was only 1 faulty sample misclassified as non-faulty and 1 non-faulty sample misclassified as faulty. It is obvious from these results that the model is capable of learning the provided features of the training data very well. The right side of the figure provides the confusion matrix for the testing dataset. This matrix also exhibits that the model was able to predict the test samples well. For the test data set, the number of correctly predicted non-faulty samples is 267 and faulty ones is 293.

Likewise, the quantity of misclassified samples is rather small; being only 3 non-defective samples being incorrectly marked as defective while no defective samples are incorrectly marked as non-defective. The significant performance on the test set illustrates the distinguishing claim of the model, where it generalizes well and has high accuracy on test data, which was not encountered before. The effective results of the model on the two datasets, the training and the testing sets, show that he understands what the most important criteria for the faults in wireless sensor networks are and has applied them properly to the problem of fault detection.

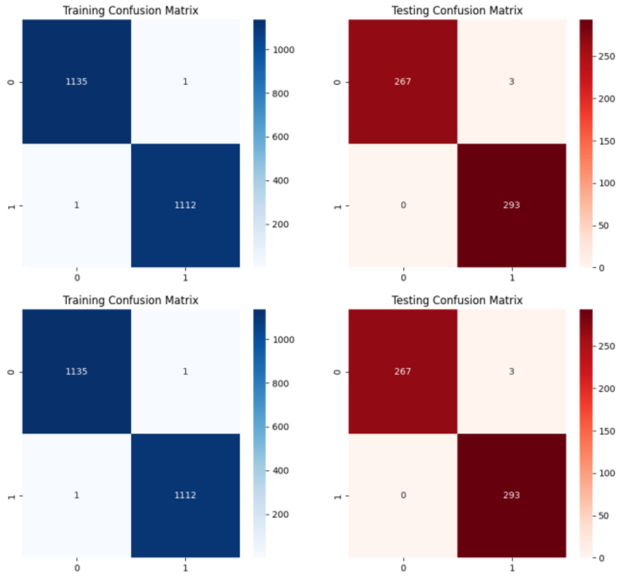

Figure 8: Confusion matrix for training and testing data

Figure 8 displays the ROC curve for both training and testing datasets. The ROC curve is a basic illustration of the performance of classification models. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) on a two-dimensional plane. While the negative FPR is on the horizontal X-axis of the graph, the Y-axis displays TPR which provides a measure of how many positive samples were positively classified. The random classifier line is the 45-degree diagonal line, which indicates the effectiveness of the model that does not differentiate positive and negative samples. If the model is better than a random classifier, the ROC curve will be above this line. The area above the line is called the acceptable area, which reveals the effectiveness of the model. The better the effectiveness of the model, the closer the ROC curve is to the diagonal axis.

The ROC curves for both training and test datasets lie in the desirable area as depicted in figure 9. This means that the model was able to tell the difference between faulty and non-faulty states in both datasets, and therefore have preformed well in both cases.

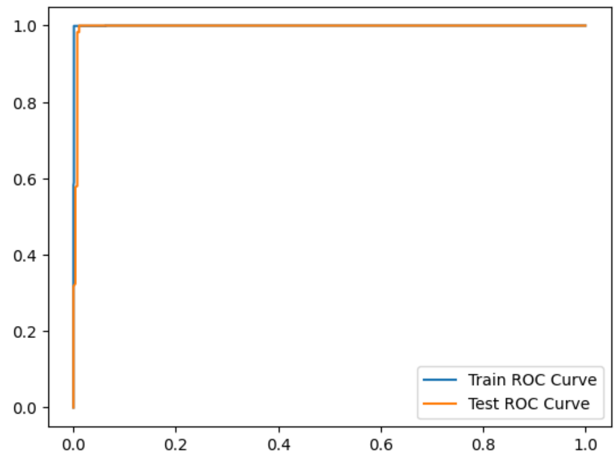

Figure 9: Receiver Operating Characteristic curve for training and testing data

Finally, Figure 10 summarizes the values obtained for the evaluation criteria from the training and testing datasets.

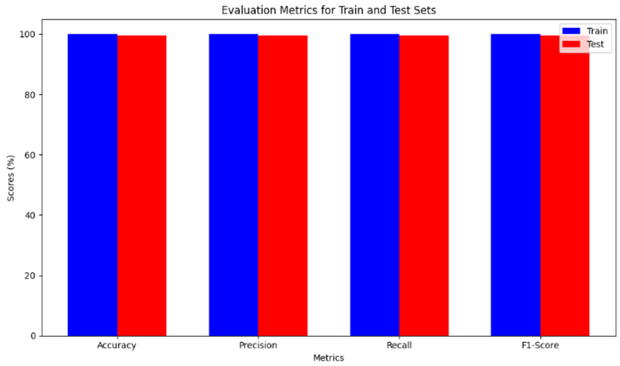

Figure 10: Values obtained for the evaluation criteria of the training and testing datasets

As the chart shows, four different parameters, which are the Accuracy, Precision, Recall, and F1-Score, have been averaged for both sets, training and testing Datos. The training set is marked by the blue bars, while the red ones indicate the testing set. From the studies, it can be seen that all four metrics fall within the average range of approximately 99%, which indicates the best performance of the model in determining the correct and incorrect states in the wireless sensor network. The Acurracies of both the training and test sets is very high and is approximately 99.63% for the training set and 99.47% for test set. This means that the model is highly accurate and very few samples are falsely detected, thus making the model very reliable and efficient in fault detection. Moreover, this finding provides information on how well balanced the model has learned the information without overfitting. The model's sucessful predictions of positive (fault) instances is represented my Precision, which for the training set is 99.63% and the testing set has 99.40%. This means, there is a very minimal amount of false positives, ensuring the model is trustworthy in terms of fault detection.

The model's ability to detect real 'faults' and the Recall calculated is very high for the model which is at 99.23% for the training set and 99.21% for the test set. This indicates that the model has been able to isolate every possible fault instance, so it is confirmed that critical samples were not missed. In relation to this, the F1-Score, which is obtained through the harmonic mean of Precision and Recall, achieved 99.23% and 99.21% for test and training set respectively This ratio indicates that recall was not enhanced at the expense of precision within the model. Thus, there is proof of the relative levels of Faults and Non-faults being classified and structured, showing clarity and structure within the model. These results confirm the model's expectation capability and performance, which is said to be precisely normal. The trained and tested data are rather similar in regards to their performance, meaning that it can be assumed the model learned the important features, without overfitting, and it is still capable of finding the faults on new data.

7. Comparison

In this part, we will perform a complete assessment for the proposed fault detection technique with respect to other methods that have been previously investigated in the field of Wireless Sensor Networks (WSNs). Our goal is to understand the strengths and the specific contribution of the CNN-HOA based approach as compared to other existing techniques. We hope to prove through comparisons that our model is more usable, sophisticated in terms of adaptability, more robust, and is updated to advanced levels of fault detection levels in WSNs.

The rapid expansion of devices with sensors due to the internet of things brings up the growing dependency on sensor data for decision-making processes across a variety of services and systems and with this technology also comes disadvantages. Among these challenges, high rates of sensor faults are greatly anticipated, bringing in the need for prolific sensor fault detection methods. Article [21] proposes a singular focus fault detection algorithm that approaches the task with the aid of a digital twin strategy. Article [21] demonstrates the efficacy of this notion by presenting a method for the detection of a specific type of fault using multiple sensors. A case was made for using a Generative Adversarial Network (GAN) to form a virtual version of the sensor itself. GANS can perform exceptionally well when given image data, so GAF encoding was employed to convert time series data into images while retaining the order of events. The discriminator network was trained on these GAF images. When put to use, the discriminator, primed to detect normal patterns, is deployed as a fault detection mechanism. In normal state conditions the virtual sensor has adequate performance, successfully generating data as expected. Out of all the models that were evaluated, this one managed to achieve the highest fault detection rates of 98.7%. With such precision, it is reasonable to treat this GAN-based, DT-inspired method as a strong contender for accomplishing sensor fault detection.

There are difficulties and issues relating to software, hardware, and communication that are expected with the deployment of wireless sensor networks (WSNs) in volatile environments. These physical sensors, positioned in remote and unreachable spots, are vulnerable to nefarious activities. The sensed information must be checked for faults to mitigate losses in a timely manner and to enable the seamless functioning of the sensor networks. For the reason of fault classification, the study [22] identifies the use of classifiers in the application of machine learning for their proficiency in dividing the sensed data into faulty and non-faulty groups. Specifically examined in this study are the faults from Denial of Service, Probe, Remote to Local, and User to Root systems. The KDD CUP 99 dataset is used for these competitions for it contains 41 features, which are divided to 3 classes, content, basic, and TCP features. A recursive feature elimination approach is used to select the most relevant features for each of the specified faults. The newcomers test in these competitions are [A2] 2.0 and classify them for evaluation: Actionable Intelligence, accuracy, F-measures, precision, and recall. The results of the experiments highlight that the Random Forest classifier has the superior outcome in terms of fault detection for Wireless Sensor Networks. Also, simulation outcomes confirm that the Multi-layer perceptron classifier gives the best outcome with an accuracy of 92 percent.

The monitoring, detection, and identification of faults in Wireless Sensor Networks (WSNs) is highly crucial because of the magnitude of shortcomings that may occur. As WSNs are heterogeneous cyber-physical systems that are prone to defects, adequate treatment and management of the issues at hand is vital. Towards the realization of this complex problem, reference [23] employed a supervised machine learning strategy. The issue of class imbalance is addressed through RUS techniques and the analysis of sensor data for contradiction is done through the ET classification algorithm.

The efficiency of the suggested CNN-HOA based methodology is thoroughly benchmarked against contemporary Machine Learning algorithms like Support Vector Machine (SVM) and Random Forest (RF). The effectiveness of the proposed approach is measured through key metrics such as Accuracy, Recall, Precision, F1 Score, and AUC ROC. The results prove the significant benefits obtained by employing HOA for CNN parameters which in turn improves the figure of merit and robustness for fault detection with WSN data. More significantly, the proposed approach of CNN-HOA shows the best results for all other models and demonstrates the best performance across all configurations.

Table 2 provides a detailed and exhaust comparative study, providing deeper insights on the results of our studies in light of the results of previous studies.

Reference	Method	Accuracy
[27]	Generative Adversarial Network (GAN) with Gramian Angular Field (GAF)	98.7 %
[28]	Multi-layer perceptron classifier	92 %
[29]	Random Under Sampling (RUS) and the Extra-Tree (ET) algorithm	96 %
Our proposed approach	LSSVM and Autoencoder	99.47

8. Conclusion

One of the many issues that the WSNs world comes with is the detection of faults in a reliable and timely manner for critical data loss. In this article, the problem has been addressed thoroughly and subsequently led to the development of a fully-functional WSN fault detection approach. Our groundwork for this began with an application of thorough data preprocessing such as: the min-max normalizaton, sample window rearrangement, data cleaning, and data partitioning. Not only did these steps help set the dataset for later analysis, but they also formed the foundation of the remaining steps of our methodology.

The core of the approach that we used was adding CNN's with the HOA. A CNN model used in this paper was designed to detect faults within WSN while learning its high-dimensional features from the data. After this, the HOA was used to set the following attributes of the CNN: the learning rate, the number of filters, and the size of the convolutional kernels. This multi-step optimization ensures that the CNN is working at its peak efficiency and accuracy.

The accuracy rates were extraordinarily high after training the CNN model on the dataset. The dataset used for training had an accuracy of 99.63%, while the testing dataset had a relatively high accuracy of 99.47%. These results, as shown in previous sections of work, confirm the accuracy and dependability of the CNN-HOA-based algorithm model developed by us for fault diagnosis of WSN.

In summary, this paper has tackled the issue of the fault detection challenge in WSNs and has given a pragmatic, realistic solution that can be implemented. The combination of CNN and HOA, intensive data preprocessing, have given our methodology the ability to accurately detect anomalies. This technique aids in improving the fault tolerance of WSN but also reduces the chances of data loss in sensitive applications of the network which further contributes to the advancement of WSN fault detection.

References

[1] Panda, M., Gouda, B. S., & Panigrahi, T. (2020). Fault diagnosis in wireless sensor networks using a neural network constructed by deep learning technique. Nature Inspired Computing for Wireless Sensor Networks, 77-101.

[2] De, D., Mukherjee, A., Das, S. K., & Dey, N. (Eds.). (2020). Nature inspired computing for wireless sensor networks. Singapore: Springer.

[3] Lee, M. H., & Choi, Y. H. (2008). Fault detection of wireless sensor networks. Computer Communications, 31(14), 3469-3475.

[4] Mazibuco, V. A., Nhung, N. P., & Linh, N. T. (2023). Fault detection in wireless sensor networks with deep neural networks. Journal of Military Science and Technology,(CSCE7), 27-36.

[5] Singh, Y., Rathi, R., Prasad, R., & Baghel, R. K. (2023, September). Fault Detection in Wireless Sensor Networks Using Autoencoder Classifier. In 2023 6th International Conference on Contemporary Computing and Informatics (IC3I) (Vol. 6, pp. 1563-1568). IEEE.

[6] Mohapatra, S., Khilar, P. M., & Swain, R. R. (2019). Fault diagnosis in wireless sensor network using clonal selection principle and probabilistic neural network approach. International Journal of Communication Systems, 32(16), e4138.

[7] Bista, R., & Chaudhary, M. (2022, December). A New Fault Detection Approach in Wireless Sensor Networks. In 2022 14th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 187-191). IEEE.

[8] Prasad, R., & Baghel, R. K. (2021). A novel fault diagnosis technique for wireless sensor network using feedforward neural network. IEEE Sensors Letters, 6(1), 1-4.

[9] Noshad, Z., Javaid, N., Saba, T., Wadud, Z., Saleem, M. Q., Alzahrani, M. E., & Sheta, O. E. (2019). Fault detection in wireless sensor networks through the random forest classifier. Sensors, 19(7), 1568.

[10] Laiou, A., Malliou, C. M., Lenas, S. A., & Tsaoussidis, V. (2019). Autonomous Fault Detection and Diagnosis in Wireless Sensor Networks Using Decision Trees. J. Commun., 14(7), 544-552.

[11] Panda, M., Gouda, B. S., & Panigrahi, T. (2020). Fault diagnosis in wireless sensor networks using a neural network constructed by deep learning technique. In Nature Inspired Computing for Wireless Sensor Networks (pp. 77-101). Singapore: Springer Singapore.

[12] Gupta, S., Kaur, G., & Chanak, P. (2021, November). A deep bi-LSTM based fault detection algorithm for WSNs. In 2021 IEEE Bombay Section Signature Conference (IBSSC) (pp. 1-5). IEEE.

[13] Adamova, A., Zhukabayeva, T., & Mardenov, Y. (2023, May). Machine Learning in Action: An Analysis of its Application for Fault Detection in Wireless Sensor Networks. In 2023 IEEE International Conference on Smart Information Systems and Technologies (SIST) (pp. 506-511). IEEE.

[14] Azzouz, I., Boussaid, B., Zouinkhi, A., & Abdelkrim, M. N. (2020, December). Multi-faults classification in WSN: A deep learning approach. In 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) (pp. 343-348). IEEE.

[15] Biswas, P., Charitha, R., Gavel, S., & Raghuvanshi, A. S. (2019, April). Fault detection using hybrid of KF-ELM for wireless sensor networks. In 2019 3rd international conference on trends in electronics and informatics (ICOEI) (pp. 746-750). IEEE.

[16] Karmarkar, A., Chanak, P., & Kumar, N. (2020, February). An optimized svm based fault diagnosis scheme for wireless sensor networks. In 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-7). IEEE.

[17] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.

[18] Wu, J. (2017). Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China, 5(23), 495.

[19] Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9, 611-629.

[20] Ketkar, N., Moolayil, J., Ketkar, N., & Moolayil, J. (2021). Convolutional neural networks. Deep learning with Python: learn best practices of deep learning models with PyTorch, 197-242.

[21] Kim, P., & Kim, P. (2017). Convolutional neural network. MATLAB deep learning: with machine learning, neural networks and artificial intelligence, 121-147.

[22] Pinaya, W. H. L., Vieira, S., Garcia-Dias, R., & Mechelli, A. (2020). Convolutional neural networks. In Machine learning (pp. 173-191). Academic Press.

[23] MiarNaeimi, F., Azizyan, G., & Rashki, M. (2021). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. Knowledge-Based Systems, 213, 106711.

[24] Moldovan, D. (2022). Binary horse optimization algorithm for feature selection. Algorithms, 15(5), 156.

[25] Miathali, P. G. (2023). Efficient Machine Learning Classifier for Fault Detection in Wireless Sensor Networks.

[26] Wardhani, L. K., Febriyanto, R. A., & Anggraini, N. (2022, September). Fault Detection in Wireless Sensor Networks Data Using Random Under Sampling and Extra-Tree Algorithm. In 2022 10th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-6). IEEE.

[27] Salah Zidi, Tarek Moulahi and Bechir Alaya "Fault detection in Wireless Sensor Networks through SVM classifier", IEEE sensor journal, Vol. 18, NO.1, Jan 2018.

[28] Shan Suthaharan, Mohammed Alzahrani, Sutharshan Rajasegarar, Christopher Leckie and Marimuthu Palaniswami, Labelled Data Collection for Anomaly Detection in Wireless Sensor Networks, in Proceedings of the Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2010), Brisbane, Australia, Dec 2010.

[29] Hasan, M. N., Jan, S. U., & Koo, I. (2023). Wasserstein GAN-based Digital Twin Inspired Model for Early Drift Fault Detection in Wireless Sensor Networks. IEEE Sensors Journal.