Electronics and Electrical Engineering

AIMS Electronics and Electrical Engineering, 8(4): 508–541.
DOI: 10.3934/electreng.2024024
Received: 03 August 2024
Revised: 19 September 2024
Accepted: 02 November 2024
Published: 29 November 2024

https://www.aimspress.com/journal/ElectrEng

Research article

Hybrid optimized artificial neural network using Latin hypercube sampling and Bayesian optimization for detection, classification and location of faults in transmission lines

Abdul Yussif Seidu, Elvis Twumasi* and Emmanuel Assuming Frimpong

Department of Electrical and Electronic Engineering, College of Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

*** Correspondence:** Email: etwumasi.coe@knust.edu.gh; Tel: +233-024-865-6932.

Abstract: This paper introduces a novel hybrid approach that integrates Latin hypercube sampling (LHS) and Bayesian optimization for optimizing artificial neural networks (ANNs) in fault detection, classification, and location for transmission lines. The proposed method advances the accuracy and efficiency of fault diagnosis in power systems, representing a significant step forward compared to conventional approaches. The test system is a 400 kV, 50 Hz, 300 km transmission system, and the simulations were carried out in MATLAB/Simulink environment. Using the strategic insight of LHS, optimal initial points were determined, which subsequently formed the basis for the Bayesian optimization to further refine the learning rate and training epochs. Within the fault detection domain, the model showcased remarkable precision when deployed on an evaluation dataset of 168 cases, accurately detecting every instance of normal and faulty scenarios. This culminated in an astounding 100% accuracy in fault detection. In terms of fault classification, the ANN model, trained on a dataset of 495 instances, revealed perfect regression coefficients across the training, testing, and validation subsets. When tested against unseen data, it demonstrated its proficiency by correctly classifying 154 out of 154 cases, showcasing a 100% F1 score. Also, the accuracy figures in terms of fault location fluctuated between 99.826% and 99.999%, with a mean absolute percentage error (MAPE) of 0.053%. The model's mean square error (MSE) stood at 0.0083, while the mean absolute error (MAE) was calculated at 0.0717. A deep dive into diverse fault types reaffirmed the model's precision, underscoring its consistent performance across various fault scenarios. The trained models were deployed on three different transmission lines and the models exhibited remarkable precision in all the cases tested. In summary, the innovative hybrid optimized ANN model, weaving together the

strengths of LHS and Bayesian optimization, signifies an advancement in the field of power system fault analysis, ushering in heightened efficiency and reliability.

Keywords: Latin; hypercube; Bayesian; optimization; transmission; artificial neural network

1. Introduction

As an indispensable part of modern civilization, power systems are the lifeline supporting various sectors, including residential, commercial, and industrial domains. The continuous and reliable functioning of these systems is, therefore, a critical concern. One of the primary challenges that threaten the integrity of power systems is the occurrence of faults in power transmission lines. These faults, if undetected or misclassified, can cause significant disruptions, leading to economic losses and potential safety hazards [1].

Fault detection, classification, and location methods play a crucial role in ensuring the reliability and stability of power transmission networks [2]. In these intricate systems, the occurrence of faults can lead to disruptive outages and potentially severe consequences, underscoring the urgency for accurate and swift fault analysis to trigger timely corrective actions. Traditional approaches often rely heavily on domain expertise and manual interventions, thereby limiting their scalability and applicability in real-time scenarios [2]. With the emergence of advanced technologies, such as artificial neural networks (ANNs) [1,3], support vector machines (SVMs) [4–6], and evolutionary algorithms [7], the landscape of fault analysis has undergone a transformative shift. These modern methodologies harness the power of machine learning, pattern recognition, and optimization techniques to significantly enhance fault detection accuracy, facilitate the classification of various fault types, and precisely locate faults within the distribution network. As power systems continue to grow in complexity and scale, these innovative approaches offer the promising potential to optimize the efficiency, precision, and speed of fault analysis, ultimately contributing to the establishment of more dependable and resilient power distribution networks.

The versatility of convolutional neural networks (CNNs) becomes evident in their capacity to adeptly learn intricate features from a wide range of data, consequently enabling accurate fault identification and precise location in power transmission networks [8]. However, the inherent black-box nature of CNNs presents challenges related to interpretability, potentially hindering a clear understanding of the decision-making processes underlying their outcomes [9]. Moreover, CNN training demands considerable computational resources and extended timeframes, potentially impeding their deployment in real-time applications [10]. Additionally, the effectiveness of CNNs heavily relies on the availability of extensive and representative labeled datasets, which could pose limitations in scenarios characterized by limited data availability [11]. Striking a balance between these advantages and challenges is paramount when harnessing CNNs for fault analysis within power transmission networks.

The backpropagation (BP) algorithm, initially proposed by [1,3], stands as a pivotal technique for enhancing fault classification, detection, and location using ANNs. Through iterative weight adjustments driven by error propagation, BP bolsters fault detection, classification, and location accuracy, enabling ANNs to grasp intricate patterns within complex data [1]. This adaptability to nonlinear relationships in power system data renders BP well-suited for fault detection

applications [12]. However, its effectiveness hinges on a well-structured architecture and sufficient labeled data [13], and its susceptibility to issues like vanishing gradients and slow convergence must be acknowledged [14]. Nonetheless, the demonstrated success of the BP algorithm solidifies its crucial role in advancing fault detection and classification [1,3].

Various methods have been explored for fault classification, detection, and location in power distribution networks. Model-based approaches offer accurate fault detection based on system models but rely on precise model representation and might struggle with model mismatch. Expert systems utilize rule-based decision-making, providing interpretability but requiring extensive domain knowledge and manual rule creation. Machine learning techniques, including support vector machines (SVMs), exhibit strong generalization capabilities but might demand large amounts of labeled data for training. Evolutionary algorithms offer a global search strategy but might have longer convergence times. Hybrid methods, which combine multiple techniques, aim to harness the strengths of different approaches, but their complexity might lead to increased computational requirements. These methods underscore the tradeoffs between accuracy, interpretability, data requirements, and computational efficiency, presenting a spectrum of choices for enhancing fault analysis in power distribution networks.

In transmission lines, several types of faults can occur, which can disrupt the normal operation of power systems. These include single-line-to-ground (SLG), line-to-line (LL), double-line-to-ground (DLG), and three-phase faults [15]. SLG faults are the most common, occurring when one phase comes in contact with the ground. LL faults involve a short circuit between two phases, while DLG faults occur when two phases are short to the ground. Three-phase faults are rare but the most severe, as they involve a short circuit between all three phases [16]. Each type of fault exhibits different characteristics in terms of voltage and current waveforms, making accurate detection and classification critical for power system reliability [15,17,18]

The meticulous calibration of learning rates and epochs is pivotal for efficient training in ANNs. As emphasized by [19,20], the right balance guarantees swift and effective convergence, safeguarding against potential pitfalls of underfitting or overfitting. Furthermore, adaptive methodologies, like those proposed by [21], have shown promise in optimizing this equilibrium, automating the intricate balance between learning rate and epoch settings. Early stopping, supported by findings from [22,23], offers an efficient strategy to prevent overfitting by monitoring validation set performance. Ultimately, leveraging these advancements in training and optimization techniques can lead to a more robust and streamlined ANN training process [24,25].

Latin hypercube sampling (LHS) is recognized for its efficient, scalable exploration of hyperparameter spaces, offering uniform and comprehensive coverage [26]. Its adaptability to various machine learning algorithms and hyperparameters sets it apart [26,27]. Moreover, its parallel processing capabilities and lack of preconceived assumptions enhance its utility in hyperparameter tuning [28]. Bayesian optimization (BO) uses probabilistic models like Gaussian processes, balancing exploration and exploitation in hyperparameter tuning [29]. Its sequential and adaptive nature ensures informed decisions and quicker convergence [30]. Despite challenges in high-dimensional spaces, its versatility and the ability to integrate prior knowledge bolster its effectiveness [28].

While various methods for fault detection, classification, and location in transmission lines exist, most suffer from tradeoffs between accuracy and computational efficiency. Techniques such as grid search and random search often face limitations like long convergence times and suboptimal

solutions [30]. Overfitting and underfitting remain significant challenges in fault detection models, where overly complex models may fit the noise in the data, and overly simple models may fail to capture essential fault patterns [3]. Balancing these issues is critical for ensuring that models generalize well to unseen data while maintaining high accuracy in fault detection, classification, and location.

LHS has been explored in other domains for efficiently covering parameter spaces [31], but its application in transmission line fault detection remains underexplored. Additionally, BO, while effective for hyperparameter tuning, is rarely combined with LHS in power system fault detection, classification, and location applications. This paper addresses these gaps by proposing a hybrid LHS–BO optimized ANN, designed to mitigate overfitting and underfitting while improving the precision, efficiency, and generalization of fault detection, classification, and location in transmission lines.

The novelty of this approach lies in the integration of LHS for diverse initial sampling of the hyperparameter space, followed by BO for targeted optimization, which balances exploration and exploitation to find optimal model configurations' hyperparameters. This hybrid strategy reduces overfitting by preventing the model from becoming overly sensitive to specific training data, while also reducing underfitting by ensuring the model complexity is appropriate for the fault detection, classification, and location tasks.

2. Materials and methods

This section presents the various theoretical concepts employed in this study for the development of the algorithm for the detection, classification, and location of transmission line faults.

2.1. Latin hypercube sampling

Latin hypercube sampling (LHS) is a systematic and efficient sampling technique used to create representative samples of a multidimensional parameter space [32]. It aims to improve the coverage and diversity of the sampled space compared to simple random sampling. LHS is particularly useful in various scientific and engineering applications, including sensitivity analysis, uncertainty quantification, and optimization, where exploring a wide range of input parameter combinations is essential [31,33]. LHS builds upon the concept of dividing each input parameter into equally probable intervals and then selecting a single value from each interval to form a sample point. The key idea is to ensure that no two sample points share the same value in the same interval for any parameter (hence "Latin" in the name). This property helps in improving the space-filling and diversity of the sample.

2.2. Bayesian optimization

Bayesian optimization (BO) is a global optimization technique used to find the optimal configuration of a costly-to-evaluate objective function [24]. It works iteratively by building a surrogate model, typically a Gaussian process (GP), of the objective function and then using an acquisition function to decide where to evaluate the true objective function [34]. The acquisition

function guides the search by balancing exploration (sampling in uncertain regions) and exploitation (sampling in regions with high predicted values).

2.2.1. Expected improvement (EI) acquisition function

The expected improvement acquisition function quantifies the potential of improving over the current best value f_{best} at each evaluation point. It is defined as follows in (1):

$$EI(x) = \begin{cases} (f(x) - f_{best}) \times \Phi(Z) + \sigma(x) \times \phi(Z), \sigma(x) > 0 \\ 0, \qquad\qquad\qquad\qquad\quad \sigma(0) = 0 \end{cases} \tag{1}$$

Where:

- $f(x)$ is the predicted mean of the GP at point x.
- f_{best} is the best-observed function value so far.
- $\sigma(x)$ is the predicted standard deviation of the GP at point x.
- $\Phi(.)$ is the cumulative distribution function of the standard normal distribution.
- $\phi(.)$ is the probability density function of the standard normal distribution.
- $Z = \frac{f(x) - f_{best}}{\sigma(x)}$

The EI value is used to guide the selection of the next point to evaluate in Bayesian optimization. It represents the potential improvement over the current best value and is influenced by the uncertainty of the GP predictions.

2.2.2. Expected improvement per second plus $(EI - PS^{+})$

EI-PS+ is an extension of the EI acquisition function for optimizing real-world systems where evaluations take time. It takes into account both the expected improvement and the time taken for the evaluation. The acquisition function is modified to account for the tradeoff between improving the objective and the time spent on evaluation, as in (2):

$$EI - PS^{+}(x) = \frac{EI(x)}{T(x)} \tag{2}$$

Where:

- $EI(x)$ is the original expected improvement.
- $T(x)$ is the expected time required to evaluate the function at point *x*.

2.3. Artificial neural networks (ANNs)

Artificial neural networks (ANNs) are a cornerstone of artificial intelligence (AI) and one of the prime tools used in machine learning [3]. They are computational models that are inspired by the human brain's interconnected network of neurons [18]. ANNs mimic biological neural networks, comprising nodes organized in layers. Neurons process and transmit information via weighted connections. ANNs excel in pattern recognition, classification, and regression. They learn by adjusting connection weights based on data, capturing intricate relationships.

2.3.1. Multi-layer feedforward neural network

A multi-layer feedforward neural network, also known as a feedforward neural network or multi-layer perceptron (MLP), is a type of ANN architecture in which information flows in one direction, from the input layer through the hidden layers to the output layer, without any cycles or loops [18]. This architecture is characterized by its ability to approximate complex nonlinear functions [1]. An illustration of a multi-layered feedforward neural network (FNN) with a single hidden layer is provided in Figure 1, showcasing inputs, outputs, biases, and the interconnected weights.

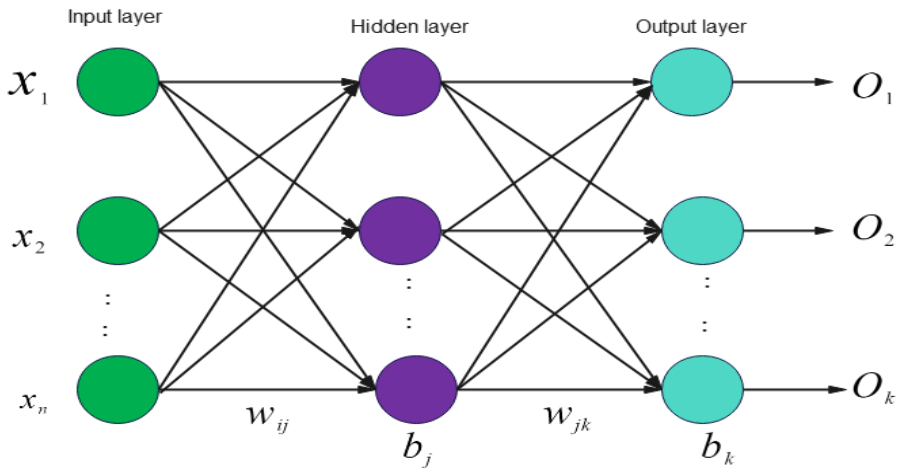

Figure 1. Architecture of FNN with one hidden layer.

In Figure 1, the input variables (x_1, x_2,… ,x_n) are represented, where w_{ij} signifies the input weights originating from the i^{th} input neuron toward the j^{th} hidden neuron. Additionally, b_j denotes the input bias associated with the j^{th} hidden neuron, w_{jk} represents the connection weight extending from the j^{th} hidden neuron to the k^{th} output neuron, and b_k signifies the bias pertaining to the k^{th} output neuron. The mathematical representation is shown in (3), where y is the output.

$$y = f(b + \sum_{i=1}^{n} w_i x_i) \tag{3}$$

2.3.2. Activation functions

Activation functions are essential for neural networks, providing the nonlinearity that shapes neuron outputs. Common functions include "tansig" (hyperbolic tangent sigmoid) and "purelin" (linear). Tansig uses the hyperbolic tangent to map inputs between -1 and 1, introducing nonlinearity that is crucial in hidden layers, with a smooth gradient for efficient learning. Purelin maintains the weighted sum of inputs without adding nonlinearity, making it useful for output layers that require continuous results, like regression tasks, where a wider range of output values is needed.

2.4. Normalization of 3-phase voltages and current

Following data collection, the next step is data preprocessing, which includes the normalization of the three-phase voltages and currents. In power systems, voltage and current measurements have

different magnitudes and variability. Standard min-max scaling, which normalizes all features to the same range (typically [0, 1]), might not capture the distinct nature of these measurements and hence is not ideal for fault data analysis for power transmission lines [35]. The range is defined according to (4) where X_{min} and X_{max} are the lower and upper bounds, respectively.

$$X_{norm} = \left(\frac{X - X_{min}}{X_{max} - X_{min}}\right) \tag{4}$$

2.4.1. Modified min-max scaling normalization

In fault data analysis for power transmission lines, modified min-max scaling is advantageous over standard scaling due to its adaptability to diverse data types [35,36], like phase voltages and currents. This method tailors scaling ranges to specific feature characteristics, enhancing model accuracy and sensitivity in detecting faults. It effectively addresses skewed distributions and prevents information loss, crucial for accurately interpreting power system data and ensuring effective fault detection. This customized approach aligns closely with the unique requirements of power transmission systems analysis.

$$X_{norm} = \left(\frac{X - X_{min}}{X_{max} - X_{min}}\right) \times (B - A) + A \tag{5}$$

Where A and B are the new X_{min} and X_{max} values, respectively.

The phase voltages and currents are both normalized with the modified min-max scaling normalization (5) in the intervals $[A_1, B_1]$ and $[A_2, B_2]$, respectively.

This preprocessing technique transforms raw data, enabling data mining algorithms to focus on structural patterns rather than magnitudes. Real-world simulations often have outliers that affect ANN performance. Removing outliers ensures the model learns from consistent data.

2.5. Trained models' performance evaluation

The metrics used to evaluate the performance of the trained models are the mean square error (MSE), mean absolute percentage error (MAPE), and mean absolute error (MAE). These metrics are computed using (6), (7), and (8), respectively.

$$MSE = \frac{\sum_{i=1}^{n}[\text{Output}_i - \text{Target}_i]^2}{\text{Number of data sets (n)}} \tag{6}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{Output}_i - \text{Target}_i}{\text{Target}_i}\right| \times 100\% \tag{7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\text{Output}_i - \text{Target}_i| \tag{8}$$

2.5.1. F1 score

The F1 score is a metric that combines precision and recall to evaluate the performance of classification models, especially for imbalanced datasets [37]. It is the harmonic mean of precision and recall, providing a single score that reflects the balance between false positives and false

negatives as in (11). Precision is the percentage of correct positive predictions out of all positive predictions made as in (9) [37,38], while recall is the percentage of correct positive predictions out of all actual positives in the dataset as shown in (10). Accuracy measures the proportion of correctly classified samples out of the total samples. In binary classification, it is calculated as in (12).

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP)+False\ Positives\ (FP)} \tag{9}$$

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP)+False\ Negatives\ (FN)} \tag{10}$$

$$F-1\ Score = 2 \times \frac{Precision\ \times Recall}{Precision\ + Recall} \tag{11}$$

$$Accuracy = \frac{True\ Positives\ (TP)+True\ Negatives\ (TN)}{Total\ Samples} \tag{12}$$

2.6. Methodology

Next, the research for the development of the algorithm for the detection, classification, and location of transmission line faults is conducted.

2.6.1. Modeling 400 kV, 300 km transmission line in MATLAB simulink

This paper employed MATLAB Simulink to model a 400 kV, 300 km transmission line [39]. This software offers an intuitive interface for dynamic system modeling using block diagrams. The transmission line is modeled using a distributed parameter line model, which is fed from sources at both ends representing Thevenin's equivalent of the two sub-stations, as shown in Figure 2. The 300 km line was divided into 5 km segments for thorough analysis. Simulation under normal and faulty conditions provided comprehensive data for training an ANN. The model was simulated in both normal and faulty conditions. Normal operation represents an unblemished, baseline state. Faulty simulations mirror real-world power system scenarios. These include diverse faults like single line-to-ground, line-to-line, double line-to-ground, and three-phase faults. Faults were introduced at various positions every 5 km, assessing how location influences behavior. These diverse conditions offered vital data for robust ANN training. Data collected included three-phase voltages and currents. Simulink's "Voltage Measurement" and "Current Measurement" blocks captured measurements at each 5 km section. The processed data was partitioned into three subsets: 70% for training, 15% for testing, and 15% for validation.

The deployment data, multiples of 20 km, were withheld from training and reserved for model testing. This practice gauges the model's performance on new unseen data, mirroring real-world scenarios. Evaluating the ANN's capability to identify line conditions and faults in these unseen data segments reflects its real-world potential. The trained and optimized ANN models were deployed on three different transmission lines—220 kV, 100 km [40], 500 kV, 200 km [41], and 400 kV, 300 km [12]—after being tested on unseen data assessing their performance against underfitting, overfitting, and accuracy.

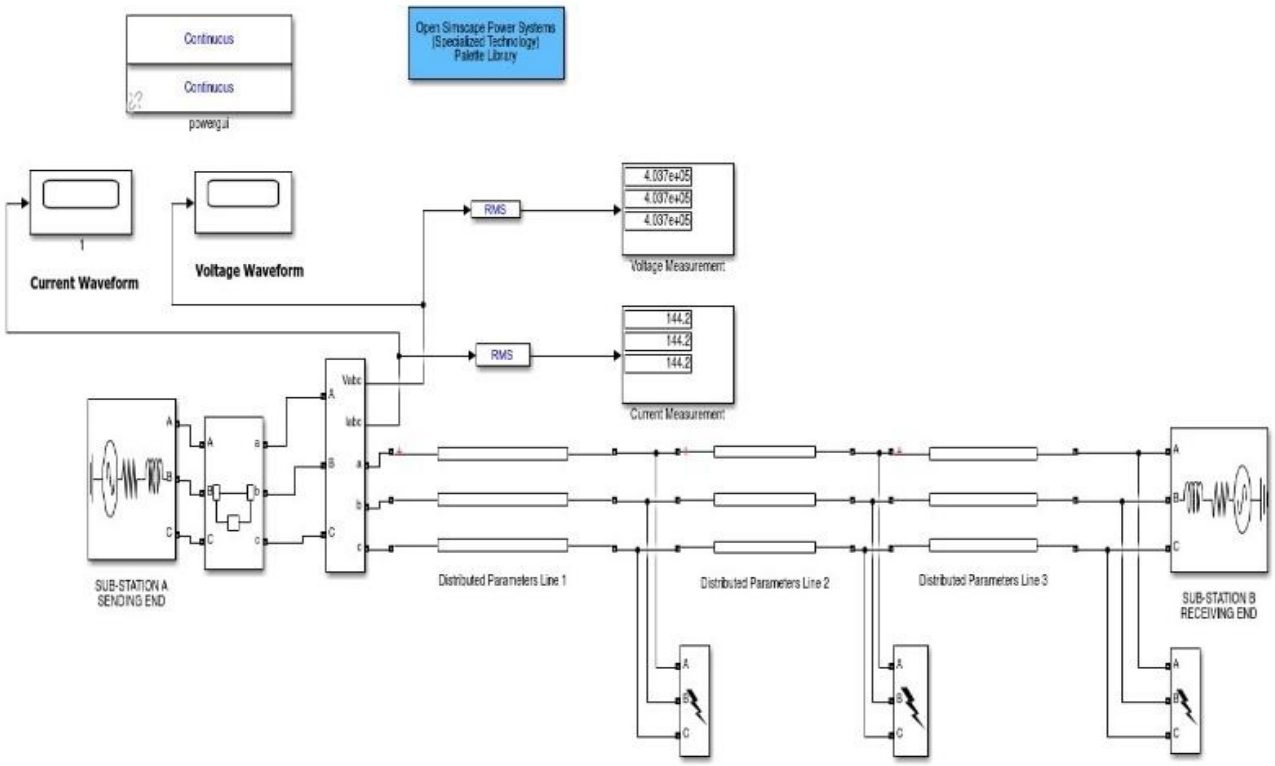

Figure 2. Simulink model of the transmission line.

2.6.2. Latin hypercube sampling (LHS) and Bayesian optimization (BO) algorithms

The LHS algorithm explores the hyperparameter space by generating samples that cover the entire range of each hyperparameter [34]. While Latin hypercube sampling (LHS) and Bayesian optimization (BO) have been applied in various optimization tasks, their combined application has not been explored especially for fault detection, classification, and location in power transmission lines. LHS is particularly effective in exploring the hyperparameter space by ensuring diverse initial conditions, while BO refines the search by focusing on the most promising areas identified by LHS. This hybrid approach significantly reduces the computational cost associated with hyperparameter tuning while improving model performance. The integration of these techniques promises a faster convergence to optimal solutions and more accurate fault classification. By evaluating the network's performance with different hyperparameter combinations, it aims to find promising regions in the hyperparameter space for subsequent optimization. The BO is illustrated in Figure 3, and the proposed framework for fault detection, classification, and location is also illustrated in Figure 4.

Latin hypercube sampling (LHS) algorithm:

1. *Initialize the number of LHS iterations and an empty array to store the LHS results.*

- *Set the desired number of LHS iterations to explore the hyperparameter space.*
- *Create an empty array to store the results of each LHS iteration.*

2. *Generate a Latin hypercube sample using the specified number of iterations and the length of the hyperparameter variables.*
   - *Generate a Latin hypercube sample with the specified number of iterations and the number of hyperparameter variables.*
3. *Start a loop for each LHS iteration.*
   - *Begin the loop to iterate over each iteration of the Latin hypercube sampling.*
4. *Inside the loop, generate hyperparameters for the current iteration by iterating over each hyperparameter variable.*
   - *Iterate over each hyperparameter variable to generate specific hyperparameter values for the current iteration.*
5. *Map the LHS sample from the unit interval to the variable range, taking into account whether the variable is of integer or continuous type.*
   - *Scale the LHS sample values from the unit interval to their corresponding ranges, accounting for variable types (integer or continuous).*
6. *Update the network's hyperparameters with the generated hyperparameters.*
   - *Update the hyperparameters of the network (e.g., learning rate, epochs) with the generated hyperparameters for the current iteration.*
7. *Train the network using the updated hyperparameters and the training data and obtain the training record.*
   - *Train the neural network using the updated hyperparameters and the training data.*
   - *Obtain the training record, which contains information about the training process (e.g., performance, validation errors).*
8. *Evaluate the network's performance by extracting the best validation performance from the training record.*
   - *Extract the best validation performance achieved during training from the training record.*
9. *Check if the current performance is better than the previous best performance. If it is, update the best hyperparameters with the current hyperparameters and update the best performance.*
   - *Compare the current performance with the previously recorded best performance.*
   - *If the current performance is better, update the best hyperparameters with the current hyperparameters and update the best performance value.*
10. *Store the current iteration's results (iteration number, hyperparameters, and performance) in the LHS results array.*
    - *Save the results of the current LHS iteration, including the iteration number, hyperparameters used, and the corresponding performance, into the LHS results array.*
11. *End the loop.*
    - *Conclude the loop for the Latin hypercube sampling*.

By incorporating training and testing within the SMBO loop, the algorithm can iteratively optimize the hyperparameters based on the performance of unseen testing data. This enables the SMBO algorithm to fine-tune the model for better generalization and improve its performance on unseen data.

2.6.3. Sequential model-based optimization (SMBO) algorithm with training and testing:

1. *Set the initial point for Bayesian optimization as the best hyperparameters obtained from the Latin hypercube sampling.*
   - *Initialize the starting point for Bayesian optimization with the best hyperparameters found in the Latin hypercube sampling step.*
2. *Specify the number of SMBO iterations.*
   - *Determine the desired number of iterations for the sequential model-based optimization loop.*
3. *Start a loop for each SMBO iteration.*
   - *Begin the loop to iterate over each iteration of the sequential model-based optimization.*
4. *Print the current SMBO iteration number.*
   - *Display the current iteration number for monitoring the progress of the SMBO loop.*
5. *Perform Bayesian optimization using the specified objective function, hyperparameter variables, and other settings.*
   - *Apply Bayesian optimization to find the optimal hyperparameters by optimizing the specified objective function using the defined hyperparameter variables and other specified settings.*
6. *Obtain the optimal hyperparameters from the Bayesian optimization results.*
   - *Extract the hyperparameters that yield the best performance from the results of Bayesian optimization.*
7. *Train the network using the optimal hyperparameters and the training data.*
   - *Use the obtained optimal hyperparameters to train the neural network using the training data.*
8. *Evaluate the network's performance on the testing data.*
   - *Assess the performance of the trained network on the testing data to evaluate its generalization ability.*
9. *Store the performance of the trained network on the testing data as the objective value for Bayesian optimization.*
   - *Save the obtained performance of the trained network on the testing data as the objective value, which will be used by Bayesian optimization to guide the search in subsequent iterations.*
10. *Update the initial point with the optimal hyperparameters found in the current iteration.*
    - *Update the initial point for the next iteration of Bayesian optimization with the hyperparameters that yielded the best performance in the current iteration.*

11. End the loop.

- *Conclude the loop for the sequential model-based optimization.*

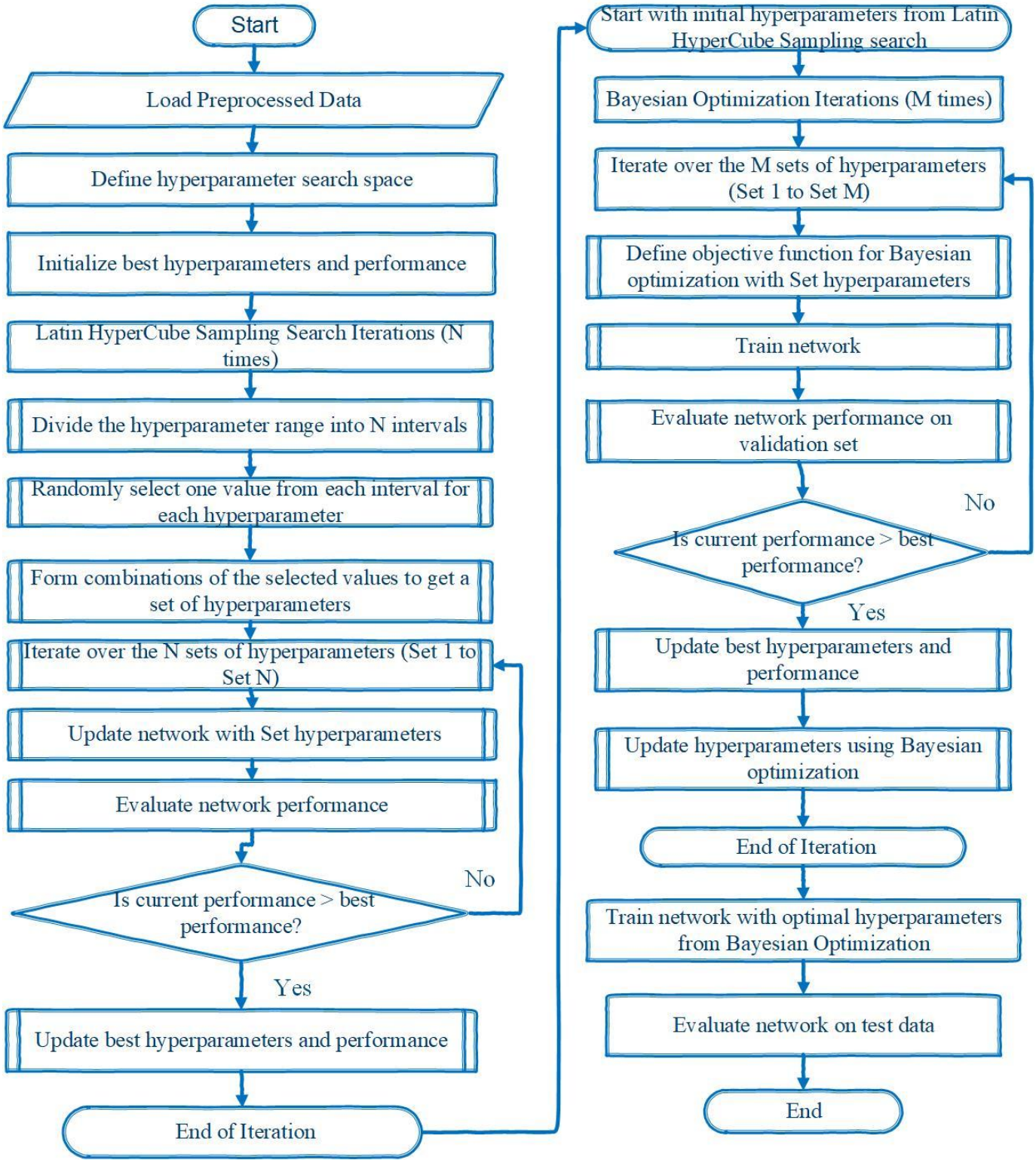

Figure 3. Flowchart of Latin hypercube sampling and Bayesian optimization algorithms.

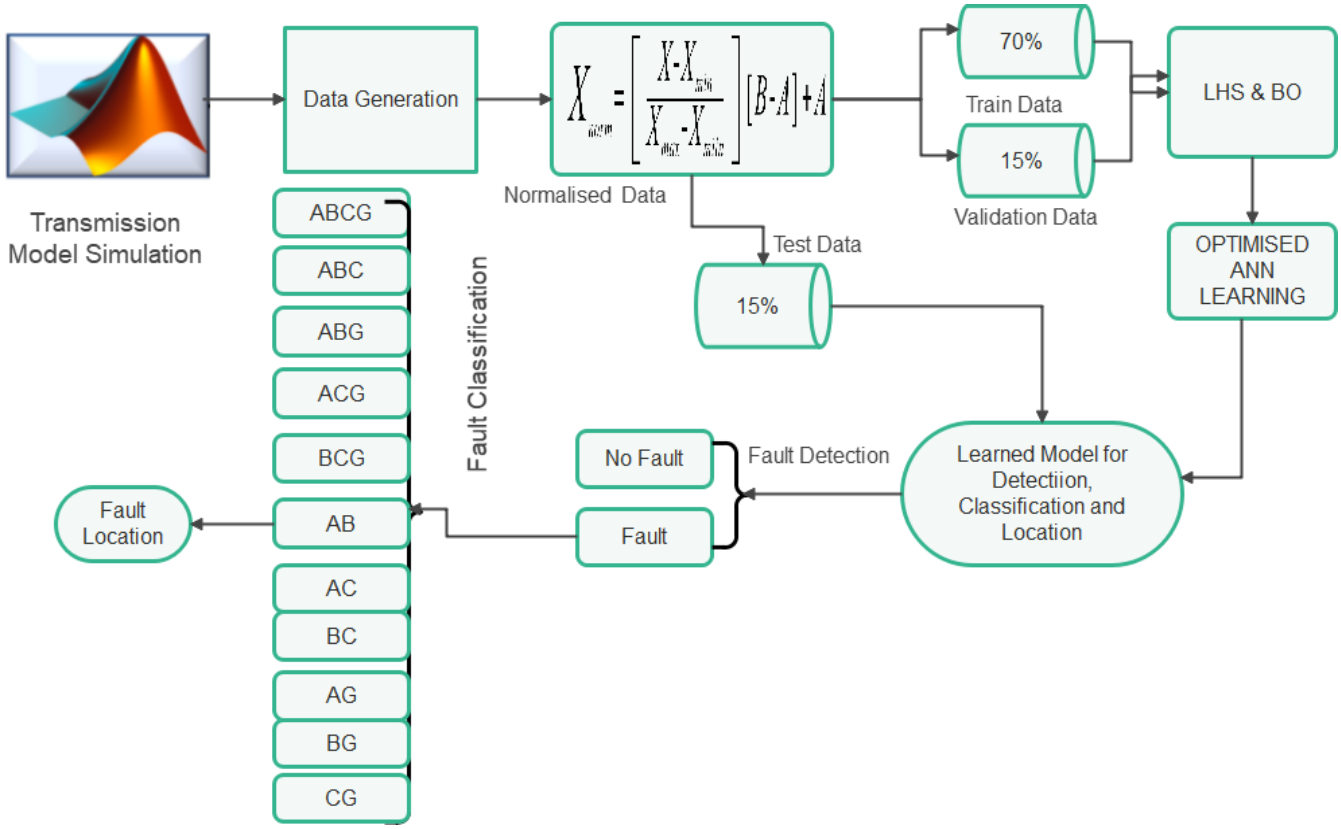

Figure 4. Proposed framework for fault detection, classification, and location.

2.7. The optimized ANN fault detector, classifier, and locator

The system comprises three neural networks (ANNs) designed to detect, classify, and locate faults in a power system, sharing common algorithms, learning functions, and performance metrics. The three neural networks used for fault detection, classification, and location were designed with varying structures to best suit the specific task. The fault detector uses a simpler network architecture (6-12-12-1), because the task of detecting whether a fault is present or not is relatively straightforward. However, the fault classifier (6-12-12-4) requires a more complex structure to accurately classify faults across multiple classes (e.g., SLG, LL, DLG, and three-phase faults). The fault locator (6-12-12-1) also uses a similar structure to the detector because it only needs to estimate a continuous value, the distance to the fault, rather than classify it. These architecture choices were made to optimize performance while keeping computational complexity manageable.

1. **Fault detector**: This 6-12-12-1 feedforward neural network identifies 11 fault conditions with a binary output (1 for fault, 0 for normal). It uses the Levenberg–Marquardt (LM) algorithm for backpropagation learning and the LEARNGDM function for efficient processing. Trained on 2646 datasets of voltage and current signals, it is evaluated using mean squared error (MSE). Data is split 70% for training, 15% for validation, and 15% for testing.
2. **Fault classifier**: This 6-12-12-4 feedforward neural network, as shown in Figure 5, determines which phase (A, B, C, or ground) has a fault. It employs the same LM algorithm and LEARNGDM function for training, evaluated with MSE. Trained on 4950 data points, it uses the same 70/15/15 data split.

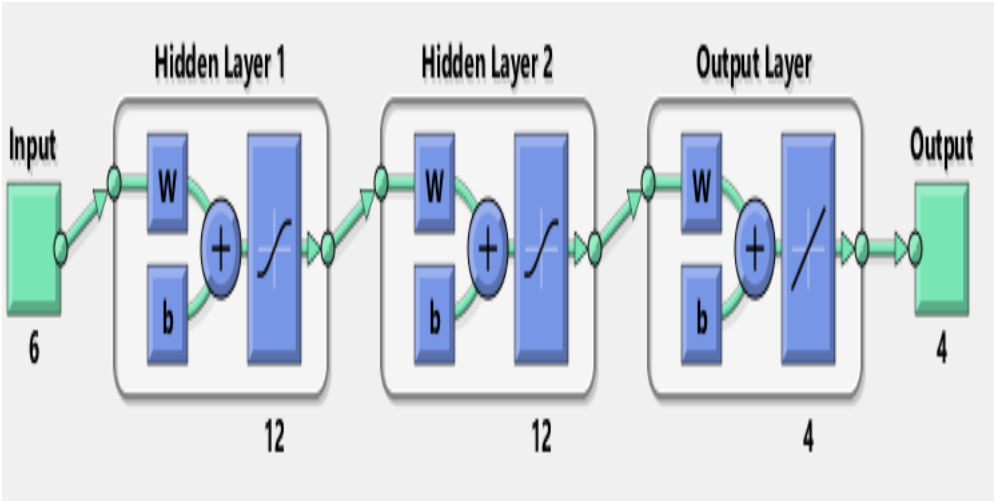

Figure 5. FNN structure of the fault classifier.

3. **Fault locator**: Similar to the detector as seen in Figure 6, this 6-12-12-1 feedforward neural network estimates the distance to a fault from substations. Using the same LM algorithm, LEARNGDM function, and TANSIG activation functions, it assesses performance with MSE. It is trained on 3465 data points, with the same data split.

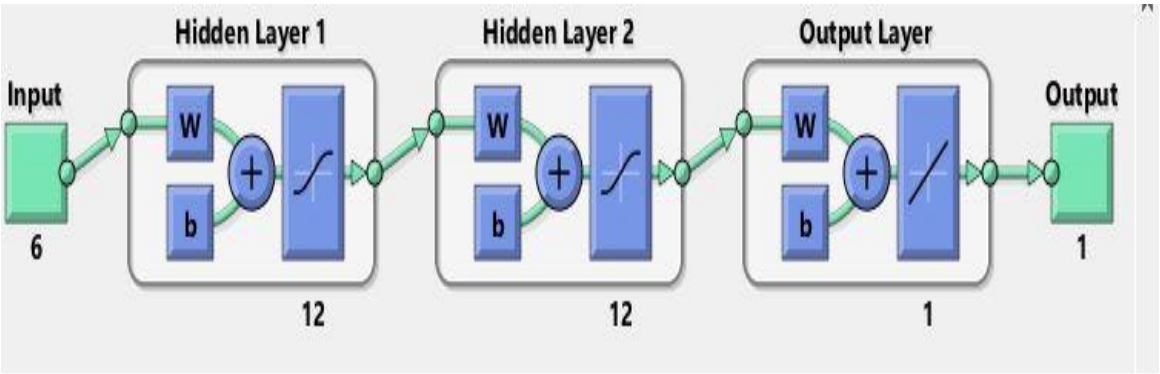

Figure 6. FNN structure of the fault detector and locator.

The consistent use of the LM algorithm for backpropagation learning and LEARNGDM for efficient processing, with performance measured by MSE, ensures a unified approach to fault detection, classification, and location in power systems.

3. Results

3.1. Preprocessing of input data

The modified min-max normalization technique is essential for preprocessing voltage and current data from power transmission lines for machine learning applications like fault location. By scaling the data, the model can effectively learn from patterns in the voltages and currents, which remain consistent regardless of fault distance. Figure 7 illustrates the transformation of raw data using the modified min-max normalization technique. This method enhances the algorithm's ability to detect

subtle patterns in voltage and current data, improving fault location accuracy across varying operational conditions.

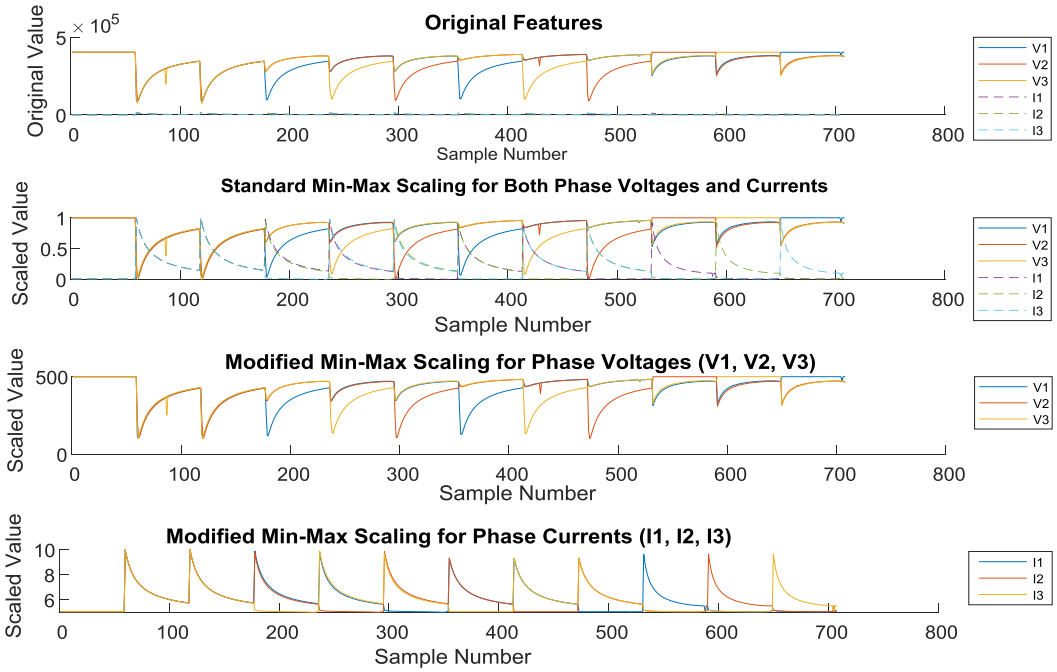

Figure 7. Waveforms of the modified min-max normalization technique.

3.2. Evaluation of the network with the best hyperparameters

After determining the best hyperparameters, the network is trained using these optimized settings to ensure the best possible performance. Once trained, the network's performance is evaluated across three subsets of the data: training, validation, and testing.

Figure 8 shows optimization results using LHS. LHS uniformly samples the parameter space. The learning rate vs. performance plot indicates most combinations achieve near-zero performance, the optimization goal, suggesting effective learning rate distribution by LHS. Similarly, for epochs vs. performance, uniform performance distribution across epochs with near-zero values implies successful optimization. The spread of data points indicates effective exploration of the parameter space by LHS, with most samples nearing the ideal performance.

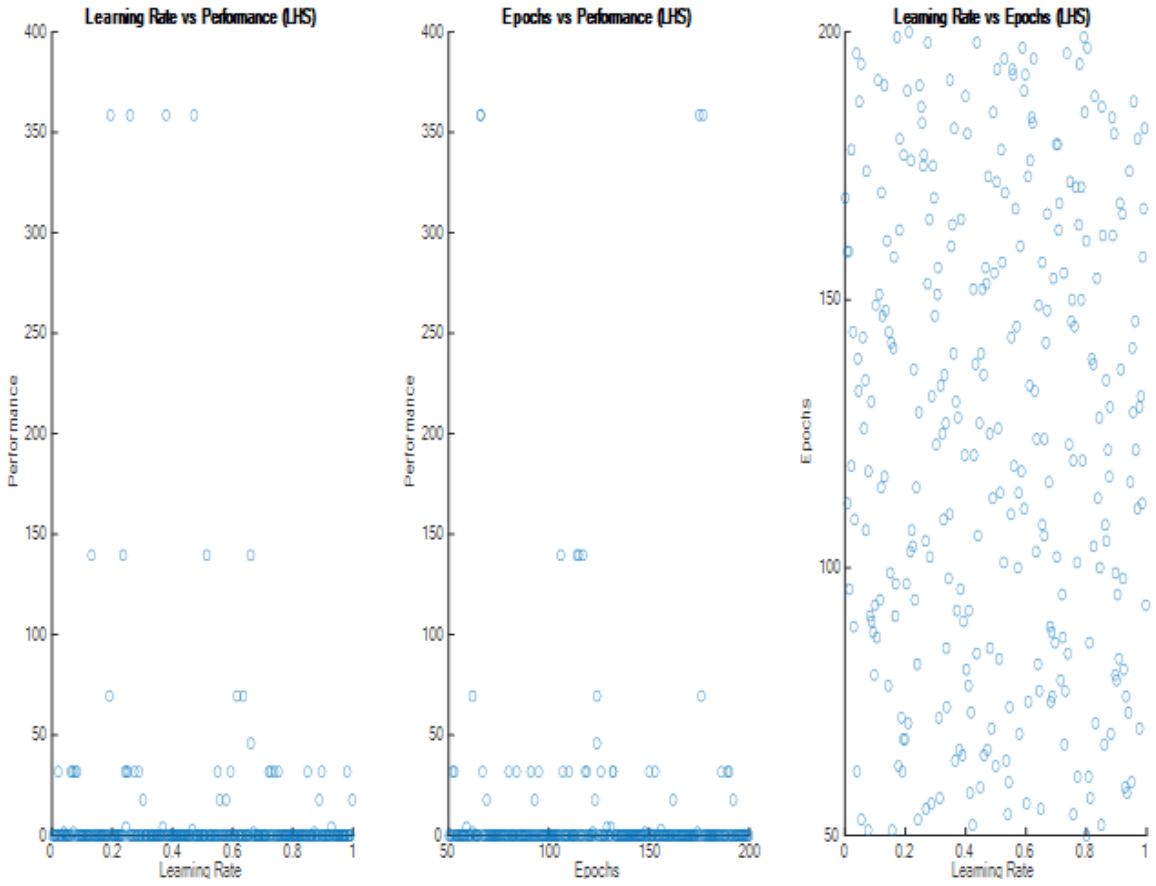

Figure 8. LHS performance metrics.

Figure 9 depicts results from Bayesian optimization over two hyperparameters: epochs and learning rate (lr), as in Figure 11. Each blue dot represents a tested hyperparameter combination, plotted against the objective function's estimated value, typically a loss function. The red line indicates the model mean, the average predicted performance. The red star denotes the "model minimum feasible", the most promising hyperparameter set. The black dot is the "next point", where the algorithm will probe next. Blue points cluster at lower learning rates and moderate-to-high epochs, indicating promising optimization regions. This visualization aids in understanding the optimization landscape and convergence behavior.

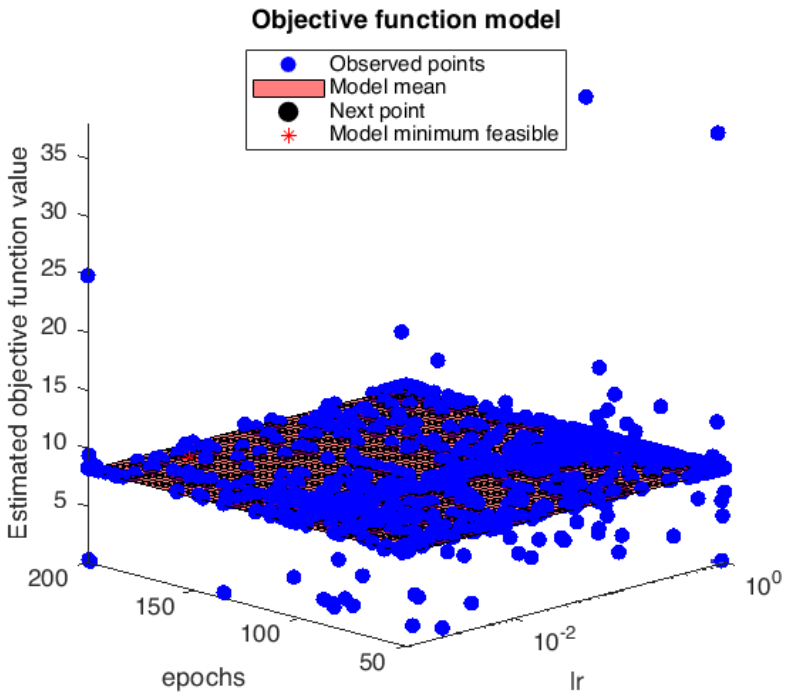

Figure 9. Objective function model performance.

The box plots in Figure 10 help to visualize the spread and central tendency of the parameters tested during optimization, providing insights into the stability of the model against these parameters. The optimization has narrowed down to a specific range for learning rates while allowing more variability in the number of epochs.

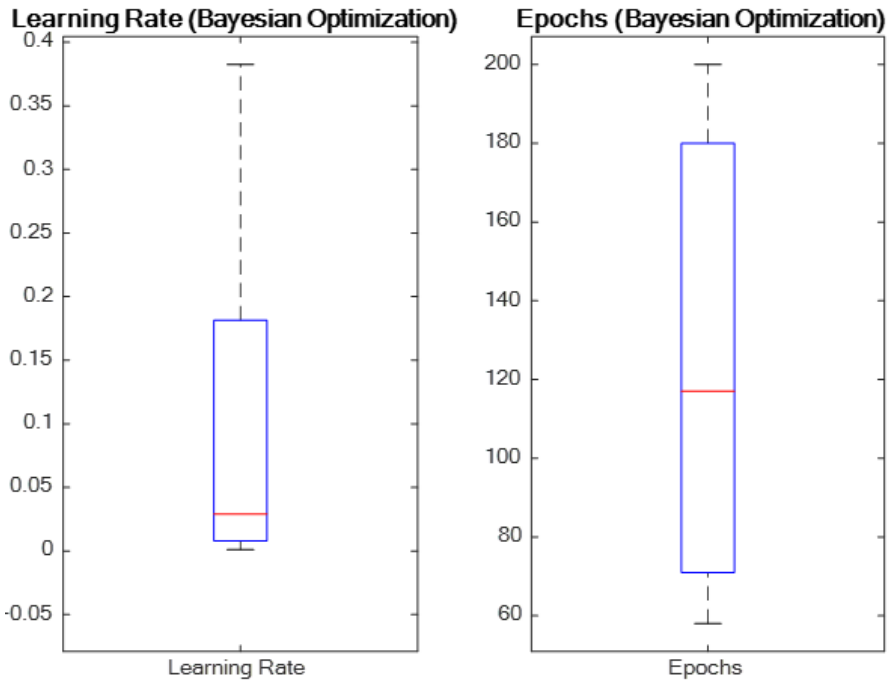

Figure 10. Boxplots of Bayesian optimization performance.

3.2.1. Fault detection

Impressively, by epoch 2, the optimized ANN demonstrated an MSE of 9.5079×10^{-11} and a regression R of 1, as shown in Figures 11 and 12. Figure 12 shows the mean squared error (MSE) across three epochs of neural network training. The MSE for training decreases significantly, while the validation MSE remains low and stable, suggesting the model is generalizing well. The test MSE is initially higher but converges toward the validation MSE, indicating consistent performance on unseen data. The "Best" mark on the graph indicates the epoch where the validation MSE is at its lowest, which is the ideal point for the model's performance. There is no sign of overfitting or underfitting, as the validation and test errors do not increase over time, and the training error converges with the validation error without becoming too low.

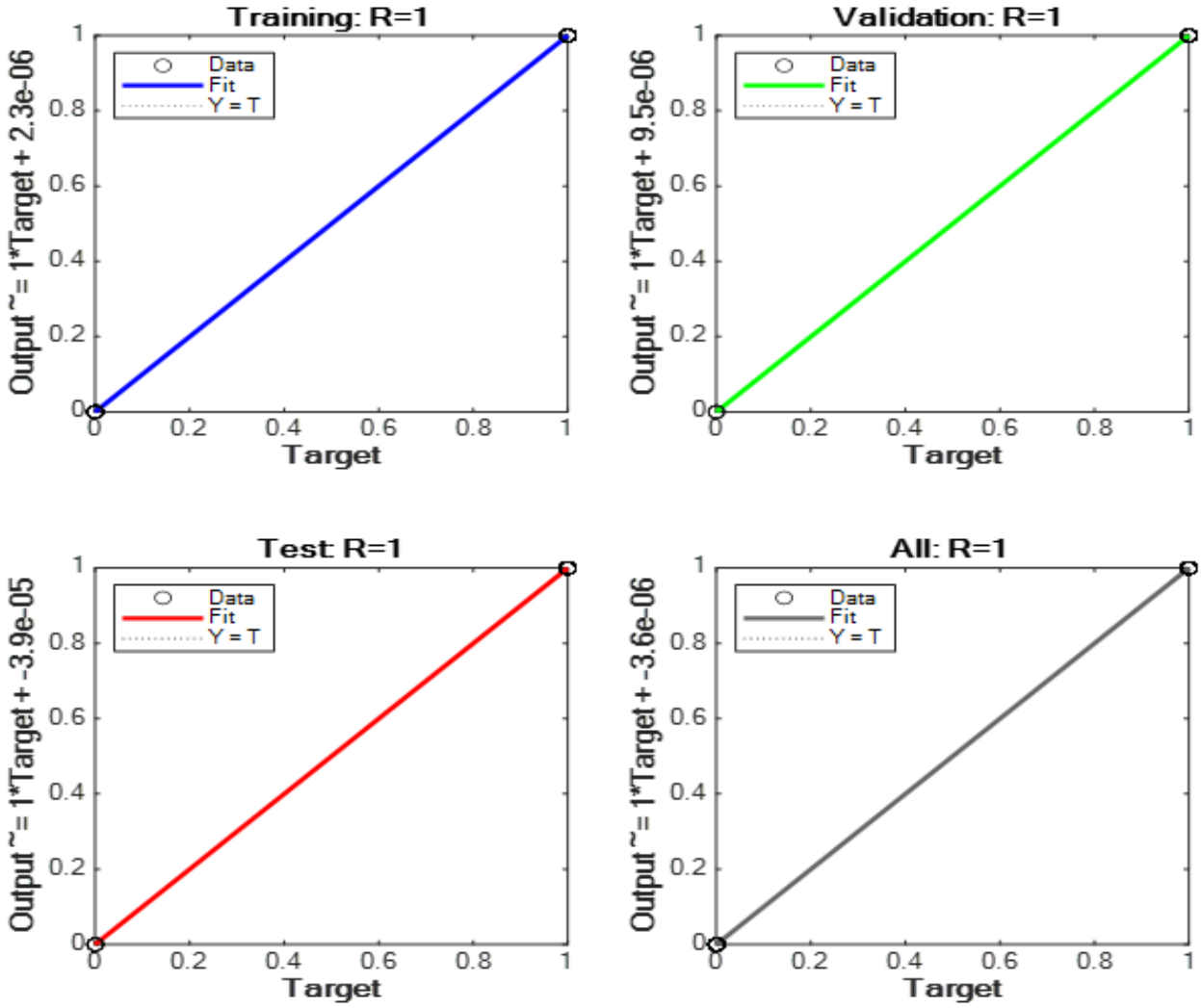

Figure 11. Training, testing, and validation regression of FNN fault detector.

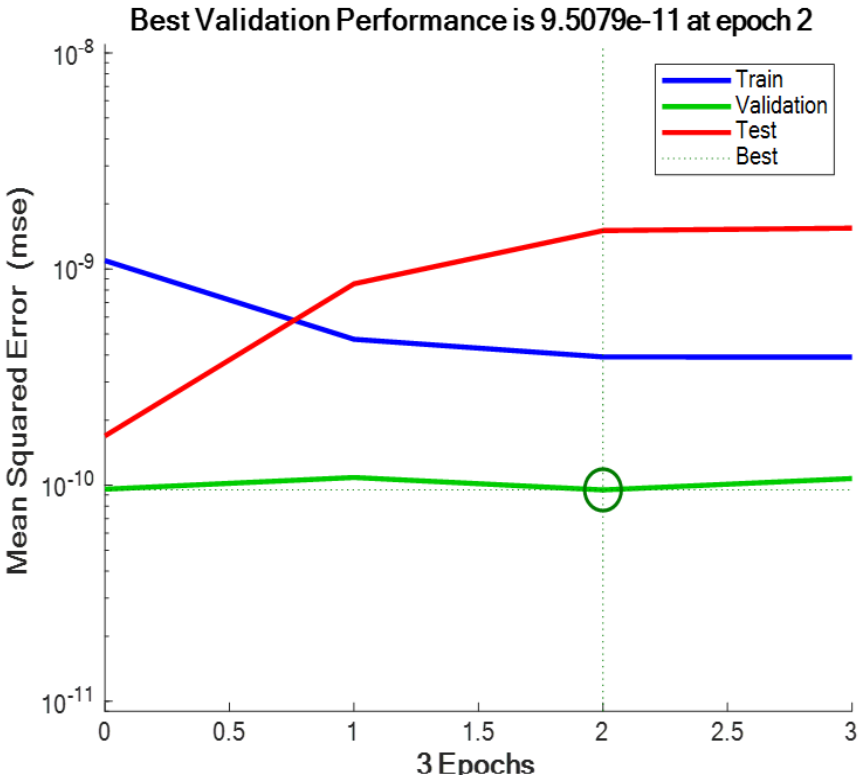

Figure 12. Best validation performance of FNN fault detector.

3.2.2. Fault classification

Figure 13 elegantly showcases the regression plots for the fault classification model in power transmission lines, each plot vividly capturing the model's stellar performance across different datasets: training, validation, testing, and all data combined. With regression values (R) for training, validation, testing, and all data hitting the perfect score of 1, the model demonstrates an impeccable correlation between its predictions and the actual target values, underlining its superior predictive accuracy.

All three errors start low and remain relatively flat throughout the epochs, as shown in Figure 14. The best performance according to the validation set is marked early, around the first epoch, as indicated by the dotted line with a circle. This flat trend suggests the model was very quickly optimized and did not significantly improve or worsen over additional training epochs, indicating stable performance across the board. The consistency between the training, validation, and test errors suggests that the network generalizes well without overfitting or underfitting.

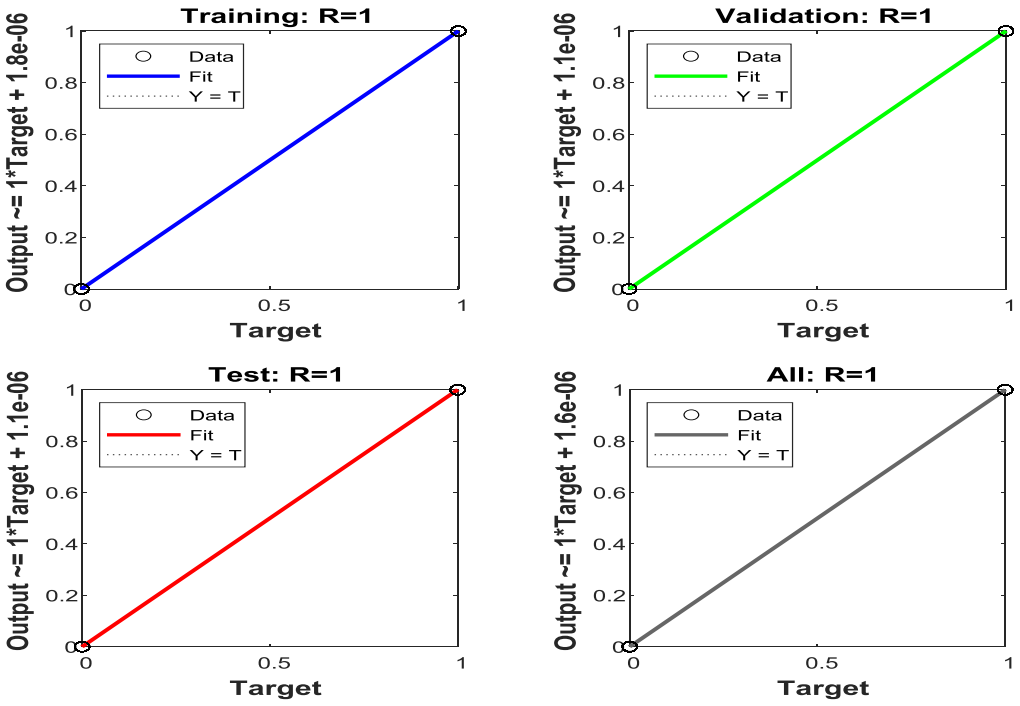

Figure 13. Training, testing, and validation regression of FNN fault classifier.

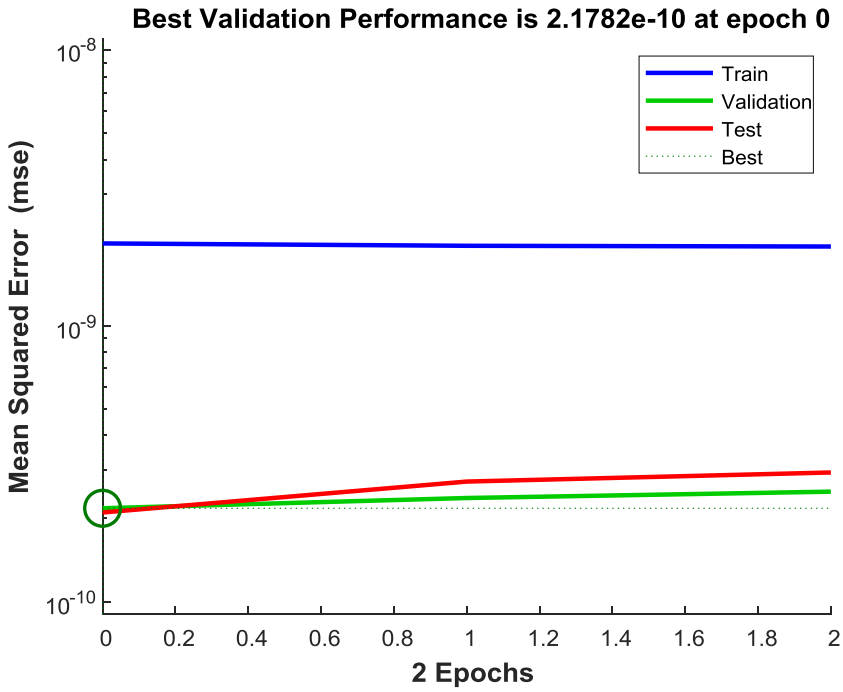

Figure 14. Best validation performance of FNN fault classifier.

3.2.3. Fault location

A strong positive correlation is evident from the training, testing, and validation regression curves of the trained model. Specifically, the model achieved a regression coefficient (R) of 0.99999

for both training and validation, while the testing data achieved a perfect R of 1.00000, and the best validation performance was 0.060262, as seen in Figures 15 and 16. The consistent performance of the model on both validation and test sets after the first epoch, without further significant drops in error, indicates that the model reached its learning capacity given the current complexity and data. There are no signs of overfitting or underfitting after the initial learning phase; overfitting would be indicated by an increasing validation error as the training error continued to decrease, and underfitting would be indicated by a high error that did not decrease.

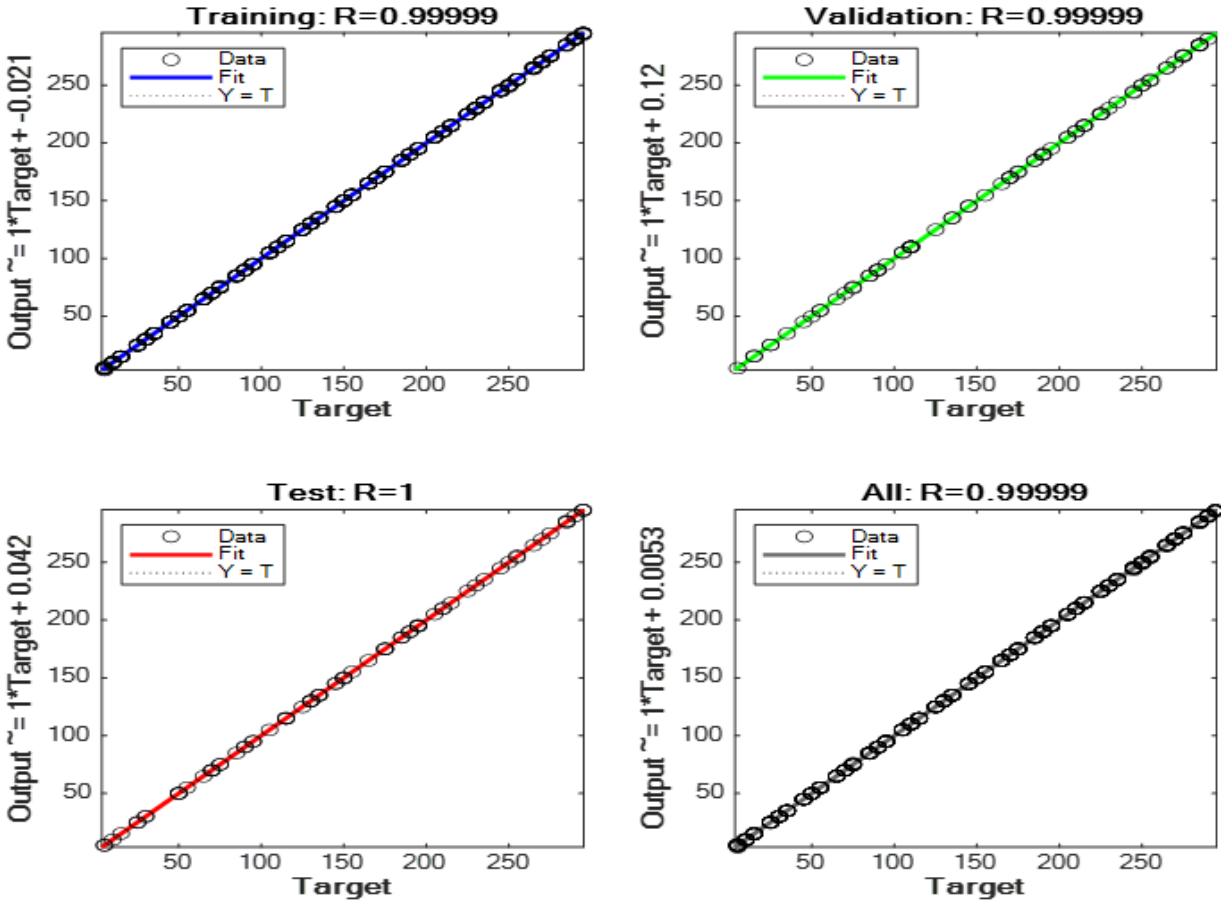

Figure 15. Training, testing, and validation regression of FNN fault locator.

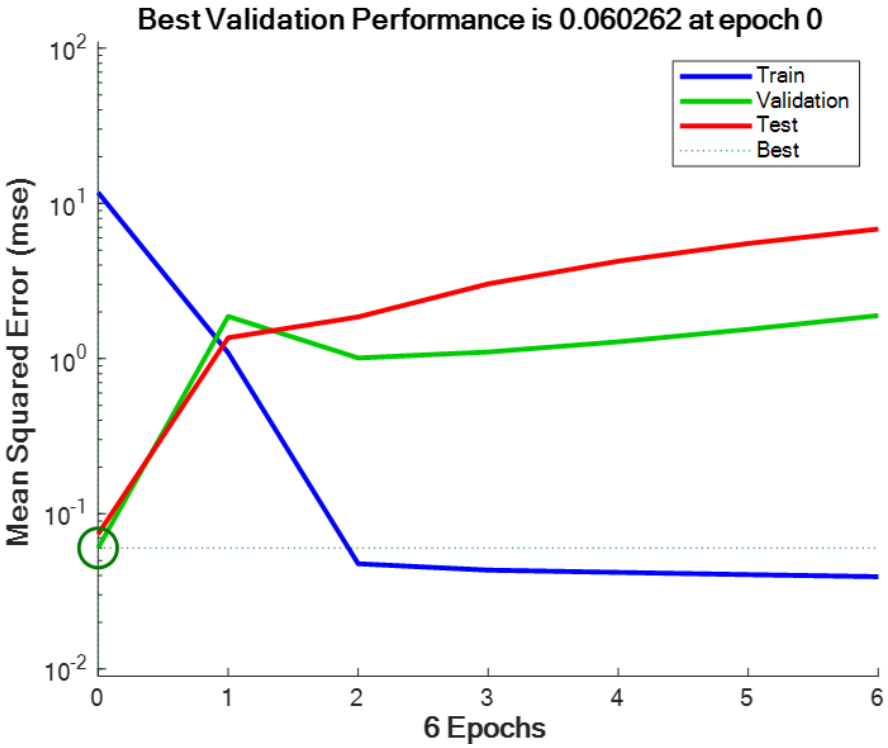

Figure 16. Best validation performance of FNN fault locator.

The training, validation, and test bars reaching close to the "Zero Error" line as seen in Figure 17 indicate that the model accurately located faults in all three datasets. Such high performance across the board is an indication that the model is highly effective at generalizing from its training data to unseen data, which is crucial for reliable real-world application in fault location on transmission lines.

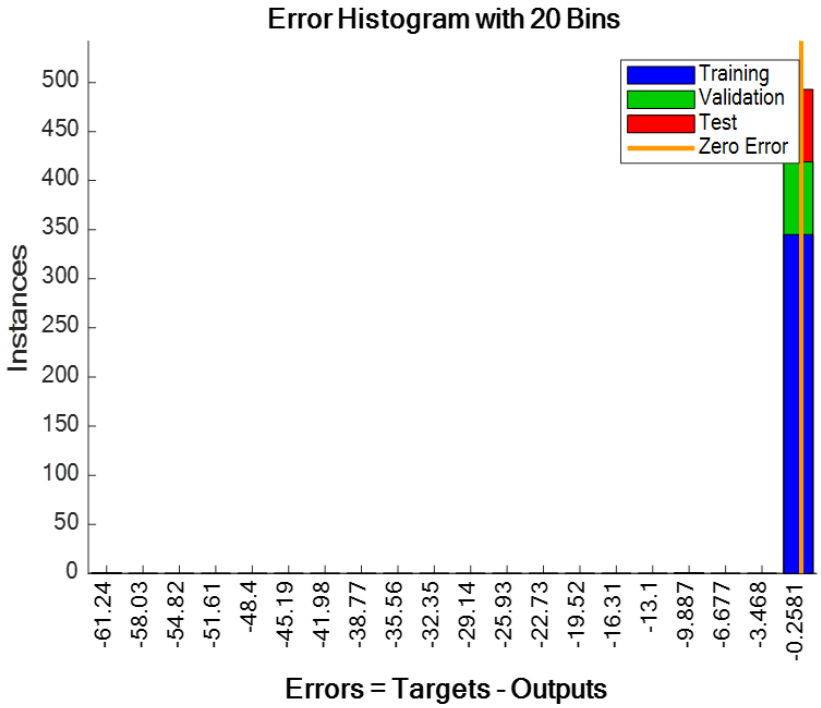

Figure 17. Histogram showing the error performance of FNN fault locator.

4. Discussion

4.1. Deployment on unseen data

In the realm of machine learning, the deployment of a trained ANN model on unseen data for evaluation stands as a cornerstone in determining the model's real-world efficacy. This critical step involves rigorously testing the model against new, previously unexposed datasets to evaluate its ability to generalize beyond the conditions of its training environment.

4.1.1. Fault detection

When tested on an evaluation dataset of 168 cases, encompassing 11 different short-circuit fault types and normal scenarios, the ANN model's precision was exemplary. It accurately identified every case, 14 normal and 154 faulty, as detailed in Figure 18. The performance metrics underscore its effectiveness, achieving consistent 100% accuracy, precision, recall, and F1 score. This exemplary performance highlights the effectiveness of the optimization techniques used in refining the ANN for fault detection, demonstrating its potential for robust applications in real-world power systems.

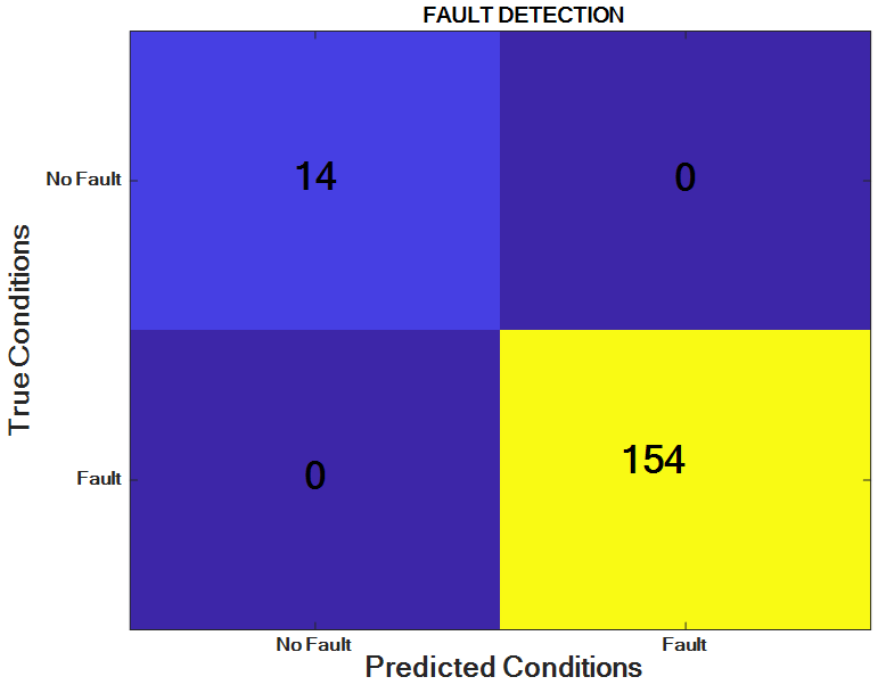

Figure 18. Confusion matrix of deployment result of FNN fault detector

4.1.2. Fault classification

The confusion matrix displayed for fault classification in power transmission systems exemplifies perfect model accuracy, with a 100% success rate in identifying various fault types, as shown in Figure 19. A confusion matrix is an essential tool in these scenarios, providing clear and organized insight into the model's classification capabilities. In this instance, the matrix indicates that out of 154 fault cases, the model correctly classified all 154, demonstrating impeccable precision.

Furthermore, the model achieved 100% in all key performance metrics: precision, recall, F1 score, and accuracy, as seen in Figure 20.

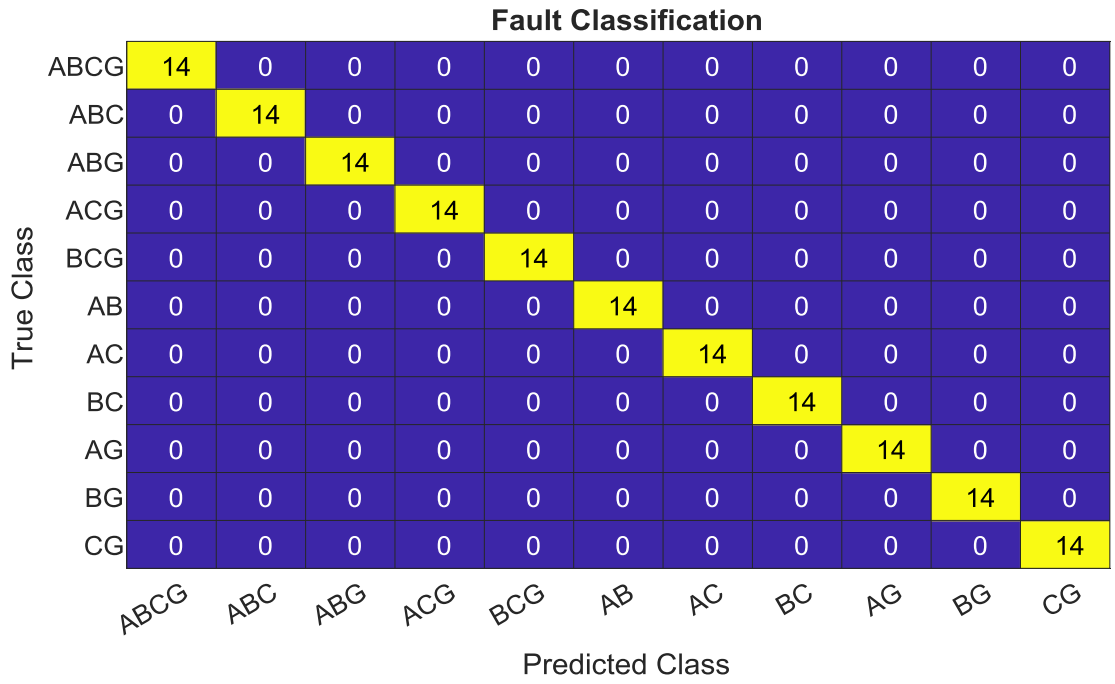

Figure 19. Confusion matrix of deployment result of FNN fault classifier.

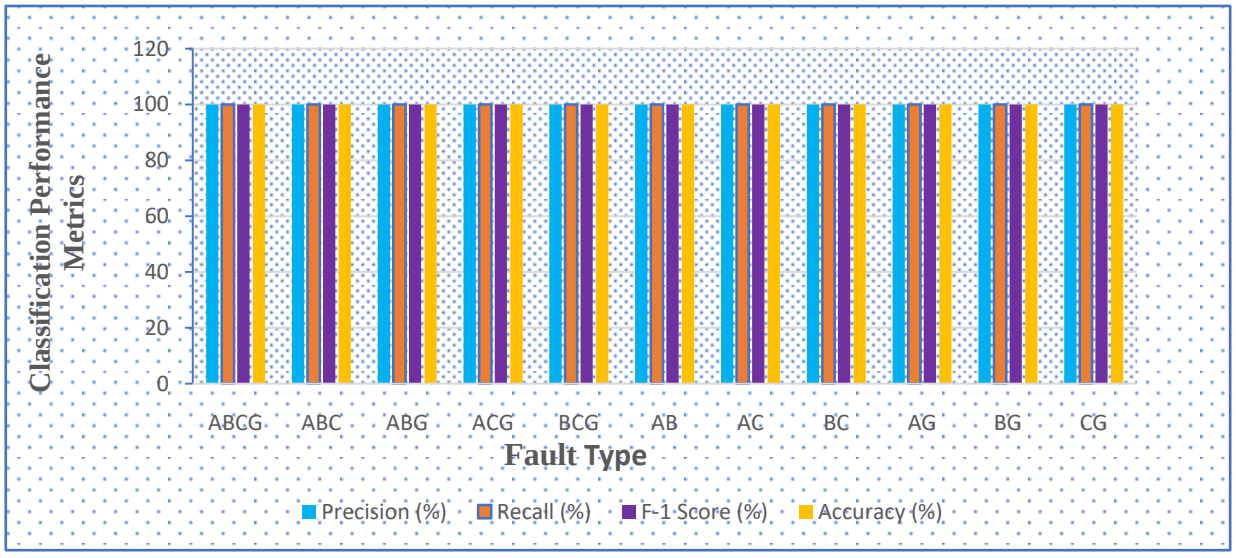

Figure 20. Performance metrics of FNN fault classifier.

4.1.3. Fault location

Model deployment on evaluation data:

When the trained model was deployed on unseen evaluation data, comprising 154 cases across 11 different short-circuit fault types, impressive results were observed. The overall minimum and maximum errors were 0.001% and 0.174%, respectively, while the overall accuracy ranged from

99.826% to 99.999%. The mean absolute percentage error (MAPE) for this dataset stood at 0.053%, meaning that the accuracy of the model was 99.947%, the MSE was found to be 0.0083, and the MAE was also found to be 0.0717. As we endeavored to unpack the intricacies of our model's performance, as shown in Figure 21, we meticulously analyzed the results in tandem with the diverse fault types present. This analytical approach was pivotal in discerning the model's nuanced competencies and potential areas for enhancement.

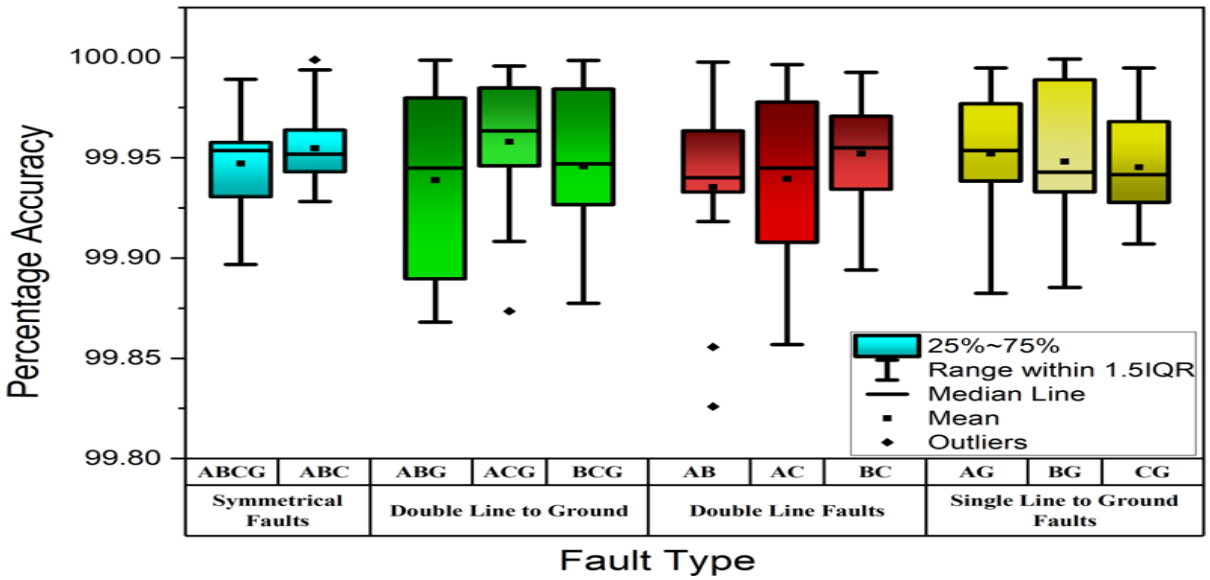

Figure 21. Boxplot of percentage accuracy of various short-circuit fault types.

Symmetrical faults analysis:

ABCG faults:

- Error range: 0.011% to 0.103%
- Accuracy: 99.897% to 99.947%
- MAPE: 0.053%

ABC faults:

- Error range: 0.001% to 0.072%
- Accuracy: 99.928% to 99.999%
- MAPE: 0.045%

Double-line-to-ground faults analysis:

ABG faults:

- Error range: 0.001% to 0.132%
- Accuracy: 99.868% to 99.999%
- MAPE: 0.061%

ACG faults:

- Error range: 0.004% to 0.127%
- Accuracy: 99.873% to 99.996%
- MAPE: 0.042%

BCG faults:

- Error range: 0.001% to 0.122%
- Accuracy: 99.878% to 99.999%
- MAPE: 0.054%

Line-to-line faults analysis

AB faults:

- Error range: 0.002% to 0.174%
- Accuracy: 99.826% to 99.998%
- MAPE: 0.065%

AC faults:

- Error range: 0.004% to 0.143%
- Accuracy: 99.857% to 99.996%
- MAPE: 0.060%

BC faults:

- Error range: 0.007% to 0.106%
- Accuracy: 99.894% to 99.993%
- MAPE: 0.048%

Line-to-ground faults analysis

AG faults:

- Error range: 0.005% to 0.118%
- Accuracy: 99.882% to 99.995%
- MAPE: 0.048%

BG faults:

- Error range: 0.001% to 0.115%
- Accuracy: 99.885% to 99.999%
- MAPE: 0.052%

CG faults:

- Error range: 0.005% to 0.093%
- Accuracy: 99.907% to 99.995%
- MAPE: 0.055%

The deployment of the model on unseen data yielded impressive results, as shown in Table 1 for line-to-line faults and Table 2 for line-to-ground faults. This granular analysis provides a comprehensive perspective on the model's performance, illustrating its adaptability and precision across a diverse array of fault scenarios in the power transmission network.

Table 1. Predicted results of line-to-line faults.

Fault type	AB		AC		BC	
Actual fault distance	Predicted fault distance	Absolute percentage accuracy	Predicted fault distance	Absolute percentage accuracy	Predicted fault distance	Absolute percentage accuracy
20	20.000	**99.998**	20.018	**99.908**	19.996	**99.978**
40	40.070	**99.826**	39.972	**99.931**	39.982	**99.955**
60	59.978	**99.964**	60.003	**99.995**	60.038	**99.936**
80	80.054	**99.933**	79.982	**99.978**	80.026	**99.968**
100	100.038	**99.962**	99.899	**99.899**	99.971	**99.971**
120	120.098	**99.918**	119.913	**99.928**	119.896	**99.913**
140	139.958	**99.970**	140.200	**99.857**	139.930	**99.950**
160	159.893	**99.933**	160.006	**99.996**	160.062	**99.961**
180	179.894	**99.941**	180.248	**99.862**	180.191	**99.894**
200	199.868	**99.934**	200.044	**99.978**	200.137	**99.931**
220	220.318	**99.856**	219.938	**99.972**	219.902	**99.955**
240	239.854	**99.939**	239.905	**99.961**	239.982	**99.993**
260	259.891	**99.958**	259.836	**99.937**	260.171	**99.934**
280	279.898	**99.963**	280.132	**99.953**	279.975	**99.991**

Table 2. Predicted results of single line-to-ground faults.

Fault type	AG		BG		CG	
Actual fault distance	Predicted fault distance	Absolute percentage accuracy	Predicted fault distance	Absolute percentage accuracy	Predicted fault distance	Absolute percentage accuracy
20	19.996	**99.978**	20.013	**99.933**	19.988	**99.938**
40	39.991	**99.977**	39.957	**99.894**	40.036	**99.909**
60	60.071	**99.882**	60.038	**99.936**	60.052	**99.913**
80	80.032	**99.960**	80.092	**99.885**	79.951	**99.939**
100	100.053	**99.947**	100.061	**99.939**	99.980	**99.980**
120	120.074	**99.938**	120.121	**99.899**	120.111	**99.907**
140	139.898	**99.927**	139.991	**99.994**	140.045	**99.968**
160	159.911	**99.944**	159.945	**99.966**	159.959	**99.975**
180	180.102	**99.944**	179.984	**99.991**	179.875	**99.930**
200	199.974	**99.987**	200.106	**99.947**	200.112	**99.944**
220	219.833	**99.924**	220.147	**99.933**	220.159	**99.928**
240	239.905	**99.960**	239.927	**99.970**	239.988	**99.995**
260	260.093	**99.964**	259.971	**99.989**	260.145	**99.944**
280	280.014	**99.995**	279.998	**99.999**	279.900	**99.964**

4.2. Evaluation of the trained ANN models on three different transmission lines

The deployment of the trained ANN models on power transmission lines of varying lengths provided a robust test of fault detection, classification, and location of the trained models' capabilities.

4.2.1. Deployment of the trained ANN model across transmission lines for fault detection

The trained ANN model was successfully deployed on three different transmission lines to evaluate its ability to detect faults in power systems. Throughout the tests, the model demonstrated flawless performance, achieving a 100% accuracy rate across all scenarios, which included a variety of short-circuit fault types and normal operating conditions. Additionally, the model not only reached perfect accuracy but also maintained 100% precision, recall, and F1 score, showcasing its exceptional capability to consistently identify and classify each fault type accurately without any errors. This comprehensive performance solidifies the ANN model as a highly effective and reliable tool for fault detection in power system applications.

4.2.2. Deployment of the trained ANN model across transmission lines for fault classification

The deployment of the trained ANN model across power transmission lines of varying lengths demonstrated outstanding fault classification capabilities, with 100% precision, recall, F1 score, and accuracy in each scenario. Initially tested on a 100 km line with 209 cases, the model accurately identified every case, excelling across multiple fault types. In subsequent tests on 200 and 300 km lines, involving 429 and 649 unseen cases, respectively, the model consistently maintained perfect

classification performance despite a higher scale and complexity, as seen in Figures 22 and 23, respectively. These deployments collectively highlight the model's robustness and adaptability in accurately detecting faults, confirming its effectiveness and reliability in practical applications.

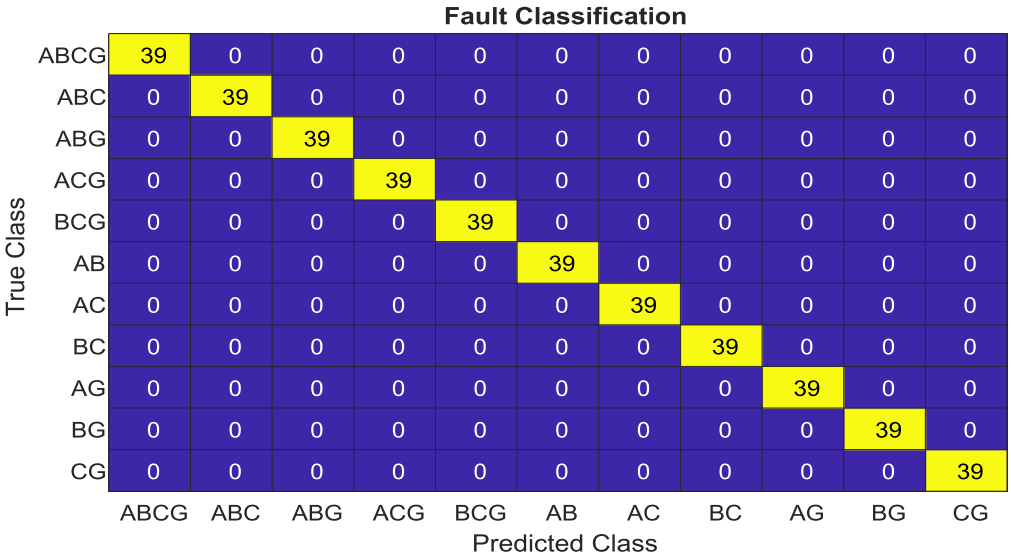

Figure 22. Confusion matrix of deployment result of FNN fault classifier on 500 kV, 200 km.

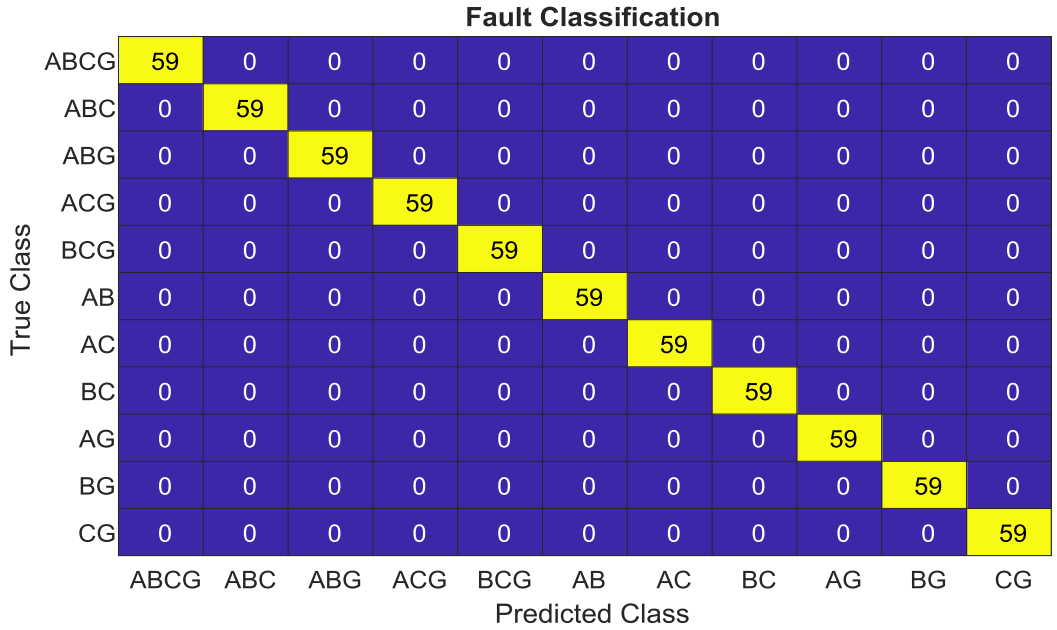

Figure 23. Confusion matrix of deployment result of FNN fault classifier on 400 kV, 300 km.

4.2.3. Deployment of the trained ANN model across transmission lines for fault location

The trained and optimized ANN model was again deployed on the same three different transmission lines after being tested on unseen data for fault location, assessing its performance against underfitting, overfitting, and accuracy. In the first deployment case, the model demonstrated remarkable accuracy and precision across 209 cases, with an impressive minimum error of 0.0003% and maximum accuracy of 99.9997%. In the second scenario, involving 429 cases, the model

exhibited noteworthy performance with a minimum error of 0.0002% and a maximum accuracy of 99.9998%. Finally, in the third deployment, with 649 cases, the model maintained commendable results, showcasing a minimum error of 0.0001% and a maximum accuracy of 99.9999%. The average percentage accuracies for the performance evaluation of the trained ANN model on the three test systems are shown in Figure 24. These findings highlight the model's reliability, versatility, and scalability in identifying short-circuit fault types within power transmission networks, with consistently high accuracy and low error rates across different deployment scenarios.

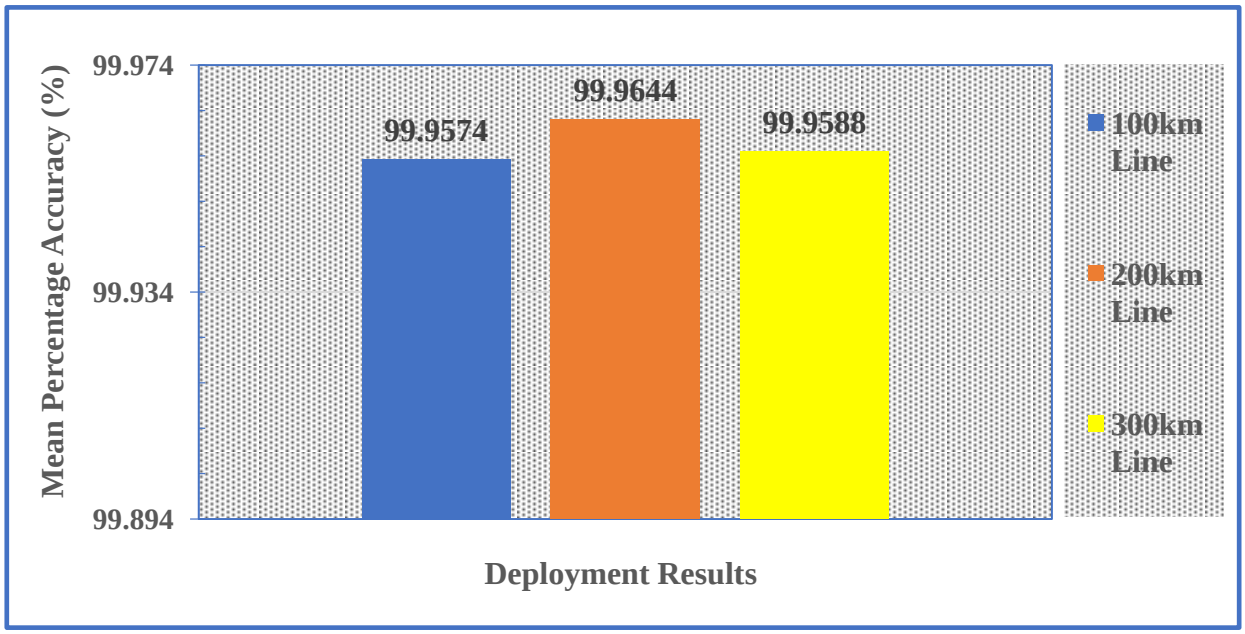

Figure 24. Performance of FNN fault locator on test systems.

4.3. Comparison of the proposed optimized ANN fault locator with related techniques in the literature

The proposed optimized ANN model for fault location in power transmission systems demonstrated a remarkably high average percentage accuracy of 99.947%. This performance notably surpasses several previous methodologies in the field, as illustrated in Table 3.

Table 3. Comparison of the proposed LHS–BO ANN fault locator with related techniques in the literature.

Technique	Percentage accuracy (%)
ANN [1]	97.56
Stockwell Transform ANN [42]	96.16
CNN-LSTM [43]	99.11
Transfer learning [44]	97.85
ANN-WT [18]	98.64
ANN [45]	99.27
SVM parameter [46]	98.96
Proposed LHS–BO ANN	99.947

5. Conclusions

The hybrid optimized ANN model, specifically engineered to meet the unique demands of power transmission systems, has demonstrated exceptional proficiency in fault detection, classification, and location. Its advanced capabilities solidify its role as a pivotal tool for the next generation of fault analysis solutions in power systems.

In the area of **fault detection**, the model exhibited flawless performance. Evaluated against a dataset of 168 cases, it achieved a perfect accuracy rate of 100%, distinguishing between normal and faulty scenarios with remarkable precision. This result highlights the extraordinary potential unlocked through the synergy of LHS and BO combined with ANN technology.

For **fault classification**, the model displayed both robustness and precision, successfully classifying 154 out of 154 cases with 100% accuracy, precision, recall, and F1 score, even on unseen data. This showcases the model's adaptability and reliability in handling various fault types.

In terms of **fault location**, the model's performance was equally remarkable, achieving accuracy rates between 99.826% and 99.999% across 154 cases. With a negligible MAPE of 0.053%, low MSE of 0.0083, and MAE of 0.0717, the model accurately located faults, whether symmetrical or line-to-ground, confirming its reliability and precision.

In conclusion, the hybrid optimized ANN model offers a powerful and versatile solution for fault detection, classification, and location in transmission systems. Its high accuracy and robust performance across varied fault scenarios make it an invaluable tool for modern power systems. By addressing the current limitations and exploring future directions, this model has the potential to significantly improve fault management and enhance the reliability of power transmission networks, heralding a new era in the field of power system fault analysis.

Recommendations and future work

While the hybrid optimized ANN model has demonstrated exceptional performance, several areas for improvement and future research are worth exploring. First, further testing with real-world transmission line data is recommended to confirm the model's robustness and generalizability outside of the simulated environment. Additionally, the model could benefit from integration with distributed power systems and microgrids, where the complexity of fault detection increases and additional data sources, such as weather conditions, may play a significant role. Although the hybrid LHS–BO approach enhances performance, optimizing computational efficiency is critical when scaling the model for real-time systems, especially those with large datasets. Future work could also focus on how noise and uncertainty in sensor measurements affect model accuracy, further strengthening its reliability in practical, less controlled environments. Addressing these aspects will enhance the model's applicability and effectiveness in real-world scenarios.

Limitations

The primary limitation of the current study is that it has been tested exclusively in simulated conditions. The model has yet to be validated with real-time data, which may include unmodeled complexities such as communication delays, non-ideal sensor data, and environmental factors that could influence its performance. Additionally, the method's computational demands, particularly in real-time fault diagnosis, need to be assessed and optimized for large-scale power systems.

Author contributions

Abdul Yussif Seidu: Conceptualization, Data curation, Investigation, Methodology, Software, Visualization, writing of original draft. Elvis Twumasi: Conceptualization, Investigation, Methodology, Supervision, Validation, Visualization, Review and Editing**.** Emmanuel Asuming Frimpong: Investigation, Supervision, Validation, Visualization, Review and Editing. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare that there is no conflict of interest in this paper.

References

1. Ogar, VN, Hussain S, Gamage KA (2023) The use of artificial neural network for low latency of fault detection and localisation in transmission line. *Heliyon* 9. https://doi.org/10.1016/j.heliyon.2023.e13376
2. Goni MO, Nahiduzzaman M, Anower MS, Rahman MM, Islam MR, Ahsan M, et al. (2023) Fast and Accurate Fault Detection and Classification in Transmission Lines using Extreme Learning Machine. *e-Prime - Advances in Electrical Engineering, Electronics and Energy* 3: 100107. https://doi.org/10.1016/j.prime.2023.100107
3. Naji HA, Fayadh RA, Mutlag AH (2023) ANN-based Fault Location in 11 kV Power Distribution Line using MATLAB. *2023 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, JEEIT*, 134–139. https://doi.org/10.1109/JEEIT58638.2023.10185849
4. Venkata P, Pandya V, Vala K, Sant AV (2022) Support vector machine for fast fault detection and classification in modern power systems using quarter cycle data. *Energy Reports* 8: 92–98. https://doi.org/10.1016/j.egyr.2022.10.279
5. Ahmadipour M, Othman MM, Bo R, Salam Z, Ridha HM, Hasan K (2022) A novel microgrid fault detection and classification method using maximal overlap discrete wavelet packet transform and an augmented Lagrangian particle swarm optimization-support vector machine. *Energy Reports* 8: 4854–4870. https://doi.org/10.1016/j.egyr.2022.03.174
6. Aiswarya R, Nair DS, Rajeev T, Vinod V (2023) A novel SVM based adaptive scheme for accurate fault identification in microgrid. *Electr Pow Syst Res* 221: 109439. https://doi.org/10.1016/j.epsr.2023.109439
7. De Santis E, Rizzi A, Sadeghian A (2018) A cluster-based dissimilarity learning approach for localized fault classification in Smart Grids. *Swarm Evol Comput* 39: 267–278. https://doi.org/10.1016/j.swevo.2017.10.007
8. Bhuyan A, Panigrahi BK, Pal K, Pati S (2022) Convolutional Neural Network Based Fault Detection for Transmission Line. *2022 International Conference on Intelligent Controller and Computing for Smart Power, ICICCS*, 1–4. https://doi.org/10.1109/ICICCSP53532.2022.9862446
9. Omar AMS, Osman MK, Ibrahim MN, Hussain Z, Abidin AF (2020) Fault classification on transmission line using LSTM network. *Indonesian Journal of Electrical Engineering and*

Computer Science 20: 231–238. https://doi.org/10.11591/ijeecs.v20.i1.pp231-238
10. Biswas S, Nayak PK, Panigrahi BK, Pradhan G (2023) An intelligent fault detection and classification technique based on variational mode decomposition-CNN for transmission lines installed with UPFC and wind farm. *Electr Pow Syst Res* 223: 109526. https://doi.org/10.1016/j.epsr.2023.109526
11. Mirshekali H, Keshavarz A, Dashti R, Hafezi S, Shaker HR (2023) Deep learning-based fault location framework in power distribution grids employing convolutional neural network based on capsule network. *Electr Pow Syst Res* 223: 109529. https://doi.org/10.1016/j.epsr.2023.109529
12. Tabari M, Sadeh J (2022) Fault location in series-compensated transmission lines using adaptive network-based fuzzy inference system. *Electr Pow Syst Res* 208: 107800. https://doi.org/10.1016/j.epsr.2022.107800
13. Naidu K, Ali MS, Abu Bakar AH, Tan CK, Arof H, Mokhlis H (2020) Optimized artificial neural network to improve the accuracy of estimated fault impedances and distances for underground distribution system. *PLoS One* 15: e0227494. https://doi.org/10.1371/journal.pone.0227494
14. Iliyaeifar MM, Hadaeghi A (2023) Extreme learning machine-based fault location approach for terminal-hybrid LCC-VSC-HVDC transmission lines. *Electr Pow Syst Res* 221: 109487. https://doi.org/10.1016/j.epsr.2023.109487
15. Kanwal S, Jiriwibhakorn S (2023) Artificial Intelligence Based Faults Identification, Classification, and Localization Techniques in Transmission Lines-A Review. *IEEE Latin America Transactions* 21: 1291–1305. https://doi.org/10.1109/TLA.2023.10305233
16. Yadav A, Dash Y (2014) An Overview of Transmission Line Protection by Artificial Neural Network: Fault Detection, Fault Classification, Fault Location, and Fault Direction Discrimination. *Advances in Artificial Neural Systems* 2014: 230382. https://doi.org/10.1155/2014/230382
17. Kanwal S, Jiriwibhakorn S (2024) Advanced Fault Detection, Classification, and Localization in Transmission Lines: A Comparative Study of ANFIS, Neural Networks, and Hybrid Methods. *IEEE Access* 12: 49017–49033. https://doi.org/10.1109/ACCESS.2024.3384761
18. Ravesh NR, Ramezani N, Ahmadi I, Nouri H (2022) A hybrid artificial neural network and wavelet packet transform approach for fault location in hybrid transmission lines. *Electr Pow Syst Res* 204: 107721. https://doi.org/10.1016/j.epsr.2021.107721
19. Hannan MA, How DN, Lipu MH, Ker PJ, Dong ZY, Mansur M, et al. (2020) SOC Estimation of Li-ion Batteries with Learning Rate-Optimized Deep Fully Convolutional Network. *IEEE T Power Electr* 36: 7349–7353. https://doi.org/10.1109/TPEL.2020.3041876
20. Liu S, Ghosh R, Min JT, Motani M (2022) Optimizing Learning Rate Schedules for Iterative Pruning of Deep Neural Networks. *arXiv preprint arXiv:2212.06144*.
21. Xu C, Liu S, Huang Y, Huang C, Zhang Z (2021) Over-the-air Learning Rate Optimization for Federated Learning. *2021 IEEE International Conference on Communications Workshops, ICC Workshops*, 1–7. https://doi.org/10.1109/ICCWorkshops50388.2021.9473663
22. Yasusi K (2016) Optimizing Neural-network Learning Rate by Using a Genetic Algorithm with Per-epoch Mutations. *2016 International Joint Conference on Neural Networks (IJCNN)*, 1472–1479. IEEE.
23. Hsieh HL, Shanechi MM (2018) Optimizing the learning rate for adaptive estimation of neural encoding models. *PLoS Comput Biol* 14: e1006168. https://doi.org/10.1371/journal.pcbi.1006168

24. Dai Z, Yu H, Low BK, Jaillet P (2019) Bayesian Optimization Meets Bayesian Optimal Stopping. *International conference on machine learning*, 1496–1506.
25. Serizawa T, Fujita H (2020) Optimization of Convolutional Neural Network Using the Linearly Decreasing Weight Particle Swarm Optimization. https://doi.org/10.11517/pjsai.JSAI2022.0_2S4IS2b03
26. Sheikholeslami R, Razavi S (2017) Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environmental models. *Environmental Modelling and Software* 93: 109–126. https://doi.org/10.1016/j.envsoft.2017.03.010
27. Shields MD, Zhang J (2016) The generalization of Latin hypercube sampling. *Reliab Eng Syst Saf* 148: 96–108. https://doi.org/10.1016/j.ress.2015.12.002
28. Wu J, Chen XY, Zhang H, Xiong LD, Lei H, Deng SH (2019) Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology* 17: 26–40. https://doi.org/10.11989/JEST.1674-862X.80904120
29. Thelen A, Zohair M, Ramamurthy J, Harkaway A, Jiao W, Ojha M, et al. (2023) Sequential Bayesian optimization for accelerating the design of sodium metal battery nucleation layers. *J Power Sources* 581: 233508. https://doi.org/10.1016/j.jpowsour.2023.233508
30. Shin S, Lee Y, Kim M, Park J, Lee S, Min K (2020) Deep neural network model with Bayesian hyperparameter optimization for prediction of NOx at transient conditions in a diesel engine. *Eng Appl Artif Intell* 94: 103761. https://doi.org/10.1016/j.engappai.2020.103761
31. Burrage K, Burrage P, Donovan D, Thompson B (2015) Populations of models, experimental designs and coverage of parameter space by Latin Hypercube and Orthogonal sampling. *Procedia Computer Science*, 1762–1771. https://doi.org/10.1016/j.procs.2015.05.383
32. Shu Z, Jirutitijaroen P (2011) Latin hypercube sampling techniques for power systems reliability analysis with renewable energy sources. *IEEE T Power Syst* 26: 2066–2073. https://doi.org/10.1109/TPWRS.2011.2113380
33. Deutsch JL, Deutsch CV (2012) Latin hypercube sampling with multidimensional uniformity. *J Stat Plan Inference* 142: 763–772. https://doi.org/10.1016/j.jspi.2011.09.016
34. Nguyen V (2019) Bayesian optimization for accelerating hyper-parameter tuning. *Proceedings - IEEE 2nd International Conference on Artificial Intelligence and Knowledge Engineering, AIKE*, 302–305. https://doi.org/10.1109/AIKE.2019.00060
35. D. Singh D, Singh B (2022) Feature wise normalization: An effective way of normalizing data. *Pattern Recogn* 122: 108307. https://doi.org/10.1016/j.patcog.2021.108307
36. Islam MJ, Ahmad S, Haque F, Reaz MB, Bhuiyan MA, Islam MR (2022) Application of Min-Max Normalization on Subject-Invariant EMG Pattern Recognition. *IEEE T Instrum Meas* 71: 1–12. https://doi.org/10.1109/TIM.2022.3220286
37. Szabó S, Holb IJ, Abriha-Molnár VE, Szatmári G, Singh SK, Abriha D (2024) Classification Assessment Tool: A program to measure the uncertainty of classification models in terms of class-level metrics. *Appl Soft Comput* 155: 111468. https://doi.org/10.1016/j.asoc.2024.111468
38. Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21: 1–13. https://doi.org/10.1186/s12864-019-6413-7
39. Borkhade AD (2014) Transmission line fault detection using wavelet transform. *International Journal on Recent and Innovation Trends in Computing and Communication* 2: 3138–3142.
40. Teja ON, Ramakrishna MS, Bhavana GB, Sireesha K (2020) Fault Detection and Classification

in Power Transmission Lines using Back Propagation Neural Networks. *Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC 2020)*, 1150–1156.
41. Fan R, Liu Y, Huang R, Diao R, Wang S (2018) Precise Fault Location on Transmission Lines Using Ensemble Kalman Filter. *IEEE T Power Deliver* 33: 3252–3255. https://doi.org/10.1109/TPWRD.2018.2849879
42. Arranz R, Paredes Á, Rodr ́guez A, Mu ̃oz F (2021) Fault location in Transmission System based on Transient Recovery Voltage using Stockwell transform and Artificial Neural Networks. *Electr Pow Syst Res* 201: 107569. https://doi.org/10.1016/j.epsr.2021.107569
43. Moradzadeh A, Teimourzadeh H, Mohammadi-Ivatloo B, Pourhossein K (2022) Hybrid CNN-LSTM approaches for identification of type and locations of transmission line faults. *Int J Electr Power* 135: 107563. https://doi.org/10.1016/j.ijepes.2021.107563
44. Shang B, Luo G, Li M, Liu Y, Hei J (2023) Transfer learning-based fault location with small datasets in VSC-HVDC. *Int J Electr Power* 151: 109131. https://doi.org/10.1016/j.ijepes.2023.109131
45. Pouabe Eboule PS, Pretorius JHC, Mbuli N (2018) Artificial Neural Network Techniques apply for Fault detecting and Locating in Overhead Power Transmission Line. *2018 Australasian Universities Power Engineering Conference (AUPEC)*, 1–6
46. Said A, Saad MH, Eladl SM, Elbarbary ZS, Omar AI, Saad MA (2023) Support Vector Machine Parameters Optimization for 500 kV Long OHTL Fault Diagnosis. *IEEE Access* 11: 3955–3969. https://doi.org/10.1109/ACCESS.2023.3235592

AIMS Press