KFUPM
Journal of Undergraduate Research International

Diamond Open Access

Article
Computer Science/Electrical Engineering

# Data-Driven Deep Learning Fault Diagnosis Model for Smart Grids Under Load Demand Uncertainty

Khalid Alfuwail, Hamza M. Anwar, Md Shafiullah*, Md Mahfuzur Rahman, and Mohamed Ali Abido

Cite https://doi.org/10.64589/juri/215357

**Submitted:** March 20, 2025 **Revised:** November 25, 2025 **Accepted:** December 09, 2025

ABSTRACT

Faults are common hazards in electricity distribution networks and should be diagnosed immediately to prevent harm and restore the power supply. This study proposes a hybrid approach for fault diagnosis that combines deep learning models and advanced signal-processing approaches. The proposed method creates a standard four-node test distribution grid in MATLAB/Simulink, extracts features using discrete wavelet transform (DWT) and short-time Fourier transform (STFT), and trains deep learning models to identify, categorize, and pinpoint faults. The results show that while both STFT and DWT perform well in fault detection, STFT demonstrates superior robustness in fault localization and categorization under noisy conditions owing to its uniform time–frequency resolution. The proposed framework uniquely integrates load demand uncertainty, measurement noise, and a hybrid MATLAB–Python pipeline, providing a more realistic evaluation than previous studies. The proposed models are not affected by variations in the pre-fault loading conditions, fault inception angle, or fault resistance. The proposed framework is suitable for deployment in real urban distribution grids with significant electrical interference. The main advantage is the reduced fault location time, which reduces outages and improves reliability.

**Keywords:** deep neural networks, fault detection, distribution grids, time frequency analysis, short-time fourier transform, wavelet transform

## 1. INTRODUCTION

Electricity is a necessity in the modern world, serving as the basis for power generation, transmission, and distribution. However, as energy demand increases, the grid becomes more crowded and thus more prone to malfunctions, such as short circuits, which can cause power outages [1–3]. These faults can disrupt daily routines and essential tasks. Therefore, it is essential to provide steady and seamless electrical transmission from utilities to consumers. Identifying the location of a grid breakdown quickly is crucial for minimizing customer minutes lost and accelerating electricity restoration. Effective methods are required to accurately identify, categorize, and pinpoint defects to ensure the safety and efficiency of electrical grids [4–6].

Grid fault detection techniques are categorized into knowledge-based, traveling-wave-based, and impedance-based methods [7]. The impedance-based approach estimates the fault distance using voltage and current measurements, offering simplicity and cost-effectiveness [8]. However, it suffers from inaccuracies owing to iterative calculations, dynamic loads, noise, and laterals [9]. The traveling wave method analyzes fault-generated pulses using the arrival time, wave velocity, phase, and polarity to locate faults from one or both line ends [10,11]. Although effective for long transmission lines, it is unsuitable for short distribution lines (accuracy: 100–500 m) and requires costly infrastructure [12].

Knowledge-based approaches, particularly machine learning (ML), are promising for fault diagnosis. ML techniques adapt by identifying patterns in the training data without explicit programming, thereby proving effective for fault locations in distribution grids [13]. They excel in handling fault complexities and can be integrated with signal processing to obtain hybrid solutions. For example, Mamuya [2] combined a wavelet transform (WT) for feature extraction with artificial neural networks (ANN) to enhance the fault detection accuracy. The versatility of ML and data-driven learning make it a robust alternative to traditional methods.

**Wavelet-based approaches:** Rizeakos et al. [19] combined continuous wavelets transform (CWT) with CNNs for fault location, whereas Naidu et al. [21] integrated discrete wavelets transform (DWT) with an ANN optimized via particle swarm optimization for high-impedance fault (HIF) location. Iliyaeifar and Hadaeghi [22] paired DWT with an extreme learning machine (ELM) for HVDC systems, and Rezaee Ravesh et al. [23] used wavelet packets transform with an ANN for hybrid transmission lines. While these approaches capture multi-resolution signal features effectively, most have been validated only on HVDC or hybrid systems with limited testing under noisy AC distribution conditions.

**Fourier-based approaches:** Grcić et al.[24] applied short-time Fourier transform (STFT) with ML classifiers in DC microgrids, but lacked ANN-based classification, whereas Shafiullah et al.[7] used S-transform for optimized fault location in distribution grids. These studies indicate the potential of STFT for fault diagnosis but leave open the question of its comparative performance against wavelet methods under noise and load variations.

**Other ML-based approaches:** Lan et al.[14] used empirical mode decompositions (EMD) with a 1D-CNN for HVDC gearbox fault detection, although only in simulations. Yu et al.[15] introduced a directed tree-based traveling wave method and Babayomi and Oluseyi[16] applied an adaptive neuro-fuzzy inference system, with both studies having limitations in repeated fault handling. Advanced models, such as histogram-based gradient boosting (HGB)[17] and graph convolutional networks (GCNs)[18], have shown promise in specific scenarios, but have not been benchmarked against hybrid STFT/DWT approaches.

**Research gap:** Across these studies, there has been limited validation of realistic and noisy AC distribution systems, insufficient analysis of repeated faults, and no systematic comparison between STFT and DWT integrated with deep learning for simultaneous detection, classification, and location tasks under-load demand uncertainty. This study addresses these limitations.

Most knowledge-based methods rely on MATLAB toolboxes, such as Statistics and Machine Learning[25] and Deep Learning[26]. While effective, exploring open-source alternatives, such as Python, could improve the accessibility and comparison of model efficacy[27]. Python offers powerful ML libraries (scikit-learn, TensorFlow, and Keras)[28–32] and signal processing tools with simpler syntax than MATLAB. Combining MATLAB signal processing with Python ML capabilities may optimize fault detection in power grids. Future work should leverage Python open-source libraries to enhance fault identification while maintaining performance.

Current fault diagnosis methods often overlook key factors, such as dynamic load variations, measurement noise, and practical deployment constraints. This study addresses these limitations by proposing a hybrid fault diagnosis approach for distribution networks that explicitly incorporates the load demand uncertainty, multiple noise levels, and a new set of statistical features. This combination, along with MATLAB–Python integration, enables a more comprehensive and realistic evaluation than existing studies. This study employs Python-based deep learning models and makes the following contributions:

- The uncertainty around the fault information and load demand is considered when modeling the test distribution feeders.
- From the collected data, features are extracted using DWT and STFT.
- Deep neural networks with several hidden layers are used to create smart defect diagnostic models (detection, classification, and localization) using Python and its modules.
- For fault location, the train-test split for both the complete dataset and individual fault types must be varied to determine the split that yields the best results.
- The sensitivity of the resulting models is evaluated while considering the network-related uncertainty and measurement noise.

The remainder of this paper is structured as follows: Section 2 details the deep learning and signal processing methods for fault detection. Section 3 describes the distribution feeder model. Section 4 explains the feature extraction and proposed fault detection approach. Section 5 compares the model results. Finally, Section 6 summarizes the study and provides the conclusions.

## 2. METHODS

### 2.1. Signal Processing Techniques.

Signal processing transforms signals to reveal key components[33]. For grid fault diagnosis, mathematical transformations (e.g., STFT and DWT) extract features from the current signals, which are then fed to the ML algorithms. This study utilizes both STFT and DWT as feature extraction tools. The following section explains the theoretical foundations of these transformers.

#### *2.1.1. Short-time Fourier Transform.*

The Fourier transform decomposes signals into frequency components but loses temporal information[34]. STFT overcomes this problem by analyzing equal-length signal segments and providing time-localized frequency data[34]. It multiplies the signal $x(t)$ by a short-duration window function $w(t)$ (e.g., Blackman or Hann). The STFT of $x(t)$ is expressed as follows:

$$X(\tau, f) = \int_{-\infty}^{+\infty} x(t)\, w(t-\tau)\, e^{-i2\pi f t} dt \tag{1}$$

where $t$ is time, $f$ is frequency, and $w(t-\tau)$ is the window function centered at $\tau$. This enables joint time-frequency analysis for fault detection.

Most computer applications, including MATLAB, use a fast Fourier transform (FFT) to compute STFT features[35]. The STFT of a discrete signal $x(n)$ is determined as follows:

$$X_m(f) = \sum_{n=-\infty}^{+\infty} x(n)\, w(n-mR)\, e^{-i2\pi f n} \tag{2}$$

The STFT process uses a frame index $n$, window $w(t)$ centered at $mR$, and $R$ sample intervals[36]. As $w(n)$ slides along the signal length $M$, the discrete Fourier transform (DFT) computes each windowed segment. Overlap $L = M - R$ occurs as adjacent segments hop $R$ samples, creating overlapping analysis windows for a continuous time-frequency representation[37].

#### *2.1.2. Wavelet transform.*

WT overcomes the Fourier transform limitations by providing a multi-resolution time-frequency analysis[38]. Using variable-width wavelets derived from a mother wavelet $\psi(t)$ (e.g., Morlet or Daubechies), it captures signal components at different scales. Unlike the fixed windows in STFT, wavelets adapt their widths to be narrow for high frequencies and wide for low frequencies. CWT slides these scalable wavelets along the signal, multiplying and analyzing them from high to low frequencies. Mathematically, the CWT of signal $f(t)$ is represented as $F(\tau, s)$, where $\tau$ is translation and $s$ is scale. This approach enables precise localization of both transient and sustained features in fault signals:

$$F(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(t)\, \psi^* \left( \frac{t - \tau}{s} \right) dt \tag{3}$$

CWT uses $\psi^*\left(\frac{t-\tau}{s}\right)$ *as the daughter wavelet, where* $\psi$ is the complex conjugate of the mother wavelet, $s$ controls the scale/frequency, and $\tau$ determines the position[39]. For computational efficiency, DWT discretizes this approach. The DWT $D[a, b]$ of signal $f[t_m]$ analyzes discrete translations (b) and dyadic scales ($a = 2^j$), enabling efficient multi-resolution analysis, as follows:

$$D[a, b] = \frac{1}{\sqrt{b}} \sum_{n=0}^{p-1} f[t_m]\, \psi \left[ \frac{t_m - a}{b} \right] \tag{4}$$

where $a$ and $b$ are both integers, the scale parameter is $s = b$, and the translation parameter is $\tau = a$[39]. To determine the approximation and detail coefficients (low- and high-frequency components, respectively) of the input signal, DWT uses a wavelet-based technique known as multi-level decomposition, in which the signal is processed through many low- and high-pass filters [4]. An example of this process is shown in Figure 1.

**2.2. Deep Neural Network.** ML enables systems to learn patterns from training data and make predictions. Inspired by biological neurons, ANNs are fundamental ML models composed of interconnected (input, hidden, and output) layers. Feed-forward neural networks (FNNs), which are the simplest ANNs, process data through weighted connections and biases between layers (Figure 2)[40]. During training, the input data propagates forward by adjusting the weights to minimize errors. Deep neural networks (DNNs) extend this approach to multiple hidden layers for complex pattern recognition. Typically, datasets are split into training and testing subsets to evaluate their performance. These architectures form the basis of advanced fault diagnosis systems.

Neural networks apply activation functions (ReLU, sigmoid, tanh, and softmax) to weighted inputs, thus introducing nonlinearity into complex pattern learning[41]. During training, the output layer errors (predicted versus desired) drive backpropagation, which is an optimization process that adjusts weights and biases to minimize loss[4]. This iterative refinement enables the DNNs to model intricate relationships. After training, the models are evaluated using unseen test data by comparing predictions against known outputs and quantifying performance through error metrics. The combination of forward propagation (feature transformation) and backward error correction (gradient-based optimization) allows deep networks to progressively improve their predictive accuracy through exposure to labeled training samples.

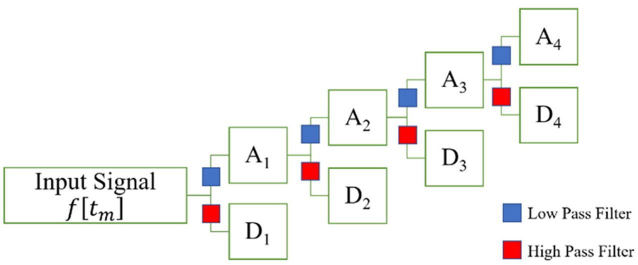


**Figure 1.** An example of multi-level decomposition up to four levels. Before each level, the signal is passed through a high-pass filter, from which the detailed coefficients ($D_i$) are obtained, and a low-pass filter, from which the detailed coefficients ($A_i$) are obtained

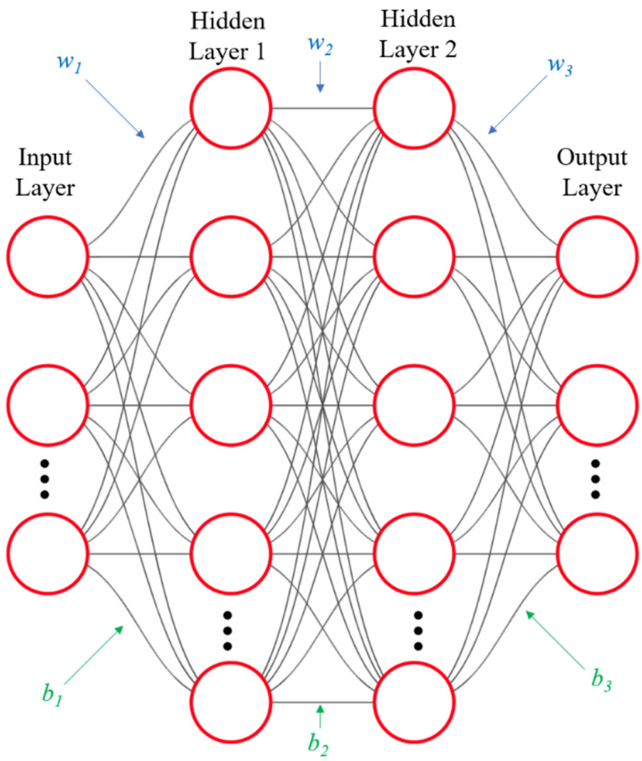


**Figure 2.** DNN model used for fault detection (binary), fault classification (seven fault types), and fault location (continuous value between 5–15 km). The input vector contains either 108 STFT-based features or 144 DWT-based features per sample

## 3. DISTRIBUTION NETWORK MODEL UNDER LOAD DEMAND UNCERTAINTY

Figure 3 shows the 60 Hz test feeder with transformers, a lumped load, a transmission line, and four nodes. The distribution grid parameters are listed in Table 1. All data in this study were generated using MATLAB/Simulink "Simscape Electrical" with discrete power systems blockset (PSB)[42]. Fault events were labeled by type, location, resistance, and inception angle directly from the simulation parameters, ensuring an accurate ground truth for training and testing. A 20 kHz sampling rate (333 samples/cycle) and two-block transmission lines enabled fault simulations at multiple locations. The dataset contained 14,000 samples per signal processing method (STFT and DWT), evenly distributed across seven fault types and non-fault cases, with four noise levels (0, 20, 30, and 40 dB). This resulted in a balanced dataset of 64,000 samples (4 noise levels × 8 conditions × 2,000 samples).

## 4. PROPOSED FAULT DIAGNOSIS METHOD

This section details the steps involved in developing the proposed fault diagnostic approach. It covers the application of flaws; the parameters of signal processing techniques, including the

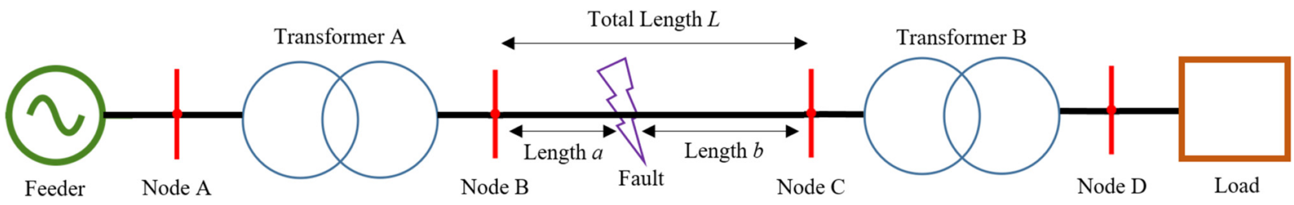


**Figure 3.** Modeled four-node distribution grid used for simulation. The system consists of transformers, a lumped load, and transmission lines. Faults are applied at different locations along the distribution line between Nodes B and C

features extracted from them; the preparation of data; and the construction of deep learning models for fault diagnosis.

**4.1. Fault Application.** Three-phase fault currents were recorded from Node B (Figure 3) for various fault types (AG, BG, CG, ABG, BCG, ACG, and ABCG) applied between nodes B and C, capturing one pre-fault and one post-fault cycle. The fault parameters (type, distance, resistance, and inception angle) and load demands were varied systematically. The non-fault data were generated by varying only the load. The MATLAB/Simulink "awgn" function created signals with 0, 20, 30, and 40 dB noise levels[4]. Figure 4 shows a sample fault current waveform. This comprehensive dataset enables a robust analysis of fault characteristics under different system conditions and noise environments.

**4.2. Feature Extraction.** This study generated 64, 000 samples (16, 000 per noise level) using STFT and DWT in MATLAB, covering seven fault types and normal conditions. Each noise level (0, 20, 30, and 40 dB) contained 2, 000 samples per fault scenario ($8\ conditions \times 2,000 = 16,000$). The subsequent sections detail the feature extraction parameters for both signal processing methods.

***4.2.1. STFT parameters.*** The MATLAB "stft" function processed three-phase current signals using a 128-sample Hann window with 96-sample overlap and 128-point FFT. This produced a time-frequency matrix for each phase. From each matrix, the following statistical features were extracted for each phase:

- The maximum values and energies within each of the 17 frequency columns, resulting in 34 features (17 maxima and 17 energy values).

**Table 1.** *Test four-node distribution grid parameters*[4]

| Component | Parameter | Value |
|---|---|---|
| Distribution Line | Length | 20 km |
| | Positive Sequence Resistance | 0.1153 Ω/km |
| | Positive Sequence Inductance | 1.05 mH/km |
| | Positive Sequence Capacitance | 11.33 nF/km |
| | Zero Sequence Resistance | 0.413 Ω/km |
| | Zero Sequence Inductance | 3.32 mH/km |
| Lumped Load | Active Power | 6 MW |
| | Reactive Power | 3 MVar (Inductive) |
| System Variation | Load Demand Uncertainty | ±5% |
| Fault Parameters | Impedance (R~f~) | 0.1 – 15 Ω |
| | Inception Angle | 0–360° |
| | Location | 5–15 km |
| Transformer A | Voltage Ratio | 33/11 kV |
| Transformer B | Voltage Ratio | 11/4.16 kV |
| Transformers A and B | Power Rating | 12 MVA |
| | Winding 1 (R, L) | 0.0434 Ω, 3.454 mH |
| | Winding 2 (R, L) | 68.88 μΩ, 5.481 μH |
| | Magnetization Branch (R~m~, L~m~) | 26042 Ω, ∞ H |
| | Connection | Y~g~-Δ |

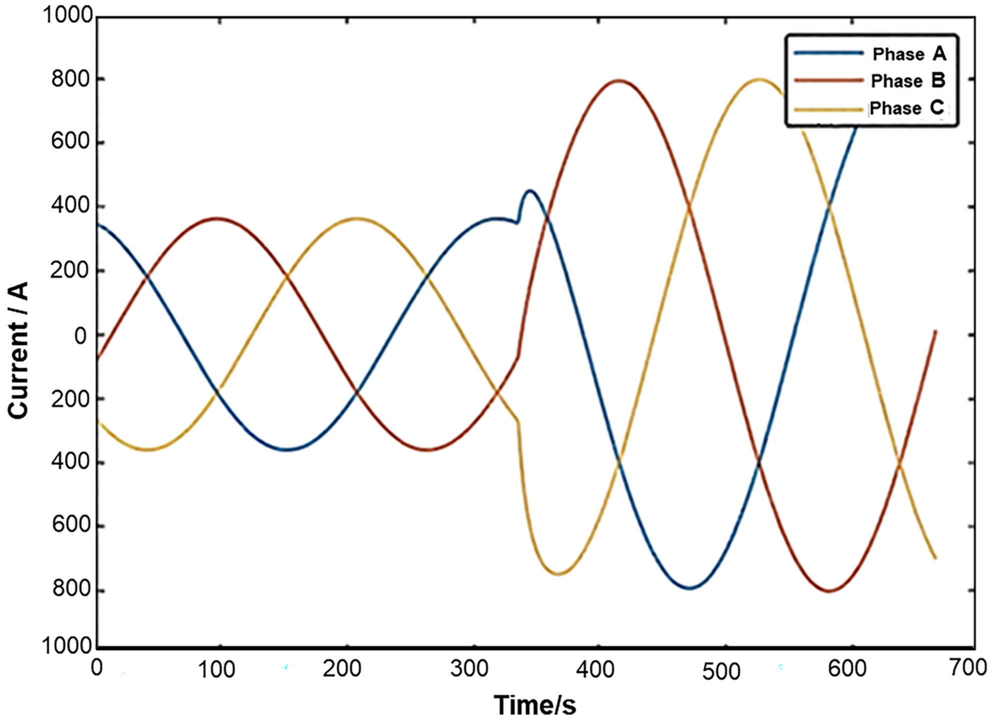


**Figure 4.** Simulated three-phase current waveform for an ABCG (three-phase-to-ground) fault

- The standard deviation of the maximum values across the time frames (rows) and frequency bins (columns), yielding two features.
- The overall mean magnitude of the time-frequency matrix, adding one final feature.

This resulted in 36 features per phase (17+17+1+1). Using three-phase signals, each sample produced 108 features (36×3). This feature set effectively captured the time-frequency characteristics and statistical properties of fault transients. The frequency range was centered during STFT computation. This feature extraction method captured both time-frequency characteristics (via STFT matrices) and statistical properties, enabling comprehensive fault representation. This approach systematically transformed raw current signals into a discriminative feature set suitable for ML classification while maintaining physical interpretability through energy and deviation metrics.

***4.2.2. DWT parameters.*** MATLAB *wavedec*, *appcoef*, and *detcoef* functions performed seven-level DWT decomposition using 'db4' wavelet, extracting one approximate and seven detailed coefficients. Six statistical measures (std, mean, entropy, energy, kurtosis, and skewness) were computed for each coefficient, yielding 144 features. The implementation details are provided in Shafiullah et al.[4].

**4.3. Input/Output Features and Data Preparation.** The extracted features were stored as .mat files and imported into Python using the SciPy *loadmat* function[45], and converted to Pandas DataFrames[31]. Each dataset included (1) binary fault indicators (1/0), (2) fault type labels $(1-7)$, and (3) fault distances from Node B. For each noise level $(0, 20, 30,\ and 40\ dB)$, separate datasets were created by combining all fault scenarios $(seven types + normal)$, resulting in eight datasets per signal processing method (STFT/DWT). Data were shuffled to prevent pattern memorization. Feature normalization (0–1 range) was performed using the scikit-learn *MinMaxScaler*[30], which is particularly critical for regression tasks, such as fault distance prediction. This preprocessing pipeline ensured compatibility with Python deep learning frameworks while maintaining the physical meaning of the fault characteristics. The structured datasets enabled both classification (fault detection/typing) and regression (location estimation) tasks, with normalized features improving the model convergence during training. The input to the deep learning models consisted of statistical features extracted from STFT or DWT, whereas the outputs varied by task: binary labels (1 = fault, 0 = no fault) for detection, multiclass labels (1–7) for fault types, and continuous values (5–15 km) for fault location. No manual feature engineering was performed and the features were fully derived from the raw current signals.

**4.4. Deep Learning Models.** The Keras library[29] was used to build three ML models per signal processing method: fault detection, classification, and localization. The classification/localization models used DNNs with two optimized hidden layers (to balance complexity and performance). The hyperparameters were systematically tuned via trial-and-error to achieve optimal results. Deep learning architectures were selected because of their enhanced pattern-recognition capabilities in complex fault-diagnosis tasks.

**4.5. Performance Evaluation Metrics.** Model performance was evaluated using task-specific metrics: accuracy for fault detection/classification (correct predictions/total samples) and regression metrics ($R^2$, mean absolute error (MAE), and root mean square error (RMSE)) for fault location[4]. Location predictions were denormalized to the 5–15 km range before metric

**Table 2.** *Detailed feature extraction pipeline from three-phase current signals*

| Step | Description | STFT-Based Features | DWT-Based Features |
|---|---|---|---|
| 1. Input | Raw three-phase current signals (Ia, Ib, and Ic), sampled for one pre-fault and one post-fault cycle. | Identical Input | Identical Input |
| 2. Signal Transform | Apply the signal processing technique. | **STFT**<br>●Creates a time-frequency matrix for each phase.<br>●Parameters: Hann window (128 samples), 96-sample overlap, 128-point FFT. | **DWT**<br>●Decomposes the signal of each phase into coefficients.<br>●Parameters: 'db4' wavelet, seven levels of decomposition. |
| 3. Raw Output | The direct result of the transform. | A complex-valued matrix X(f, t) for each phase, showing the signal strength at different frequencies (f) and times (t). | A set of eight coefficient vectors for each phase:<br>●A7: Approximation coefficients (low-frequency trend).<br>●D1-D7: Detailed coefficients (high-frequency details, D1 is the highest frequency). |
| 4. Feature Calculation (the key difference) | Conversion of the raw output into a set of numbers (features) for the artificial intelligence (AI) model. | For the time-frequency matrix of each phase:<br>1. Maxima: Find the maximum value in each of the 17 frequency columns → 17 features.<br>2. Energy: Calculate the energy in each of the 17 frequency columns → 17 features.<br>3. Standard Deviations:<br>- std(max_values_across_time) → 1 feature<br>- std(max_values_across_frequency) → One feature<br>→ Subtotal per phase:<br>17 + 17 + 1 + 1 = 36 features. | For the set of coefficients of each phase:<br>●Take all eight coefficient vectors (A7, D1, D2, D3, D4, D5, D6, and D7).<br>●For each vector, calculate six statistical measures:<br>Standard Deviation, Mean, Entropy, Energy, Kurtosis, and Skewness.<br>→ Subtotal per phase:<br>8 vectors ×6 statistics = 48 features. |
| 5. Total Features per Sample | Combine features from all three phases. | 3 phases ×36 features = 108 features. | 3 phases ×48 features = 144 features. |
| 6. Final Step | Prepare the final dataset for the AI model. | All features from all samples are normalized to a [0, 1] range. | All features from all samples are normalized to a [0, 1] range. |

calculation. To ensure reproducibility despite Keras stochastic weight initialization, a fixed random seed (42) was used for all the experiments. The optimal models were selected after multiple trials to confirm their stable performance, and the reported results were obtained from these finalized models. The optimal models were selected after multiple trials when the performance stabilized, thereby ensuring consistent metrics. This approach accounted for the inherent variability of neural networks, while identifying the most reliable configurations. The evaluation framework provided a comprehensive assessment of both classification precision (accuracy) and regression fidelity ($R^2$, MAE, and RMSE) for a complete fault diagnosis.

**4.6. Deep Learning Models Development.** The models were optimized using systematic hyperparameter tuning. The neuron counts in the hidden layers followed heuristic rules[46], with STFT-based models using 108 neurons (matching input features) in layer 1 and 55 neurons (half input count) in layer 2. The DWT models employed different input fractions (96, 144, and 72 - two-thirds ratios). The fault location models used similar proportional schemes. Table 2 lists the tested neuron combinations. The selection process evaluated the performance across these configurations, adhering to established neural network design principles[46]. Optimal architectures balanced complexity and generalization, with neuron counts systematically derived from the input dimensionality rather than from arbitrary choices.

For fault classification, optimal neuron configurations were determined through performance comparisons. The STFT model achieved the peak accuracy with 108 (layer 1) and 60 (layer 2) neurons, showing minimal loss (Figure 5). Comparable results were obtained for the 140/60, 140/70, and 120/55 combinations. The DWT model performed best with 130 and 80 neurons (Figure 6). The regression models followed the same optimization process but used $R^2$ and RMSE metrics instead of accuracy/loss. Table 3 lists the neuron counts selected for the STFT/DWT regression models. This systematic approach balanced model complexity and predictive performance while avoiding overfitting, with architectural decisions guided by empirical results rather than arbitrary selection.

**Table 3.** *Iterations of hidden layer neuron counts for the fault location and classification models*

| Signal Processing Technique | | Fault Classification (Multi-class Classification) | | Fault Location (Regression) | |
|---|---|---|---|---|---|
| | STFT | DWT | STFT | DWT | |
| Number of | Layer 1 | 90, 100, 111, 120, 130, 140 | 120, 130, 144, 150, 160 | 90, 100, 111, 120, 130, 140 | 100, 110, 120, 130, 144, 150 |
| hidden neurons | Layer 2 | 30, 40, 55, 60, 70 | 50, 60, 72, 80, 96 | 30, 40, 55, 60, 74, 80 | 50, 60, 72, 80, 96 |

This study systematically evaluated the activation functions and found that ReLU and tanh were the most effective for hidden layers across models. The output layers used the following task-specific functions: sigmoid (binary detection), softmax (multi-class classification), and linear (regression for fault location). The Adam optimizer was employed universally, except for the DWT classification model, in which RMSprop yielded better results.

For fault location, identified as the most critical challenge, additional optimization was performed on the following: (1) training sample size and (2) train-test splits. The complete dataset contained 14,000 samples (2,000 per fault type), of which 9,800 samples (70%) were initially allocated for training. Experimental increments (10% steps) revealed optimal training sizes: 7,840 samples (80%) for the STFT models (noting a performance dip at 90%) versus the full 9,800 samples for the DWT models.

Figure 7 shows how model performance (measured by $R^2$) varied with the quantity of training data. The 70/30% train-test split provided sufficient validation data while maximizing learning. This rigorous optimization process, particularly for fault location tasks, ensured a robust model performance by balancing data utilization against overfitting risks. The results demonstrate that the optimal training set sizes can vary significantly between different signal processing approaches in fault diagnosis systems.

This study evaluated train-test splits in 10% increments to optimize model performance. For STFT models, a 70/30% train-test split proved optimal, whereas the DWT models performed best with an 80/20% split. This analysis was conducted for each fault type using the previously determined ideal training sample sizes. Figure 8 shows the performance of the DWT model for one fault type, revealing the 80/20% split yielded the lowest RMSE, with the 70/30% and 90/10% splits showing comparable results. A systematic evaluation confirmed that these splits maximized generalization across both signal processing methods while preventing overfitting. Similar patterns emerged for the other fault types, although their plots were excluded for brevity.

A binary classification problem, "fault equivalent to 1" or "no-fault equivalent to 0," was the outcome of fault detection. The fault type, denoted by a number between one and seven, was the outcome of the multi-class classification problem. It is

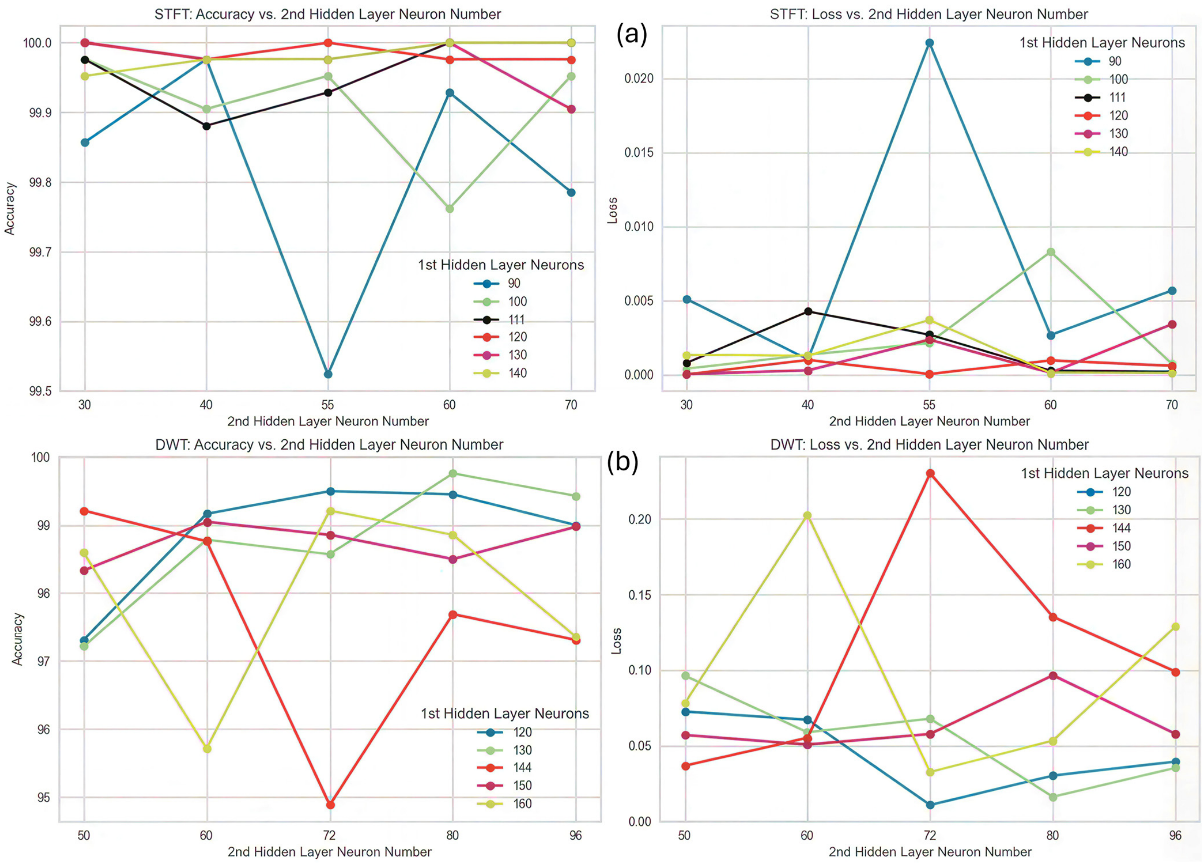


**Figure 5.** Model accuracy and loss for various neuron counts in the (a) STFT-based and (b) DWT-based fault classification models

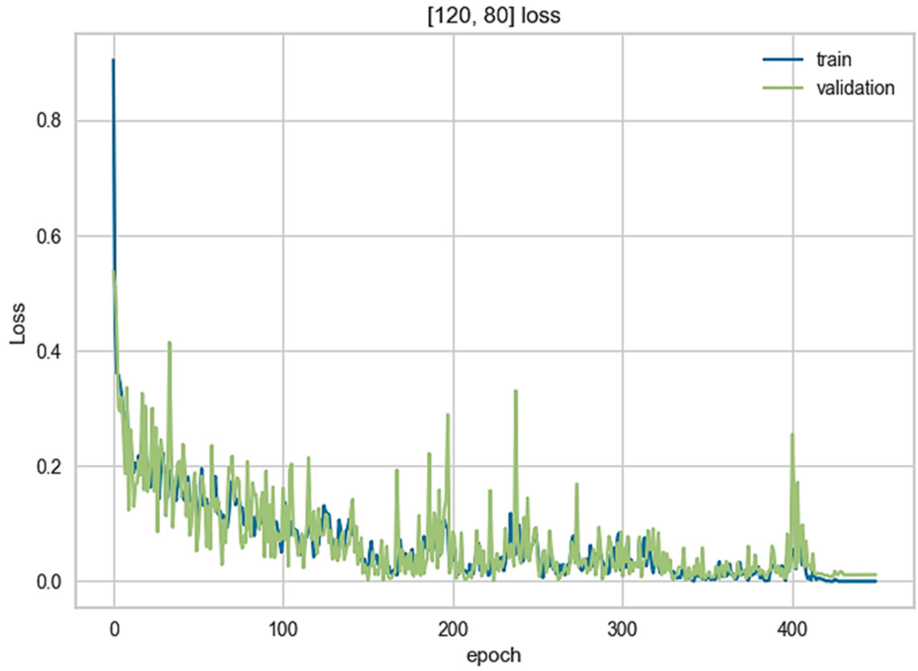


**Figure 6.** Training curve of the optimum DWT classification model

worth noting that non-faulty samples from the dataset were eliminated prior to fault classification. A decimal value that should fall between 5–15 (the range in kilometers where the faults are placed) was produced by the regression issue of fault placement. This number may exceed this range depending on the accuracy of the model. The selected parameters of the deep learning models are presented in the Table 4.

The fault detection and classification models used relatively few epochs because additional iterations provided diminishing returns in accuracy while increasing the computation time. The optimal epoch count was determined when the training curve plateaued. For these models, the training data were further split (70/30%), with 33% allocated for validation, to plot the learning curves. The 70/30% train-test split followed the conventional practice. Figure 9 shows a flowchart of the complete fault diagnosis methodology, summarizing the systematic approach from data processing to model evaluation. This balanced design ensured efficient training without overfitting while maintaining robust performance validation using dedicated validation datasets.

**Table 4.** *Fault detection model parameters*

| Task | Model | Hidden Layers & Neurons | Activation Functions | Loss Function | Optimizer | Learning Rate | Batch Size | Epochs |
|---|---|---|---|---|---|---|---|---|
| Fault Detection | STFT-based | 10 | Hidden: ReLU<br>Output: Sigmoid | Binary Cross-Entropy | Adam | 0.001 | 32 | 20 |
| | DWT-based | 10 | Hidden: ReLU<br>Output: Sigmoid | Binary Cross-Entropy | Adam | 0.001 | 32 | 20 |
| Fault Classification | STFT-based | [111, 60] | Hidden 1: Tanh<br>Hidden 2: ReLU<br>Output: Softmax | Categorical Cross-Entropy | Adam | 0.001 | 64 | 50 |
| | DWT-based | [130, 80] | Hidden 1: Tanh<br>Hidden 2: ReLU<br>Output: Softmax | Categorical Cross-Entropy | RMSprop | 0.0005 | 64 | 450 |
| Fault Location | STFT-based | [120, 55] | Hidden 1: Tanh<br>Hidden 2: ReLU<br>Output: Linear | Mean Squared Error | Adam | 0.001 | 128 | 1000 |
| | DWT-based | [110, 60] | Hidden 1: Tanh<br>Hidden 2: ReLU<br>Output: Linear | Mean Squared Error | Adam | 0.001 | 128 | 1000 |

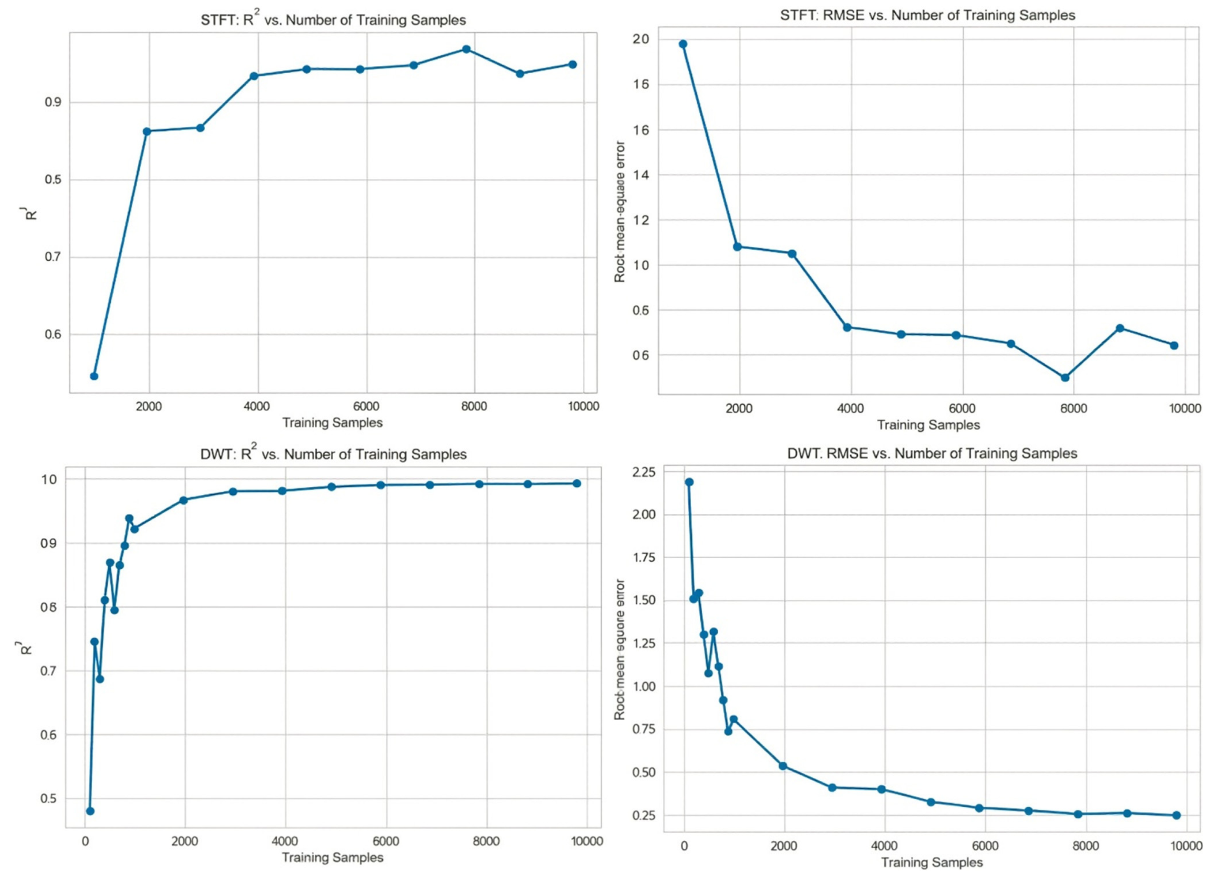


**Figure 7.** $R^2$ values and RMSEs of both regression models for various numbers of training samples

## 5. RESULTS AND DISCUSSION

This section presents the outcomes of the developed deep learning method combined with advanced signal processing techniques for fault diagnosis in the distribution grid. First, Table 5 shows that both the STFT and DWT models accurately detected faults under both noisy and noise-free measurements. In all cases, they were able to distinguish between faulty and non-faulty cases. Given the ease of discovering faults, this was anticipated. The human eye can easily distinguish between defective and non-faulty current signals. As shown in Figure 4, the current surges sharply at a fault, resulting in a significant increase in the amplitude in the current plot versus time.

Both the STFT and DWT models achieved a high accuracy (99.66% and 98.11%, respectively) across noise levels (0–40 dB). STFT delivered flawless classification (100%) under noise-free conditions and near-perfect accuracy (>99.6%) with noise, excelling for multi-phase faults. This advantage arises from the uniform time–frequency resolution of STFT, which better preserves the transient signatures of faults under noise, making it more resilient for localization tasks than the

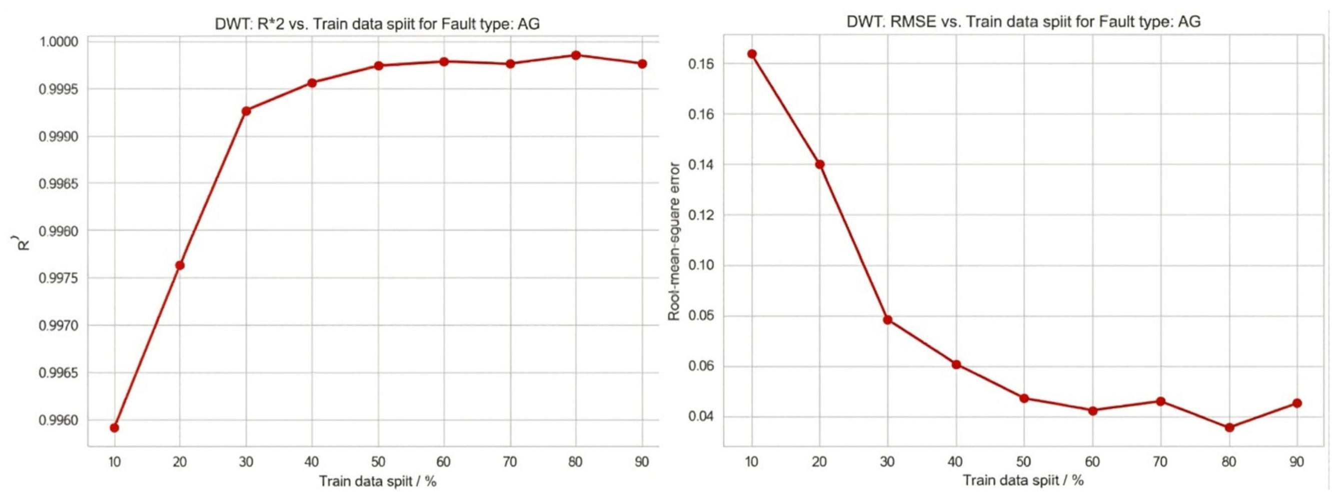


**Figure 8.** $R^2$ values and RMSEs for various training data splits of the DWT regression model for the fault type "AG"

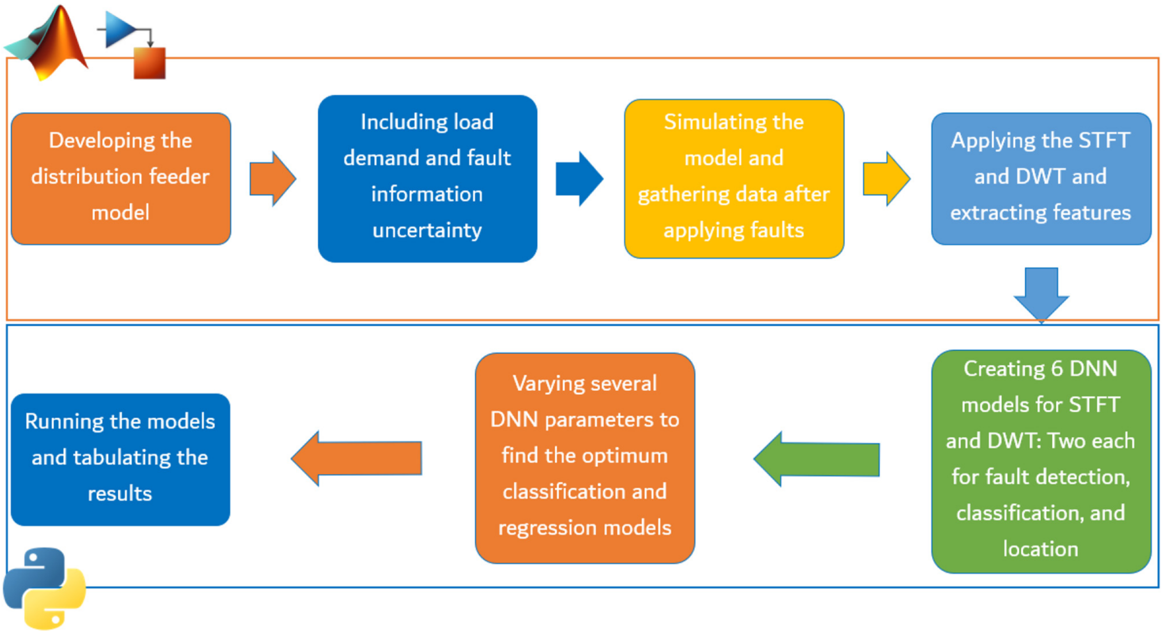


**Figure 9.** Complete fault diagnosis method

scale-dependent resolution of DWT. Minor dips occurred in the BG (20/40 dB) and ABCG (20 dB) faults. DWT also performed robustly, with perfect results in noise-free and 40 dB cases but slight inaccuracies in the CG, ACG, and BCG classifications. Both models maintained consistent accuracy (STFT: 99–100%, DWT: 98–100%), demonstrating strong noise immunity. The superior performance of STFT suggests that its time-frequency analysis better captures fault features, although both methods are reliable (<2% error) for real-world applications.

In addition to the accuracy, the precision, recall, and F1-score were computed for each fault type to evaluate the classification performance more rigorously. The STFT model achieved a macro-average precision of 0.997, recall of 0.996, and F1-score of 0.996 at a noise level of 30 dB. The confusion matrix (Figure 10) confirmed the high separability between classes, particularly for

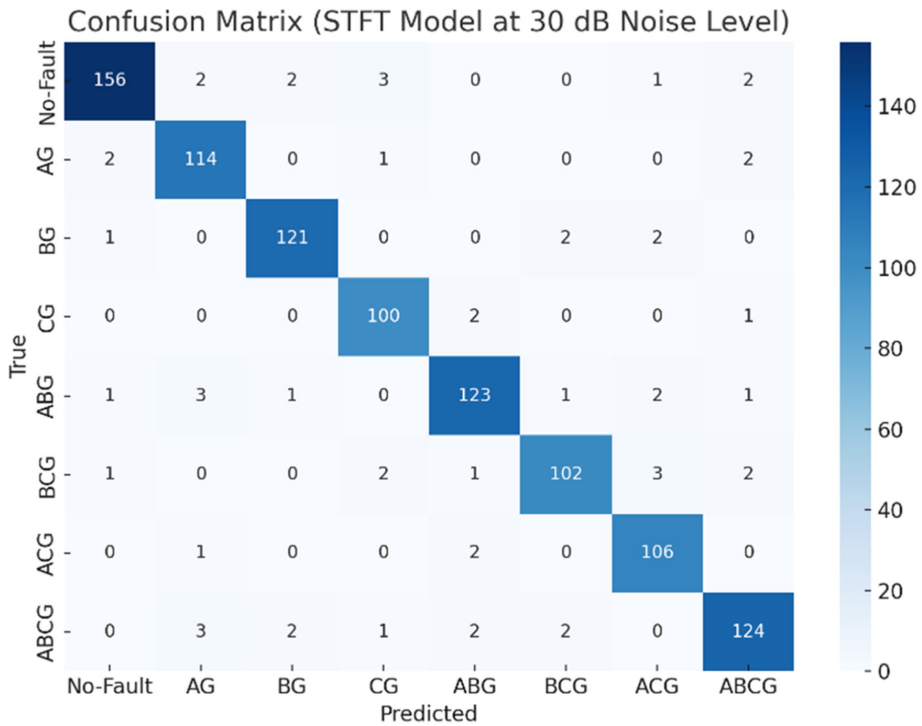


**Figure 10.** Confusion matrix for fault classification using the STFT model under 30 dB noise. The model achieved strong separability across all fault types, with minimal misclassification

**Table 5.** *Precision of the models for fault classification and detection*

| Fault type | 0 dB | 20 dB | 30 dB | 40 dB | | | | |
|---|---|---|---|---|---|---|---|---|
| | STFT | DWT | STFT | DWT | STFT | DWT | STFT | DWT |
| No-Fault | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| AG | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| BG | 100% | 100% | 99.66% | 100% | 100% | 100% | 99.82% | 100% |
| CG | 100% | 99.84% | 100% | 99.67% | 100% | 100% | 100% | 100% |
| ABG | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| BCG | 100% | 100% | 100% | 98.78% | 99.83% | 100% | 100% | 100% |
| ACG | 100% | 100% | 100% | 99.67% | 100% | 98.11% | 100% | 100% |
| ABCG | 100% | 100% | 99.82% | 100% | 100% | 100% | 100% | 100% |

the three-phase and ground faults. The high classification accuracy, along with a low $R^2$ value for fault location, reflects the difference in task complexity; classification is a discrete labeling task that is less sensitive to noise, whereas regression-based location estimation is highly sensitive to signal distortion. In particular, DWT loses location accuracy in noisy environments owing to its scale–frequency trade-off, whereas STFT maintains its performance by retaining more uniform time–frequency details.

The fault location analysis revealed distinct performance differences between the STFT- and DWT-based models. Although both achieved excellent results under noise-free conditions (DWT: $R^2 = 0.995$, STFT: $R^2 = 0.973$), their noise robustness varied significantly. The STFT model maintained a consistently high accuracy across all noise levels ($R^2 > 0.89$, $RMSE < 0.93$), demonstrating superior stability. In contrast, the DWT performance degraded sharply with noise, with $R^2$ dropping to 0.203 and RMSE >2.5 at higher noise levels.

The $R^2$ metric (range: $0 - 1$) indicated that STFT predictions closely matched actual values (≥89% variance explained), while DWT became unreliable in noisy conditions. RMSE measurements (lower=better) showed that the average location error of STFT remained below 0.93 regardless of noise, whereas DWT errors tripled in noisy scenarios. This suggests that the time-frequency localization of STFT better preserves the fault distance information under signal degradation.

Notably, DWT outperformed STFT under ideal (noise-free) conditions, achieving a near-perfect $R^2(0.995)$. However, this advantage disappeared even with moderate noise. The robust performance of the STFT model across all test conditions makes it suitable for practical implementations where noise is inevitable. These results highlight the critical importance of testing fault location algorithms under realistic and noisy conditions rather than ideal scenarios.

The severe performance degradation of the DWT-based model at 20 dB noise ($R^2 = 0.2029$) can be attributed to the fundamental properties of the wavelet transform. DWT provides multi-resolution analysis, offering a high time resolution but poor frequency resolution at high frequencies, and vice-versa. Under significant noise, the high-frequency noise components can corrupt the detail coefficients (D1–D4) that are crucial for pinpointing the precise transients caused by a fault. Unlike STFT, which maintains a uniform time-frequency resolution across the spectrum, the adaptive windowing of DWT may amplify the influence of noise in specific sub-bands, effectively drowning out the salient features needed for accurate distance regression. This inherent sensitivity makes the DWT model vulnerable to noisy conditions, whereas the consistent windowing approach of STFT is far more robust for the fault location task.

Both models exhibited unexpected performance variations across noise levels. Surprisingly, 40 dB noise yielded better results than 30 dB; STFT achieved higher $R^2$ (0.97 versus 0.96) and lower RMSE (0.52 versus 0.60), with similar trends for DWT. However, increasing the noise from 0–40 dB degraded the overall performance. The $R^2$ value of the STFT model decreased from 0.973 (noise-free) to 0.968 (40 dB), whereas its RMSE remained stable (<0.93). The DWT model suffered more severe degradation: $R^2$ decreased sharply from 0.995 (0 dB) to 0.903 (40 dB), and the RMSE increased from 0.202 to 0.902.

Table 6 shows that higher noise (20 dB) caused greater performance loss than moderate levels (30/40 dB). STFT demonstrated resilience ($R^2$>0.89 at 40 dB), while DWT became unstable ($R^2$ as low as 0.203), suggesting the time-frequency analysis of STFT better preserves the fault location accuracy under distortion. The non-linear noise impact (inverse trend at 30/40 dB) warrants further study. The consistency of STFT favors noisy environments, though real-world validation is needed. Computational costs from STFT-deep learning integration may challenge

**Table 6.** *Fault location model results*

| | STFT | | | | DWT | | | |
|---|---|---|---|---|---|---|---|---|
| Performance Metrics | 0 dB | 20 dB | 30 dB | 40 dB | 0 dB | 20 dB | 30 dB | 40 dB |
| $R^2$ | 0.9733 | 0.8966 | 0.9561 | 0.9675 | 0.9951 | 0.2029 | 0.7069 | 0.9032 |
| RMSE | 0.4701 | 0.9213 | 0.6070 | 0.5221 | 0.2015 | 2.5703 | 1.5842 | 0.9018 |
| MAE | 0.3351 | 0.6611 | 0.4259 | 0.3719 | 0.1296 | 1.7758 | 1.0648 | 0.6145 |

real-time use; solutions such as model pruning or edge optimization could be useful. Future work should enhance interpretability via tools such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model Agnostic Explanation (LIME) to build trust in critical infrastructure applications.

**Practical implications:** For deployment in real distribution grids, the proposed models can be integrated with phasor-measurement units (PMUs) or feeder-mounted sensors to provide near-real-time fault detection, classification, and localization. The strong noise resilience of the system, particularly with the STFT-based features, suggests its viability in urban environments with significant electrical interference. The main operational benefit is the reduced fault location time, which directly shortens outage durations and improves service reliability. The challenges include ensuring computational efficiency for edge devices and validating the performance using field data, which will be the focus of future work.

## 6. CONCLUSIONS

In this study, an intelligent fault diagnosis system that combines STFT/DWT signal processing with deep learning was developed for distribution grids. The MATLAB/Simulink test feeder simulated diverse faults under varying loads and noise levels (0–40 dB). STFT extracted 108 time-frequency features, whereas DWT derived 144 wavelet-based features per sample. These features trained deep learning models for fault detection, classification, and location. The evaluation showed that both techniques achieved high accuracy; fault detection was near-perfect, whereas classification favored STFT (99.66% versus 98.11% for DWT). For location, the STFT demonstrated superior noise robustness ($R^2$ >0.89 across all noise levels) compared to DWT, whose performance degraded sharply ($R^2$ : 0.995–0.203) under noise. The consistent RMSE of STFT (<0.93) confirmed its reliability in practical noisy environments. Future work should explore (1) validation using larger real-world grids with distributed renewable generation, (2) optimization of STFT parameters (window type/length) for enhanced feature extraction, and (3) alternative transforms such as Stockwell for comparative analysis. The adaptability of the framework to evolving grid complexities, including the high penetration of inverter-based resources, can further strengthen its diagnostic utility.

## AFFILIATIONS AND AUTHOR DETAILS


### Undergraduate Authors

**Khalid Alfuwail** – *Department of Aerospace Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia;* 0009-0002-4442-3454
Email: g202045220@kfupm.edu.sa

**Hamza M. Anwar** – *Department of Control & Instrumentation Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia; Interdisciplinary Research Center for Sustainable Energy Systems, King Fahd University of Petroleum & Minerals, Dhahran 31216, Saudi Arabia;*
Email: g201928690@kfupm.edu.sa

### Corresponding Author

**Md Shafiullah** – *Research Mentor, Assistant Professor, Department of Control & Instrumentation Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia;* 0000-0003-2282-5663
Email: shafiullah@kfupm.edu.sa

### Authors

**Md Mahfuzur Rahman** – *Assistant Professor, Department of Information and Computer Science, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia; Interdisciplinary Research Center for Intelligent Secure Systems, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia* 0000-0002-2871-9119
Email: mdmahfuzur.rahman@kfupm.edu.sa

**Mohamed Ali Abido** – *Professor, Department of Electrical Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia; Interdisciplinary Research Center for Sustainable Energy Systems, King Fahd University of Petroleum & Minerals, Dhahran 31216, Saudi Arabia;* 0000-0001-5292-6938
Email: mabido@kfupm.edu.sa


## DECLARATIONS OF COMPETING INTEREST

The authors affirm that they have no known competing financial interests or interpersonal conflicts that could have affected the research presented in this study.

## ACKNOWLEDGEMENTS

The authors acknowledge research support received from the IRC for Sustainable Energy Systems under Grant No. INSE2512, at King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia.

## REFERENCES


(1) Taft, J. D. *Fault Intelligence: Distribution Grid Fault Detection and Classification*. (Richland, Sep. 2017).

(2) Mamuya, Y. D., Lee, Y. D., Shen, J. W., Shafiullah, M. & Kuo, C. C. Application of machine learning for fault classification and location in a radial distribution grid. *Appl. Sci.* **10**, (2020).

(3) Zhang, C. L. *et al.* Fault diagnosis and removal for hybrid power generation systems based on an ensemble deep learning diagnostic method with self-healing strategies. *Int. J. Hydrogen Energy* **109**, 1297–1313 (2025).

(4) Shafiullah, M., AlShumayri, K. A. & Alam, M. S. Machine learning tools for active distribution grid fault diagnosis. *Adv. Eng. Softw.* **173**, (2022).

(5) Shafiullah, M., Abido, M. A. & Al-Mohammed, A. H. Intelligent fault diagnosis for distribution grid considering renewable energy intermittency. *Neural Comput. Appl.* **34**, 16473–16492 (2022).

(6) Wu, H. & Wang, T. A novel fault diagnosis method of smart grids based on artificial immune spiking neural P systems considering false data injection attacks. *Electr. Power Syst. Res.* **244**, 111572 (2025).

(7) Shafiullah, M., Abido, M. A. & Abdel-Fattah, T. Distribution grids fault location employing ST based optimized machine learning approach. *Energies* **11**, (2018).

(8) Stefanidou-Voziki, P., Sapountzoglou, N., Raison, B. & Dominguez-Garcia, J. L. A review of fault location and classification methods in distribution grids. *Electr. Power Syst. Res.* **209**, 108031 (2022).

(9) Shafiullah, M. & Abido, M. A. A Review on Distribution Grid Fault Location Techniques. *Electr. Power Compon. Syst.* **45**, 807–824 (2017).

(10) Jia, H. An Improved Traveling-Wave-Based Fault Location Method with Compensating the Dispersion Effect of Traveling Wave in Wavelet Domain. *Math. Probl. Eng.* **2017**, 1–11 (2016).

(11) Baseer, M. A. Travelling Waves for Finding the Fault Location in Transmission Lines. *J. Electr. Electron. Eng.* **1**, 1–19 (2013).

(12) Dutta, R. & Samantaray, S. R. Assessment of impedance based fault locator for AC micro-grid. *Renew. Energy Focus* **26**, 1–10 (2018).

(13) Labrador Rivas, A. E. & Abrão, T. Faults in smart grid systems: Monitoring, detection and classification. *Electr. Power Syst. Res.* **189**, 106602 (2020).

(14) Lan, S., Chen, M. J. & Chen, D. Y. A Novel HVDC Double-Terminal Non-Synchronous Fault Location Method Based on Convolutional Neural Network. *IEEE Trans. Power Deliv.* **34**, 848–857 (2019).

(15) Yu, K. *et al.* A Novel Traveling Wave Fault Location Method for Transmission Network Based on Directed Tree Model and Linear Fitting. *IEEE Access* **8**, 122610–122625 (2020).

(16) Babayomi, O. O. & Oluseyi, P. O. Intelligent Fault Diagnosis in a Power Distribution Network. *Adv. Electr. Eng.* **2016**, 1–10 (2016).

(17) Sahu, S. K. *et al.* Machine learning based adaptive fault diagnosis considering hosting capacity amendment in active distribution network. *Electr. Power Syst. Res.* **216**, 109025 (2023).

(18) Chen, K., Hu, J., Zhang, Y., Yu, Z. & He, J. Fault Location in Power Distribution Systems via Deep Graph Convolutional Networks. *IEEE J. Sel. Areas Commun.* **38**, 119–131 (2020).

(19) Rizeakos, V. *et al.* Deep learning-based application for fault location identification and type classification in active distribution grids. *Appl. Energy* **338**, 120932 (2023).

(20) Hong, Y. Y. & Cabatac, M. T. A. M. Fault Detection, Classification, and Location by Static Switch in Microgrids Using Wavelet Transform and Taguchi-Based Artificial Neural Network. *IEEE Syst. J.* **14**, 2725–2735 (2020).

(21) Naidu, K. *et al.* Optimized artificial neural network to improve the accuracy of estimated fault impedances and distances for underground distribution system. *PLoS One* **15**, e0227494 (2020).

(22) Iliyaeifar, M. M. & Hadaeghi, A. Extreme learning machine-based fault location approach for terminal-hybrid LCC-VSC-HVDC transmission lines. *Electr. Power Syst. Res.* **221**, 109487 (2023).

(23) Rezaee Ravesh, N., Ramezani, N., Ahmadi, I. & Nouri, H. A hybrid artificial neural network and wavelet packet transform approach for fault location in hybrid transmission lines. *Electr. Power Syst. Res.* **204**, 107721 (2022).

(24) Grcić, I., Pandžić, H. & Novosel, D. Fault detection in dc microgrids using short-time fourier transform. *Energies* **14**, 277 (2021).

(25) The MathWorks Inc. *Statistics and Machine Learning Toolbox.* (mathworks.com).

(26) The MathWorks Inc. *Deep Learning Toolbox.* (mathworks.com).

(27) Van Rossum, G. & Drake, F. L. *Python 3 Reference Manual.* (CreateSpace, Scotts Valley, CA, 2009).

(28) Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).

(29) Chollet, F. *Keras.* (keras.io).

(30) Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

(31) McKinney, W. Data Structures for Statistical Computing in Python. in *Proceedings of the 9th Python in Science Conference* 56–61 (2010).

(32) Hunter, J. D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **9**, 90–95 (2007).

(33) Oppenheim, A. V. & Schafer, R. W. *Discrete Time Signal Processing.* (Prentice Hall, 1998).

(34) Kehtarnavaz, N. *Digital Signal Processing System Design: LabVIEW-Based Hybrid Programming.* (Elsevier, 2008).

(35) Heideman, M. T., Johnson, D. H. & Burrus, C. S. Gauss and the History of the Fast Fourier Transform. *IEEE ASSP Mag.* **1**, 14–21 (1984).

(36) Smith, J. O. *Spectral Audio Signal Processing.* (W3K Publishing, 2011).

(37) The MathWorks Inc. *stft.* (mathworks.com).

(38) Nicoll, A. *The Wavelet Transform for Beginners.* (YouTube).

(39) Srivastava, M. *Improving Signal Resolution and Reducing Experiment Time in Electron Spin Resonance Spectroscopy via Data Processing Methods.* (Cornell University, 2018).

(40) LeNail, A. NN-SVG: Publication-Ready Neural Network Architecture Schematics. *J. Open Source Softw.* **4**, 747 (2019).

(41) Brownlee, J. *Using Activation Functions in Neural Networks.* (Machine Learning Mastery).

(42) The MathWorks Inc. *Simscape Electrical Toolbox.* (mathworks.com).

(43) Kachhepati, B. *Application of Short Time Fourier Transform (STFT) in Power Quality Monitoring and Event Classification.* (New Mexico State University, 2016).

(44) Mishra, S., Bhende, C. N. & Panigrahi, B. K. Detection and Classification of Power Quality Disturbances Using S-Transform and Probabilistic Neural Network. *IEEE Trans. Power Deliv.* **23**, 280–287 (2008).

(45) Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).

(46) Heaton, J. in *Introduction to Neural Networks with Java, 2nd Edition* (ed. Smith, K.) 159–159 (Heaton Research, Inc, 2008).