

Transmission Line Fault Location Analysis Using Deep Network Designer

Abdul Malek Saidina Omar, Muhammad Khusairi Osman, Siti Sarah Mat Isa,
Siti Solehah Md Ramli, Mohaiyedin Idris, Mohamad Adha Mohamad Idin, Kamarulazhar Daud*

Electrical Engineering Studies, Universiti Teknologi MARA Pulau Pinang Branch, Permatang Pauh Campus,
13500 Permatang Pauh, Pulau Pinang, Malaysia

*Corresponding author's email: kamarul395@uitm.edu.my

Article info:

Received: 30 December 2025

Revised: 16 May 2026

Accepted: 27 June 2026

DOI:

[10.69650/rast.2026.265489](https://doi.org/10.69650/rast.2026.265489)

Keywords:

Deep Network Designer
Deep Neural Networks
Long Short-Term Memory
Root Mean Square Error
Time-Series Analysis
Transmission Line Fault
Location

ABSTRACT

Fault location in transmission lines is essential for maintaining power system reliability, reducing outage duration, and improving maintenance efficiency. Conventional fault-location methods, including impedance-based and traveling-wave approaches, often experience degraded performance under high fault resistance, waveform distortion, and noisy measurement conditions. To address these limitations, this study proposes a low-code deep learning framework for transmission-line fault location estimation using a long short-term memory (LSTM) network implemented in MATLAB Deep Network Designer (DND). A 100 km, 400 kV three-phase transmission line was modeled in MATLAB-Simulink, and fault cases were generated by varying fault types, locations, resistances, and inception angles. The resulting three-phase current signals were normalized and arranged as input sequences for regression-based fault-distance estimation. The dataset comprised 7,200 simulated cases per noise condition and was divided into training, validation, and testing subsets in an 80:10:10 ratio. The model was trained using the noise-free dataset and then evaluated under three signal conditions: noise-free, 30 dB additive white Gaussian noise (AWGN), and 20 dB AWGN, in order to assess cross-noise generalization. The proposed LSTM model achieved average RMSE values of 0.8575 km, 1.0000 km, and 1.1330 km, respectively, while maintaining R^2 values above 99.86% and an overall ± 1 km accuracy of 89.44%. Comparative analysis against CNN and CNN-LSTM models indicates that the proposed approach provides competitive accuracy with lower implementation complexity in DND. These results show that the proposed LSTM-DND framework is effective for single-fault location estimation under simulated operating conditions, although further validation with mixed-noise and field-recorded data is still required.

1. Introduction

The reliability, stability, and operational efficiency of high-voltage transmission systems are crucial for maintaining a continuous and secure supply of electrical energy. Modern power systems are vast, interconnected, and increasingly complex, spanning hundreds of kilometers with multiple substations and distribution nodes. Under these circumstances, fault occurrences such as short circuits, insulation breakdowns, or line-to-ground faults pose significant risks to system stability and equipment safety. The ability to quickly and accurately locate faults on transmission lines is essential not only for minimizing power interruptions but also for reducing repair times, improving system resilience, and ensuring grid reliability [1, 2].

Traditional fault location methods, such as impedance-based, traveling wave, and phasor measurement unit (PMU)-based approaches, have been widely used for decades. Impedance-based methods estimate the fault distance by calculating the apparent impedance at the relay point; however, their accuracy is often affected by fault resistance, system nonlinearity, and load variations [3]. Similarly, traveling wave techniques determine the fault location based on the time difference between fault-generated waves reaching line terminals. Although highly accurate, they require high sampling frequencies and precise synchronization between measuring devices, leading to increased cost and implementation complexity [4].

Furthermore, PMU-based methods offer synchronized phasor measurements and improved temporal resolution but remain vulnerable to data latency, communication delays, and synchronization errors [5].

Despite advancements in signal processing, these traditional methods often fail to deliver consistent performance in complex, noisy environments or under varying fault resistances. High fault impedance, for instance, can significantly distort current and voltage waveforms, leading to inaccurate distance estimation. In addition, real-world transmission networks often contain multiple power sources and complex topologies, further complicating analytical fault location methods. These limitations have driven the shift toward intelligent, data-driven approaches capable of learning nonlinear relationships from large datasets [6, 7].

In recent years, artificial intelligence (AI) and machine learning (ML) techniques have emerged as powerful alternatives for power system fault analysis. Machine learning algorithms such as Support Vector Machines (SVMs), Decision Trees (DTs), and Random Forests (RFs) have been applied to classify and locate faults using features extracted from current and voltage signals [8]. However, these conventional ML models rely heavily on manual feature extraction, which requires expert domain knowledge and may limit scalability

when applied to diverse system conditions. Furthermore, static ML models are often insufficient to capture the temporal dependencies inherent in fault signal sequences.

To overcome these challenges, deep learning (DL) models have gained considerable traction in recent power system studies due to their capability for automatic feature learning and high accuracy in complex nonlinear problems. Among the popular architectures, Convolutional Neural Networks (CNNs) have been widely used for feature extraction from time–frequency or spectrogram representations of current signals [6, 7, 9, 11]. However, CNNs are inherently limited when dealing with raw sequential data, as they primarily focus on spatial features and lack long-term temporal memory. This limitation is particularly significant in transmission line fault analysis, where fault signals exhibit strong temporal dependencies.

Recurrent Neural Networks (RNNs) and their advanced variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, are designed to address this problem. LSTM networks, in particular, have shown exceptional capability in modeling temporal patterns in sequential data by retaining information over long periods through their memory cell architecture [10–12]. This makes LSTM highly suitable for analyzing power system fault signals, which often contain transient features that evolve dynamically during fault initiation and propagation. Previous studies have demonstrated the potential of LSTM-based models for both fault classification and fault location tasks, achieving superior accuracy compared to conventional CNN-based approaches [9–12, 14].

Nevertheless, despite the promising results of DL models, several practical challenges persist in real-world deployment. These include the need for large, high-quality datasets, model interpretability, and computational requirements for training and inference. Additionally, integrating DL models into existing relay systems requires compatibility with current hardware and communication protocols. MATLAB's Deep Network Designer (DND) offers an efficient solution to some of these challenges by providing a user-friendly, low-code interface for building, training, and testing deep neural networks. DND allows researchers and engineers to visually design and deploy neural networks without requiring extensive programming expertise, facilitating faster prototyping and experimentation.

Although recent studies have demonstrated the usefulness of DL in power-system fault analysis, three limitations remain evident in the current literature. First, many studies in [6–12, 14, 15] focus primarily on fault detection or fault classification rather than direct regression-based fault distance estimation. Second, studies that address fault location often rely on more complex hybrid architectures or additional signal transformations, which increase implementation burden and computational cost. Third, the use of low-code development environments for building and evaluating DL-based fault location models has received limited attention. Motivated by these gaps, this study develops an LSTM-based fault location framework in MATLAB Deep Network Designer using simulated three-phase current sequences under varying fault resistance, fault inception angle, and noise conditions.

The present study focuses on a 100 km, three-phase high-voltage transmission line modeled in MATLAB-Simulink under ten fault conditions, including single line-to-ground, double line, double line-to-ground, and three-phase faults. Fault location, resistance, and inception angle are varied to simulate a broad range of operating conditions, while the generated current signals are used to train an LSTM network in DND for regression-based fault distance prediction. To assess cross-noise generalization, the model is trained on the noise-free dataset and tested under noise-free, 30 dB AWGN, and 20 dB AWGN conditions. The model performance is evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), coefficient of determination (R^2), and ± 1 km accuracy, and is compared with benchmark architectures including CNN and CNN-LSTM variants.

The main contributions of this study are threefold:

- The development of a low-code LSTM regression workflow for transmission-line fault location using MATLAB Deep Network Designer.
- The generation of a simulation-based dataset covering ten fault types under varying fault location, fault resistance, fault inception angle, and noise conditions; and
- A comparative evaluation against CNN and CNN-LSTM architectures to assess the trade-off between prediction accuracy, robustness, and implementation simplicity.

Overall, this study aims to provide a practical and reproducible DL framework for transmission-line fault location estimation under simulated single-fault conditions. By combining MATLAB-Simulink-based data generation with LSTM-based sequential learning in DND, the proposed framework contributes to ongoing efforts to improve intelligent fault analysis while maintaining relatively low implementation complexity [6–12, 14, 15].

This paper is organized as follows: Section 2 presents the methodology, including system modeling, data preparation, LSTM architecture design, and training strategy. Section 3 discusses experimental results and performance metrics. Section 4 concludes the findings and proposes directions for future research, including integration of fault classification and location in a unified model.

2. Methodology

The proposed methodology for accurate transmission line fault location using a DL-based Long Short-Term Memory (LSTM) framework comprises five main stages:

- Modeling of the high-voltage transmission line in MATLAB-Simulink,
- Fault signal generation and data collection,
- Data preprocessing and feature preparation,
- LSTM architecture design and training using MATLAB's Deep Network Designer (DND), and
- Model testing, performance evaluation, and comparative analysis.

An overview of the workflow is shown in the research methodology diagram (Fig.1), which illustrates the flow from simulation to DL model deployment. Each stage is explained in detail in the following subsections.



Fig. 1 Fault Location Analysis in Transmission Line using DND.

2.1 Transmission Line Modeling

A high-voltage three-phase transmission line model was constructed in MATLAB-Simulink shown in Fig.2 to replicate the electrical behavior of a long-distance transmission network. The model consists of two voltage sources representing the sending and receiving ends, connected through a 100 km, 400 kV transmission line modeled using the distributed-parameter π -section configuration. This representation accurately reflects the distributed resistance (R), inductance (L), capacitance (C), and conductance (G) per unit length of the line, enabling realistic fault simulations [1, 2].

The equivalent π -model allows for accurate representation of the voltage and current distribution along the line, particularly under faulted conditions where line impedance and fault resistance interactions are nonlinear. The model includes measurement blocks for capturing three-phase current and voltage waveforms at the sending end, which serve as input data for subsequent fault location analysis.

Table 1 Transmission Line Parameter Setting.

Parameters	Value
Three-Phase Source	400 kV
Frequency	50 Hz
Line Length	100 km
Positive- and zero-sequence resistances	$0.01273 + j0.3864 \Omega/\text{km}$
Positive- and zero-sequence inductances	$0.9337 + j4.1264 \text{ mH}/\text{km}$
Positive- and zero-sequence capacitances	$12.74 + j7.751 \text{ nF}/\text{km}$

The transmission line parameters, summarized in Table 1, are derived from standard IEEE test system values and adjusted to emulate typical Malaysian grid configurations. Key parameters include:

- Line voltage: 400 kV
- System frequency: 50 Hz
- Line length: 100 km
- Positive-sequence resistance (R_1): $0.01273 \Omega/\text{km}$
- Positive-sequence inductance (L_1): $0.9337 \text{ mH}/\text{km}$
- Positive-sequence capacitance (C_1): $12.74 \text{ nF}/\text{km}$

To ensure model realism, a source impedance was added to each end to represent the Thevenin equivalent of connected systems, while the load was modeled as a balanced three-phase impedance load.

2.2 Fault Simulation and Parameter Variation

A programmable three-phase fault block was integrated into the MATLAB-Simulink model to simulate fault events at different positions along the transmission line.

The simulated fault categories included single line-to-ground faults (A-G, B-G, and C-G), line-to-line faults (A-B, B-C, and C-A), double line-to-ground faults (A-B-G, B-C-G, and C-A-G), and three-phase faults (A-B-C and A-B-C-G). These ten fault classes were selected to provide broad coverage of common transmission-line fault scenarios.

Fault location was varied over ten predefined positions along the 100 km line, measured from the sending end. To avoid ambiguity, the exact locations used in the experiments should be listed explicitly in Table 2. Fault resistance was varied over nine discrete levels from 0Ω to 400Ω . The lower resistance values represent solid or near-solid faults with large current magnitudes, whereas the higher values represent increasingly resistive fault conditions that are more difficult to localize because of reduced fault-current magnitude and stronger waveform distortion. The upper part of the resistance range was included to stress-test the model under challenging high-impedance conditions rather than to imply that all such values are equally common in practice.

The fault inception angle was varied over eight values between 0° and 90° to represent different fault initiation instants within the voltage cycle. For each parameter combination, the simulation duration was set to 0.2 s, with fault application at 0.1 s and fault clearing at 0.15 s. Three-phase current signals were recorded throughout the simulation using Scope and To Workspace blocks and then exported to MATLAB for further processing.

Using ten fault types, ten fault locations, nine resistance levels, and eight inception angles, the data generation process produced 7,200 simulated fault cases per noise condition. This systematic parameter variation ensured broad coverage of single-fault operating scenarios and provided a structured dataset for subsequent training, validation, and testing of the proposed regression model [3-4].

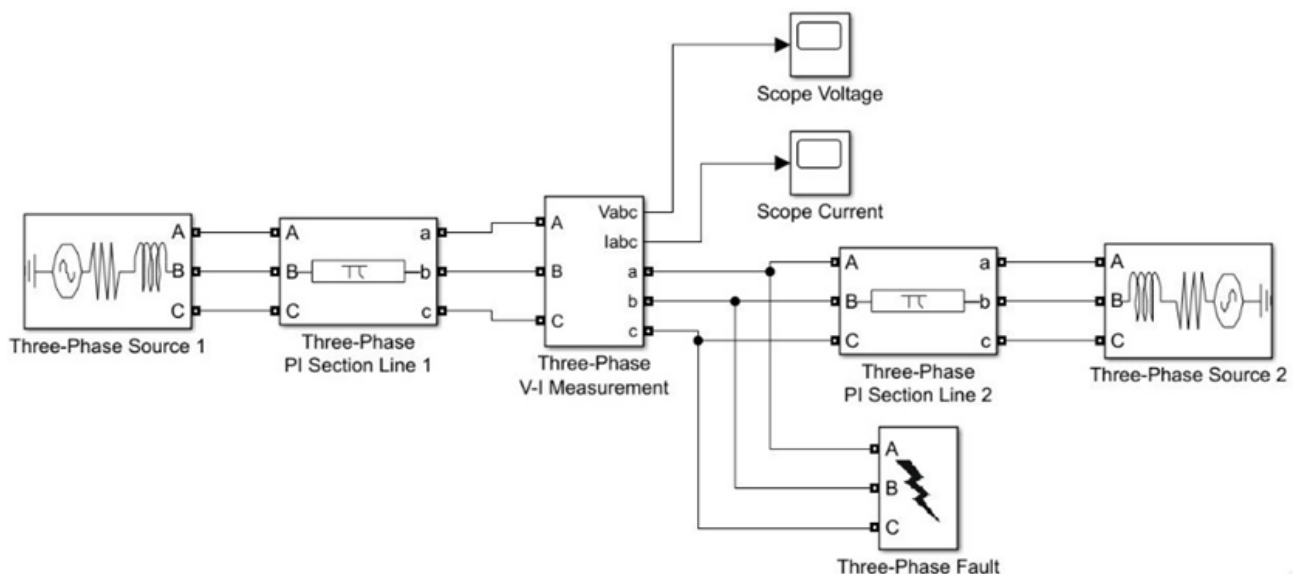


Fig. 2 Transmission Line System Modeling.

Table 2 Fault Simulation Parameters and Dataset Composition.

Parameters	Value
Transmission line length	100 km
System voltage	400 kV
System frequency	50 Hz
Fault types	A-G, B-G, C-G, A-B, B-C, C-A, A-B-G, B-C-G, C-A-G, A-B-C / A-B-C-G
Fault locations	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Fault resistance levels	0, 10, 30, 50, 100, 150, 200, 300, 400
Fault inception angles	0°, 10°, 20°, 30°, 45°, 60°, 80°, 90°
Simulation duration	0.2 s
Fault inception time	0.1 s
Fault clearing time	0.15 s
Sampling frequency	10 kHz
Cases per noise condition	7,200
Total cases across three conditions	21,600
Training / validation / testing split	80% / 10% / 10%
Samples per split (per condition)	5,760 / 720 / 720

2.3 Dataset Generation and Noise Injection

Fault signals are generated by systematically running the simulation for each fault type across a range of parameters. For each scenario, three-phase current signals are recorded using scope blocks and exported to the MATLAB workspace as sample in Fig. 3. To test the robustness of the LSTM model under different signal qualities, three datasets were prepared:-

- Dataset A (Noise-Free): Ideal fault current signals without noise contamination.
- Dataset B (30 dB): Fault current signals with 30 dB white Gaussian noise added to simulate mild interference conditions.
- Dataset C (20 dB): Fault current signals with 20 dB white Gaussian noise added to represent high-noise environments.

The robustness evaluation is conducted under these three noise conditions to provide a tiered stress-test that balances interpretability and experimental complexity. The ideal case establishes the upper-bound performance by isolating modeling error from measurement noise, while 30 dB and 20 dB represent moderate and more challenging noise environments, respectively, enabling systematic observation of performance degradation.

These SNR values are also widely adopted in prior transmission-line fault analysis studies, where 20 dB and 30 dB AWGN are commonly injected as benchmark noise levels for evaluating immunity and comparing learning-based approaches.

The noise injection was performed using MATLAB's `awgn` (`I`, `SNR`) function in MATLAB/Simulink (R2024b) to superimpose AWGN at the desired SNR. In this study, AWGN was used as a controlled baseline disturbance model to evaluate sensitivity to measurement contamination. However, it does not fully represent all practical grid disturbances, such as harmonic distortion, switching transients, impulsive noise, or sensor-related non-idealities, which should be considered in future robustness studies. [5, 6].

Each dataset was composed of time-series data representing three-phase currents (I_a , I_b , I_c) sampled at 10 kHz. To maintain uniform input lengths across samples, all signals were truncated or zero-padded to the same number of samples before normalization. The datasets were labeled according to their corresponding fault locations, which serve as continuous target values for regression.

The dataset was then divided into training (80%), validation (10%), and testing (10%) subsets. The training set was used to fit model parameters, the validation set was used for hyperparameter tuning and early stopping, and the testing set evaluated the model's final performance [7].

2.4 Data Preprocessing and Normalization

The three-phase current signals obtained from the Simulink simulations were preprocessed before being fed into the LSTM model. First, each signal was normalized using min-max normalization so that its amplitude lay within the range [0,1], as given by (1). This step reduced scale variation across samples and improved numerical stability during training.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Each simulated event was sampled at 10 kHz. The normalized three-phase current signals I_a, I_b, I_c were then segmented into fixed-length sequence windows representing 0.1 s of data. At a sampling rate of 10 kHz, each sequence therefore contained 1000 samples per phase. A 50% overlap was used between adjacent windows in order to preserve temporal continuity and reduce information loss at the window boundaries. Each input sample was arranged as a sequence matrix of size 1000×3 , where the three columns corresponded to phases A, B, and C, respectively.

To maintain consistent input dimensions across the dataset, sequences shorter than the required length were zero-padded, while longer sequences were truncated. No handcrafted features were extracted from the current signals. Instead, the LSTM network was allowed to learn discriminative temporal representations directly from the normalized raw sequences. This choice simplified implementation and preserved the transient information associated with fault initiation and propagation [8].

2.5 LSTM Architecture Design Using Deep Network Designer

The proposed fault-location model was implemented as a sequence-to-one regression network in MATLAB DND. DND provides a graphical, low-code environment for constructing, visualizing, and training DL models, which makes it suitable for rapid prototyping and engineering-oriented experimentation [9].

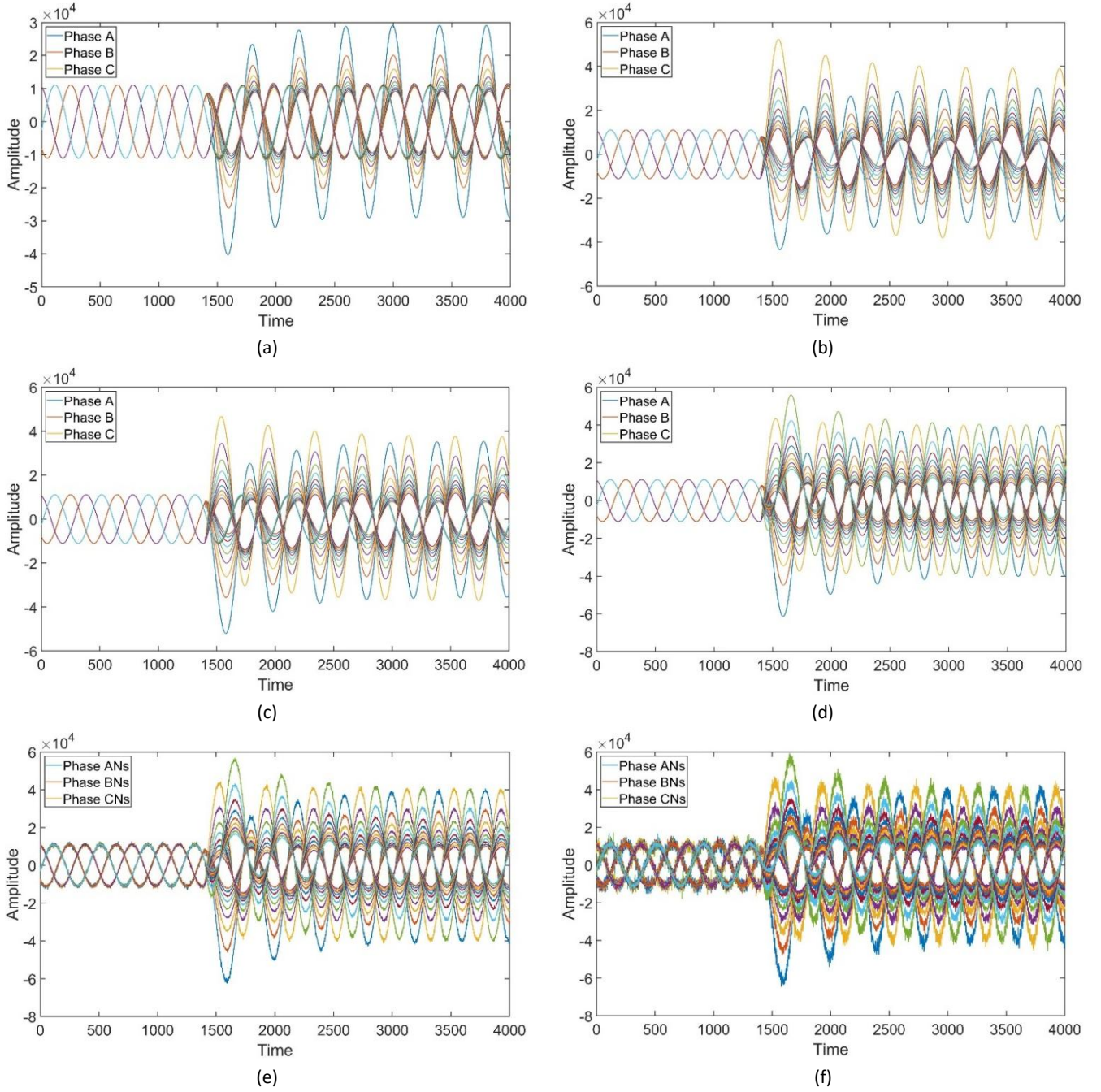


Fig. 3 Fault Signal for (a) SLGF, (b) DLF, (c) DLFG, (d) LLL/LLLG, (e) LLL/LLLG with 30 dB and (f) LLL/LLLG with 20 dB.

The final LSTM regressor consisted of a sequence input layer, a single LSTM layer with 128 hidden units, a fully connected layer with 64 neurons, a dropout layer with a dropout probability of 0.5, a final fully connected layer with one neuron, and a regression output layer. The sequence input layer received the three-phase current sequence for each simulated fault case, while the LSTM layer captured temporal dependencies in the signal evolution following fault inception. The fully connected layers performed nonlinear mapping from the learned temporal representation to the final scalar output, corresponding to the estimated fault location in kilometers.

The network was trained using the Adam optimizer with an initial learning rate of 0.001, a mini-batch size of 32, a maximum of 100 epochs, and a gradient threshold of 1.0. Early stopping was applied when the validation loss did not improve for 10 consecutive epochs. The regression loss was evaluated using Root Mean Square Error (RMSE), as defined in Equation (2). These

hyperparameters were selected empirically based on validation performance rather than through exhaustive hyperparameter optimization.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (2)$$

The chosen configuration achieved stable convergence with minimal overfitting, as observed from the training and validation loss curves. The use of dropout regularization further improved model generalization under unseen test conditions, including noisy signals [10, 11]. Hyperparameter selection was performed through empirical manual tuning guided by validation-set performance and training stability, rather than through exhaustive grid search or Bayesian optimization.

Table 3 LSTM architecture and training configuration.

Component	Setting
Input type	Sequence-to-one regression
Input channels	3 (Ia, Ib, Ic)
LSTM layers	1
Hidden units	128
Fully connected layers	1
FC layer sizes	64, 1
Dropout rate	0.5
Output layer	Regression
Optimizer	Adam
Initial learning rate	0.001
Mini-batch size	32
Maximum epochs	100
Gradient threshold	1
Early stopping	Validation patience = 10
Hyperparameter strategy	Empirical manual tuning

2.6 Model Training, Validation, and Testing

The LSTM model was trained using Dataset A, which contains the noise-free fault-current signals. Datasets B and C, containing 30 dB and 20 dB AWGN, respectively, were used to evaluate cross-noise generalization after training. This strategy was chosen to examine how well a model learned from clean simulated data could retain performance under progressively noisier test conditions.

It should be noted that the present training strategy evaluates generalization under AWGN only. The study does not yet include mixed-noise training, harmonic distortion, impulsive transients, or field-recorded disturbances, and therefore the robustness claims in this paper are limited to the controlled simulated conditions considered here. Nevertheless, the proposed framework can be extended to noisy-data training by augmenting the training set with multiple disturbance types, including AWGN, harmonic components, transient spikes, and random timing perturbations. Such mixed-noise training is expected to improve robustness by exposing the LSTM model to a broader range of waveform variations during optimization.

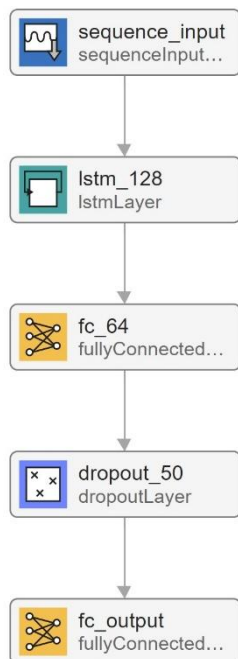


Fig. 4 Deep Network Designer Model.

The complete dataset for each condition was divided into training, validation, and testing subsets in an 80:10:10 ratio. The training set was used to update the network parameters, the validation set was used for monitoring convergence and early stopping, and the testing set was reserved for final performance evaluation. To improve repeatability, each experiment was repeated five times with different random initializations, and the average RMSE and R^2 values were recorded. The trained model was then saved for subsequent inference and comparative evaluation.

2.7 Comparative Model Development

For benchmarking, alternative models - 1D CNN, 2D CNN (spectrogram-based), and CNN-LSTM hybrids were also implemented. Each was trained under identical conditions using the same datasets and hyperparameters.

- The 1D CNN processed the time-domain signals using convolutional filters for local pattern extraction.
- The 2D CNN required conversion of time-domain signals into spectrogram images, increasing computational cost.
- The CNN-LSTM hybrid combined CNN layers for spatial feature extraction followed by LSTM layers for temporal sequence learning.

Although hybrid models achieved slightly higher accuracy in some cases, they required longer training times and more memory resources. In contrast, the LSTM-only model provided a balanced trade-off between accuracy, computational efficiency, and implementation simplicity [12, 14].

2.8 Validation Strategy and Evaluation Metrics

The model performance was assessed using four key metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), coefficient of determination (R^2), and accuracy within ± 1 km. These metrics were selected to evaluate prediction precision, average error magnitude, goodness-of-fit, and practical error tolerance.

- RMSE indicates the average magnitude of prediction error while giving greater weight to larger deviations.
- MAE measures the average absolute difference between predicted and actual fault locations.
- R^2 indicates how well the regression predictions explain the variance in the target values.
- Accuracy within ± 1 km represents the percentage of predictions that fall within a 1 km error band around the actual fault location, which is used here as a practically interpretable performance measure for the studied transmission-line case.

In this study, RMSE, MAE, and ± 1 km accuracy are considered the most operationally meaningful indicators, while R^2 is interpreted as a complementary goodness-of-fit metric rather than as a standalone measure of practical performance.

A high R^2 (>0.95) and low RMSE (<2 km) across datasets demonstrate that the model effectively captures both linear and nonlinear relationships between fault signal features and distance.

2.9 Implementation Advantages of Deep Network Designer

Using MATLAB's Deep Network Designer significantly simplifies the workflow for DL model creation and deployment. Unlike traditional coding approaches, DND allows users to design networks using a drag-and-drop interface, visualize intermediate layer activations, and tune hyperparameters interactively.

In addition, DND integrates seamlessly with MATLAB-Simulink, enabling direct import of simulation data and export of trained models into embedded systems or real-time hardware. This integration facilitates the practical application of the proposed fault location model in real-world protection devices such as intelligent electronic relays, substation monitoring systems, and SCADA platforms [13].

The following section presents detailed results, performance comparisons, and analysis under different noise conditions, providing insights into the model's strengths and limitations when applied to realistic fault scenarios.

3. Result and Discussion

The proposed Long Short-Term Memory (LSTM) framework for fault location estimation was implemented and tested using MATLAB's Deep Network Designer (DND) environment. The results were obtained by training and evaluating the model on three datasets corresponding to varying signal noise levels: noise-free, 30 dB, and 20 dB white Gaussian noise. The model performance was analyzed in terms of convergence behavior, prediction accuracy, robustness to noise, and comparison with alternative DL architectures.

3.1 Training Performance and Convergence Behavior

The LSTM model was trained for 100 epochs using the Adam optimizer, with training and validation losses monitored to ensure convergence and generalization. Fig. 5 illustrates the training and validation RMSE curves for the noise-free dataset. The model demonstrated a steady decrease in RMSE during the initial epochs, stabilizing after approximately 60 epochs. The validation RMSE closely followed the training RMSE curve, indicating that the model generalized well to unseen data without significant overfitting.

The final RMSE achieved on the validation dataset was approximately 0.85 km, while the training RMSE converged to

around 0.82 km. The early stopping criterion was triggered when no improvement was observed in validation performance for 10 consecutive epochs, preventing unnecessary training iterations. The total training time for the LSTM network was about 12 minutes on a standard workstation equipped with an Intel i7 processor and 16 GB RAM, demonstrating computational efficiency suitable for research and educational environments.

The smooth convergence curves confirm the effectiveness of the chosen hyperparameters (learning rate = 0.001, batch size = 32) and network architecture (128 LSTM units, 64 fully connected neurons). The use of a dropout layer (50%) further enhanced model generalization by mitigating overfitting. Similar convergence patterns were observed for noisy datasets (30 dB and 20 dB), albeit with slightly higher residual loss values due to the presence of noise-induced variations.

3.2 Fault Location Accuracy Across Noise Conditions

Table 4 summarizes the performance metrics of the LSTM model under different noise conditions and fault types, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), coefficient of determination (R^2), and the percentage of predictions within ± 1 km of the actual location. Under ideal noise-free conditions, the model achieved an average RMSE of 0.8575 km, a mean MAE of 0.615 km, and an R^2 value of 99.92%, with approximately 89.44% of predictions within ± 1 km accuracy. These results indicate a high correlation between predicted and actual fault locations.

The R^2 values remained high across all three test conditions, indicating that the proposed model captured the general relationship between the input current sequences and the target fault distances. However, because the regression target is bounded within a fixed 0–100 km interval, R^2 should be interpreted together with RMSE, MAE, and ± 1 km accuracy rather than in isolation. In this study, the kilometer-scale error metrics provide a more operationally meaningful interpretation of model performance than R^2 alone.

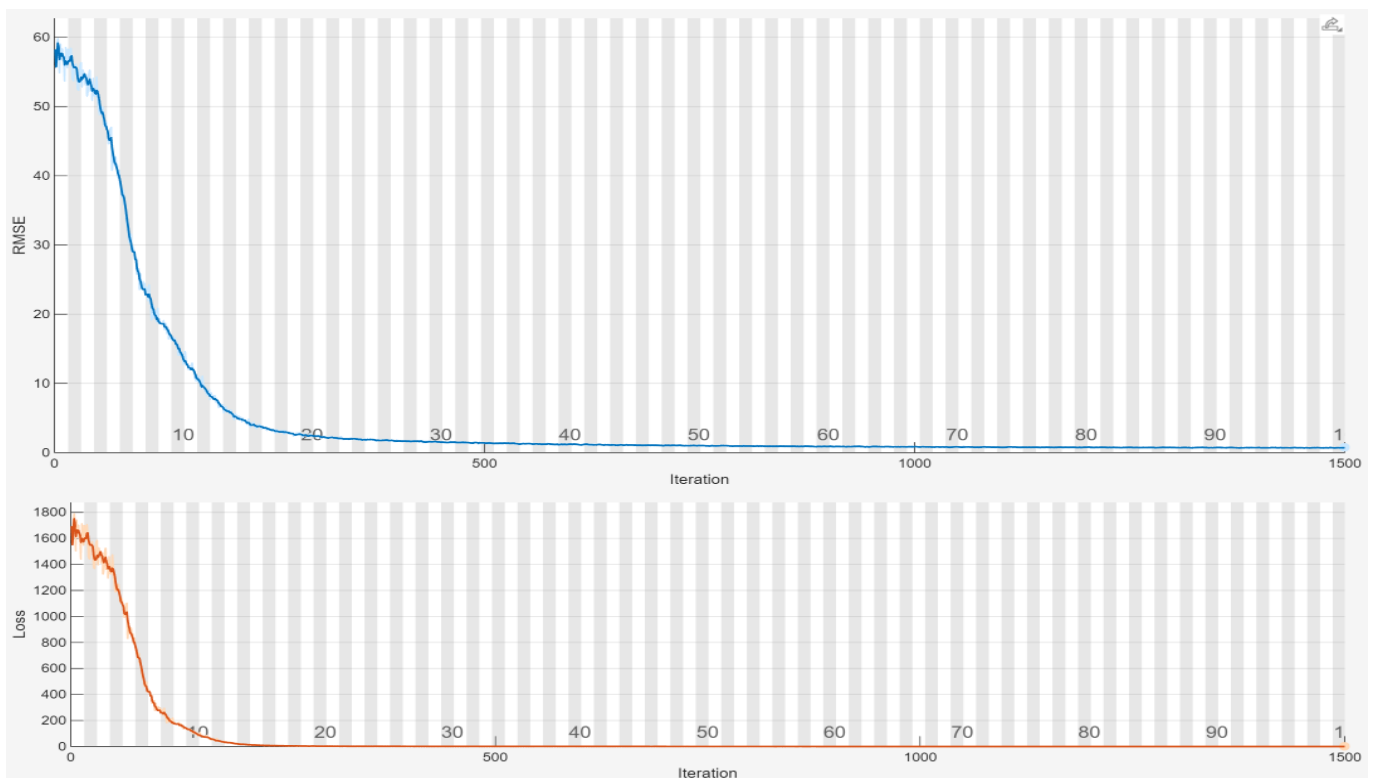


Fig. 5 Regression Training for Fault Location Estimation.

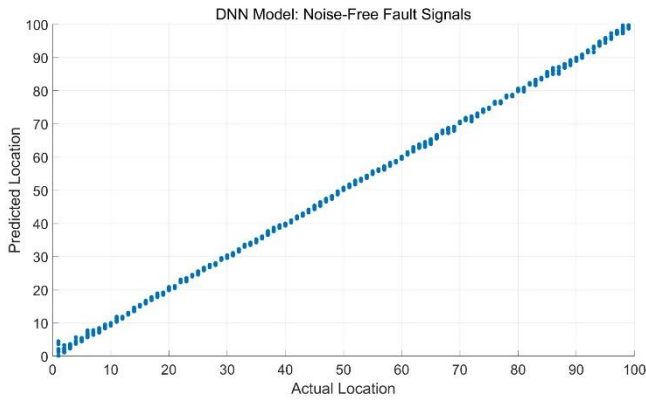


Fig. 6(a) Regression output for Dataset A (noise-free) showing predicted vs. actual fault distances.

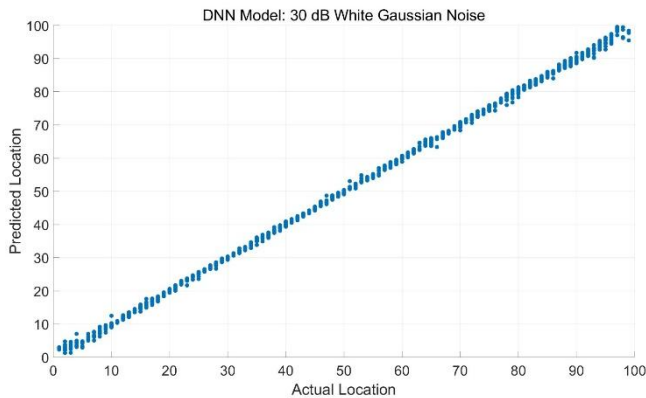


Fig. 6(b) Regression output for Dataset B (30 dB White Gaussian Noise) showing predicted vs. actual fault distances.

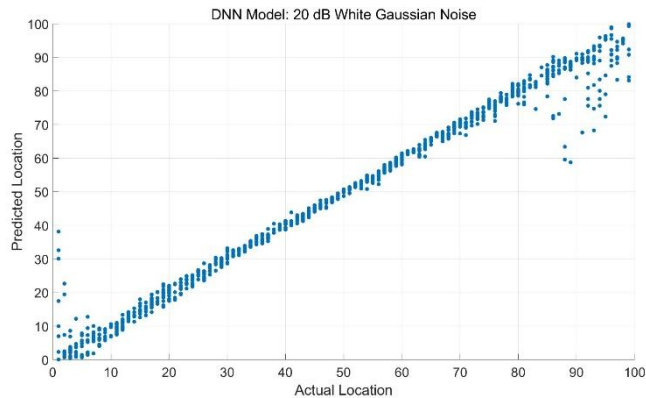


Fig. 6(c) Regression output for Dataset C (20 dB White Gaussian Noise) showing predicted vs. actual fault distances.

Under noisy conditions (30 dB and 20 dB datasets), the RMSE values increased slightly to 1.000 km and 1.133 km, respectively. Despite the added noise, the R^2 values remained above 99.86%, indicating the model's strong robustness and generalization capability. Fig. 6 presents regression scatter plots for the three datasets, where the close alignment of data points along the 45° diagonal line demonstrates the LSTM model's accuracy in predicting fault distances across various scenarios.

The analysis shows that single line-to-ground faults (A-G, B-G, C-G) yielded the lowest RMSE and highest accuracy, with errors typically below 1 km. Complex faults involving multiple phases (A-B-G, A-B-C-G) showed slightly higher RMSE values due to overlapping signal features and increased waveform distortion. However, the deviation remained minimal (<2 km), confirming the model's effectiveness even in challenging conditions.

The ± 1 km accuracy metric provides an intuitive indication of how often the predicted fault location falls within a narrow error band around the true distance. For the 100 km transmission line considered in this study, this measure helps interpret the practical usefulness of the regression results. However, it should not be interpreted as a universal industry benchmark, since acceptable fault-location tolerances may vary across utilities, line lengths, instrumentation quality, and operational practice.

3.3 Impact of Noise on Model Robustness

Evaluating performance under noisy measurements is important because practical transmission-line signals are often degraded by electromagnetic interference, switching activity, instrumentation effects, and communication disturbances. In this study, noise robustness was assessed using a controlled AWGN-based stress test involving noise-free, 30 dB, and 20 dB signal conditions.

The results show a gradual and limited degradation in performance as the signal-to-noise ratio decreases. The average RMSE increased from 0.8575 km in the noise-free case to 1.0000 km at 30 dB and 1.1330 km at 20 dB, while the average R^2 remained above 99.86% across all three datasets. These results indicate that the model preserved most of its predictive capability even when the current sequences were corrupted by substantial Gaussian noise. However, this behavior should be interpreted as robustness to AWGN within the simulated framework, rather than as complete robustness to all real-world measurement disturbances.

The limited performance degradation can be attributed to the temporal modeling capability of the LSTM layer, the normalization of the input signals, and the use of dropout regularization.

Table 4 Overall Fault Location Performance of Proposed Method.

Fault Type	RMSE			MAE	R^2 (%)	Approximate Accuracy within ± 1 km (%)
	Ideal	30 dB	20 dB	Average	Average	Average
A-G	0.6362	0.7914	0.8723	0.4333	99.96	94.34
B-G	0.6769	0.8037	0.8832	0.4423	99.96	96.06
C-G	0.6730	0.8129	0.9010	0.4456	99.96	95.76
A-B	0.8470	0.9762	1.1427	0.6553	99.91	87.98
B-C	0.8701	0.9923	1.1586	0.6603	99.94	85.45
C-A	0.8953	1.0235	1.1854	0.6754	99.94	86.87
A-B-G	0.9603	1.1081	1.2619	0.7028	99.89	86.51
B-C-G	0.9583	1.1045	1.2497	0.6952	99.89	89.60
C-A-G	0.9645	1.1174	1.2711	0.7010	99.93	85.86
A-B-C / A-B-C-G	1.0935	1.2437	1.4036	0.8392	99.86	85.99
Average	0.8575	1.0000	1.1330	0.6150	99.92	89.44

Nevertheless, the present study does not include non-Gaussian noise, harmonic distortion, impulsive noise, switching transients, or sensor saturation effects. Therefore, the current robustness results should be interpreted as AWGN-specific rather than fully representative of real transmission-line measurements. A practical next step would be to retrain the model using mixed-noise augmented datasets so that the LSTM is exposed during training to multiple disturbance types rather than to clean data only. This approach is expected to improve robustness and reduce performance degradation under real operating conditions.

3.4 Comparative Evaluation with Other Architectures

To benchmark the proposed LSTM model, additional DL architectures were implemented, including 1D CNN, 2D CNN using spectrogram inputs, and CNN-LSTM hybrids. All models were evaluated using the same datasets and the same general train/validation/test protocol. However, it should be noted that the comparison reflects a common experimental framework rather than exhaustive architecture-specific hyperparameter optimization for every model.

As summarized in Table 5, the CNN-LSTM hybrid achieved the lowest average RMSE under ideal conditions, but at the cost of increased architectural complexity and additional preprocessing or computational burden. In particular, the 2D CNN required conversion of time-domain signals into spectrograms, which introduced extra preprocessing steps and increased memory demand. By contrast, the proposed LSTM model operated directly on normalized raw current sequences and achieved an average RMSE of 0.8575 km and MAE of 0.6150 km, thereby offering competitive accuracy with a simpler workflow.

The LSTM model therefore represents a favorable trade-off between predictive accuracy, noise tolerance, and implementation simplicity, especially in the context of MATLAB Deep Network Designer. A fully fair comparison would ideally also include training time, parameter count, memory usage, and inference latency for all models. These measures should be added wherever they can be obtained consistently.

Table 5 Performance comparison between the proposed LSTM model and benchmark DL architectures.

Model	RMSE (Avg)	MAE (Avg)
CNN (1D)	1.3300	0.9585
CNN (2D)	1.2219	0.8727
CNN-LSTM (1D)	0.8431	0.5324
CNN-LSTM (2D)	1.0780	0.7275
LSTM (Proposed Method)	0.8575	0.6150

3.5 Error Analysis and Model Interpretability

To further examine model behavior, an error distribution analysis was carried out using the residual difference between actual and predicted fault locations. For the noise-free dataset, most residuals were concentrated within ± 1 km, indicating tight agreement with the true distance values. Under 30 dB and 20 dB noise conditions, the residual spread increased slightly, but the majority of predictions still remained within a practically useful error band. To improve the quantitative presentation of this analysis, the revised manuscript supplements the text with distribution-based visualization, such as a box plot or cumulative error plot, rather than relying only on qualitative description.

With respect to interpretability, the present study uses DND diagnostic visualization only as a qualitative aid. While intermediate activations suggest that the model responds strongly to transient variations after fault inception, this should not be interpreted as a full explainability analysis. Accordingly, the present manuscript treats interpretability as preliminary and identifies SHAP, relevance propagation, or other explainable AI techniques as important directions for future work rather than presenting interpretability as fully established.

3.6 Comparison with Previous Studies

Table 6 compares the proposed model's results with recent fault location studies employing different DL architectures.

Table 6 Proposed Method against Recent Studies on DL-Based Fault Location.

Reference Paper	Method	Notes
Moradzadeh et al. [11]	CNN, LSTM, C-LSTM	Fault type and location identification under varying fault impedance
Belagoune et al. [12]	LSTM classification and regression	Fault detection, classification, and fault-distance prediction in a multi-machine system
Fan et al. [14]	CNN-LSTM	Single-ended fault-location prediction from voltage/current sequences
Proposed Method	LSTM using DND	Direct regression using three-phase current sequences; RMSE = 0.8575 km under the noise-free condition

The comparison demonstrates that the proposed LSTM-based DND approach is well positioned relative to existing DL methods for transmission-line fault analysis. Moradzadeh et al. [11] employed CNN, LSTM, and convolutional-LSTM architectures to identify both fault type and location under varying fault impedance, while Belagoune et al. [12] demonstrated that LSTM can support fault detection, classification, and regression-based distance estimation in a large-scale multi-machine power system.

Similarly, Fan et al. [14] applied a CNN-LSTM architecture to single-ended fault-location estimation using voltage and current measurements, highlighting the effectiveness of hybrid spatial-temporal feature learning. In contrast, the proposed method employs a comparatively simpler LSTM regression architecture implemented directly in MATLAB DND and uses only three-phase current sequences as inputs, avoiding additional convolutional layers, image-based transformations, or manual feature extraction.

Despite this reduced architectural complexity, the proposed model achieved an RMSE of 0.8575 km under noise-free conditions, demonstrating that accurate fault-distance estimation can be obtained using a streamlined sequence-learning framework. Therefore, the main strength of the proposed approach is not merely numerical accuracy, but its balance between prediction performance, implementation simplicity, and reproducibility within a low-code environment. This makes it particularly suitable for rapid development and simulation-based fault-location studies, while the different system configurations and datasets used in the cited studies mean that their reported performance should be interpreted as methodological comparisons rather than as strictly equivalent numerical benchmarks.

3.7 Real-World Considerations and Challenges

While simulation-based validation provides controlled conditions for testing, applying the model to real transmission systems introduces additional challenges. Real-world fault data often exhibit greater variability and contain noise sources beyond Gaussian interference, such as harmonics, switching surges, and data acquisition errors [10].

Furthermore, data availability is limited since actual fault events are infrequent, and utility companies may impose restrictions on accessing operational records due to cybersecurity or privacy concerns [11]. Other practical challenges include:

- Data quality and labeling: Real fault records may be incomplete or contain inaccuracies in recorded fault distances.
- Instrumentation inconsistencies: Variations in current transformers (CT) and potential transformer (PT) characteristics may introduce bias.
- Synchronization errors: Phase angle differences between measurement points can impact accuracy in long-distance lines.

Addressing these challenges requires integrating real-world datasets with simulation-generated data using transfer learning or domain adaptation techniques to bridge the gap between synthetic and real environments. This hybrid data approach has been shown to enhance generalization and robustness in recent AI-based power system studies [12, 13].

The present model is validated only for single-fault events occurring at one location per simulation run. Simultaneous faults at multiple locations were not included in the generated dataset and therefore fall outside the validated scope of this work. Similarly, the model uses three-phase current signals only. Although this choice is practical for protection-oriented applications, future extensions may benefit from incorporating additional channels such as voltage measurements, PMU-based synchro phasors, or breaker-status information in order to improve performance under more complex operating conditions.

In practical deployment, the proposed LSTM model may be affected by non-ideal measurement conditions such as instrument gain errors, CT/PT saturation, synchronization mismatch, and missing or distorted samples. These effects can alter waveform amplitude, phase alignment, and transient shape, thereby reducing the consistency between the training distribution and real operating data. In the present study, such non-idealities were not explicitly simulated; therefore, the reported results should be interpreted as valid under controlled simulated conditions only. Nevertheless, the sequence-based nature of the LSTM model provides some tolerance to moderate waveform perturbations because it learns temporal patterns over an entire observation window rather than relying on a single instantaneous measurement. To improve robustness against these practical disturbances, future work should incorporate data augmentation and domain-adaptation strategies, including random gain scaling, timing jitter, saturation-like waveform clipping, and sensor-bias perturbations during training, together with validation using field-recorded fault data.

3.8 Practical Implications

From a practical perspective, accurate fault-distance estimation can support faster inspection and maintenance planning by narrowing the likely faulted region along the transmission line. For long

transmission corridors, improved localization can reduce search time and support more efficient restoration planning. However, the present findings are based on simulated single-fault conditions and therefore should be interpreted as evidence of practical potential rather than as full validation for field deployment.

Average Interruption Duration Index) and SAIFI (System Average Interruption Frequency Index). Furthermore, the low computational footprint of the LSTM model allows for integration into digital relays or local substation processors, enabling real-time fault diagnostics without reliance on high-end computing hardware [14, 15]. The framework's compatibility with MATLAB-Simulink also facilitates integration into existing simulation-based training systems for educational or operator training purposes. Engineers can retrain the model using new fault datasets, ensuring continuous improvement and adaptation to evolving grid conditions.

3.9 Computational Complexity and Real-Time Feasibility

The computational cost of the proposed model is moderate because the network consists of a single LSTM layer with 128 hidden units, one intermediate fully connected layer with 64 neurons, and a scalar regression output. This yields a relatively compact architecture compared with deeper hybrid CNN-LSTM models. On the workstation used in this study (Intel i7 CPU, 16 GB RAM), the complete training process required approximately 12 minutes for 100 epochs. Inference is substantially lighter than training because only forward propagation is required.

Since the model operates directly on normalized three-phase current sequences without spectrogram generation or handcrafted feature extraction, its runtime burden is lower than that of 2D CNN-based approaches. Although this study does not yet include hardware-level deployment benchmarking, the compact architecture and direct time-series input suggest that the method is suitable for near-real-time fault-location support in relay-adjacent or substation-level processing environments, subject to future validation on embedded hardware.

3.10 Summary of Findings

The analysis indicates that the proposed LSTM-based framework exhibits:

- Good fault-location accuracy under simulated conditions, with average RMSE remaining below 1.2 km across all tested datasets;
- Stable performance degradation under AWGN, indicating useful cross-noise generalization within the controlled signal conditions considered;
- Relatively low implementation complexity, since the model operates directly on normalized raw current sequences without manual feature extraction; and
- Practical usability for simulation-driven prototyping through MATLAB Deep Network Designer.

These findings support the potential of the LSTM-DND framework for simulated single-fault location estimation, while also highlighting the need for broader validation using mixed-noise and field-recorded data.

4. Conclusion

This study developed an LSTM-based fault location estimator for a simulated 100 km, 400 kV transmission line using MATLAB Deep Network Designer. The proposed framework used

three-phase current sequences generated in MATLAB-Simulink under ten fault types, multiple fault locations, multiple resistance levels, and varying fault inception angles. Using an 80:10:10 train/validation/test split, the model was trained on the noise-free dataset and evaluated under noise-free, 30 dB AWGN, and 20 dB AWGN conditions.

The results show that the proposed model achieved average RMSE values of 0.8575 km, 1.0000 km, and 1.1330 km for the three test conditions, respectively, while maintaining R^2 values above 99.86% and an overall ± 1 km accuracy of 89.44%. These findings indicate that the LSTM model can estimate the location of single transmission-line faults with good accuracy under the simulated scenarios considered in this work. Comparative evaluation further showed that the proposed LSTM approach offers competitive performance relative to CNN-based alternatives, while requiring less preprocessing and offering a simpler workflow in DND.

At the same time, the present study has several important limitations. The robustness analysis is restricted to AWGN-corrupted simulated data, the training strategy does not include mixed-noise learning, and multi-fault scenarios were not considered. The current validation does not yet include field-recorded fault data. In addition, the current study does not explicitly model instrument errors, CT/PT saturation, or synchronization mismatch, which remain important sources of domain shift between simulation and field deployment. Therefore, the present findings should be interpreted as evidence of strong performance under controlled simulated single-fault conditions rather than full proof of real-world deployment readiness. Future work will focus on mixed-noise training, multimodal inputs, explainable AI analysis, multi-fault scenarios, and validation using utility fault records. A particularly important extension is mixed-noise training, in which AWGN, harmonic distortion, transient spikes, and measurement perturbations are incorporated into the training set to improve robustness beyond the AWGN-only case.

Acknowledgements

The authors would like to extend their sincere appreciation to the Electrical Engineering Studies, Universiti Teknologi MARA, Pulau Pinang Branch, Permatang Pauh Campus, for their valuable support and guidance throughout the course of this research. The authors also gratefully acknowledge the administrative assistance provided by the university, which contributed significantly to the successful completion of this work.

This research was financially supported by the Conference Support Fund (CSF) under Universiti Teknologi MARA, Malaysia (Ref. 100-IPSiS 9/11/8 Code 250001040001).

References

- [1] Lopes, F. V., Mouco, A., Fernandes, R. O. and Neto, F. C., Real-World case studies on transmission line fault location feasibility by using M-Class phasor measurement units. *Electric Power Systems Research*. 196 (2021) 107261, doi: <https://doi.org/10.1016/j.epsr.2021.107261>.
- [2] Liu, Y., Lu, D., Vasilev, S., Wang, B., Lu, D. and Terzija, V., Model-based transmission line fault location methods: A review. *International Journal of Electrical Power & Energy Systems*. 153 (2023) 109321, doi: <https://doi.org/10.1016/j.ijepes.2023.109321>.
- [3] Yu, K., Zeng, J., Zeng, X., Liu, F., Zu, Y., Yu, Q. and Zhuo, C., A novel traveling wave fault location method for transmission network based on time linear dependence. *International Journal of Electrical Power & Energy Systems*. 126 (2021) 106608, doi: <https://doi.org/10.1016/j.ijepes.2020.106608>.
- [4] Ngoc Hung, T., Methods for Fault Location in High Voltage Power Transmission Lines: A Comparative Analysis. *International Journal of Renewable Energy Development*. 11 (2022) 1134-1141, doi: <https://doi.org/10.14710/ijred.2022.46501>.
- [5] Alencar, G. T. d., Santos, R. C. d. and Neves, A., A new robust approach for fault location in transmission lines using single channel independent component analysis. *Electric Power Systems Research*. 220 (2023) 109281, doi: <https://doi.org/10.1016/j.epsr.2023.109281>.
- [6] Shukla, P. K. and Deepa, K., Deep learning techniques for transmission line fault classification – A comparative study. *Ain Shams Engineering Journal*. 15 (2024) 102427, doi: <https://doi.org/10.1016/j.asej.2023.102427>.
- [7] de Alencar, G. T., dos Santos, R. C. and Neves, A., A fault recognition method for transmission systems based on independent component analysis and convolutional neural networks. *Electric Power Systems Research*. 229 (2024) 110105, doi: <https://doi.org/10.1016/j.epsr.2023.110105>.
- [8] Omar, A. M. S., Osman, M. K., Ibrahim, M. N., Hussain, Z. and Abidin, A. F., Fault classification on transmission line using LSTM network. *Indonesian Journal of Electrical Engineering and Computer Science*. 20 (2020) 231–238, doi: <https://doi.org/10.11591/ijeecs.v20.i1.pp231-238>.
- [9] Wang, X., Zhou, P., Peng, X., Wu, Z. and Yuan, H., Fault location of transmission line based on CNN-LSTM double-ended combined model. *Energy Reports*. 8 (2022) 781-791, doi: <https://doi.org/10.1016/j.egy.2022.02.275>.
- [10] Rafique, F., Fu, L. and Mai, R., LSTM autoencoders based unsupervised machine learning for transmission line protection. *Electric Power Systems Research*. 221 (2023) 109432, doi: <https://doi.org/10.1016/j.epsr.2023.109432>.
- [11] Moradzadeh, A., Teimourzadeh, H., Mohammadi-Ivatloo, B. and Pourhossein, K., Hybrid CNN-LSTM approaches for identification of type and locations of transmission line faults. *International Journal of Electrical Power & Energy Systems*. 135 (2022) 107563, doi: <https://doi.org/10.1016/j.ijepes.2021.107563>.
- [12] Belagoune, S., Bali, N., Bakdi, A., Baadji, B. and Atif, K., Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems. *Measurement*. 177 (2021) 109330, doi: <https://doi.org/10.1016/j.measurement.2021.109330>.
- [13] The MathWorks, Inc. *Design Deep Neural Networks Using Deep Network Designer*, <<https://www.mathworks.com/help/deeplearning/ref/deepnetworkdesigner-app.html>> (R2024b).
- [14] Fan, R., Yin, T., Huang, R., Lian, J. and Wang, S., Transmission Line Fault Location Using Deep Learning Techniques. in *2019 North American Power Symposium (NAPS)*. (2019) 1-5, doi: <https://doi.org/10.1109/NAPS46351.2019.9000224>.
- [15] Fahim, S. R., Sarker, S. K., Mueen, S. M., Das, S. K. and Kamwa, I., A deep learning based intelligent approach in detection and classification of transmission line faults. *International Journal of Electrical Power & Energy Systems*. 133 (2021) 107102, doi: <https://doi.org/10.1016/j.ijepes.2021.107102>.