

## Article

# Intelligent Fault Detection and Classification Schemes for Smart Grids Based on Deep Neural Networks

Ahmed Sami Alhanaf <sup>1,\*</sup> , Hasan Huseyin Balik <sup>2</sup>  and Murtaza Farsadi <sup>2</sup> <sup>1</sup> Department of Computer Engineering, Yildiz Technical University, Istanbul 34220, Turkey<sup>2</sup> Department of Computer Engineering, Faculty of Engineering, Istanbul Aydin University, Istanbul 34295, Turkey; hasanbalik@gmail.com (H.H.B.); murtazafarsadi@aydin.edu.tr (M.F.)

\* Correspondence: ahmed.alhanaf@std.yildiz.edu.tr

**Abstract:** Effective fault detection, classification, and localization are vital for smart grid self-healing and fault mitigation. Deep learning has the capability to autonomously extract fault characteristics and discern fault categories from the three-phase raw of voltage and current signals. With the rise of distributed generators, conventional relaying devices face challenges in managing dynamic fault currents. Various deep neural network algorithms have been proposed for fault detection, classification, and location. This study introduces innovative fault detection methods using Artificial Neural Networks (ANNs) and one-dimension Convolution Neural Networks (1D-CNNs). Leveraging sensor data such as voltage and current measurements, our approach outperforms contemporary methods in terms of accuracy and efficiency. Results in the IEEE 6-bus system showcase impressive accuracy rates: 99.99%, 99.98% for identifying faulty lines, 99.75%, 99.99% for fault classification, and 98.25%, 96.85% for fault location for ANN and 1D-CNN, respectively. Deep learning emerges as a promising tool for enhancing fault detection and classification within smart grids, offering significant performance improvements.

**Keywords:** smart grid (SG); fault classification and detection; deep neural networks; ANN; CNN



**Citation:** Alhanaf, A.S.; Balik, H.H.; Farsadi, M. Intelligent Fault Detection and Classification Schemes for Smart Grids Based on Deep Neural Networks. *Energies* **2023**, *16*, 7680. <https://doi.org/10.3390/en16227680>

Academic Editors: Nicu Bizon and Mihai Oproescu

Received: 7 October 2023

Revised: 2 November 2023

Accepted: 16 November 2023

Published: 20 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Smart Grids (SG) integrate intelligent communication facilities and interconnected devices to control and monitor energy demand, enhancing power supply stability and continuity [1]. Smart grids provide benefits such as demand management, network reliability, fault/outage detection, energy recovery, and grid stability through balancing production and consumption [2]. Distributed Generators (DGs) represented by renewable energy sources are a growing solution for meeting increasing energy demand. They offer numerous advantages, including enhanced power generation efficiency, decreased environmental pollution, minimized transmission line losses, and improved voltage profile [3]. Transmission line protection involves fault detection, classification, and location determination, with early detection being crucial for minimizing damage and ensuring uninterrupted power supply. Fault classification and detection in transmission lines identifies fault types, enabling efficient maintenance processes and minimizing downtime in addition to isolating the faulty zone [4,5].

Most of the prior research has concentrated on the development of algorithms that enhance the accuracy and efficiency of fault detection and classification in power systems. These studies have primarily utilized signal processing techniques as well as machine/deep learning approaches. Important features can be obtained from direct measurements, such as RMS values, voltages, or currents, as well as from transformed signals, such as wavelets, Fourier transform, and others. However, the direct measurements have certain limitations in terms of accuracy during the initial post-fault cycles and their susceptibility to minor variations (outliers) in power quality during normal operation. Conversely, transformed

signals suffer from a delay in the processing time. Despite the success of utilizing transformed signals, they suffer from significant time delay, which can be crucial in certain training operations, particularly real-time training operations. There are many methods to collect data from given networks, such as Power Measurement Unit (PMU) and Supervisory Control and Data Acquisition (SCADA). The primary objective of a SCADA system is to collect real-time data from remote locations and present operators with a graphical interface for monitoring and controlling processes. PMUs facilitate the detection of Frequency Disturbance Events (FDEs) by measuring synchronized phasors in real time, which is essential for fault classification and detection that rely on voltage/current variations and FDEs [6–9].

The measurement of synchro phasors by PMUs plays a significant role in dynamically monitoring transient processes within energy supply systems, making a valuable contribution in this regard. The integration of PMUs in power systems greatly enhances the opportunities for monitoring and analyzing the dynamics of the power system [10–12]. Compared to traditional SCADA measurements, PMUs have a higher sampling rate, allowing for signals to be collected at up to 60 samples per second and FDEs to be captured with higher precision and speed seconds [13]. This is attributed to the higher sampling rate of PMUs, which typically ranges from 30 to 120 samples per second, surpassing the sampling rate of traditional SCADA measurements taken at intervals of 2 to 4 s [10–12,14]. Despite the fact that there are many types of faults, such as unsymmetrical faults (L-G,LL-G,LL) and symmetrical faults (LLL-LLLG) [15], deep learning algorithms can detect fault type, identify fault location and determine faulty parts, learn incipient failure and their causes, and predict the pattern of faults. Fault detection, classification, and location techniques can be either data-driven or model-driven, as noted in [16,17].

Data-driven techniques use data mining and machine learning algorithms to analyze large amounts of data to detect and classify faults. Model-driven techniques use mathematical models of the power system to detect and locate faults. The selection of the appropriate technique depends on the specific application and the available data, as each technique comes with its own set of advantages and disadvantages. The smart grid utilizes advanced communication and information technologies to improve reliability, flexibility, and efficiency, but it is susceptible to faults that can lead to power outages. Fault detection, classification, and localization are critical tasks in smart grid maintenance, allowing for quick fault identification and power restoration.

Data-driven approaches, such as machine learning-based techniques, are useful for detecting faults in highly nonlinear systems, as they are not dependent on system structure [7,18]. With the rapid advancements in deep learning and parallel computing hardware, data-driven methods have emerged as highly promising solutions for real-time fault diagnosis. Furthermore, data-driven algorithms demonstrate strong noise resistance capabilities, making them particularly well-suited for tackling complex classification problems. Fault detection and classification tasks have demonstrated the potential of Deep Neural Networks (DNNs), including Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs) [19–21].

In recent years, there has been a notable increase in the complexity of Artificial Neural Network (ANN) structures, with various architectures being designed to cater to different application scenarios. The fault detection capabilities of ANN-based methods are advantageous, as they do not rely on a pre-existing knowledge base. This allows them to quickly and precisely detect, locate, and classify faults within the power system [21,22]. CNNs have been used for fault classification, and they can be combined with other techniques, such as Discrete Wavelet Transform (DWT), to develop fault classification approaches [19].

CNNs are commonly used for classification and computer vision tasks [23]. Deep learning algorithms can be trained on labeled fault data to automatically detect, classify, and locate faults in the smart grid in real time [24–26]. ANNs are suitable for handling various data types, while CNNs are ideal for image-based data [24,27]. These algorithms have gained significant usage in fault detection and diagnosis tasks across different systems, such as power distribution grids, transmission lines, and photovoltaic modules. The main

goal of these approaches is to create an artificial neural network system that can rapidly identify and classify different types of faults in transmission lines as soon as they occur. ANNs and CNNs can diagnose faults in the smart grid in real time, outputting the fault diagnosis, including the fault type and location, enabling operators to take the necessary actions. The use of these models has the potential to enhance the reliability and efficiency of power systems, minimizing downtime and improving customer satisfaction. A study introduced a deep neural network (ANN and CNN)-based approach for detecting and isolating faults in microgrids without the need to shut down the entire system. These algorithms have the potential to improve the reliability and efficiency of smart grids [28–30]. The algorithms used current and voltage measurements, which were pre-processed to detect characteristic changes in current and voltage signals caused by faults in the network.

The proposed model's Deep Neural Network (DNN) algorithm [31] has the capability to detect faults and determine fault location in medium or low voltage in transmission systems as well as distribute systems. The aim of the proposed data-driven model is to precisely identify the Faulty Line Identifier (FLI), Fault Class Types (FCT), and Fault Locations Estimator (FLE) in smart grids using Deep Neural Networks (DNNs) including but not limited to ANNs and CNN. The proposed data-driven model has the capability to detect various types of faults, such as single-line to ground, double-line, double-line to ground, and three-phase faults. Furthermore, the proposed scheme has the capability to identify fault class, faulty line, and the fault location in the grid simultaneously in the whole system. The proposed method in this study simplifies the process by eliminating the pre-processing steps as in [32,33], feature engineering, and conversion of signals, resulting in a more efficient approach. This avoids the unnecessary steps of converting voltage and current signals into grayscale images or other transformations to extract the meaningful features from signals, which was required in many previous studies.

This paper makes significant contributions in the following aspects:

- Introducing innovative fault detection methods using Artificial Neural Networks (ANNs) and one-dimensional Convolutional Neural Networks (1D-CNNs) for smart grids.
- Isolating faults in microgrids without the need to shut down the entire system.
- Simplifying the fault detection process by eliminating pre-processing steps, feature engineering, and signal conversion, resulting in a more efficient approach.
- Improving the reliability and efficiency of smart grids by accurately identifying the Faulty Line Identifier (FLI), Fault Class Types (FCT), and Fault Locations Estimator (FLE) using DNNs.
- Achieving optimal accuracy rates in fault detection, classification, and location for both ANN and 1D-CNN models in the IEEE 6-bus system.
- Offering significant performance improvements in fault detection and classification within smart grids by reducing testing time as much as possible, which is considered a crucial factor in real-time scenarios when using deep learning techniques.

The upcoming sections of this paper will be structured as follows: Section 2 provides an overview of the system and details the data generation process. In Section 3, the implementation and training of the proposed models are described, along with a discussion on the performance of the multi-class fault classification. Section 4 further examines the proposed model. Lastly, Section 5 presents a summary of the findings, concluding the paper.

## 2. System Simulation and Data Collection

The IEEE N-bus system is a standard power system used in electrical engineering to evaluate various power system analysis and control algorithms. To generate a data-driven model of this system in MATLAB, a mathematical model is built based on collected data from the system. Figure 1 shows the single-line configuration of the IEEE 6-bus system, which comprises three traditional voltage sources with a voltage rating of 132 kV and a frequency of 60 Hz. Data were gathered across a spectrum of scenarios, encompassing different expected load profiles. Additionally, data were collected in situations where

distributed sources, such as photovoltaic systems or variable wind turbines, were connected. This inclusion aimed to enhance the robustness and reliability of the study's findings.

The IEEE 6-bus test case represents a simple approximation of the distribution power system. It has six buses, four generators, three loads, and seven transmission lines. The transmission lines are modeled as medium lines with three-phase pi section lines. Additionally, the system includes three loads that consume both active and reactive power at bus 2, bus 5, and bus 6. To simulate any type of fault, we used the three-phase fault block, and, by selecting any phase (A, B, C) and ground G, it was possible to set the fault type, while the current and voltage were collected from both sides of the transmission line during fault. The line model divided it into two sections. The purpose of dividing the line into two parts is to change the values of resistance and inductance in a certain way, through which we can change the location of the fault during the simulation process.

Deep learning algorithms are used to detect patterns and relationships in the simulated data, and the learned model can predict the system's behavior under different conditions. In this study, MATLAB/SIMULATION is used to simulate the system with fault blocks generating data. Table 1 provides details of the data generators, line data, and load data used in the simulation. The system consists of six pi-section lines with a length of 100 km. The utility grid in the system has a rating of 1000 MVA, operates at 132 kV voltage, and has a base frequency of 60 Hz.

Building a data-driven model of the IEEE 6-bus system using MATLAB can help engineers and researchers better understand the system's behavior and develop more effective control and optimization strategies [34]. Time series data voltage and current signals for simulations were generated using MATLAB/SIMULINK 2022b. All experiments were conducted on a system equipped with an Intel CORE i7-10510U CPU, operating at 1.80 GHz (base clock) and 2.30 GHz (turbo boost), along with 16 GB of RAM. The training progress was implemented using the Python scikit-learn library, Jupyter (notebook), and Google Colab environments.

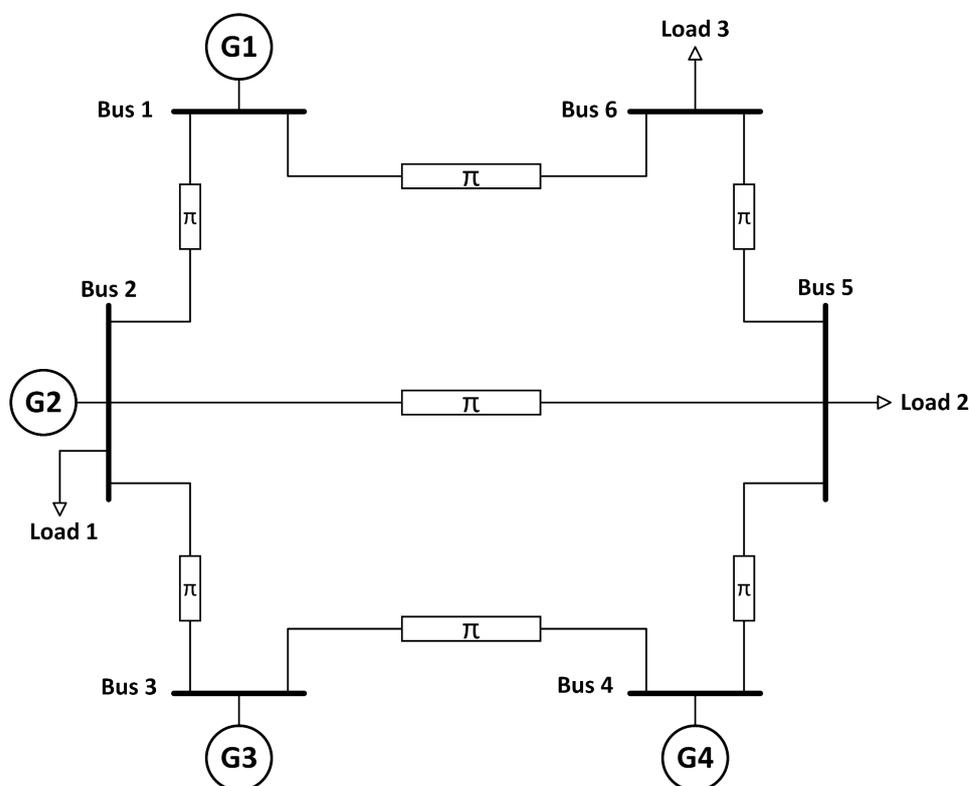


Figure 1. Single-line diagram of IEEE 6-bus.

**Table 1.** IEEE 6-bus system, line data, generator information, and load data.

Line Data				Generator Data				LoadData				
Line No.	Bus Code	R (p.u)	X (p.u)	Generator No.	P+jQ (p.u)	Voltage (p.u)	MVA	Bus Type	Bus No.	P (MW)	Q (MVAR)	
1	1-2	0.05	0.20	G1	0+j0	1.00	100	slack	At Bus 2	20	10	
2	2-3	0.10	0.50	G2	0.1+j0.05	1.041	15	PV	At Bus 5	40	15	
3	3-4	0.20	0.80	G3	0.30+j0.2	1.0190	40	PV	At Bus 6	30	10	
4	4-5	0.10	0.30	G4	0.2+j0.1	1.071	30					
5	5-6	0.20	0.40									
6	6-1	0.10	0.15									
7	2-5	0.20	0.50									

The simulation system proposed in this study captures voltage and current signals in three phases, which are sampled at a frequency of 2.5 kHz [34], resulting in 41 samples during a single cycle ( $2500 \div 60$ ). To obtain sufficient data for training and evaluating the proposed model's performance, various fault and non-fault scenarios are simulated by adjusting the system's parameters and settings. Table 2 presents detailed configurations for both fault and non-fault scenarios. These simulations are conducted to ensure the model's versatility and ability to detect and classify different types of faults that may occur in the smart grid system.

**Table 2.** Configuration for fault possibility cases during simulation.

Parameters	Possible Configuration	No. of Cases
Fault class	A-G, B-G, C-G, AB, BC, AC, AB-G, BC-G, AC-G, ABC, and Normal	11
Faulty Line	Line Line1, Line2, Line3, Line4, Line5, Line6, and Line7	7
Fault Location	10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90% of line length	10
Fault Resistanc (Rf)	0.1, 10, and 50	3

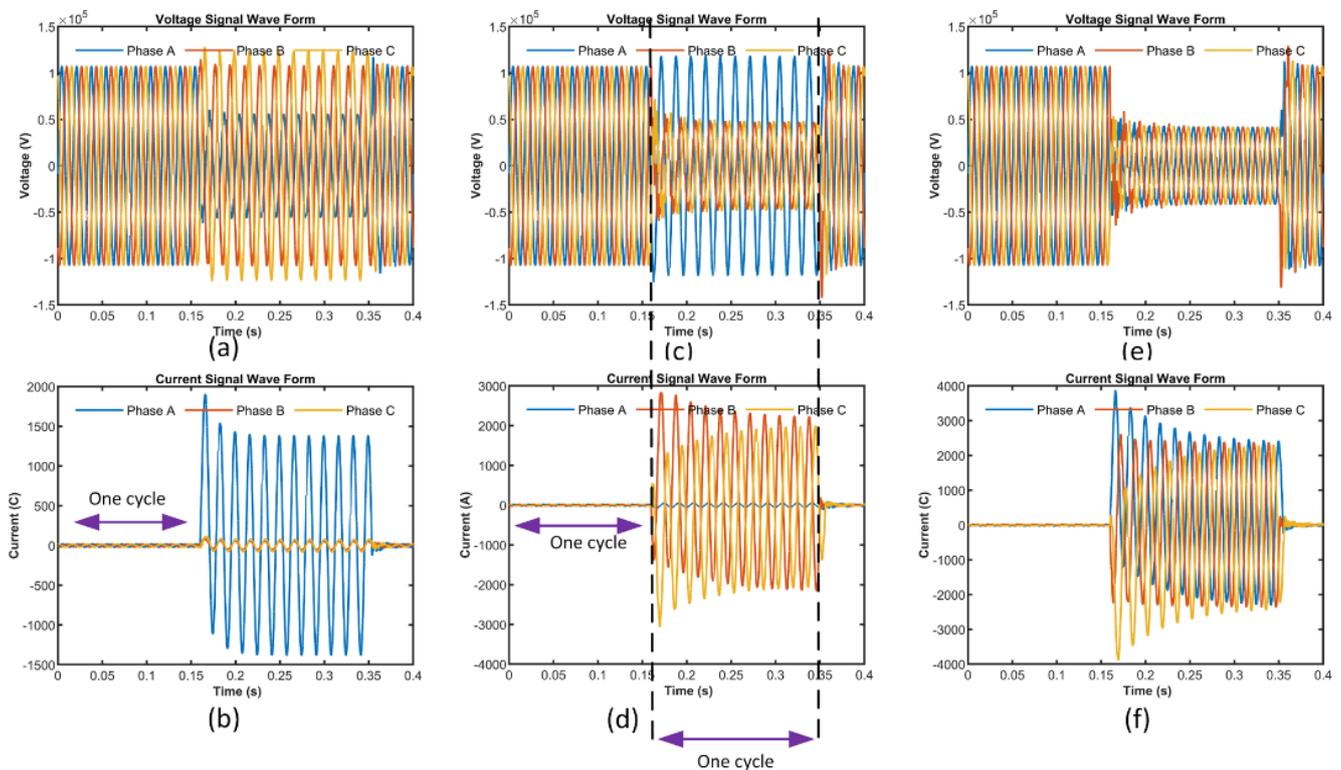
Table 3 provides an overview of the cases and labels for each class. The fault class comprises 8610 cases, which were generated by adjusting system parameters and configurations. On the other hand, the non-fault class consists of 2310 cases. To ensure accuracy, overlapping values have been eliminated. The table displays the total number of sampled data for three categories: 10 Fault Class (FCT), 7 Faulty Line (FLI), and 10 Fault Location (FLE). Various fault types were simulated by modifying fault resistances in different lines and locations within the system.

The data collection process accounted for different fault resistance values, including low and high settings. Fault resistance, along with inception angle, is a critical parameter for detecting faults in electrical systems, as it undergoes changes when a fault occurs. These changes lead to voltage and current variations in the affected section. Consequently, incorporating different fault resistance values in the data collection process is vital for enhancing the accuracy of fault detection models.

**Table 3.** Total no. of samples for FCT, FLI, and FLE.

Fault Class Type (FCT)	No. of Distribution	Faulty Line Identifier (FLI)	No. of Distribution	Fault Location Estimator (FLE)	No. of Distribution
Normal	15,120	Normal	15,120	Normal	15,120
AG	10,080	Line1–2	14,400	10% Line Length	11,200
BG	10,080	Line2–3	14,400	20% Line Length	11,200
CG	10,080	Line3–4	14,400	30% Line Length	11,200
ABG	10,080	Line4–5	14,400	40% Line Length	11,200
ACG	10,080	Line5–6	14,400	50% Line Length	11,200
BCG	10,080	Line6–1	14,400	60% Line Length	11,200
ABC	10,080	Line2–5	14,400	70% Line Length	11,200
AB	10,080	-	-	80% Line Length	11,200
AC	10,080	-	-	90% Line Length	11,200
BC	10,080	-	-	-	-

Figure 2 depicts the waveform of three-phase voltage and current signals in different simulated fault cases, each with varying fault resistances. The occurrence of the fault is at  $t = (1/60)$  s, and the voltage and current waveforms undergo noticeable modifications for different resistance values after the fault. When a fault occurs, the system impedance decreases, which leads to a rapid surge in fault current. The rise in current levels has the potential to inflict damage on electrical components and can also affect the overall stability of the system. Furthermore, fault voltage plays a critical role in determining the type and location of the fault. During a fault, the voltage waveform experiences significant changes, and the magnitude of these alterations depends on the fault type and its location. By analyzing the changes in the voltage waveform, it becomes possible to determine both the type and location of the fault.



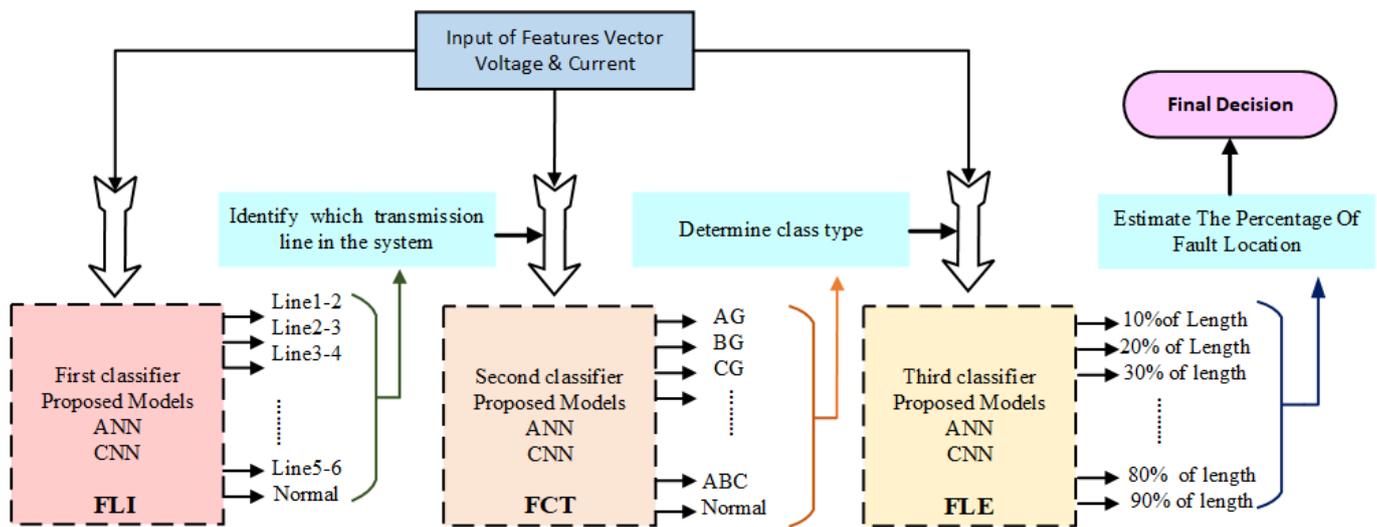
**Figure 2.** Waveforms of three-phase voltages and currents on different fault types as follows: (a) three-phase voltages of Phase A to ground fault, (b) three-phase currents of Phase A to ground fault (AG), (c) three-phase voltages of Phase B to Phase C to ground fault (BCG), (d) three-phase currents of Phase B to Phase C to ground fault (BCG), (e) three-phase voltages of Phase A to Phase B to Phase C to ground fault (ABCG), (f) three-phase currents of Phase A to Phase B to Phase C to ground fault (ABCG).

### 3. The Proposed Methodology

As mentioned in the previous section, we provided an overview of the generated dataset, followed by an explanation of the data collecting approach. Lastly, we present the theoretical principles and architecture of the proposed deep learning model. The training and validation datasets consist of input feature spaces, each comprising extended sequences of high-dimensional vectors derived from the real values of three-phase voltage and current signals measured at various locations.

In this study, we introduce three novel models. The first model serves as a classifier tasked with detecting faults and determining the specific faulty section. Subsequently, a second classifier is created for each identified section, facilitating the classification of faults based on their types, thus enabling fault diagnosis within that particular region. Finally,

a third classifier model is utilized to estimate the distance at which the fault occurred within the identified section, as illustrated in Figure 3.



**Figure 3.** Flow chart of transmission line fault classification method based on ANN and CNN in general.

### 3.1. Artificial Neural Network (ANN)

The aim of this research is to develop an ANN model capable of classifying all fault types, including no-fault scenarios, in a distribution network that may be penetrated with Distributed Generators (DGs). Figure 4 outlines the proposed methodology through a schematic block diagram. The dataset used for this study consists of raw samples of three-phase voltage and current in fault and no-fault scenarios, each labeled accordingly. Deep Neural Networks (DNNs) have become a robust model for non-linear statistical modeling, particularly in the field of fault detection and classification.

ANNs comprise various architectures, including Feed-Forward Neural Networks (FNNs) and Recurrent Neural Networks (RNNs). Among them, Long Short-Term Memory (LSTM) networks are particularly suitable for time sequence classification since they can address problems associated with exploding and vanishing gradients [35]. The proposed model has undergone two training methods: Normal Classification (NC) and Cross-Fold Validation (CV). NC involves dividing the dataset into two groups: a training set (80%) and a testing set (20%), whereas CV involves partitioning the dataset into 10 folds, with each fold having a distinct training and testing set.

The generated dataset is partitioned into a training set and a testing set. The training set is used to train the proposed model, while the testing set is used to evaluate the model performance after the training process to show the effectiveness of the models. Due to the varying dataset distribution in each fold, cross-validation guarantees that all data are utilized for both training and testing during the training process. In the training phase, in this study, the proposed ANN model is trained in offline phase using a training set with their corresponding actual labels.

The proposed methodology consists of two phases, namely, the training phase and the testing phase, which are illustrated in Figure 5. During the training phase, the model extracts features from the raw data and generates trained weights or trainable parameters. During the testing phase, the model operates in online mode, where it takes raw data as input without undergoing any feature extraction. Leveraging the trained parameters, the model then predicts the label of unseen or new data as output, eliminating the need for any additional preprocessing steps.

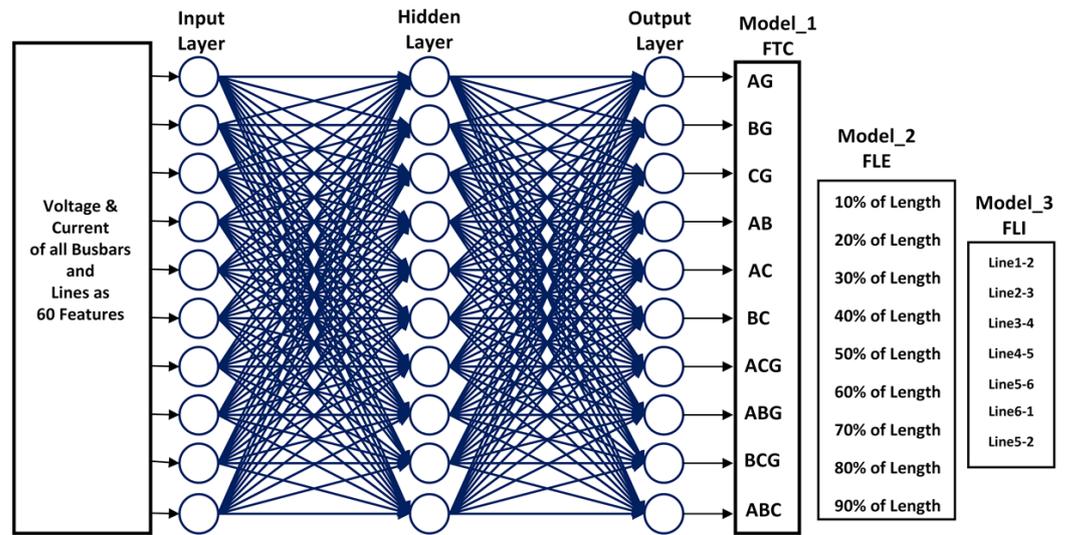


Figure 4. Proposed ANN architecture for fault classification, identification, and location.

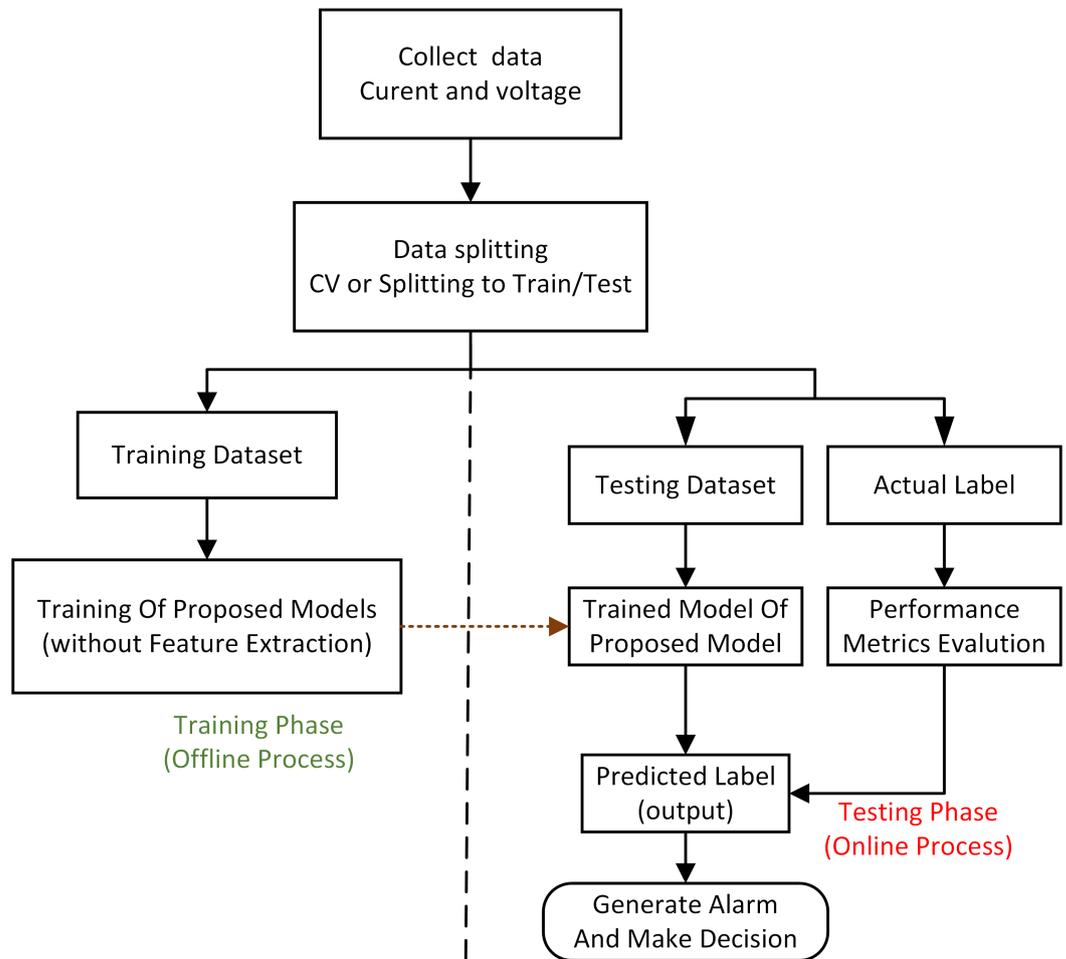


Figure 5. Flow chart of transmission line data training and testing based on proposed methods.

To evaluate the performance of the proposed model, the predicted and actual labels of the test data are compared. The architecture of the proposed ANN model comprises five layers (L0–L4), including an input layer and an output layer. For the FCT classifier, the model has a total of 187,059 trainable parameters. For the FLI, it has 176,520 trainable parameters, and for the FLE, it has 177,546 trainable parameters, as depicted in Table 4.

**Table 4.** Structures of the ANN models.

Layer No.	Layer Detail	Output Shape	Training Parameters
L0	Input data	(60 × 1)	-
L1	dense (Dense)	(None, 128)	7808
L2	dense_1 (Dense)	(None, 256)	33,024
L3	dense_2 (Dense)	(None, 512)	131,584
L4	dense_3 (Dense)	FTC (None, 11)	5643
		FLI (None, 08)	4104
		FLE (None, 10)	5130

Various performance metrics, such as Accuracy, Recall, Sensitivity, Specificity, Precision, and F1-Score, based on True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) as shown in Equations (1)–(5) are utilized to evaluate the proposed model (FCT, FLI, and FLE). In the context of binary and multiclass classification, the abbreviations have the following meanings:

- TP (True Positives) represents the number of positive elements correctly predicted as positive.
- FP (False Positives) denotes the number of negative elements incorrectly predicted as positive.
- FN (False Negatives) represents the number of positive elements incorrectly predicted as negative.
- TN (True Negatives) denotes the number of negative elements correctly predicted as negative.

The performance metric formulas are as follows:

$$\text{Total Accuracy} = \frac{\text{sum of TP}}{\text{sum of confusion matrix}} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \text{sensitivity} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (4)$$

$$f\text{-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

The article discusses the suitability of different evaluation metrics for different types of datasets. Accuracy is recommended for symmetrical datasets, while F1-Score is better for uneven class distributions. Precision measures the confidence level in true positives, and Recall measures the coverage of all positives. Specificity can be used to avoid false positives in critical cases.

The performance of the proposed model was assessed through 10-fold cross-validation, and the outcomes are presented in Tables 5 and 6. Additionally, Figure 6 displays the training curve for fold number 5.

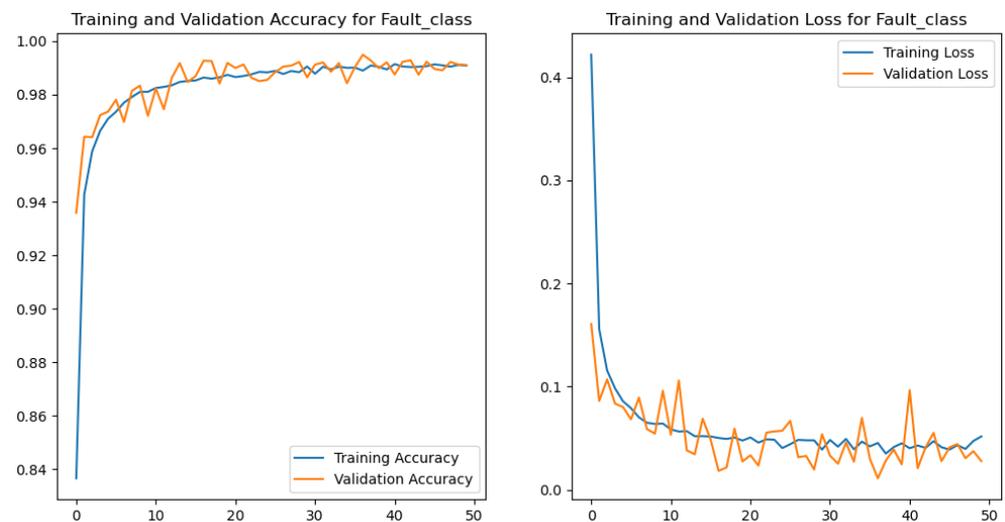
To assess the performance of the ANN model for fault detection, classification, and fault location, various configurations were employed. These configurations included different settings of parameters and hyper-parameters for the model's layers. By exploring diverse combinations, the aim was to optimize the model's effectiveness and achieve the best possible results for the specific tasks of fault detection, classification, and localization. Parameters are important for making predictions, while hyper-parameters are crucial for optimizing the model.

**Table 5.** The results of the 10-fold cross-validation for fault classification FCT using the ANN.

Fold No.	Accuracy	Precision	Recall	F1-Score
Fold-1	0.99344	0.960	0.990	0.975
Fold-2	0.99438	0.990	0.990	0.990
Fold-3	0.99570	1.000	1.000	1.000
Fold-4	0.99537	1.000	1.000	1.000
Fold-5	0.99089	0.900	1.000	0.947
Fold-6	0.98968	0.980	0.990	0.985
Fold-7	0.99266	1.000	1.000	1.000
Fold-8	0.99444	0.990	0.990	0.990
Fold-9	0.99299	0.990	0.990	0.990
Fold-10	0.99692	0.980	0.990	0.985
Average	0.99365	0.974	0.994	0.986

**Table 6.** The performance result of fold no. 5 for the proposed ANN.

Class	Precision	Recall	F1-Score	Accuracy	FP	FN	Support
Normal	1.000	1.000	1.000	0.9950	0	7	1495
AG	0.990	1.000	0.995	1.000	3	6	996
BG	0.970	1.000	0.985	0.999	8	8	1072
CG	1.000	0.990	0.995	1.000	5	8	1066
AB	0.990	0.990	0.990	0.9960	18	7	941
AC	0.9900	1.000	0.995	0.999	13	11	1015
BC	1.000	0.990	0.995	0.994	3	25	984
ABG	0.990	0.960	0.975	0.980	27	8	1040
ACG	0.970	0.990	0.980	0.975	2	27	990
BCG	0.990	1.000	0.995	0.989	0	42	982
ABC	1.000	0.990	0.995	0.978	42	24	1011
Average	0.990	0.9918	0.991	0.9914			11,592

**Figure 6.** Training and loss curve of fold no. 5 for fault classification type using ANN model.

The hyper-parameter settings of the proposed models are presented in Table 7. In this work, we have used a well-known optimizer ‘Adam’ and categorical-cross entropy as a loss function. Table 8 compares the performances of different ANN models, and the results show that increasing the number of convolutional layers improves fault classification performance but leads to an increase in trainable parameters. The proposed model achieves superior performance with fewer trainable parameters, achieving a balance between performance and complexity.

**Table 7.** Hyper-parameter configuration of proposed method.

Hyperparameter	ANN			CNN		
	FCT	FLI	FLE	FCT	FLI	FLE
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Epoch number	50	50	100	100	50	100
Learning rate	0.001	0.001	0.0001	0.001	0.001	0.0001
Batch size	128	128	128	128	128	128
Loss function	categorical	categorical	categorical	categorical	categorical	categorical
Activation function	RELU	RELU	tanh	RELU	RELU	tanh
Output layer	Softmax	Softmax	Softmax	Softmax	Softmax	Softmax

**Table 8.** Performance results of different ANN models for FCT, FLI, and FLE in the studied system.

ANN Model	Trainable Parameters	Normal Classification (NC)			Cross-Validation (CV) 10-Fold		
		FCT Accuracy %	FLI Accuracy %	FLE Accuracy %	FCT Accuracy %	FLI Accuracy %	FLE Accuracy %
1 Denes+ FC	9227	99.36	99.45	97.87	97.85	99	94.58
2 Denes+ FC	43,659	99.6	99.78	98	98.45	99.15	95.32
3 Denes+ FC	178,059	99.75	99.99	98.25	98.99	99.55	96.75

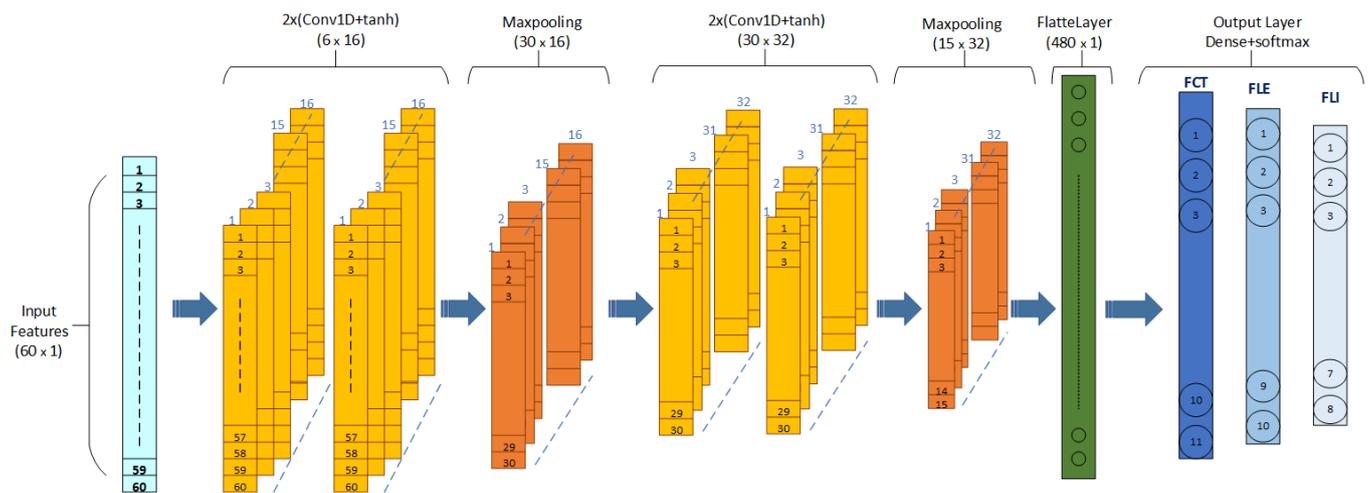
### 3.2. Convolution Neural Network (CNN)

The study proposes a deep learning algorithm using one dimensional CNNs for Fault Type Classification (FTC), Line Faulty Detection (LFI), and Fault Location Estimation (FLE) based on raw data samples. CNNs have been designed to provide specific features to neural networks, which enhance their ability to process image inputs and improve their feature extraction efficiency. Typically, these networks are used for recognizing 2D images. However, the proposed CNN in this study utilizes 1D convolutional layers that can directly handle numerical datasets without requiring any preprocessing. Previous studies, as in [36,37], have commonly depicted color information in an image by utilizing multiple channels for each pixel. The structure of 1DCNNs consists of different layers that carry out functions such as feature extraction and reducing network size.

This study examines the network architecture and clarifies how the CNN model functions. The 1DCNN architecture consists of three primary layers: the convolutional layer, pooling layer, and fully connected layer. The training process involves adjusting a set of filters that move across the input matrix and calculate the dot product between the input and filter values. In this study, we examine the 1DCNN architecture and explain how the model works. The proposed 1DCNN model is composed of ten layers, denoted as L1 to L10, out of which four are dense or weighted layers, as outlined in Table 9. The architecture of the CNN model is visually depicted in Figure 7.

**Table 9.** Architectural details of proposed CNN models.

Layer No.	Layer Detail	Kernal Size	Output Shape	Trainable Parameters
L1	Input Raw Data Sample	-	60 × 1	-
L2	conv1d_2+ Relu	1 × 3	60 × 16	64
L3	conv1d_3+ Relu	1 × 3	60 × 16	784
L4	MaxPooling1D	1 × 2	30 × 16	-
L5	conv1d_4+ Relu	1 × 3	15 × 64	1586
L6	conv1d_5+ Relu	1 × 3	15 × 64	3104
L7	MaxPooling1D	1 × 2	15 × 32	-
L8	Batch_Normalization (BN)	-	15 × 32	128
L9	Flatten	-	480	-
L10	Dense + SoftMax	-	11	5291



**Figure 7.** Architecture of the proposed 1DCNN model.

The proposed 1DCNN model is fed with raw data, representing direct measurements of three-phase voltages and currents (60 features) collected from all buses and transmission lines. These measurements are used without any preprocessing and represent data for a single cycle. The proposed models in this study have two convolutional stacked layers, with each layer containing two 1D convolutional layers. The primary objective of these convolutional layers is to automatically extract important features from the input voltage and current during offline phase. The main function of using Batch Normalization (BN) in the CNN is to address the internal covariate shift input features problem that arises during training. Covariate shift occurs when the distribution of the input to a layer changes, which makes it difficult for the network to learn and slows down the training process. BN addresses this problem by normalizing the input features in each batch, thereby reducing the internal covariate shift and stabilizing the network's training process. Additionally, BN can also act as a regularizer by reducing the generalization error of the network and improving its overall performance [38].

The activation function serves as a mathematical operation that converts the input into a non-linear output. It plays a critical role as a gate, regulating the flow of information between the input, the hidden neuron, and the output in a neural network. In each convolutional layer, the rectified linear activation function (Relu or tanh) is employed for its advantageous properties. Convolutional layers utilize the mathematical operation of convolution, which differs from the matrix multiplication used in other types of neural networks. The one-dimensional convolution operation can be mathematically represented by Equation (6), as follows:

$$C_j = f \left[ \sum_{i=1}^n (x_{i+j} \times W_i) + b \right] \quad (6)$$

In this context, the convolution output layer is denoted by  $C_j$ , and the activation function used is  $f$ .  $x$  and  $W$  represent the mini batch of input data and filters, respectively, and  $b$  is bias of each neuron.  $BN(x)$  represents batch normalization applied to input  $x$ ,  $n$  refers to the dimension of the filter, and  $b$  represents the bias term. The 1D convolutional layers at L2, L3, L5, and L6 utilize sets of learnable filters. The first stacked convolutional layer employs 16 filters, while the second stacked convolutional layer uses 32 filters. As previously mentioned, both the 10-fold cross-validation method and custom train/test split were employed.

For cross-validation, the model was trained for 50 epochs, while for custom splitting, it was trained for 100 epochs, with a batch size of 20. Table 10 presents the performance metrics of the proposed models. It can be observed that the accuracy of each model is

consistently high, exceeding 0.99, except for the fault location model (FLE), at about 0.96, which exhibits a slightly lower accuracy compared to the other models. The various CNN configurations are implemented with various selections of the number of convolutional layers to analyze the performance behavior of the CNN models for fault classification, detection, and fault location. Furthermore, Table 10 shows that as the number of convolutional layers increases, the performance for fault classification, detection, and fault location improves. As the number of convolutional layers increases, the number of trainable parameters also increases. However, the suggested model outperforms others while maintaining a lower number of trainable parameters, mainly due to the utilization of stacked convolutional layers. This approach achieves a balance between model performance and complexity, resulting in improved performance without excessive complexity in terms of trainable parameters.

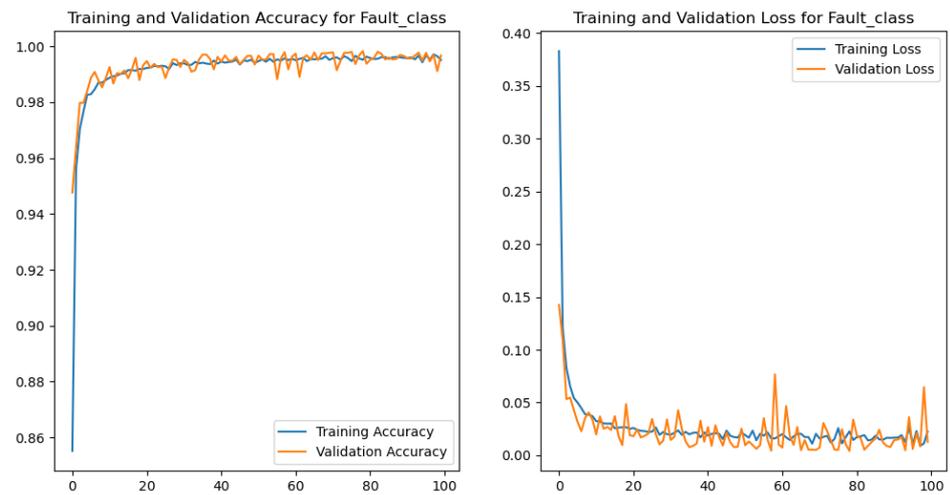
**Table 10.** Performance results of different CNN models for FCT, FLI, and FLE in the studied system.

CNN Model	Trainable Parameters	Normal Classification (NC)			Cross-Validation (CV) 10-Fold		
		FCT Accuracy %	FLI Accuracy %	FLE Accuracy %	FCT Accuracy %	FLI Accuracy %	FLE Accuracy %
4 Denes+ FC	10,394	99.83	99.98	89.45	99.34	99.7	92.57
6 Denes+ FC	28,698	99.998	99.97	95.41	99.32	99.99	96.85

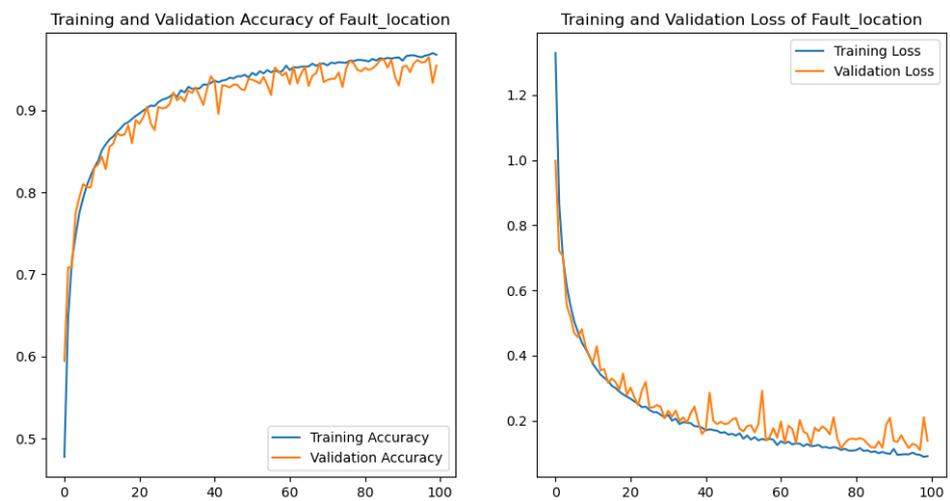
Figure 8a illustrates the accuracy and loss curves for Fault Class Type (FCT) during training and testing cases across 50 epochs for line faulty and 100 epochs for fault class and fault location. It can be observed that initially, the testing curve attempts to closely align with the training curve. However, around epoch 40, a plateau in model performance becomes apparent. The observed behavior suggests that the initial learning rate requires more epochs to achieve model convergence with higher performance. The model struggles to learn the problem efficiently using the initial learning rate, as shown in Figure 8b,c. As a result, adopting an adaptive learning strategy becomes crucial to address this issue.

By reducing the learning rate at that point, the model achieves better performance on both the training and test sets with quicker convergence. Eventually, the proposed models achieve convergence with higher performance after approximately 30 epochs in cases of line faulty, while in cases of fault location, the model converges at 60, as evidenced by the training curve closely aligning with the testing curve. This indicates that there is no need for further reduction of the learning rate or additional training epochs, as the model has reached a satisfactory level of performance.

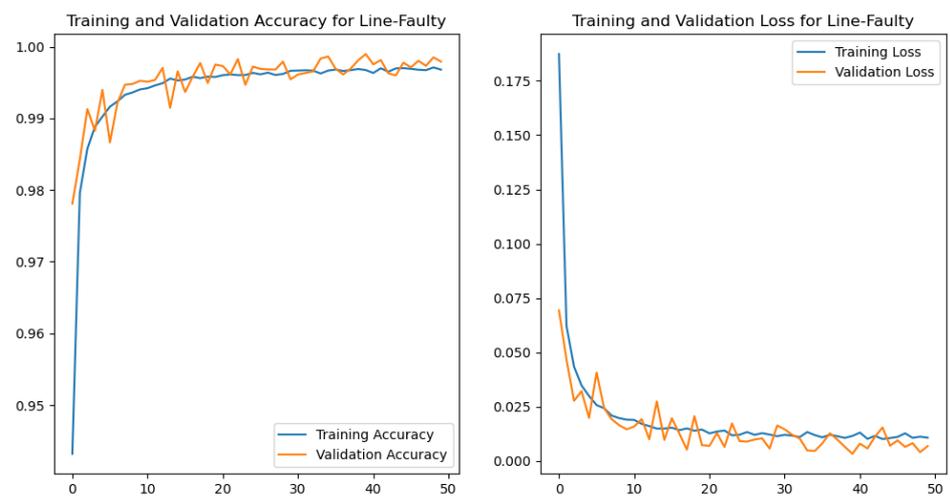
Figure 9 depicts the confusion matrix, providing a comprehensive evaluation of the classification performance. It represents the predicted results in the matrix's rows and the actual results in the columns. The diagonal elements represent accurately classified samples, while the off-diagonal elements indicate misclassified samples. In Figure 9a, the Fault Class Type response (FCT) for the testing data is showcased. Although a few acceptable misclassified errors are observed when compared to the extensive test data, the overall classification performance remains noteworthy. Furthermore, Figure 9b illustrates the Line Faulty Identifier (LFI), and Figure 9c displays the Fault Location Estimation (FLE) for nine predefined locations.



(a) Training and loss curve of FTC with NC.

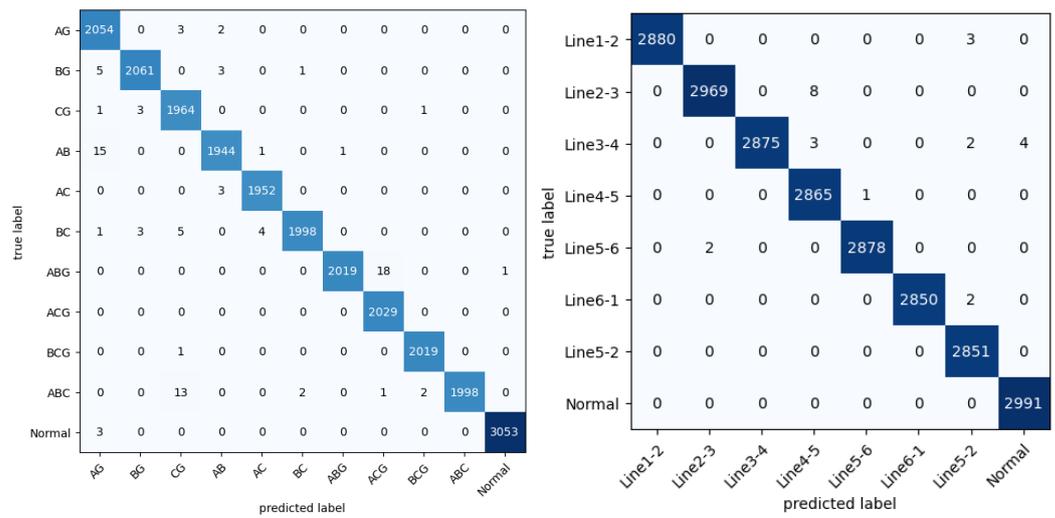


(b) Training and loss curve of LFI with NC.



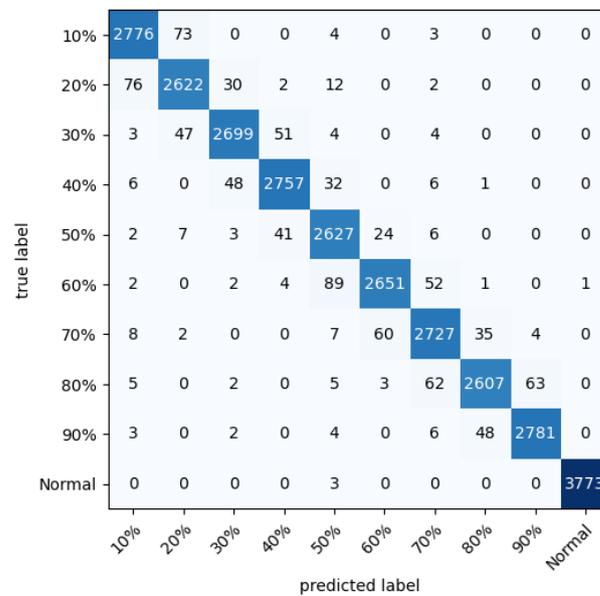
(c) Training and loss curve of FLE with NC.

Figure 8. Training and loss curve for training progress.



(a) Confusion matrix of FCT with NC.

(b) Confusion matrix of LFI with NC.



(c) Confusion matrix of FLE with NC.

Figure 9. Confusion matrix of proposed CNN using testing data.

#### 4. Discussion

The proposed models were extensively compared with various previous methods using data generated from a simulated system. In this comprehensive analysis, both the classification accuracy and computational time cost were considered as the basis for comparison. The proposed models are compared with several traditional machine learning methods, which include Support Vector Machine (SVM) [39,40], Decision Tree (DT), and KNN [40,41] classification models.

In these studies, significant features are obtained either directly or through transforming the voltage and current signals into another domain. These extracted features are then utilized to train machine learning algorithms effectively. Furthermore, the proposed models were also compared with deep learning algorithms such as CNN [42], which uses wavelet-based features and Gated Recurrent Unit (GRU) [43], which are considered as modern deep learning methods that process the raw input and voltage samples similarly to the proposed models. The study performs a comprehensive comparison of the performance

and computational time cost between the considered previous methods and the proposed models. The outcomes of this comparison are outlined in Table 11.

**Table 11.** Comparison of different methods in terms of accuracy and computational time.

Classification Model	Training Phase (Offline)			Testing Phase (Online)	
	Accuracy	Feature Extraction (s)	Training Time (s)	Feature Extraction (s)	Testing Time (s)
SVM [40]	0.9961	406.87	124.23	48.21	4.70
KNN [41]	0.9902	406.87	0.02	48.21	0.097
DT [41]	0.999	406.87	0.15	48.21	0.006
GRU [43]	0.9979	-	1350.42	-	4.57
CNN [42]	0.9988	-	882.16	-	1.21
Proposed ANN	0.999	-	1466	-	0.097
Proposed CNN	0.999	-	1512	-	0.1035

Table 11 illustrates that the accuracy of the proposed model is similar to other models, with the DT model achieving the same accuracy as the proposed model. However, this does not necessarily mean that the computational time is the same. In the training phase, machine learning models typically required less computational time compared to deep learning models, as well as the proposed models. However, manual feature extraction for machine learning models can add extra computational time and effort.

The findings indicate that the automatic feature extraction process of DNN models makes deep learning algorithms computationally more intensive during model training. However, during online mode, these models are specifically designed for their intended task, which is to predict faulty class labels for unseen data or test cases. In this mode of execution, the models are utilized solely for making predictions and effectively managing real-time scenarios, without the need for any additional training or model adjustments. On the other hand, conventional machine learning models typically demand considerably more computational time during the testing phase in online mode. Therefore, the proposed models achieve comparable performance with a total computational time of only 0.097–0.1035 s during testing. Once trained, these deep learning models do not require feature extraction or features in online mode.

## 5. Conclusions

This study addresses the crucial task of fault detection, classification, and location within multi-machine power systems, a critical aspect of smart grids. Simulated faulty data, derived from actual values of three-phase voltages and currents, were meticulously collected, preprocessed, and classified using deep learning algorithms. Novel classifier models based on Artificial Neural Networks (ANNs) and 1D Convolution Neural Networks (1D-CNNs) were introduced for intelligent fault identification, classification, and location within the IEEE 6-bus system in different scenarios.

These models demonstrated notable precision, achieving accuracies of 99.99%, 99.98% for identifying faulty lines, 99.75%, 99.99% for fault classification, and 98.25%, 96.85% for fault location for ANN and 1D-CNN, respectively. Importantly, this study stands out due to its comprehensive approach, encompassing the entirety of the network rather than focusing on isolated subsections. Our approach sampled faults without specific restrictions, such as feature selection and data reduction, thereby mitigating the computational burden of condition-based pattern selection. This renders our approach generalizable for fault analysis in power transmission networks, consequently bolstering computational efficiency during both the training and testing phases.

Additionally, we conducted a comprehensive comparative analysis of our models against established fault analysis techniques [44–50], affirming their superior performance. Table 12 provides a concise representation of our model’s notable fault analysis efficacy. It is worth noting that the combination of voltage and current signals enhanced accuracy by utilizing their behavior and features during the fault (transient time). In contrast,

image-based methods, while potentially more accurate, introduce a higher computational burden, underscoring the appeal of employing raw data (1D) directly. This research contributes to advancing fault detection, classification, and localization, which holds significant implications for enhancing power system reliability within smart grids.

**Table 12.** Comparison of different methods.

Model Name	Input Parameter	Data Type	Learning Type	FCT Accuracy
DBN [44]	Voltage	Signal	Unsupervised+ Supervised	99.00%
DSE-SVM [45]	Current	Signal	Supervised	96.00%
PNN [46]	Voltage	Signal	Supervised	99.33%
LSTM [47]	Voltage	Signal	Supervised	99.77%
SAT-CNN [48]	Voltage and current	Image	Unsupervised	99.72%
MobileNetV3-SA [49]	Voltage and current	Image	Supervised	99.90%
Waveform Encoding and Segmentation [50]	Voltage and current	Image	Supervised	99.77%
The proposed ANN	Voltage and current	Signal	Supervised	99.75%
The proposed CNN	Voltage and current	Signal	Supervised	99.99%

**Future Work:** Through the envisaged methodology, we intend to broaden the scope of our research to encompass diverse and intricate network structures. It is important to acknowledge that our proposed approach may be susceptible to external noise or the influence of switching loads. We also plan to work with varying dataset sizes and implement a combined algorithm strategy to mitigate the occurrence of false trips or inadvertent malfunctions in the protection system. Finally, the scalability assessment will be explored in our proposed method on larger systems to verify its effectiveness and robustness when applied to more extensive and complex scenarios.

**Author Contributions:** Conceptualization, A.S.A., H.H.B. and M.F.; Methodology, A.S.A., H.H.B. and M.F.; Software, A.S.A.; Validation, A.S.A., H.H.B. and M.F.; Investigation, A.S.A., H.H.B. and M.F.; Data curation, A.S.A.; Writing—original draft, A.S.A., H.H.B. and M.F.; Writing—review & editing, A.S.A., H.H.B. and M.F.; Supervision, H.H.B. and M.F.; Administration H.H.B. and M.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** For those interested, the dataset for the IEEE 6-bus network can be accessed at the following URL: [https://drive.google.com/file/d/1o63\\_qx2g69qsWUJ5hawex0SrGvILGEOF/view?usp=sharing](https://drive.google.com/file/d/1o63_qx2g69qsWUJ5hawex0SrGvILGEOF/view?usp=sharing).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

DNN	Deep Neural Network	R	Positive Sequence Resistance
ANN	Artificial Neural Network	X	Positive Sequence Reactance
CNN	Convolution Neural Network	DBN	Deep Believe Network
NC	Normal Classification	SA	Shuffle Attention
CV	Cross Fold Validation	DSE	Dynamic State Estimation
FCT	Fault Class Type	SVM	Support Vector Machine
LFI	Line Faulty Identifiers	PNN	Probabilistic Neural Network
FLE	Fault Location Estimator	FNNs	Feed Forward Neural Network

## References

- Gupta, M.; Wadhvani, R.; Rasool, A. A real-time adaptive model for bearing fault classification and remaining useful life estimation using deep neural network. *Knowl.-Based Syst.* **2023**, *259*, 110070. [CrossRef]
- Azad, S.; Sabrina, F.; Wasimi, S. Transformation of smart grid using machine learning. In Proceedings of the 2019 29th Australasian Universities Power Engineering Conference (AUPEC), Nadi, Fiji, 26–29 November 2019; pp. 1–6.

3. Azeroual, M.; Boujoudar, Y.; Bhagat, K.; El Iysaouy, L.; Aljarbouh, A.; Knyazkov, A.; Fayaz, M.; Qureshi, M.S.; Rabbi, F.; Markhi, H.E. Fault location and detection techniques in power distribution systems with distributed generation: Kenitra City (Morocco) as a case study. *Electr. Power Syst. Res.* **2022**, *209*, 108026. [[CrossRef](#)]
4. Sreelekha, V.; Prince, A. ANFIS-Based Fault Distance Locator with Active Power Differential-based Faulty Line Identification Algorithm for Shunt and Series Compensated Transmission Line using WAMS. *IEEE Access* **2023**, *11*, 91500–91510. [[CrossRef](#)]
5. Foba, V.J.; Boum, A.T.; Mbey, C.F. Optimal reliability of a smart grid. *Int. J. Smart Grid* **2021**, *5*, 74–82.
6. Rajesh, P.; Kannan, R.; Vishnupriyan, J.; Rajani, B. Optimally detecting and classifying the transmission line fault in power system using hybrid technique. *ISA Trans.* **2022**, *130*, 253–264. [[CrossRef](#)]
7. Mirshekali, H.; Dashti, R.; Keshavarz, A.; Shaker, H.R. Machine learning-based fault location for smart distribution networks equipped with micro-PMU. *Sensors* **2022**, *22*, 945. [[CrossRef](#)]
8. Li, B.; Delpha, C.; Diallo, D.; Migan-Dubois, A. Application of Artificial Neural Networks to photovoltaic fault detection and diagnosis: A review. *Renew. Sustain. Energy Rev.* **2021**, *138*, 110512. [[CrossRef](#)]
9. Phadke, A.G.; Bi, T. Phasor measurement units, WAMS, and their applications in protection and control of power systems. *J. Mod. Power Syst. Clean Energy* **2018**, *6*, 619–629. [[CrossRef](#)]
10. Cui, M.; Wang, J.; Tan, J.; Florita, A.R.; Zhang, Y. A novel event detection method using PMU data with high precision. *IEEE Trans. Power Syst.* **2018**, *34*, 454–466. [[CrossRef](#)]
11. Saldaña-González, A.E.; Sumper, A.; Aragüés-Peñalba, M.; Smolnikar, M. Advanced distribution measurement technologies and data applications for smart grids: A review. *Energies* **2020**, *13*, 3730. [[CrossRef](#)]
12. Thilakarathne, C.; Meegahapola, L.; Fernando, N. Real-time voltage stability assessment using phasor measurement units: Influence of synchrophasor estimation algorithms. *Int. J. Electr. Power Energy Syst.* **2020**, *119*, 105933. [[CrossRef](#)]
13. Shadi, M.R.; Ameli, M.T.; Azad, S. A real-time hierarchical framework for fault detection, classification, and location in power systems using PMUs data and deep learning. *Int. J. Electr. Power Energy Syst.* **2022**, *134*, 107399. [[CrossRef](#)]
14. Tshenyego, O.; Samikannu, R.; Mtengi, B.; Mosalaosi, M.; Sigwele, T. A Graph-Theoretic Approach for Optimal Phasor Measurement Units Placement Using Binary Firefly Algorithm. *Energies* **2023**, *16*, 6550. [[CrossRef](#)]
15. Wang, W.; Yin, H.; Chen, C.; Till, A.; Yao, W.; Deng, X.; Liu, Y. Frequency disturbance event detection based on synchrophasors and deep learning. *IEEE Trans. Smart Grid* **2020**, *11*, 3593–3605. [[CrossRef](#)]
16. Duan, N.; Stewart, E.M. Frequency event categorization in power distribution systems using micro pmu measurements. *IEEE Trans. Smart Grid* **2020**, *11*, 3043–3053. [[CrossRef](#)]
17. Guo, J.; Wan, J.L.; Yang, Y.; Dai, L.; Tang, A.; Huang, B.; Zhang, F.; Li, H. A deep feature learning method for remaining useful life prediction of drilling pumps. *Energy* **2023**, *282*, 128442. [[CrossRef](#)]
18. Samanta, A.; Chowdhuri, S.; Williamson, S.S. Machine learning-based data-driven fault detection/diagnosis of lithium-ion battery: A critical review. *Electronics* **2021**, *10*, 1309. [[CrossRef](#)]
19. Pan, P.; Mandal, R.K.; Rahman Redoy Akanda, M.M. Fault classification with convolutional neural networks for microgrid systems. *Int. Trans. Electr. Energy Syst.* **2022**, *2022*, 8431450. [[CrossRef](#)]
20. Karić, A.; Konjić, T.; Jahić, A. Power system fault detection, classification and location using artificial neural networks. In *Advanced Technologies, Systems, and Applications II: Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT), Banja Vrucica, Bosnia and Herzegovina, 25–28 May 2017*; Springer: Cham, Switzerland, 2019; pp. 89–101.
21. Bishal, M.R.; Ahmed, S.; Molla, N.M.; Mamun, K.M.; Rahman, A.; Al Hysam, M.A. ANN Based Fault Detection & Classification in Power System Transmission line. In Proceedings of the 2021 International Conference on Science & Contemporary Technologies (IC SCT), Dhaka, Bangladesh, 5–7 August 2021; pp. 1–4.
22. Shakiba, F.M.; Shojaee, M.; Azizi, S.M.; Zhou, M. Real-time sensing and fault diagnosis for transmission lines. *Int. J. Netw. Dyn. Intell.* **2022**, *1*, 36–47. [[CrossRef](#)]
23. Wen, L.; Gao, L.; Li, X.; Zeng, B. Convolutional neural network with automatic learning rate scheduler for fault classification. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–12. [[CrossRef](#)]
24. Mohd Amiruddin, A.A.A.; Zabiri, H.; Taqvi, S.A.A.; Tufa, L.D. Neural network applications in fault diagnosis and detection: An overview of implementations in engineering-related systems. *Neural Comput. Appl.* **2020**, *32*, 447–472. [[CrossRef](#)]
25. 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.* **2020**, *10*, 4965. [[CrossRef](#)]
26. Alvarez, G.P. Real-time fault detection and diagnosis using intelligent monitoring and supervision systems. In *Fault Detection, Diagnosis and Prognosis*; IntechOpen: London, UK, 2020.
27. Gao, H.X.; Kuenzel, S.; Zhang, X.Y. A hybrid ConvLSTM-based anomaly detection approach for combating energy theft. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–10. [[CrossRef](#)]
28. Alrifayy, M.; Lim, W.H.; Ang, C.K.; Natarajan, E.; Solihin, M.I.; Juhari, M.R.M.; Tiang, S.S. Hybrid deep learning model for fault detection and classification of grid-connected photovoltaic system. *IEEE Access* **2022**, *10*, 13852–13869. [[CrossRef](#)]
29. Yoon, D.H.; Yoon, J. Deep learning-based method for the robust and efficient fault diagnosis in the electric power system. *IEEE Access* **2022**, *10*, 44660–44668. [[CrossRef](#)]
30. Zhang, S.; Zhang, S.; Wang, B.; Habetler, T.G. Deep learning algorithms for bearing fault diagnostics—A comprehensive review. *IEEE Access* **2020**, *8*, 29857–29881. [[CrossRef](#)]

31. Coban, M.; Tezcan, S.S. Feed-Forward Neural Networks Training with Hybrid Taguchi Vortex Search Algorithm for Transmission Line Fault Classification. *Mathematics* **2022**, *10*, 3263. [[CrossRef](#)]
32. Vyas, B.Y.; Das, B.; Maheshwari, R.P. Improved fault classification in series compensated transmission line: Comparative evaluation of Chebyshev neural network training algorithms. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *27*, 1631–1642. [[CrossRef](#)]
33. Azizi, S.; Jegarluei, M.R.; Dobakhshari, A.S.; Liu, G.; Terzija, V. Wide-area identification of the size and location of loss of generation events by sparse PMUs. *IEEE Trans. Power Deliv.* **2020**, *36*, 2397–2407. [[CrossRef](#)]
34. Nitve, B.; Naik, R. Steady state analysis of IEEE-6 Bus System Using PSAT power toolbox. *Int. J. Eng. Sci. Innov. Technol. (IJESIT)* **2014**, *3*, 197–203.
35. Xiang, L.; Wang, P.; Yang, X.; Hu, A.; Su, H. Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism. *Measurement* **2021**, *175*, 109094. [[CrossRef](#)]
36. Aziz, F.; Haq, A.U.; Ahmad, S.; Mahmoud, Y.; Jalal, M.; Ali, U. A novel convolutional neural network-based approach for fault classification in photovoltaic arrays. *IEEE Access* **2020**, *8*, 41889–41904. [[CrossRef](#)]
37. Ko, T.; Kim, H. Fault classification in high-dimensional complex processes using semi-supervised deep convolutional generative models. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2868–2877. [[CrossRef](#)]
38. Liu, M.; Wu, W.; Gu, Z.; Yu, Z.; Qi, F.; Li, Y. Deep learning based on batch normalization for P300 signal detection. *Neurocomputing* **2018**, *275*, 288–297. [[CrossRef](#)]
39. Manohar, M.; Koley, E. SVM based protection scheme for microgrid. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), Kerala, India, 6–7 July 2017; pp. 429–432.
40. Abdelgayed, T.S.; Morsi, W.G.; Sidhu, T.S. A new approach for fault classification in microgrids using optimal wavelet functions matching pursuit. *IEEE Trans. Smart Grid* **2017**, *9*, 4838–4846. [[CrossRef](#)]
41. Mishra, D.P.; Samantaray, S.R.; Joos, G. A combined wavelet and data-mining based intelligent protection scheme for microgrid. *IEEE Trans. Smart Grid* **2015**, *7*, 2295–2304. [[CrossRef](#)]
42. Rai, P.; Londhe, N.D.; Raj, R. Fault classification in power system distribution network integrated with distributed generators using CNN. *Electr. Power Syst. Res.* **2021**, *192*, 106914. [[CrossRef](#)]
43. James, J.; Hou, Y.; Lam, A.Y.; Li, V.O. Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks. *IEEE Trans. Smart Grid* **2017**, *10*, 1694–1703.
44. Ola, S.R.; Saraswat, A.; Goyal, S.K.; Jhahharia, S.; Mahela, O.P. A technique using Stockwell transform based median for detection of power system faults. In Proceedings of the 2018 IEEE 8th Power India International Conference (PIICON), Kurukshetra, India, 10–12 December 2018; pp. 1–5.
45. Xie, J.; Meliopoulos, A.S.; Xie, B. Transmission line fault classification based on dynamic state estimation and support vector machine. In Proceedings of the 2018 North American Power Symposium (NAPS), Fargo, ND, USA, 9–11 September 2018; pp. 1–5.
46. Mukherjee, A.; Chatterjee, K.; Kundu, P.K.; Das, A. Probabilistic neural network-aided fast classification of transmission line faults using differencing of current signal. *J. Inst. Eng. (India) Ser.* **2021**, *102*, 1019–1032. [[CrossRef](#)]
47. Liang, H.; Liu, Y.; Sheng, G.; Jiang, X. Fault-cause identification method based on adaptive deep belief network and time–frequency characteristics of travelling wave. *IET Gener. Transm. Distrib.* **2019**, *13*, 724–732. [[CrossRef](#)]
48. Fahim, S.R.; Sarker, Y.; Sarker, S.K.; Sheikh, M.R.I.; Das, S.K. Self attention convolutional neural network with time series imaging based feature extraction for transmission line fault detection and classification. *Electr. Power Syst. Res.* **2020**, *187*, 106437. [[CrossRef](#)]
49. Xi, Y.; Zhang, W.; Zhou, F.; Tang, X.; Li, Z.; Zeng, X.; Zhang, P. Transmission line fault detection and classification based on SA-MobileNetV3. *Energy Rep.* **2023**, *9*, 955–968. [[CrossRef](#)]
50. Yuan, J.; Jiao, Z. Faulty Feeder Detection for Single Phase-to-Ground Faults in Distribution Networks Based on Waveform Encoding and Waveform Segmentation. *IEEE Trans.* **2023**, *14*, 4100–4115. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.