Deep Learning–Based Fault Detection and Classification in AC Microgrids using CNNs

Rakesh Sahu[1], **Deepak Kumar Lal**[2†],
Pratap Kumar Panigrahi[1], and **Rudranarayan Pradhan**[3], Non-members

ABSTRACT

Since there are issues with bidirectional power flow, fault current levels are extremely low, and operating conditions are dynamic in nature, with high challenges in terms of fault detection and classification with distributed energy resources in AC microgrid. To maintain system reliability and assure promptness in fault detection and response, it is extremely important to properly identify common faults like LG fault, LL fault, LLG fault, and three-phase faults. A new method based on deep learning and a deep learning method based on a convolutional neural network are presented in this paper. With this method, there would not be a need to specifically extract features. Since a large amount of data was simulated in terms of a modified 7-bus AC system created in RTDS software and tested in different fault conditions and operating conditions, a large amount of data was generated. The model finally uses a dense layer and a SoftMax classification. A comparison was made on its performance of the conventional machine learning algorithms such as Multilayer Perceptron Classifier, Decision Tree Classifier, Naïve Bayes Classifier, and the Support Vector Machine. In the simulation, the performance of the proposed model was found to be superior to the existing methods with an accuracy of up to 99%, thereby clearly distinguishing between symmetry and asymmetry through the analysis of the confusion matrix used in the model.

Keywords: Microgrids, Fault Detection, CNN, Deep Learning, Power System Protection

1. INTRODUCTION

The electric power sector has seen tremendous changes with the advent and incorporation of various forms of renewable energy resources. With regards to this, some major role has to be played by the improvement seen in microgrid installation and its ability to manage localized electric power generation and consumption with control and associated advantages. For instance, with its advantages in grid-connected and electric power generation and consumption without connectivity with other resources, as mentioned in the research articles in references [1-2], various issues are created, mainly with complex fault detection and classification issues, as mentioned in the research articles in references [3-5].

AC microgrids can encounter a range of faults: line-to-ground faults (LG), line-to-line faults (LL), double line-to-ground faults (LLG), and three-phase faults (LLL). These faults can induce a range of problems in AC MG's and can go undetected in a timely manner when not identified in a proper and efficient way. Traditional methods of protection using overcurrent relays can cause problems in MG situations, especially in inverter-dominated situations where low fault levels characterize [6-8].

Some of the ML-based approaches such as SVM, DT, NB, and MLP have been implemented on the microgrid fault analysis based on voltage and current measurements. Although these methods improve the fault detection accuracy yet there are certain limitations regarding these algorithms since their performance is highly dependent on handcrafted feature extraction, thereby limiting adaptability to varying operating conditions [9–13]. The requirement to predefine domain-specific features such as symmetrical components, harmonic numbers, and wavelet coefficients makes these approaches noise sensitive, operating condition sensitive, and model assumption sensitive, thus lowering their adaptability to different fault conditions [14-19]. Aiming at these challenges, this work investigates a CNN-based framework for fault detection and classification in AC microgrids. Finally, the proposed model is tested on a modified IEEE 7-bus AC microgrid simulated in RTDS (Real-Time Digital Simulator) using three-phase current signals under diverse fault scenarios [20]. The CNN automatically learns features that are discriminative from raw waveforms, hence obviating manual feature design [21]. This offers a very fair comparison against NB, DT, SVM, and MLP, displaying higher accuracy and robustness of the CNN approach [11-12,20].

Convolutional Neural Networks (CNNs) have advantages over recurrent models (LSTM and GRU) when it comes to capturing short-term transient features found in

Manuscript received on January 30, 2026; revised on May 8, 2026; accepted on June 11, 2026. This paper was recommended by Associate Editor Chawasak Rakpenthai.

[1]The authors are with Department of Electrical & Electronics Engineering, GIET University, Gunupur, Odisha, India.

[2]The author is with Department of Electrical Engineering Veer Surendra Sai University of Technology, Burla, Odisha, India.

[3]The author is with School of Electrical Sciences, Odisha University of Technology and Research, Bhubaneswar, Odisha, India.

[†]Corresponding author: dklal_ee@vssut.ac.in

©2026 Author(s). This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License. To view a copy of this license visit: https://creativecommons.org/licenses/by-nc-nd/4.0/.

Digital Object Identifier: 10.37936/ecti-eec.2026243.263488

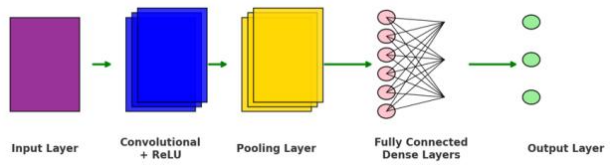

***Fig. 1**: Architecture of Proposed CNN based fault detection Model.*

***Table 1**: Sample dataset for an LLG fault event.*

Time (s)	Current (Phase-A)	Current (Phase-B)	Current (Phase-C)
0.000000	0.014057	-0.193897	0.160845
0.000167	0.020231	-0.195189	0.157184
0.000333	0.026372	-0.195777	0.152925
0.000500	0.032466	-0.195657	0.148070
0.001000	0.050306	-0.191012	0.129998

the currents from faults that occur in a microgrid because fault events are characterized by abruptly changing signals rather than being defined by long sequences of data. This means that CNN models can process data more efficiently than recurrent models do and are therefore more suitable for real-time applications. Furthermore, CNNs typically have fewer parameters than recurrent models and can take advantage of parallel processing capabilities, both of which significantly reduce the time required to generate inferences from CNN models as opposed to recurrent models.

The main contributions of this work can be listed as follows:

I. CNN-based architecture performs better than MLP-based architecture in microgrid fault classification in comparison to other ML models.
II. Simulations for the proposed approach are carried out using an IEEE MG standard test system, making the comparison with existing literature feasible.
III. It provides a scalable real-time fault detection method with minimal preprocessing needs, reducing processing complexity and time [1, 11, 20].

The rest of the paper is arranged in the following manner. Section 2 describes the methodologies adopted. Section 3 includes the results and discussion. Section 4 states the conclusion along with future scope.

2. METHODOLOGY

As depicted in Fig. 1, the proposed CNN model is composed of an input block, convolutional and ReLU neural blocks, a dense neural block, and an output block. The CNN is responsible for automatically learning the input signal representation. This is an improvement since there is no need for manual extraction of relevant features. In comparison with the MLP, the CNN relies on local receptive fields. This is an improvement since the CNN can effectively capture the correlated behaviours of the input measured signals. Using a multi-class classification approach with the proposed CNN architecture allows for differentiating several types of faults (LG, LL, LLG, LLL) present in the system. A SoftMax function will be used in the output layer to calculate the probability corresponding to each class of fault.

The input layer ingests raw three-phase current signals or waveform segments ($I_a, I_b,\ I_c$) without losing the temporal characteristics of the signals. Convolutional layers with ReLU activation extract local fault-related features using small sliding kernels. The nonlinear behaviour of ReLU function is given by:

$$f(x) = max(0, x) \tag{1}$$

Acceleration of convergence and mitigation of vanishing-gradient issues are enhanced by ReLU. Pooling layers perform a dimensionality reduction while computing local maxima or averages, emphasizing dominant fault features and making the models more robust to noise. The flatten feature maps feed the fully connected layers that learn high-level fault representations from the three-phase currents. An output layer creates normalized probabilities for fault and no-fault classes by:

$$P\,(y = k \mid x) = \frac{e^{zk}}{\sum_{j=1}^{K} e^{z}} \tag{2}$$

Since the denominator $\sum_{j=1}^{K} e^{z}$ ensures normalization, all probabilities add up to 1, where $P\,(y = k \mid x)$ is the probability that the input x belongs to class k, z_k is the raw output (logit) of the neuron corresponding to class k, and K is the total number of classes.

Dataset creation and gathering, signal preprocessing and data augmentation, CNN model design, data flow pipeline, training and hyperparameter optimization, model evaluation, and comparative analysis are the six main stages of the methodology.

2.1 Creation and Gathering of datasets

The advanced CNN-based Fault classification scheme is designed based on the IEEE 7-bus system modelled and simulated by the RTDS Tool. Three-phase current measurements were captured under various fault scenarios to guarantee real-world data acquisition with high fidelity. A few instances of current measurement are displayed in Table 1, describing the characteristics of the IEEE-7bus system just after the occurrence of an LLL Fault. A variety of distributed energy sources, loads, transmission lines, as well as protective devices, are included in the model to obtain reliable data through realistic simulation. From the perspective of the IEEE standards-based model, the most common Faults encountered by a distribution model are classified into four cases: LG Faults, LL Faults, LLG Faults, and the three-phase Faults classified as LLL Faults.

Each type of fault was also designed based on the different nature of the current waveform. The fault simulations were conducted by varying fault resistance from

0.001 Ω to 10 Ω, covering both solid and high-impedance fault conditions. This variation ensures distinguishable electrical signatures between closely related faults such as LL and LLG, thereby improving labelling accuracy and classification robustness.

For instance, LG faults exhibit unbalanced currents due to single-phase grounding, the LL faults exhibit currents due to phase-to-phase currents, the LLG faults exhibit the combination of phase currents, while the LLL faults exhibit currents due to the balanced short circuit. For the sake of dataset generalization, the faults were simulated under various conditions. For instance, the faults were assumed for different places, considering both the grid-connected mode as well as the islanded mode, along with the involvement of inverter-based distributed generators due to their contributory nature.

All the simulation scenarios sampled the three-phase current signals at a high frequency in the kilohertz range for the capture of post-fault transients. Accordingly, time-synchronized waveforms (I_a, I_b, I_c) were logged and labelled at each operating condition and fault type (LG, LL, LLG, or LLL), resulting in a supervised learning dataset. Each entry in the data includes time stamps, phase current magnitudes, and the associated class label. The proposed dataset offers a variety of operation modes and fault location including fault resistances, thereby improving generalisation and allowing unobserved fault scenarios to be addressed reliably.

2.2 Signal Preprocessing and Data Augmentation

The main aim is to provide reliable microgrid systems, which rely on the accuracy of fault detection and classification. Traditional methods of waveform and impedance-based examination usually require extensive feature engineering and do not perform well even under advanced engineering with noisy or nonlinear signals. Deep learning architectures, especially CNN, offer a more reliable option since they are able to extract features automatically from the raw signal patterns. However, the pre-processing and augmentation of the existing signal data are necessary pipeline steps in providing a reliable classification evaluation.

2.2.1 Signal Preprocessing

The implementation of analysis begins from processing current in three-phase form in a form useful for training a CNN. The fault dataset contains high-resolution current signals that were collected during fault conditions, which were denoted as LG, LL, LLG, and LLL. Furthermore, in order to normalize each of the current signals from each of the three phases, Z-normalization was employed. Then, the signals are converted into two-dimensional representations using sliding window segmentation and reshaping while preserving the temporal dynamics and inter-phase relationships. Finally, the dataset is divided into training, validation, and testing sets, typically in a 70:20:10 ratio, to ensure a reliable generalization performance.

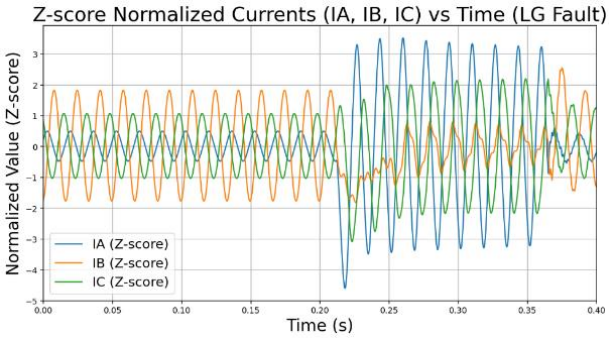

***Fig. 2**: Z-score normalized currents (Ia, Ib, Ic).*

Three-phase currents (I_a, I_b, and I_c) sampled at high resolution usually make up the recorded signals. A mathematical formula applies z-score normalization to each phase current in order to lessen the impact of scale variations:

$$X_i^* = \frac{X_i - \mu}{\delta} \tag{3}$$

Where μ, δ is the mean and standard deviation, and Xi is the instantaneous current sample. By doing this, bias in gradient updates during CNN training is prevented and all features are guaranteed to lie on a comparable scale with zero mean and unit variance.

The Z-score normalization technique was performed independently on each phase, resulting in zero mean and unit variance, to avoid favouring high magnitude features, which helps to improve convergence stability. By ensuring that all input features are equal in magnitude. There is no bias towards higher magnitude signals during training.

Figure 2 indicates the Z-score normalized three-phase currents (I_a, I_b, I_c). A natural fluctuation in the levels of the currents occurs based on the operating environment; thus, the fluctuation in the magnitude of the currents has to be normalized in order to prevent large magnitude currents from affecting the learning operations performed in the system significantly. The currents have a typical fluctuation under a normal environment, slightly deviating from zero in a natural manner. At a specific time period of 0.2 seconds, the fault diverts the currents in the system from their normal course while observing a distortion in the three currents flowing through the system.

Initially, the three-phase currents (I_a, I_b, I_c) are converted into a single dimension time series data. To take advantage of spatial feature creation, the three-phase current signals will be segmented into fixed-length segments of some defined length (for example, 100 samples) in a sliding window method with a defined overlap. Each segment will then be re-shaped into two-dimensions of size (100 × 3) where the rows of the re-shaped data will represent time samples and the columns of re-shaped data will represent phase currents (e.g., currents in phases A, B and C). This representation maintains both the temporal dynamics of the signal and

the relationship between the phases, allowing for the use of convolutional kernels to effectively extract features from the segment created.

Fault contribution from inverter connected distributed energy resources (DER) is technically limited due to control limitations and protection limitations. In one way, the operation of synchronous generators is different from inverter-based generation because the inverter-based generation injects a controlled amount of current to the grid. The normal limit of current is a little greater than the rated current (between 1.2 to 2 times the rated current). The lower fault current contribution provides additional difficulty with the use of traditional protection methods. The proposed CNN model will learn how to detect minor differences in the transient characteristics of waveforms that can still be identified with a low fault contribution.

2.2.2 Data Augmentation

Large datasets are necessary for deep learning models to prevent overfitting. Fault datasets, however, are frequently restricted to a small number of recorded or simulated cases. Data augmentation techniques are used to address this:

i. Noise Injection: Gaussian noise, $n(t) \sim N(0, \sigma_n^2)$, is added to current signals:

$$I'(t) = I(t) + n(t) \tag{4}$$

Noise with no mean as represented by $\mu = 0$, indicating zero mean there for no signal bias, is instead bounded by intervals of noise spread (determined by σ_n^2) *of which there is a certain intensity* (σ_n is the standard deviation of the noise). This in turn means fluctuations, of which most fit between the borders of $[-2\sigma_n, +2\sigma_n]$, are kept around zero, thus improving the signal robustness against the measurement errors.

ii. Time Shifting: CNN filters' temporal invariance is enhanced by slight changes in the signal window that simulate faults that occur at marginally different time instants.

iii. Scaling and Amplitude Variation: For the expected change in the severity of the faults to be modelled, random scaling factors to the samples already obtained are added.

iv. Class Balancing through Oversampling: To balance the data set and avoid any further bias on the part of the classifier, classes of faults such as infrequent classes of fault, which occur infrequently—for example, infrequent classes of LLL fault—are artificially boosted.

To address the class imbalance between normal and fault conditions, oversampling the minority class (fault conditions) have been done. A weighting scheme to the input classes while training to effectively penalise false classification of infrequent faults is also applied. This will ensure that the classifier is functionally robust to the detection of rare, but critical, faults.

2.3 Graphical Representation of Faulty Datasets (LG, LL, LLG, LLL)

The accurate classification of a faulty scenario depends heavily upon a good representation of the associated data. Within this study, the three-phase current signal associated with a LG, LL, LLG, and LLL faulty scenario was extracted from a simulated system, where it was subsequently normalized, as shown in Fig. 3, following a standardization process that utilized a z-score.

For LG faults, significant unbalance is seen in one of the phases, while the other phases experience minor unbalance. For LLG faults, there are coupled disturbances in two phases caused by combined phase-to-phase and phase-to-ground faults. The critical case is when there are disturbances in all three phases, as seen in the case of LLL faults. These conspicuous wave patterns are significant "features" that are easily picked up by the CNN to identify the faults.

2.4 CNN Model Design

The architecture of the CNN as presented in Figure 1 below will be able to learn discriminatory features from the input signals compared to other machine learning methods that rely greatly on the use of linguistic rules to create feature functions for classification processes that have the tendency to decline due to their linear properties compared to the nonlinear behaviour of microgrid systems. Pre-processed and Z-score-normalized input currents such as Ia, Ib, and Ic of the three-phase currents will be presented to the CNN as two-dimensional signals.

The Convolutional layers turn the raw 3-phase current signal into multi-dimension Features Maps, which highlight local transient patterns that are indicative of faults. Through multiple conv layers, the model learns to identify low-level variations in signals and higher-level discriminative patterns to classify accurately the various fault types.

2.4.1 Convolutional Layer

Learnable kernels are used in the convolutional operation to extract local features from the input signal. The following is an expression for how a convolutional filter operates on a signal window:

$$h_{i,j}^{(k)} = f\left(\sum_{m=1}^{M}\sum_{n=1}^{N} W_{m,n}^{(k)} \bullet X_{i+m,j+n} + b^{(k)}\right) \tag{5}$$

Where X is the input signal segment, $b^{(k)}$ is the bias term, f $(\cdot)$ is the activation function (usually ReLU), $h_{i,j}^{(k)}$ is the feature map at position $(i,\ j)$ for k^{th} filter, and $W_{m,n}^{(k)}$ are the convolutional kernel weights. The CNN can identify abrupt changes or spikes in currents, which are indicative of fault situations, thanks to this process.

2.4.2 Activation Function

According to equation-1, nonlinearity is introduced in the model using the Rectified Linear Unit (ReLU). ReLU

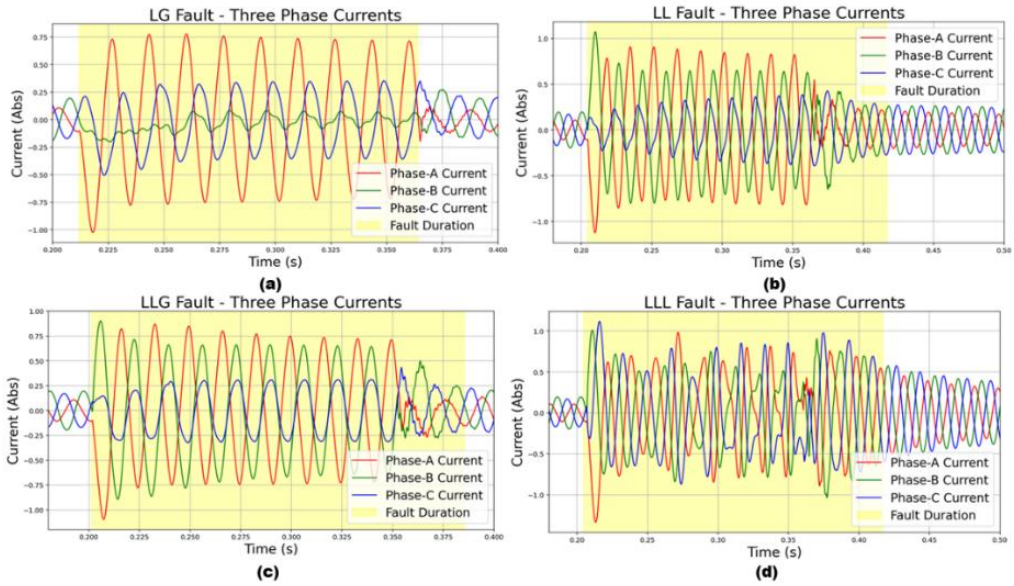

***Fig. 3**: Representation of faulty data set (a) LG, (b) LL, (c) LLG and (d) LLL fault.*

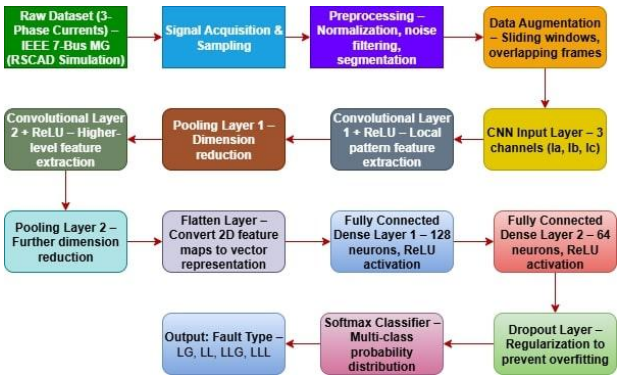

***Fig. 4**: Data Flow Pipeline for CNN-Based Fault Detection.*

generally aids in resolving the vanishing gradient problem, which speeds up the training phase's converging process.

2.4.3 Pooling Layer

Pooling layers minimize computing complexity while maintaining key features by reducing the spatial dimension of the feature maps. One popular strategy is max-pooling:

$$P_{ij} = \max_{(m,n) \in \Omega} h_{i+m,j+n} \tag{6}$$

In this case, the neighbourhood (window) over which pooling is done is defined by $(m, n) \in \Omega$, $h_{i+m,j+n}$ is the feature map input values at location $(i + m,\ j + n)$, and P_{ij} is the pooled output value at position $(i,\ j)$.

2.4.4 Fully Connected Layer

Higher-level fault patterns are transferred to the flattened feature vector:

$$z^{(s)} = f(W^{(s)} a^{(s-1)} + b^{(s)}) \tag{7}$$

The computation of the sth layer's output from the preceding layer is explained in equation 9.

Where $W^{(s)}$ is the weight matrix for layer s, $f\ (\cdot)$ is the activation function applied element-wise, and $z^{(s)}$ is the output (activations) of the sth layer. In order to change activation thresholds, it includes the learnable parameters that link layer $(s - 1)$ to layer s, $a^{(s-1)}$ input activations, and $b^{(s)}$bias vector for layer s, which are added to each neuron.

2.4.5 SoftMax Classifier

When multi-class classification is desired, CNNs (and other neural networks) frequently use the SoftMax function in their output layer. This function classifies multiple types of faults (LG, LL, LLG, LLL) at the output layer of the classification model and converts all of the network's output probabilities into normalized probability values for all class types.

2.5 Data Flow Pipeline for CNN-based Fault Classification

The complete process involved in the presented fault classification methodology is presented in Fig. 4, highlighting all the key steps involved in the process from data acquisition to final decision-making. This process concentrates strongly on the methodology involved in ensuring proper fault identification.

First of all, the three-phase current signal is extracted from the simulated IEEE 7-bus microgrid network on the RTDS platform under high-frequency sampling to maintain transient faults. In the pre-processing of the signals, the signals will be segmented, filtered, normalized by the Z-score method to reduce noise effects. To deal with the imbalanced problems, the machine learning model will be trained on different sets of data by different methods like injection of noise and shifting of signal.

2.6 Training and Hyperparameter Optimization

In essence, the performance of the developed CNN-based fault classifier not only relies on the architecture of the classifier itself but is also subject to the considered strategy for its training as well as the choice of corresponding hyper-parameters involved in the proposed method. To achieve an adequate performance level of the classifier in terms of the minimization of errors in the developed classifier, corresponding hyper-parameters are adequately adjusted.

The CNN utilizes 2 one-dimensional (1D) convolutional layers with 64 and 128 filters, each employing a kernel size of three and a stride value of one. The ReLU (Rectified Linear Unit) function provides the underlying non-linear activation function used in both 1D convolutional layers. Additionally, each one-dimensional convolutional layer has a subsequent max pooling and dropout layer to effect dimension reduction and reduce the likelihood of overfitting. Once convolutional feature maps have been created, they are flattened and passed through a fully connected (dense) layer composed of 100 neurons, which will create a class label via the sigmoid function to apply classification. The model is trained using an Adam optimizer and a learning rate of .001. The number of training epochs is set to 30 and the batch size is 32 in order to produce stable convergence and strong generalization capabilities.

2.6.1 Dataset Preparation for Training

The dataset contains three-phase current signals that have been segmented and encoded into six input channels—for every input sample, each has input dimensions of 100×6. The shapes will separate this dataset into training and test sets:

- Xtrain: (4720, 100, 6), Ytrain: (4720,)
- Xtest: (1180, 100, 6), Ytest: (1180,)

Here, six features encode absolute and normalized values across phases, and each training instance relates to a temporal window of currents with 100 samples. CNN kernels can capture localized transients and fault-induced distortions because of this structured representation, which maintains temporal evolution.

To train the CNN, mini-batch stochastic gradient descent (SGD) is used; the model weights are updated iteratively through smaller segments of the training data. The following parameters were used:

- Epochs (30): An epoch is one full run through the training program. A compromise between convergence and processing efficiency is achieved through training for 30 epochs.
- Batch size (32): Before updating weights at each stage, the network analyses 32 samples. This mini-batch method preserves computing speed while smoothing noisy gradients.
- Validation splits (0.2): To provide real-time overfitting monitoring, 20% of training data is excluded for validation.

The categorical cross-entropy loss function, which is defined as follows:

$$L = -\frac{1}{M}\sum_{i=1}^{M}\sum_{c=1}^{C} Y_{i,c}\log\left(\mathrm{P}\left(\mathrm{Y}=\mathrm{c} \mid \mathrm{X_i}\right)\right) \tag{8}$$

Where, $Y_{i,c}$ is the true label, $P\left(Y = c \mid X_i\right)$ is the CCN's SoftMax probability output, Xi is the ith input samples and M is the number of samples, while C is the number of classes.

2.6.2 Hyperparameter Optimization

The key to enhancing generalization and preventing underfitting or overfitting is hyperparameter adjustment. The following elements were adjusted in this study:

a) Learning Rate (η): During gradient descent updates, the learning rate regulates the step size.

$$W \leftarrow W - \eta\frac{\partial L}{\partial W} \tag{9}$$

Where L is the loss function, $\frac{\partial L}{\partial W}$ receives the gradient of the loss with respect to the weights, η is the learning rate, and W is CNN model's weights (parameters).
This research study will use a uniform learning rate (0.001) for the Adam optimizer to ensure that the model will have a stable & consistent convergence during training.

b) Batch Size: Studies comparing batch sizes of 16, 32, and 64 showed that 32 offered the best balance between computing cost and stability, reducing loss curve fluctuations.

c) Number of Epochs: According to training curves, accuracy peaked after 25 epochs, however overfitting and needless iterations were avoided by stopping early at 30.

d) Dropout Rate: Dropout layer (0.3) is applied to all of the convolutional layers between previous and next layers to decrease the risk of overfitting by randomly turning off neurons during training & increase the generalization capability of the model.

e) Activation Functions: Convolutional and fully linked layers used Rectified Linear Units (ReLU).

The enhanced CNN outperformed more conventional methods like SVM and Decision Trees, achieving nearly flawless classification accuracy. The hyperparameter selections, notably the combination of batch size, adaptive learning rate, and dropout, were important to attaining stability and generalization. The success of the training technique was confirmed by the final model's strong performance across all fault classes.

2.6.3 Cross-validation and Generalization

To test the CNN model's ability to generalize from the training data, it was subjected to a variety of modified training conditions (e.g., different load levels, different line impedances, and different fault locations). The

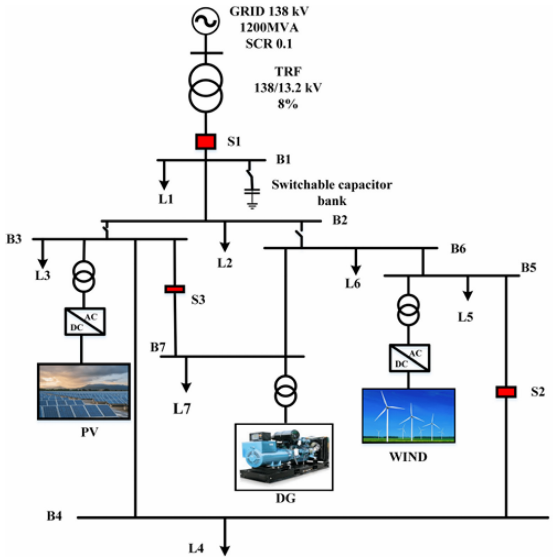

***Fig. 5**: IEEE 7-bus microgrid test system.*

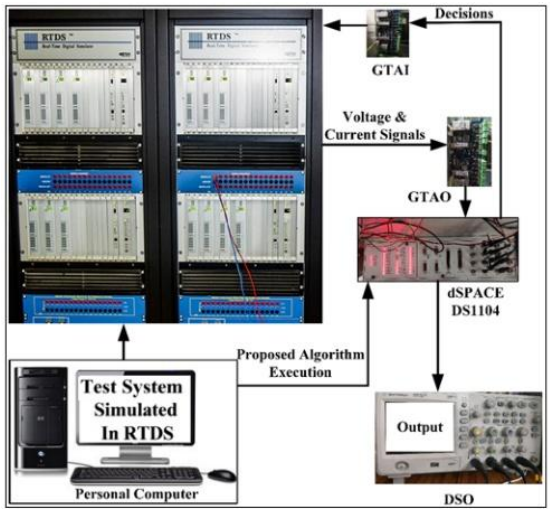

***Fig. 6**: RTDS microgrid test system.*

purpose of this type of testing is to confirm that the model has not overfitted the training data and can provide generalized results that apply to different configurations of microgrids.

2.7 System description

Figure 5 represents a medium-voltage AC microgrid connected to the utility grid via a 138/13.2 kV transformer, during weak grid conditions with low short-circuit ratio. The microgrid is represented as a distribution network with multiple distributed resources such as a solar cell-based renewable energy system, wind energy system, and a diesel generator connected to the microgrid through power electronic converters at various buses. Switching links such as breakers and feeders provide network flexibility to have radial or meshed network configurations as well as to switch seamlessly from grid-connected to islanded modes of operations. The presence of natural variability due to renewables, bidirectional power flow, as well as changing operational modes of the microgrid make this case study suitable to test robust fault schemes.

Figure 6 shows the RTDS-based real-time microgrid simulation environment used for testing the proposed algorithm. A simulation of a microgrid is created on the RTDS, and the actual RT signals for voltages and currents are connected to a dSPACE DS1104 Real-Time Controller to implement the proposed algorithm on a closed-loop Hard-In-the-Loop setup.

2.8 Machine Learning Methods

In order to verify the effectiveness of the proposed CNN model, its performance was compared against some well-established machine learning techniques that are regularly employed in fault classification problems, namely a Multilayer Perceptron (MLP), Decision Tree (DT), Support Vector Machine (SVM), and Naïve Bayes (NB).

2.8.1 Decision Tree (DT)

The Decision Tree (DT) is a supervised learning algorithm that uses hierarchical decision rules to partition the feature space. The predicted fault classes are assigned to the leaf nodes, and when stated for the internal nodes, it uses a split criterion, such as information gain or Gini index, to determine the best attribute to separate the data. Although DTs are easy to interpret and computationally efficient, they may overfit to the datasets when features are most overlapping or contain noise.

Decision trees provide a tree-like model of decisions by recursively dividing the dataset into smaller subsets according to attribute values. A split criterion, like the Gini Index or Information Gain, is used at every node. The Gini Index:

$$Gini\,(t) = 1 - \sum_{j=1}^{C} P_i^2 \tag{10}$$

Where C is the number of classes and P_i is the probability of class j for node t.

2.8.2 Support Vector Machine (SVM)

The basic principle of Support Vector Machines is to find a hyperplane within a high dimensional space that separates classes associated with possible faults and maximizes the margin between fault classes. SVM has the ability to handle non-linear decision regions using kernel mappings such as linear, polynomial, or radial basis function kernels. Although SVM is capable of effectively classifying data, tweaking the parameters of kernels and being computation intensive on large datasets is a major issue.

"Support Vector Machines" are a type of classifier based on margin classification using a hyperplane to obtain maximum margin between classes of data to

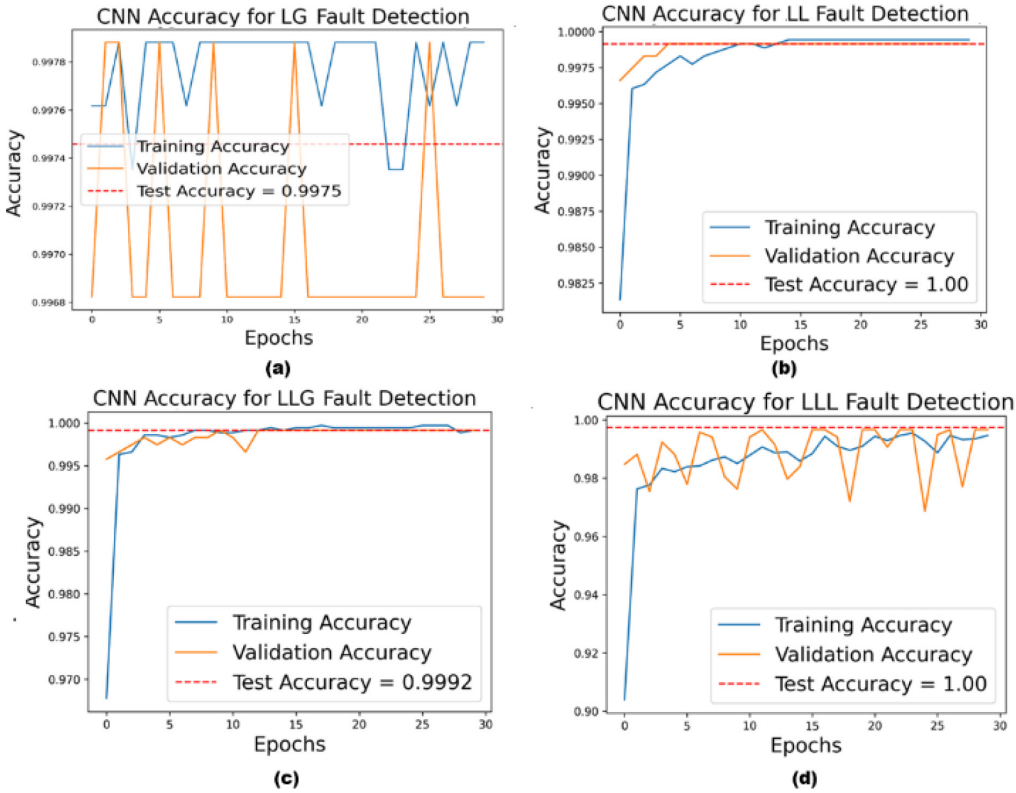

***Fig. 7**: Accuracy curves for training, validation, and test set of the proposed CNN for all fault classes and types: (a) LG; (b) LL; (c) LLG; (d) LLL.*

effectively separate them. The decision function of SVM for a linearly separable dataset is:

$$f(x) = sign(w \bullet x + b) \tag{11}$$

In above equation, x denotes the feature vector, w denotes the weight vector and b denotes bias.

The introduction of a kernel function $K(x_i, x_j)$ for non-linear classification

$$K(x_i, x_j) = \varphi(x_i) \bullet \varphi(x_j) \tag{12}$$

When the incoming data is transformed from the original space into a higher-dimensional space using the mapping function $\varphi(x)$. input feature vectors x_i, x_j, or two data samples.

2.8.3 Naïve Bayes (NB)

Assuming conditional independence between features, the Naïve Bayes classifier implements Bayes' theorem. Given the measured current or voltage features, it calculates the posterior probability of each class for fault classification. NB's independence assumption may oversimplify the intricate relationships found in power system problems, despite the fact that it is computationally light and works well on short datasets.

The foundation of naïve Bayes classifiers is the Bayes Theorem, which assumes conditional independence among features. For a feature vector $Y = (y1, y2, yn)$ and a class C_k:

$$P(B_k \mid Y) = \frac{P(Y \mid B_k)\ P(B_k)}{P(Y)} \tag{13}$$

Given the observed data, $(Y \mid B_k)$ is the likelihood where the probability of observing data Y if the class is B_k, and $P(B_k \mid Y)$ is the posterior probability and the chance that the sample Y belongs to class B_k. $P(Y)$ is the normalizing constant, and $P(B_k)$ is the prior probability.

2.8.4 Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP), a feedforward neural network used in foundational research comprises an input layer, one or more hidden layers with nonlinear activation functions, and an output layer. For training, backpropagation is applied in order to minimize classification error. MLPs are capable of modelling nonlinear features typical of defect indications, but their reliance on custom input features makes them less scalable compared to Convolutional Neural Networks (CNNs), which automatically learn hierarchical features from raw data. Each neuron in an MLP takes a weighted sum of its inputs and undergoes a nonlinear activation as defined in Equation 9. The final output layer utilizes Equation 2, a SoftMax function, to compute its output for multi-class classification.

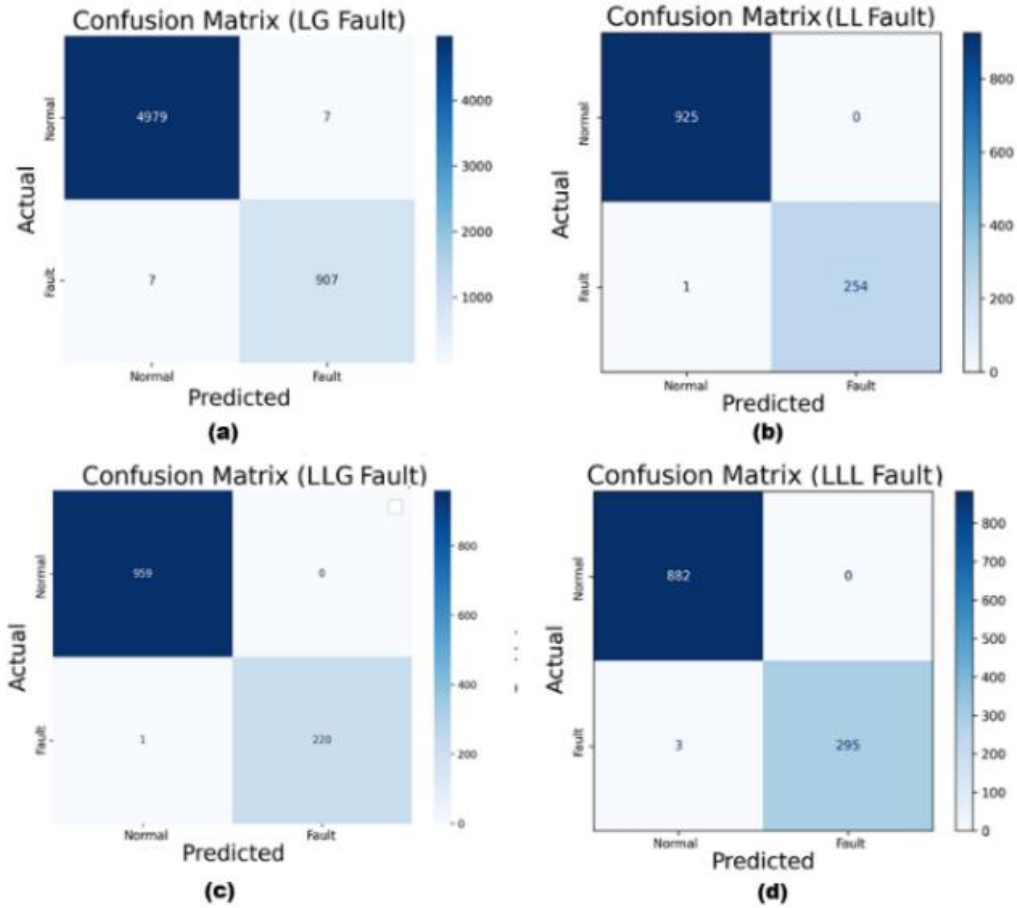

Fig. 8: Confusion matrices of the CNN classifier for (a) LG, (b) LL, (c) LLG, and (d) LLL fault scenarios.

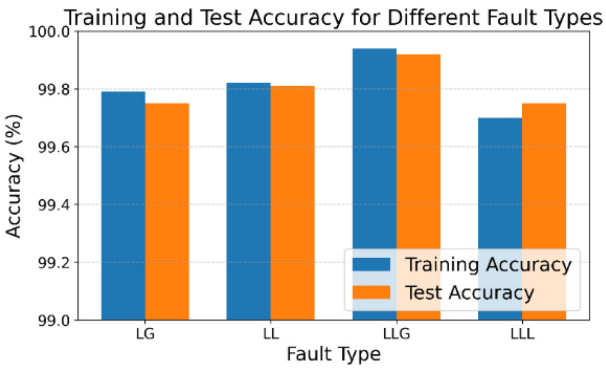

Fig. 9: Bar chart comparison of training and testing accuracy in the CNN model against the respective fault categories.

3. RESULTS AND DISCUSSION

Datasets depicting LG, LL, LLG, and LLL failures in an IEEE 7-bus microgrid were used to assess the suggested CNN model. Z-score normalized current signals were used as inputs during the 30-epoch, 32-batch training period.

Figure 7 training and validation accuracy curves demonstrate smooth convergence devoid of divergence, suggesting that overfitting was effectively reduced. CNN attained accuracies of 99.75% for LG, 99.81% for LL, 99.90% for LLG, and 99.75% for LLL faults, according to test findings, which support this finding. These results demonstrate a great capacity for generalization, as they nearly match training accuracy.

Figure 8 graphs all four of the confusion matrices for all four types of fault classes (LG, LL, LLG, and LLL). A high degree of concentration along the primary diagonal indicates an accurate classification of both fault classes and normal operating conditions, with virtually no misclassifications. Therefore, with the ability to robustly discriminate all four fault classes, and very few false negatives (0%) and false positives (0%), the proposed CNN model has the necessary integrity to detect both symmetrical and asymmetrical faults. Additionally, the confusion matrix indicates that misclassifications of closely related fault classes (LL and LLG), were insignificant. This reinforces the model's ability to detect very subtle differences in fault signatures, even when they are under similar transient waveform conditions; thus, increasing the reliability of the proposed fault classification framework.

Table 2 shows a summary of the training and testing accuracy obtained through the developed CNN model for these four types of faults. As indicated in Table 2, above 99 percent accuracy is obtained through this CNN

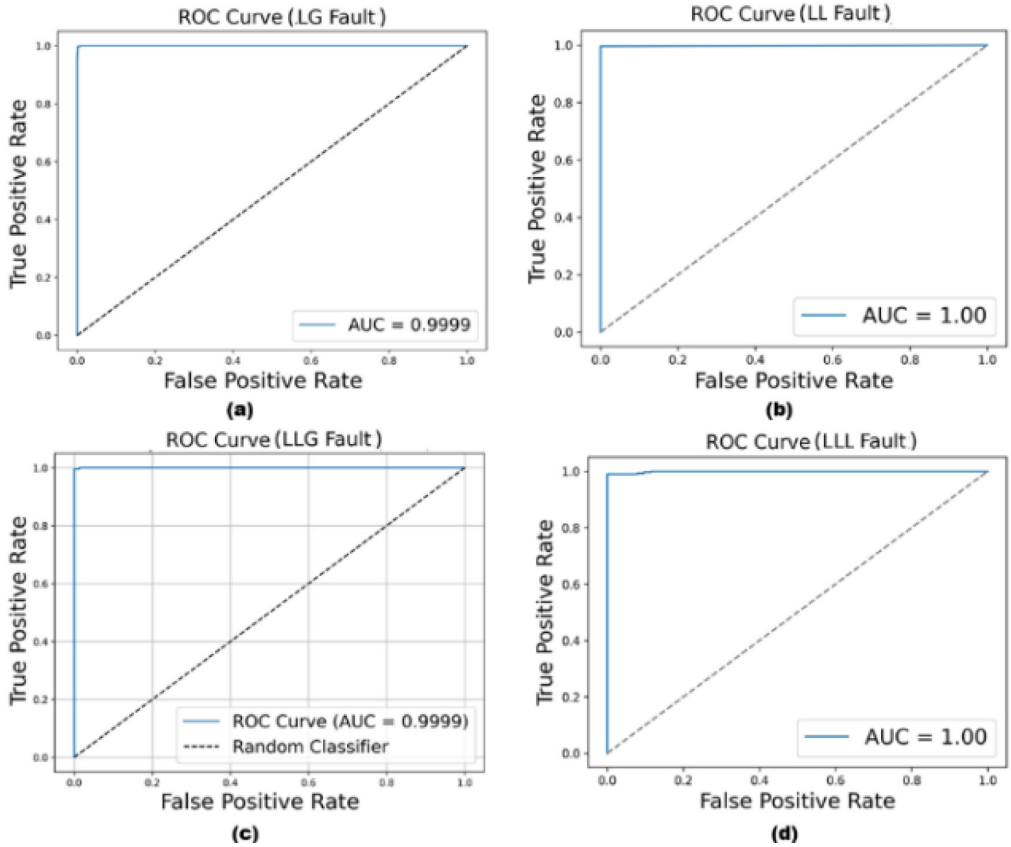

***Fig. 10**: ROC curves for each fault types: (a) LG; (b) LL; (c) LLG; (d) LLL.*

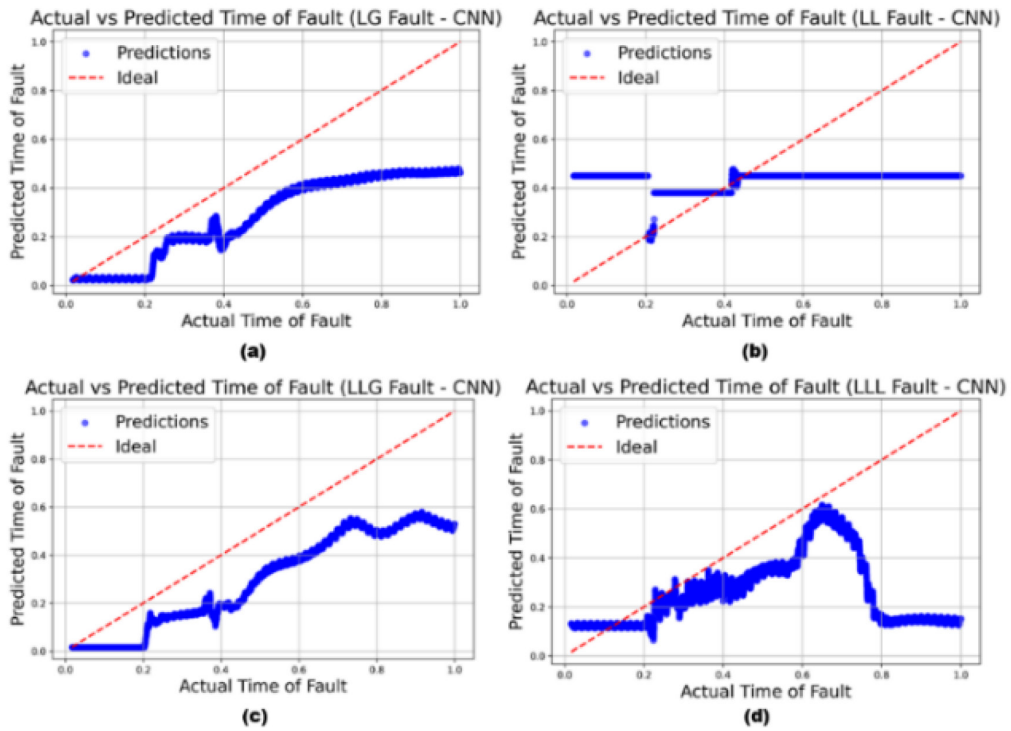

***Fig. 11**: Actual versus predicted fault occurrence times using the CNN model for (a) LG; (b) LL; (c) LLG; (d) LLL faults.*

model while training and testing. Furthermore, when such faults like LG and LL occur, balanced CNN model performance is generated because there is a minimum 0.1 percent differentiation between training and testing

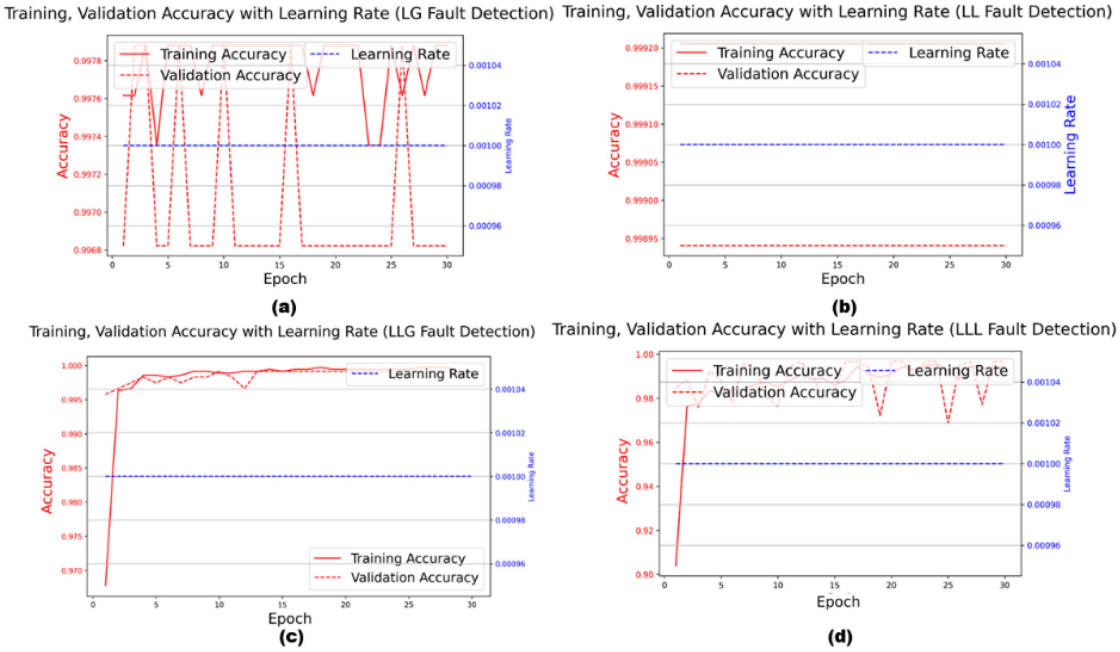

***Fig. 12**: Training and validation accuracy with learning rate scheduling for CNN-based fault detection: (a) LG; (b) LL; (c) LLG; (d) LLL.*

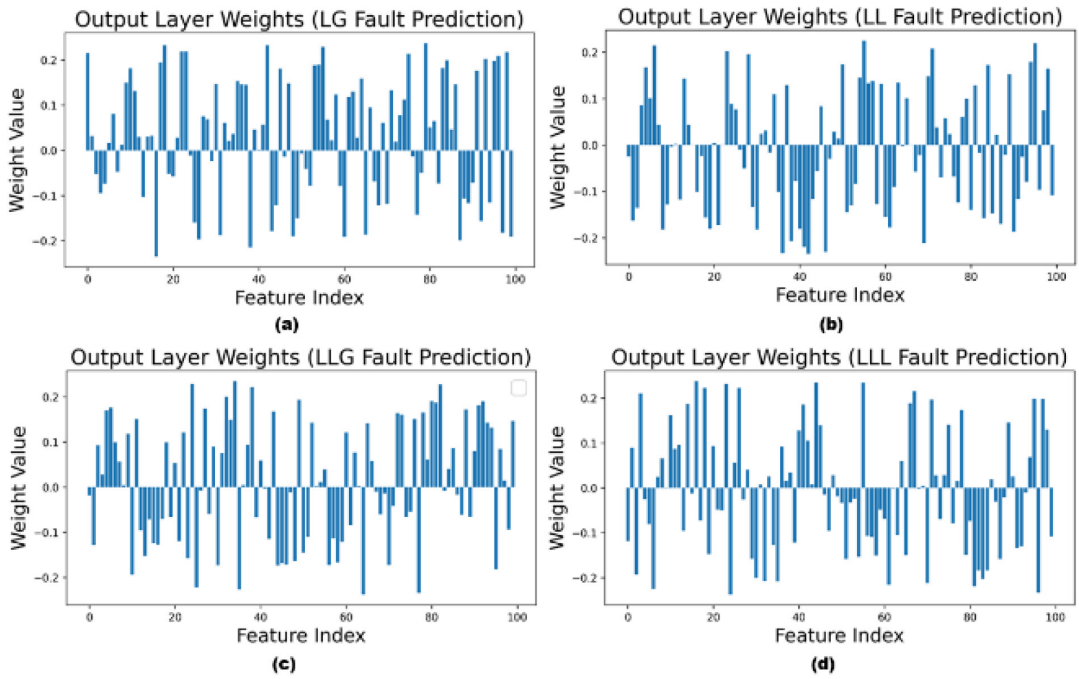

***Fig. 13**: Output layer weight distributions of the CNN model for different fault categories: (a) LG; (b) LL; (c) LLG; (d) LLL.*

***Table 2**: Testing & training performance of the proposed CNN model.*

Fault Type	Training Samples	Testing Samples	Training Accuracy (%)	Test Accuracy (%)
LG	1180	295	99.79	99.75
LL	1180	295	99.82	99.81
LLG	1180	295	99.85	99.92
LLL	1180	295	99.70	99.75

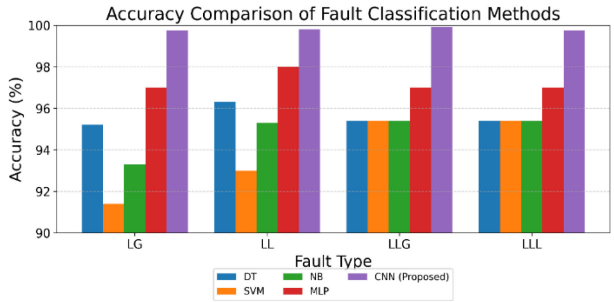

***Fig. 14**: Bar chart comparison of fault classification accuracy.*

accuracy. In contrast, when LLG faults happen, this developed model guarantees maximum test accuracy of 99.92 percent. In other instances when LLL faults occur, this developed model guarantees 99.70 percent and 99.75 percent for training and testing accuracy.

Further, a pictorial representation of this comparison has already been shown through the bar chart provided in Figure 9. It can be seen that the CNN almost provided the same output for both the training and testing sets in all the categories, proving the effectiveness of the proposed method in extracting discriminative features.

ROC curves were plotted for every defect type to further validate model performance, as seen in Figure 10. In all the scenarios, the ROC curve is along the boundary line in the top left corner, resulting in area under curve (AUC) values close to 1. This clearly explains that the model is performing impressively with respect to both true and false positive rates for all kinds of faults, whether symmetrical or asymmetrical.

The proposed CNN model's robustness was assessed in the presence of noise by injecting Gaussian noise into the test signals. The model obtained a classification accuracy of 99.75% at a 30 dB SNR and 99.15% at a 20 dB SNR, respectively, while its classification accuracy under a noise-free condition was 99.91%. Given the minimal degree of degradation in performance observed, it is apparent that this model provides reliable fault classification under noisy conditions. Furthermore, because this CNN has been able to learn invariant and discriminative features from distorted signals, it has been demonstrated that the model is robust for operation in a typical microgrid environment.

Figure 11 shows the combination of actual and predicted occurrences of faults by the CNN for LG, LL, LLG, and LLL faults. The concentration of predicted occurrences on the graph's diagonal confirms that there is strong correlation and accuracy in the prediction. Some minor non-linearities are well-explained by distortions introduced to the signals, current limiting, as well as the gradual nature that may be followed by the symmetrical faults. Based on the above discussion, the results are conclusive that the developed CNN method has accurately identified the type and timing of the faults. Figure 12 presents the training and validation accuracy trends, with a fixed learning rate for LG, LL, LLG, and LLL fault detection. The rapid convergence and the proximity of the training and validation curves indicate efficient learning with strong generalization. Minor validation fluctuations are attributed to transient fault dynamics, while the stable learning rate provides for smooth and reliable optimization in all fault categories.

Figure 13 Distribution of the weights learned in the output layer of the CNN for different classes of faults. The output layer weights for the CNN are represented in the above figure, which display the distribution of the weights for different classes of faults. These distributions show the boundaries for the decision made by the CNN on the extracted high-level features. In this case, the output layer weights of the CNN are not linked one-on-one to the input features of the signal; hence, they are not used as such in the interpretation of feature importance.

The benchmarking results shown in Table 3 highlight the superiority of the proposed CNN approach in dealing with all types of faults. The proposed CNN approach scores a maximum accuracy of 99.92% in the LLG fault set while maintaining an accuracy above 99.5% in LG, LL, and LLL fault types, which further signifies its superiority over other existing approaches.

Measurement of inference time per sample was utilized to evaluate the computational efficiency of the proposed CNN model. Results demonstrated that average execution time for each fault classification was 18.6 ms which meets the requirement of real time protection of less than 10-20 ms. Therefore, this provides sufficient assurance of the ability of the proposed CNN model to be deployed within a real-time protection system.

All models were trained and tested using the same dataset and identical preprocessing conditions to ensure a fair comparison. The results of the benchmark testing conducted on the proposed CNN method in Table 3 reveals that the proposed CNN method has proven to be superior at handling all four fault types. Specifically, the proposed CNN method achieved a high level of accuracy (99.92%) on the LLG fault data the mean accuracy values of the LG, LL, and LLL fault types are also over 99.50%, which indicates that the proposed CNN method is much better than other methods that have been used in practice. Although traditional classifiers have been used widely in classification tasks, recent work has also shown that some deep neural architectures (ResNet and attentional) work well with time series data. However, the majority of these models have a much greater computational cost and longer inference time than the proposed CNN. Despite this, the proposed CNN has similar or better classification accuracy with a lower computational overhead, making it more appropriate for use as a real-time protection system for microgrids.

The suggested CNN model exhibited high accuracy in classifying data but yet has some limitations. Measurement noise robustness was shown using Gaussian noise injected at differing signal-to-noise ratios; however, the real-world microgrid environment creates additional

***Table 3**: Accuracy benchmarking of conventional classifiers compared with CNN model.*

Fault Type	Decision Tree (DT) [20]	SVM [20]	Naïve Bayes (NB) [20]	MLP [20]	CNN (Proposed)
LG	95.2%	91.4%	93.3%	97.0%	99.75%
LL	96.3%	93.0%	95.3%	98.0%	99.81%
LLG	95.4%	95.4%	95.4%	97.0%	99.92%
LLL	95.4%	95.4%	95.4%	97.0%	99.75%

uncertainties from, for example, sensor inaccuracies, communication time delay(s), and changing operational conditions. In addition, high-impedance faults and different fault resistance values add complication to fault signatures; therefore, this can be especially problematic in inverter dominated systems because the fault current could be lower than normal. Additionally, the dynamic behaviour of IBR based distributed energy resources is complicated, and may not be represented accurately in a simulation data set. Therefore, future research/efforts will be made to extend the proposed CNN model through HIL validation and real-time deployment, which will allow for determination if it is a reliable process to use when operating in practical conditions.

4. CONCLUSION

This study used an IEEE modified 7-bus test system to validate the performance of a CNN-based framework for fault detection and classification in AC MGs. The suggested method replaces the manual feature engineering needed by traditional classifiers like DT, SVM, NB, and MLP by automatically extracting discriminative features from raw three-phase current signals. The comparative results show that CNN consistently performs better than conventional techniques, maintaining accuracy above 99.5% across all fault categories and reaching up to 99.9% classification accuracy for LLG faults. The suggested framework's suitability for real-time microgrid protection is demonstrated by the robust generalization capability and trustworthy confidence estimation offered by outputs. Future research will concentrate on hardware-in-the-loop validation for real-world implementation, high-impedance fault analysis, and hybrid deep learning models.

REFERENCES

[1] M. M. Zaben, M. Y. Worku, M. A. Hassan, and M. A. Abido, "Machine learning methods for fault diagnosis in ac microgrids: A systematic review," *IEEE access*, vol. 12, pp. 20260-20298.

[2] B. Deshmukh, D. K. Lal, and S. Biswal, "A reconstruction-based adaptive fault detection scheme for distribution system containing AC microgrid," *International Journal of Electrical Power Energy System*, vol. 147, p.108801, 2023.

[3] S. K. Maurya, R. K. Panda, A. Mohapatra and A. Sharma, "ΔV-ΔI Plane-Based Line Faults Detection and Classification in DC Microgrid," *IEEE Transactions on Power Delivery*, vol. 33, no.3, pp. 1755-67, 2022.

[4] Z. Ali, Y. Terriche, L. Q. N. Hoang, S. Z. Abbas, M. A. Hassan, M. Sadiq, C. L. Su, and J. M. Guerrero, "Fault management in DC microgrids: A review of challenges, countermeasures, and future research trends," *IEEE Access*, vol. 9, pp. 128032-128054, 2021.

[5] M. K. Ngwenyama, P. F. Le Roux, and J. L. Ngoma, "Conventional Method for Electrical Transmission System Fault Location Detection," in *6th Asia Conference on Power and Electrical Engineering*, pp. 69-76, 2021.

[6] X. Zheng, Y. Zeng, M. Zhao, and B. Venkatesh, "Early identification and location of short-circuit fault in grid-connected AC microgrid," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 2869–2878, 2021.

[7] M. A. Jarrahi, H. Samet, and T. Ghanbari, "A novel change detection and fault classification method for AC microgrids," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3987–3998, 2020.

[8] A. Arunan, R. Jayashri, and E. Ambikairajah, "Centralized voltage-signal-based fault detection and classification for islanded AC microgrids," In *2019 7th International Conference on Smart Grid (icSmartGrid)*, pp. 33–38, 2019.

[9] A. Srivastava, and S. K. Parida, "A robust fault detection and location prediction module using support vector machine and Gaussian process regression for AC microgrid," *IEEE Transactions on Industry Applications*, vol. 58, no.1, pp. 930–939, 2021.

[10] S. Wang and H. Chen, "A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network," *Applied Energy*, vol. 235, pp. 1126–1140, 2019.

[11] R. Pradhan, P. P. Agarwal, J. Behera, and S. Sen, "Intelligent Fault Diagnosis in AC Microgrids Using Short Time Fourier Transform (STFT) and Kernel-Based Algorithms," In *Fourth International Conference on Power, Control and Computing Technologies (ICPC2T)*, pp. 944-949, 2025.

[12] M. Faisal, K. Imran, H. Rehman, and H. A. Qureshi, "Hardware supported fault detection and localization method for AC microgrids using mathematical morphology with state observer algorithm," *IEEE Access*, vol. 12, pp. 12446–12457, 2024.

[13] N. Bouchiba, and A. Kaddouri, "Fault detection and localization based on Decision Tree and Support vector machine algorithms in electrical power transmission network," in *2nd IEEE International Conference on Advanced Electrical Engineering*, 2022.

[14] J. Ma, X. Zhang, and X. Peng, "Simulation evaluation of lightning and non-lightning fault identification of transmission line," in *2022 5th International Conference on Energy, Electrical and Power Engineering*, IEEE, pp. 452-458, 2022.

[15] S. N. V. Bramareswara Rao, Y. V. P. Kumar, M. Amir, et al., "Fault detection and classification in hybrid energy-based multi-area grid-connected microgrid clusters using discrete wavelet transform with deep neural networks," *Electrical Engineering*, vol. 107, no. 7, pp. 8481–8498, 2025.

[16] N. Peter, P. Gupta, and N. Goel, "Fault detection and identification of fault location in hybrid microgrid using artificial neural network," in *10th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 686-691, 2023.

[17] A. Bhalla, B. R. Bhalla, E. Purwar, D. Sosic, and Z. Stojanovic, "Detection, Classification and Localization of Faults in LVDC Microgrid Using ANN," in *2023 IEEE Belgrade PowerTech*, 2023.

[18] R. Pradhan, and P. Jena, "An innovative fault direction estimation technique for AC microgrid," *Electrical Power Systems Research*, vol. 215, Art. 108997, 2023.

[19] M. R. Zaidan, "Power system fault detection, classification and clearance by artificial neural network controller," In *2019 Global Conference for Advancement in Technology (GCAT)*.

[20] R. Pradhan, S. Mishra, A. Mahapatra, and A. Chhotaray, "Advanced fault detection and classification via fully connected neural network using multilayer perceptron in AC microgrid," *Sādhanā*, vol. 50, Art. 146, 2025.

[21] Y. Wang, B. Zhang, R. Wang, J. Desong, Y. Cui, Z. Sun, and H. Liu, "Fault reconfiguration control strategy of islanded marine ranching power supply system based on deep reinforcement learning," *International Journal of Electrical Power & Energy Systems*, vol. 169, Art. no. 110796, Aug. 2025.

Rakesh Sahu is pursuing a Ph.D. degree in Electrical Engineering at GIET University. He obtained his M. Tech degree in Power Electronics and Drives with First Class distinction Department of EEE, GIET University, Gunupur, Odisha, India. His research interests include power systems, fault detection and classification, renewable energy systems, artificial intelligence applications in electrical engineering, deep learning, and smart grid technologies. He has published several research papers in reputed journals and conferences related to electrical power systems and machine learning-based fault analysis.

Deepak Kumar Lal born in 1984 and received the B.Tech. Degree from the BPUT, Rourkela, Odisha, India in 2008 and M. Tech. Degree in Power System Engineering from Department of Electrical Engineering, National Institute of Technology, Jamshedpur, India in 2010. He finished the Ph.D. Degree in Electrical Engineering from Veer Surendra Sai University of Technology, Burla, Odisha, India in 2018. Since 2011, he is working as Assistant professor in the Department of Electrical Engineering, Veer Surendra Sai University of Technology, Burla, Odisha, India. His research interests include automatic generation control, power system protection, distribution system operation & control and renewable energy integration.

Pratap Kumar Panigrahi is working as Professor in the Department of Electrical & Electronics Engineering, GIET University, Gunupur, Odisha, India. His main research interests are in smart grids, distributed generation theory and practice fields, power system stability, protection schemes and advanced computational techniques. He has delivered supervision of numerous undergraduate level and postgraduate research projects.

Rudranarayan Pradhan (Member, IEEE) received the MTech degree in Power System Engineering from VSSUT, Burla, Odisha, India, in 2009 and a Ph.D. degree in Power System Engineering from the Department of Electrical Engineering, Indian Institute of Technology, Roorkee, India, in the year 2024. Since 2013, he has been an Assistant Professor at the School of Electrical Sciences, Odisha University of Technology and Research Bhubaneswar, India. His research interests include power system protection, microgrid, advanced relaying techniques, signal processing applications for power system relaying, and issues of renewables integration to power system.