Overview of Fault Analysis in 220-kV Transmission Network Using Deep Learning

a Project Report

submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

by

Gunjan Kumar

(Enrolment No. 211020012008)

Supervisor

Dr. K. Manjunath

Department of Electrical and Computer Science & Engineering

Institute of Infrastructure, Technology, Research and Management,

Ahmedabad, Gujarat - 380026

May 2025

DECLARATION BY STUDENT

I declare that this written submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: May 4, 2025

Place: Department of Electrical and Computer Science Engineering, IITRAM, Ahmedabad

Gunjan Kumar

Enrolment No. 211020012008

APPROVAL CERTIFICATE

This project report entitled "**Overview of Fault Analysis in 220-kV Transmission Network Using Deep Learning" submitted by Mr. Gunjan Kumar, Enrolment No.** 211020012008 is approved for acceptance in the Department of Electrical and Computer Science & Engineering, at the Institute of Infrastructure, Technology, Research and Management, Ahmedabad.

Examination Committee

Dr. Manjunath K. (**Supervisor**)
Assistant Professor,
Dept. of Electrical and Computer Science & Engineering,
IITRAM, Ahmedabad

Dr. Raghavendra H. Bhalerao (**Member- I**)
Assistant Professor,
Dept. of Electrical and Computer Science & Engineering,
IITRAM, Ahmedabad

Dr. Krupa Shah (**Member-I I**)
Assistant Professor,
Dept. of Electrical and Computer Science & Engineering,
IITRAM, Ahmedabad

ACKNOWLEDGEMENTS

First and foremost, I want to sincerely thank Dr. K. Manjunath, my supervisor from the Division of Electrical Engineering, for his strong leadership, priceless counsel, and unfailing support during this project. His knowledge and experience have greatly influenced the direction and caliber of my work.

Additionally, I want to express my gratitude to the entire professors and staff of IITRAM, Ahmedabad's Electrical Engineering Department for giving me the tools I needed and a comfortable setting in which to conduct my research.

For their unwavering love, support, and encouragement, I am so grateful to my guardians for their unwavering support, love, and back. Their sacrifices and faith in my abilities have always motivated me to succeed. I would want to express my sincere gratitude to my friends and coworkers, who have been a tremendous source of support and friendship. Their prompt conversations and suggestions have enhanced my job and added enjoyment to this trip. Lastly, I want to express my gratitude to everyone who directly or indirectly helped achieve this extent of success. This task would not have been feasible without your offer of help.

DEDICATION

This section is dedicated to my parents, whose unwavering encouragement, love, and faith in my abilities have continued to inspire me throughout my academic career. I also dedicate this work to my teacher, Dr. K. Manjunath, whose guidance, expertise, and vital tidbits of knowledge allowed me to finish it. Finally, I want to thank all of my friends and coworkers for their unwavering support, insightful advice, and for helping to make this trip unforgettable. Without you all, this achievement would not have been possible.

ABSTRACT

This report presents the development and evaluation of an advanced deep learning-based Fault Assessment System for a 220 kV power transmission line. The system is designed to automatically detect and classify faults such as no fault, single line-to-ground (LG), double line (LL), line-to-line-to-ground (LLG), and three-phase faults (LLL) using real-time voltage and current signal data.Multiple deep learning architectures including LSTM, FNN, MLP, FCN, and 1D CNN are trained and validated to perform accurate fault classification.

The project utilizes an expansive dataset of electrical signals generated from a Simulink-based simulated power system. These signals are preprocessed and divided into training and testing sets. Model performance is evaluated using metrics such as accuracy, precision, recall, F1-score, confusion matrix, and cross-validation techniques. Real-time capability is further enhanced by integrating the trained model into Simulink with a Graphical User Interface (GUI), enabling operators to monitor system behavior and interact with fault classifications in real time.

The results demonstrate the potential of deep learning-based approaches in improving fault detection accuracy and enhancing the overall reliability of power transmission systems. This research contributes to the growing field of predictive maintenance and intelligent fault management, with significant implications for real-time monitoring and decision-making in modern power grids.

Keywords: Power System Fault Classification, Deep Learning, 220 kV Transmission Line, Continuous Wavelet Transform (CWT), Real-Time Monitoring, LSTM, FCN, Predictive Maintenance, MATLAB Simulink.

Abbreviations

CWT	Continuous Wavelet Transform
FCN	Fully Convolutional Network
FNN	Feedforward Neural Network
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
1D CNN	One-Dimensional Convolutional Neural Network
GUI	Graphical User Interface
IITRAM	Institute of Infrastructure Technology Research and Management
LL	Line-to-Line Fault
LG	Line-to-Ground Fault
LLG	Line-to-Line-to-Ground Fault
LLL	Three Phase Fault

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Electric power systems are critical infrastructure for the functioning of modern society and play a key role in the reliable delivery and distribution of electricity. However,malfunctions, such as short circuits, insulation breakdowns, and ground faults, can result in serious disturbances, damage, and power interruptions. Correct and prompt fault diagnosis is key for grid stability, operational safety and service preservation.

Conventional fault detection methods by means of fixed thresholds, phasor quantity and rule-based logic are no longer in favour. These approaches are frequently not so suitable for nonlinear and time-varying faults, when the modern power grid is considered, and noise or load change is introduced.

Objective of the Study: This project presents a deep learning-based framework for multi-class fault classification in 220-kV power transmission lines. The framework is capable of identifying the following fault types:

- No Fault (NF)
- Line-to-Ground Fault (LG)
- Line-to-Line Fault (LL)
- Line-to-Line-to-Ground Fault (LLG)
- Line-to-Line-to-Line Fault (LLL)

The method uses a variety of deep learning architectures (1D CNN, FCN, FNN, LSTM and MLP) for accurate fault classification in a 220kV Transmission Line. The key goals are:

- To automate fault classification using deep learning with minimal manual feature engineering.
- To build a generalized fault diagnosis model that performs well under varying fault conditions.
- To deploy the trained model into a MATLAB-based GUI for real-time decision support.

The integration of time-frequency feature extraction and CNN classification enables accurate, fast, and scalable fault diagnosis suitable for real-world deployment.

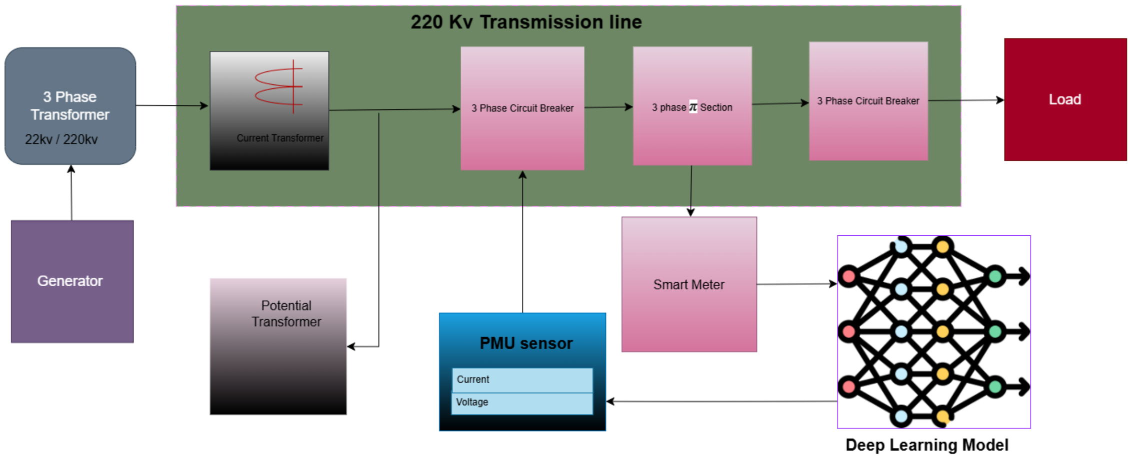

Figure 1.1: Block Diagram of Fault Assessment using DL Model

Chapter 2

Literature Review

Recently, considerable effort has been made for intelligent systems to implement a fault-detection and classification (FDC) technology in power transmission network systems. With the large size and complex structure of power grids, the traditional methods are limited in speed, applicability, and real-time fault diagnosis. This has resulted in a shift of the paradigm towards data-driven and deep learning approaches that improve accuracy, scalability and operational efficiency of the diagnoses.

2.1 Signal Processing Techniques for Fault Analysis

Wavelet techniques have achieved widespread use in fault diagnosis by those characteristics of their capturing transitory oscillations in non-stationary signals. The CWT can further transform the voltage and current waveform into time-frequency domain which allows the detector to identify fault-stimulated patterns

The CWT of a signal $x(t)$ is mathematically defined as:

$$W_x(a, b) = \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t - b}{a} \right) dt$$

where $\psi(t)$ is the mother wavelet, a is the scaling factor, and b is the translation parameter.

Such spectrograms, when treated as inputs of ML-based models, facilitate more effective feature learning that retains the temporal and frequency characteristics associated with fault occurrences.

2.2 Deep Learning Approaches in Power system Fault Classification

2.2.1 1D-CNN Models

1D Convolutional Neural Networks have shown effectiveness in learning from time-series data like voltage and current waveforms. They automatically extract hierarchical features through convolution operations and reduce reliance on manual preprocessing.

A 1D-CNN performs convolution as:

$$(y * w)(t) = \sum_{\tau} x(t - \tau)w(\tau)$$

where $x(t)$ is the input, $w(\tau)$ is the kernel, and $y(t)$ is the output feature.

2.2.2 FNN Models

Feedforward Neural Network (FNN) is the most basic artificial neural networks with a directional flow from input units to output units without having cycles in the connections between the units. They work great on simple static data (eg., fixed-size input vectors) and they are easy to train on classification using supervised learning.

2.2.3 MLP Models

Multi-Layer Perceptrons (MLP) is a kind of feedforward network with one or several layers in the middle. Each layer is made up of non-linear neurons, so the network is able to learn complex patterns. MLPs are computationally lightweight and are very good in processing operation in real-time.

2.2.4 LSTM Models

LSTMs (Long Short-Term Memory) networks arespecial kinds of RNN that are capable of learning long-term dependencies in sequential data. They are very useful for time-series processes (e.g. variations in system voltages/currents with fault conditions etc) for which accurate recognition of temporal patterns are required.

2.2.5 FCN Models

Fully Convolutional Networks Fully Convolutional Networks(FCNs) contain no fully con- nected layers. Because of this, spatial or temporal relationships in the data can be preserved, they are thus good at sequence-to-label problem. For fault classification, FCNs use temporal convolution to capture fault signatures as soon as possible.

Chapter 3

Methodology

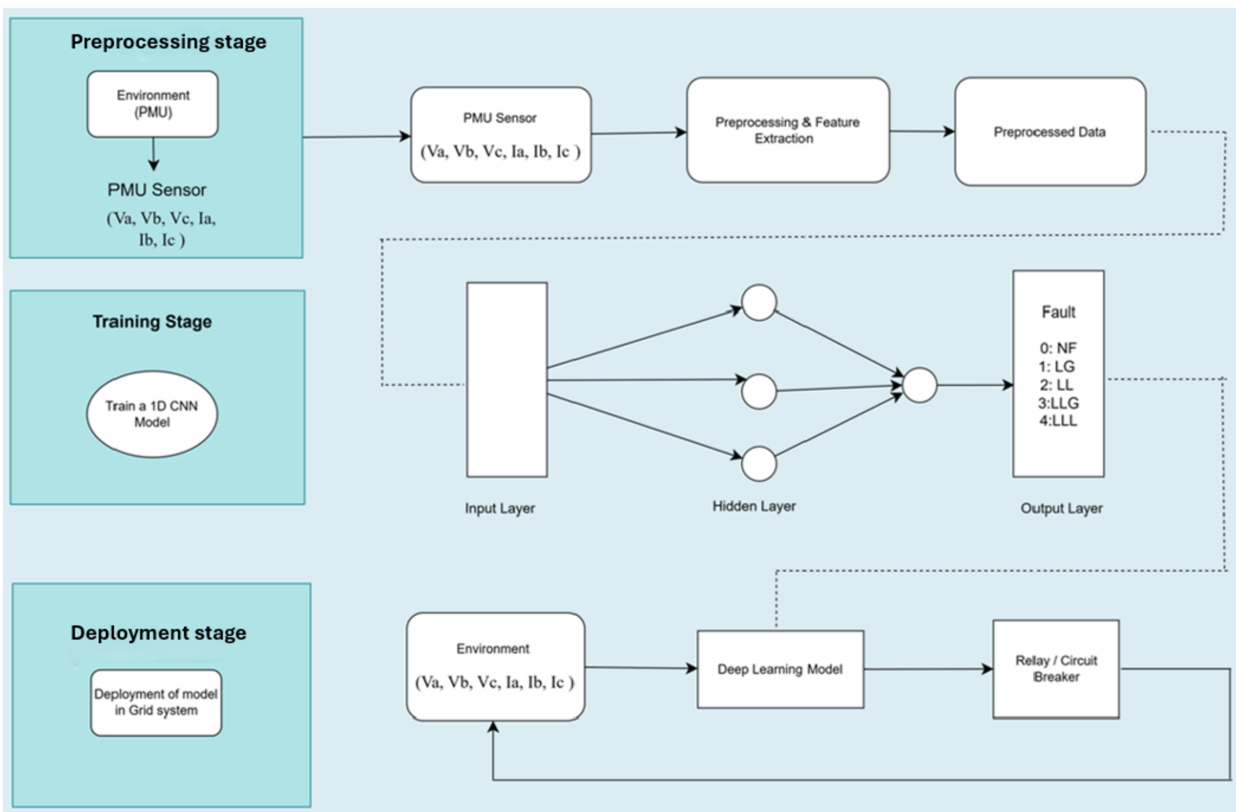

Figure 3.1: Flow chart Diagram of Fault Assessment using DL Model

3.1 Data Collection and Preprocessing

3.1.1 Experimental Setup in MATLAB

The experimental setup for fault detection in an 220 kV power system was modeled using MATLAB/Simulink, as shown in Figure **??** The system consists of multiple components designed to simulate real-world conditions for fault analysis. Key elements include:

- **Three-Phase Source and Transformer:** The system includes two transformers with two windings to step up and step down voltages across the transmission line.
- **Three-Phase Fault Block:** This block enables controlled simulation of various fault conditions, such as NF, LG, LL, LLG, and LLL faults, at predefined locations on the transmission line.
- **Voltage and Current Measurement:** Voltage and current measurements are taken at critical points using the Three-Phase VI Measurement block to capture transient behaviors during fault events.
- **Transmission Line:** The transmission line is represented by a PI Section Line model to emulate realistic power line dynamics.
- **Data Logging:** The measured voltage and current signals are sent to workspace variables for further processing and analysis.

The setup allowed for synchronized data collection under diverse fault scenarios, facilitating precise labeling for training the deep learning-based fault classification model.

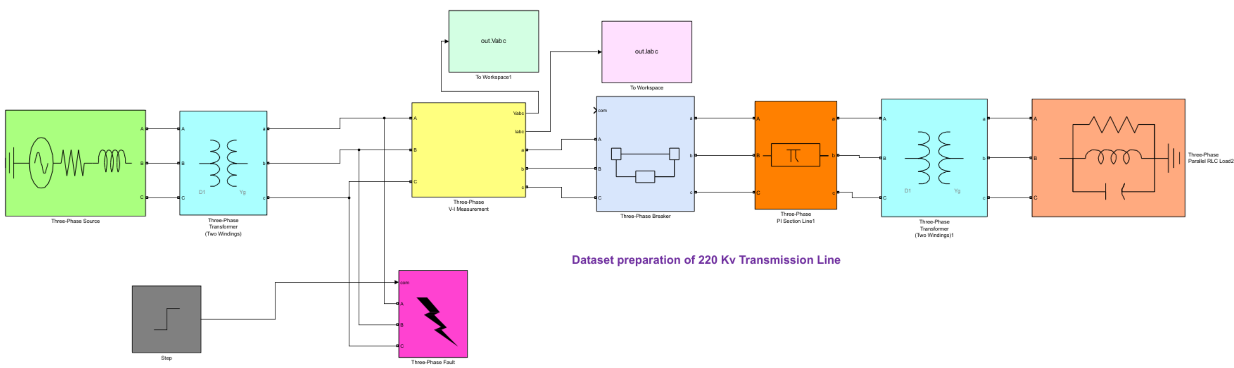

Figure 3.2: MATLAB Simulink Model for Fault Detection in an 11 kV Power System.

3.2 System Specification

3.2.1 Three-Phase Source

- **Configuration:** Y-grounded
- **Peak Phase Voltage (Vm):** 25 kV
- **Base Voltage (Vbase):** 25 kV (phase-to-phase)

- **Frequency:** 50 Hz
- **Impedance:**(R_s): 0.8929 Ω & (L_s): 16.58 mH
- **Generator Type:** Unity Power Factor

3.2.2 Three-Phase Transformers

Transformer 1

- Winding 1 Connection (ABC terminals): Delta (D1)
- Winding 2 Connection (abc terminals): Y-grounded
- Power Rating: 250 MVA
- Voltage Rating: 22 kV / 220 kV
- Impedance: $R_1 = 0.002\,\text{p.u.}$, $L_1 = 0.08\,\text{p.u.}$

Transformer 2

- Winding 1 Connection (ABC terminals): Delta (D1)
- Winding 2 Connection (abc terminals): Y-grounded
- Power Rating: 250 MVA
- Voltage Rating: 220 kV / 33 kV
- Impedance: $R_1 = 0.002\,\text{p.u.}$, $L_1 = 0.08\,\text{p.u.}$

3.2.3 Transmission Line (Three-Phase π-Section)

- Line Length: 100 km
- Frequency: 50 Hz
- Positive and Zero Sequence Impedance (per km):

  Resistance: $R_1 = 0.01273\ \Omega/\text{km}$, $R_0 = 0.3864\ \Omega/\text{km}$ Inductance: $L_1 = 0.9337\ \text{mH}/\text{km}$, $L_0 = 4.1264\ \text{mH}/\text{km}$ Capacitance: $C_1 = 12.74\ \text{nF}/\text{km}$, $C_0 = 7.7519\ \text{nF}/\text{km}$

3.2.4 Fault Module

- Fault Type: Three-Phase Fault
- Fault Resistance: 0.01 Ω
- Ground Resistance: 10 Ω
- Snubber Resistance: 1 mΩ
- Snubber Capacitance: ∞

3.2.5 Loads

- Configuration: Y-grounded
- Nominal Phase Voltage: 33 kV
- Active Power (P): 220 MW
- Inductive Reactive Power (Q_L): 22 kVAR
- Capacitive Reactive Power (Q_C): 22 kVAR

3.2.6 Measurement System

- Voltage Measurement: Phase-to-ground
- Current Measurement: Per phase

3.3 Dataset Parameter

3.3.1 Dataset Details

Fault Type	**Time Stamp (s)**	**No. of Data Points**	**Fault Label**
NF (No Fault)	0 – 89.9999	2,29,898	0
LG (ag)	90 – 119.9999	76,941	1
LG (bg)	120 – 149.9999	77,225	1
LG (cg)	150 – 179.9999	54,602	1
LL (ab)	180 – 209.9999	87,777	2
LL (bc)	210 – 239.9999	83,309	2
LL (ca)	240 – 269.9999	63,635	2
LLG (abg)	270 – 299.9999	60,559	3
LLG (bcg)	300 – 329.9999	66,789	3
LLG (cag)	330 – 359.9999	60,720	3
LLL (abc)	360 – 449.9999	220,165	4
Total	–	**1,081,620**	–

3.3.2 Dataset Fault scenario

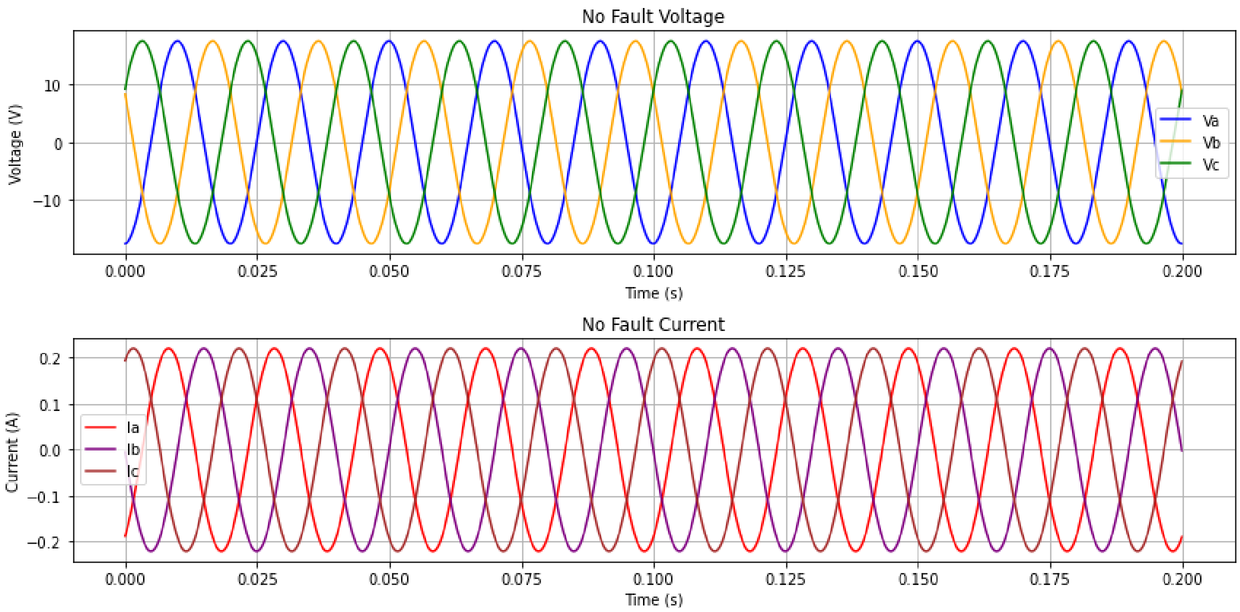

Figure 3.3: No Fault

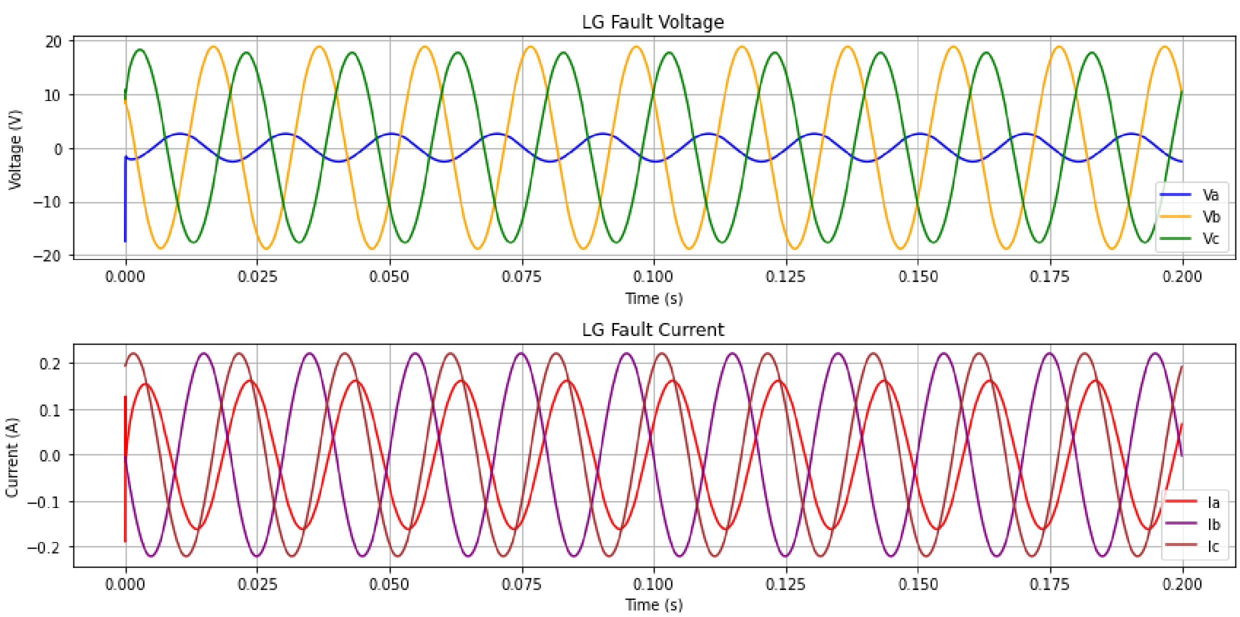

Figure 3.4: LG Fault

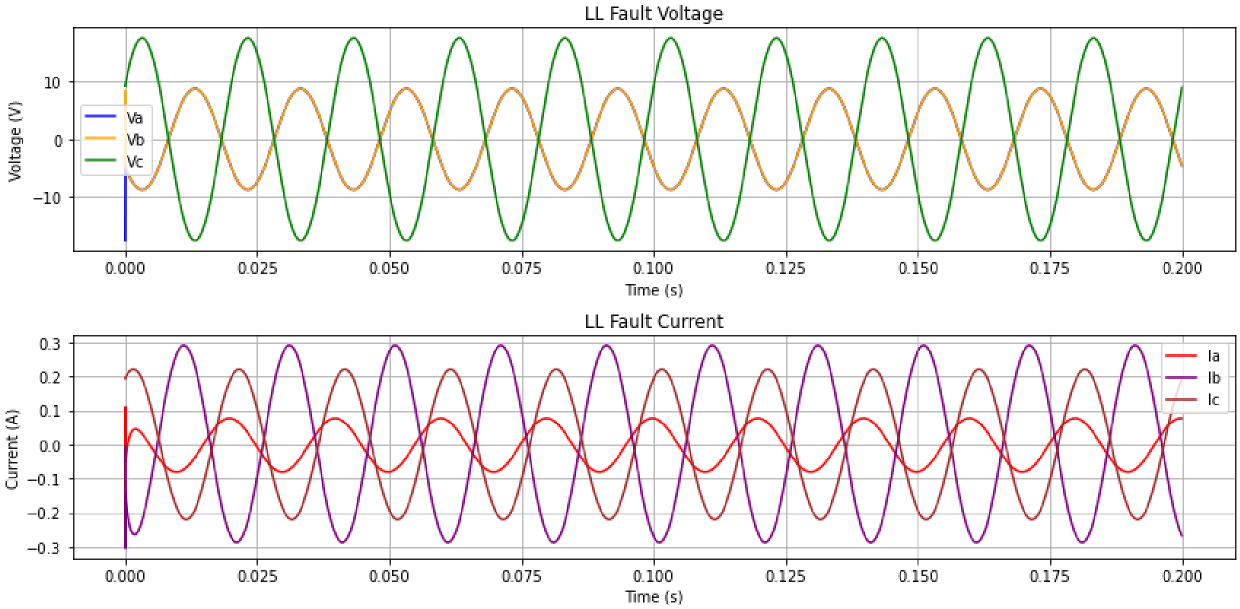

Figure 3.5: LL Fault

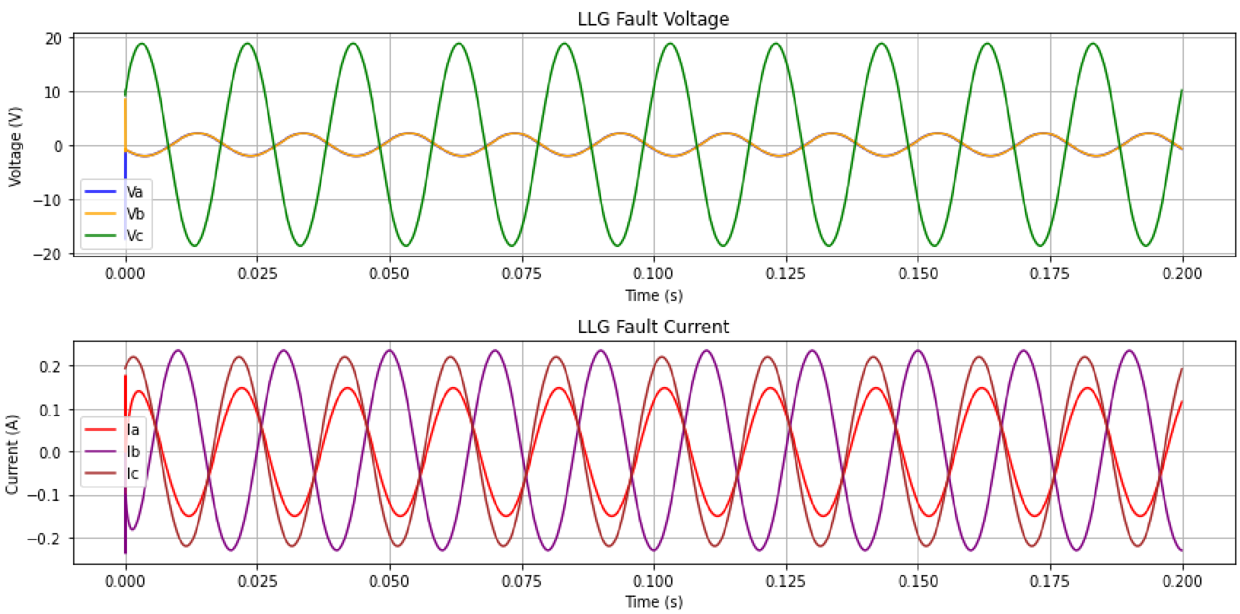

Figure 3.6: LLG Fault

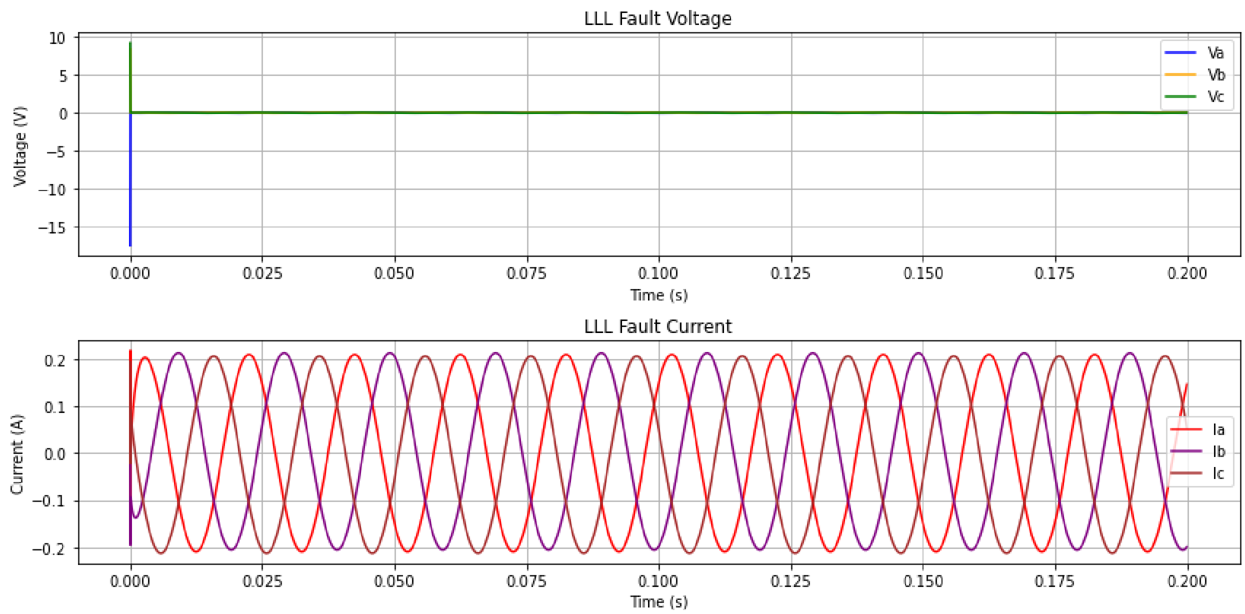

Figure 3.7: LLL Fault

3.4 Data Quantity and Labeling

he data set was thoughtfully prepared and consisted of over 11,00,00 information centers. Each information point was classified as concurring with certain Fault conditions from the reenactment time intervals. Fault labeling system was represented as follows:

- **0 – 89.9999 s**: No-Fault
- **90 – 179.9999 s**: LG Fault
- **180 – 269.9999 s**: LL Fault
- **270 – 359.9999 s**: LLG Fault
- **360 - 449.9999 s**: LLL Fault

This precise labeling was crucial for training and assessing the fault classification model effectively.

3.4.1 Data Preparation

3.4.1.1 Data Preprocessing with Pandas

The dataset, stored in `.mat` format, was initially loaded and processed using the `scipy.io` library in Python. The structured array extracted from the `.mat` file was then converted into a Pandas `DataFrame` for efficient data manipulation and cleaning. This preprocessing ensured the dataset was appropriately structured and ready for further analysis and model training.

3.4.1.2 Handling Missing Data

Colossal columns which have NaN (Not-a-Number) values were identified and removed from the dataset. This data cleaning process ensured the veracity and integrity of the dataset, reducing the chances of errors while conducting the modelling.

3.4.1.3 Data Splitting into Training and Testing Sets

Model's performance was assessed by dividing dataset into specific training and testing subsets. A portion ratio of train-test-validation part was 80:10;10, where 80% of the data were used to train the model, 10% was used as the test set, and a further 10% was set aside for validation.The preparing set was utilized to prepare the profound learning show, whereas the testing and approval set surveyed the model's generalization to concealed data.

3.4.1.4 Multi-Class Conversion

For multi-class classification, the output data were modified to ensure compatibility with the softmax function. This comprised labeling Fault categories as numbers or one-hot vectors, prepping the info to be used with the multi-class classification modell

3.5 Feature Extraction

Feature extraction is an automated procedure that uses five deep learning models—1D Convolutional Neural Network (1D-CNN), Long Short-Term Memory (LSTM), Multilayer Perceptron (MLP), Fully Convolutional Network (FCN), and Feedforward Neural Network (FNN)—to extract discriminative features from raw vibration signals. These models, which addressed a particular failure pattern without the use of handmade features, were trained using time-domain forward segments from the CWRU bearing dataset after processing.

3.5.0.1 1D-CNN for Feature Extraction

1D-CNN architecture was featured in order to learn local temporal dependencies across time in the signal. The architecture consisted of multiple convolutional and pooling layers, which made it possible for to extract local fault features such as sharp transients and periodic variations in an automatic fashion. This enabled to differentiate between normal, inner race fault, outer race fault, and ball fault.

3.5.0.2 LSTM for Sequential Pattern Learning

A Long short-term memory (LSTM) was used to learn long-term temporal dependency from the vibration data. Its memory cell design enabled the model to learn an evolutionary path of the signal pattern over time, enhancing the model sensing capability for slight variations of the signal pattern corresponding to diverse failure types

3.5.0.3 MLP for Hierarchical Feature Abstraction

A basic MLP architecture was utilized as a baseline model. It was composed of deep layers encoding non-linear combinations of input features Though it lacked spatial or sequential awareness, MLP still provided competitive results due to its ability to model general feature interactions.

3.5.0.4 FCN for End-to-End Representation Learning

The FCN model, composed entirely of convolutional layers without any fully connected layers, was used to preserve the temporal resolution of input signals while learning hierarchical features. Its global average pooling layer directly linked feature maps to class outputs, promoting efficient learning.

3.5.0.5 FNN for Direct Time-Domain Classification

The Feedforward Neural Network (FNN) was trained on raw signal values as input. It provided a straightforward architecture to benchmark the effectiveness of more complex deep learning models. Although FNN does not explicitly model sequence or spatial relations, it still contributed to the ensemble's diversity.

3.5.1 Feature Representation and Mapping

To automate the process of extracting discriminative features from raw vibration signals, five deep learning architectures were implemented: 1D Convolutional Neural Network (1D-CNN), Long Short-Term Memory (LSTM), Multilayer Perceptron (MLP), Fully Convolutional Network (FCN), and Feedforward Neural Network (FNN).

3.5.1.1 1D-CNN for Feature Extraction

The 1D-CNN model was used to capture local temporal dependencies in the signal. The architecture included multiple convolutional layers followed by pooling layers, which enabled automatic extraction of localized fault features such as sharp transients and periodic variations. This helped distinguish between conditions like normal, inner race fault, outer race fault, and ball fault.

3.5.1.2 LSTM for Sequential Pattern Learning

The LSTM model was employed to capture long-term temporal dependencies in the vibration data. Its memory cell architecture made it effective in learning the progression of signal patterns over time, improving the model's sensitivity to subtle changes indicative of different fault types.

3.5.1.3 MLP for Hierarchical Feature Abstraction

A simple MLP architecture was applied to serve as a baseline model. It consisted of dense layers that learned nonlinear combinations of input features. Though it lacked spatial or sequential awareness, MLP still provided competitive results due to its ability to model general feature interactions.

3.5.1.4 FCN for End-to-End Representation Learning

The FCN model, composed entirely of convolutional layers without any fully connected layers, was used to preserve the temporal resolution of input signals while learning hierarchical features. Its global average pooling layer directly linked feature maps to class outputs, promoting efficient learning.

3.5.1.5 FNN for Direct Time-Domain Classification

The Feedforward Neural Network (FNN) was trained on raw signal values as input. It provided a straightforward architecture to benchmark the effectiveness of more complex deep learning models. Although FNN does not explicitly model sequence or spatial relations, it still contributed to the ensemble's diversity.

Chapter 4

Model Architecture

This chapter details the architecture of the proposed model for real-time fault classification in a 220-kV transmission system. The selected deep learning architecture is a Fully Convolutional Network (FCN), known for its efficiency in time-series data modeling and suitability for real-time deployments.

The combination of CWT-based highlight extraction, CNN-based design, and cautious preparing methodologies guaranteed the models capability to precisely distinguish and classify issues within the control framework.

4.1 Model Initialization

The FCN model was constructed using TensorFlow/Keras in a sequential manner. It leverages multiple Conv1D layers to extract temporal features from the six-phase voltage and current signals. The design avoids any fully connected layers until the output stage, enabling better generalization and computational efficiency.

4.1.1 Input Layer

The model receives a 1D input of shape (6, 1), representing six time-step samples across one feature channel. These correspond to preprocessed voltage and current measurements from the transmission line. Input preprocessing involves:

$$Y[i,j] = \sum_{m=1}^{M} \sum_{n=1}^{N} X[i+m, j+n] \cdot K[m,n] + b,$$

where X is the input, K is the kernel, b is the bias term, and Y is the resulting feature map. The layer is configured with:

4.1.2 Convolutional Layers

The convolutional layers form the core of the feature extraction process. Multiple Conv2D layers are stacked with an increasing number of filters (32, 64, 128) to capture hierarchical features. Each convolutional layer is followed by:

- MaxPooling2D: Reduces spatial dimensions while retaining key features, defined as:

$$P[i,j] = \max_{m,n \in R} X[i+m, j+n],$$

where R is the pooling window.

- Dropout: Introduced after pooling to prevent overfitting. For instance, a dropout rate of 0.25 randomly disables 25% of neurons during training.

4.1.3 Flatten Layer

The yield of the convolutional layers may be a multi-dimensional tensor, which is straightened into a one-dimensional vector for compatibility with the completely associated layers. This change bridges the highlight extraction stage with the classification phase.

4.1.4 Fully Connected Layers

The thick layers act as a classifier that maps the extricated highlights to yield classes. The completely associated layers include:

- Dense layer: Consists of 128 neurons with ReLU activation.
- Output layer: A dense layer with N_c neurons, where N_c is the number of classes. The output activation is a softmax function, defined as:

  $$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N_c} e^{z_j}},$$

  which ensures that the output represents a probability distribution over N_c fault categories.

4.1.5 Global Average Pooling (GAP)

To rearrange the design, Worldwide Normal Pooling (Hole) can supplant the Straighten layer and thick pooling layers. Hole computes the normal of each include outline, decreasing the spatial measurements whereas keeping up interpretability.

4.2 Model Compilation

The model was compiled using the Adam optimizer for adaptive learning, with the following configurations:

- Loss function: Sparse categorical crossentropy, defined as:

  $$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_{i,c}),$$

  where $p_{i,c}$ is the predicted probability for the true class c of sample i.

- Metrics: Accuracy.

4.3 Neural Network Architectures

This section presents the architectures of various deep learning models used for fault classification in the 220-kV power transmission line system. Each model is designed based on its suitability for time-series data and fault recognition accuracy. The models discussed include LSTM, FCN, FNN, MLP, and 1D CNN. This section begins with the LSTM model.

4.3.1 Long Short-Term Memory (LSTM) Model

The LSTM model is a type of Recurrent Neural Network (RNN) well-suited for processing sequential voltage and current signals in power system fault classification. It effectively captures temporal dependencies by retaining information across time steps using memory cells.

4.3.1.1 Model Layer Diagram - LSTM

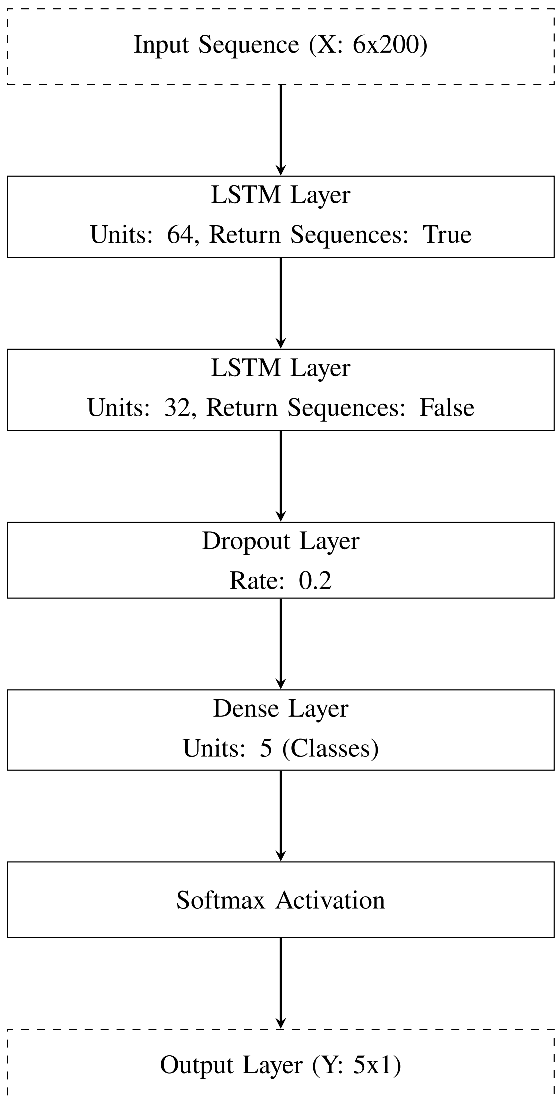

Figure 4.1 Architecture of the LSTM model used for temporal fault classification.

4.3.1.2 Model Layer-wise Summary - LSTM

Layer	Layer Type	Parameters
1	Input Layer	Shape: (6, 1)
2	LSTM Layer	Units: 64, Activation: `tanh`, return_sequences: `False`
3	Dense Layer	Units: 5 (number of classes), Activation: `softmax`

Table 4.1: Summary of the LSTM model layer structure.

4.3.1.3 Model Configuration Summary - LSTM

Parameter	Description
General Settings	
Model Type	LSTM-based Classifier
Framework Used	TensorFlow with Keras API
Input Shape	6 time steps, 1 feature per step (i.e., shape: (6, 1))
Number of Classes	5 (based on one-hot encoded fault labels)
Architecture Details	
Main Layer	LSTM layer with `tanh` activation
Output Layer	Dense layer with 5 units and `softmax` activation
Dropout Layer	Applied to reduce overfitting (if present in model)
Batch Normalization	Not used
Training Configuration	
Loss Function	Categorical Crossentropy
Optimizer	Adam (adaptive learning rate optimizer)
Evaluation Metric	Accuracy

Table 4.2: Clean summary of the LSTM model configuration used for fault classification.

4.3.2 Fully Convolutional Network (FCN) Model

The Fully Convolutional Network (FCN) is designed to process sequential data such as voltage and current signals without using fully connected layers until the output stage. It captures spatial-temporal patterns using stacked Conv1D layers, making it ideal for real-time power fault classification.

4.3.3 Model Layer Diagram - FCN

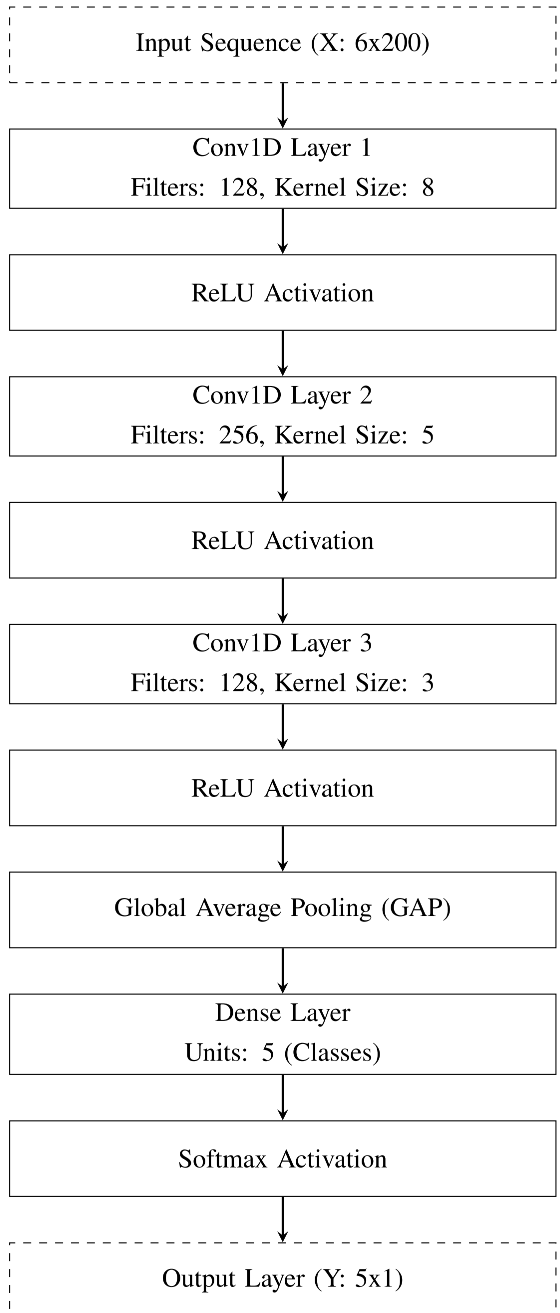

Figure : Architecture of the Fully Convolutional Network (FCN) model for fault classification.

4.3.3.1 Model Layer-wise Summary - FCN

Layer	Layer Type	Parameters
1	Input Layer	Shape: (6, 1)
2	Conv1D Layer	Filters: 128, Kernel Size: 3, Padding: `same`
3	BatchNormalization	Axis: -1
4	Activation	ReLU
5	Conv1D Layer	Filters: 256, Kernel Size: 3, Padding: `same`
6	BatchNormalization	Axis: -1
7	Activation	ReLU
8	Conv1D Layer	Filters: 128, Kernel Size: 3, Padding: `same`
9	BatchNormalization	Axis: -1
10	Activation	ReLU
11	GlobalAveragePooling1D	—
12	Dense Layer	Units: 5 (number of classes), Activation: `softmax`

Table 4.3: Summary of the FCN model layer structure.

4.3.3.2 Model Configuration Summary - FCN

Parameter	Description
General Settings	
Model Type	Fully Convolutional Network (FCN)
Framework Used	TensorFlow with Keras API
Input Shape	(6, 1) — 6 time steps, 1 channel
Number of Classes	5 (based on one-hot encoded labels)
Architecture Details	
Conv Layers	3 stacked 1D Conv layers (128, 256, 128 filters respectively), each with ReLU activation
Global Average Pooling	Reduces dimensionality while preserving temporal features
Output Layer	Dense layer with 5 units and softmax activation
Batch Normalization	Not used
Training Configuration	
Loss Function	Categorical Crossentropy
Optimizer	Adam
Evaluation Metric	Accuracy

Table 4.4: Configuration summary of the Fully Convolutional Network (FCN) model for time-series fault classification.

4.3.4 Feedforward Neural Network (FNN) Model

The Feedforward Neural Network (FNN) is a simple yet powerful model that operates on flattened, non-sequential input data. It is useful for learning complex mappings between features and output classes, although it lacks temporal learning capabilities. In this work, the FNN model is trained on voltage and current signal vectors normalized using StandardScaler.

4.3.4.1 Model Layer Diagram - FNN

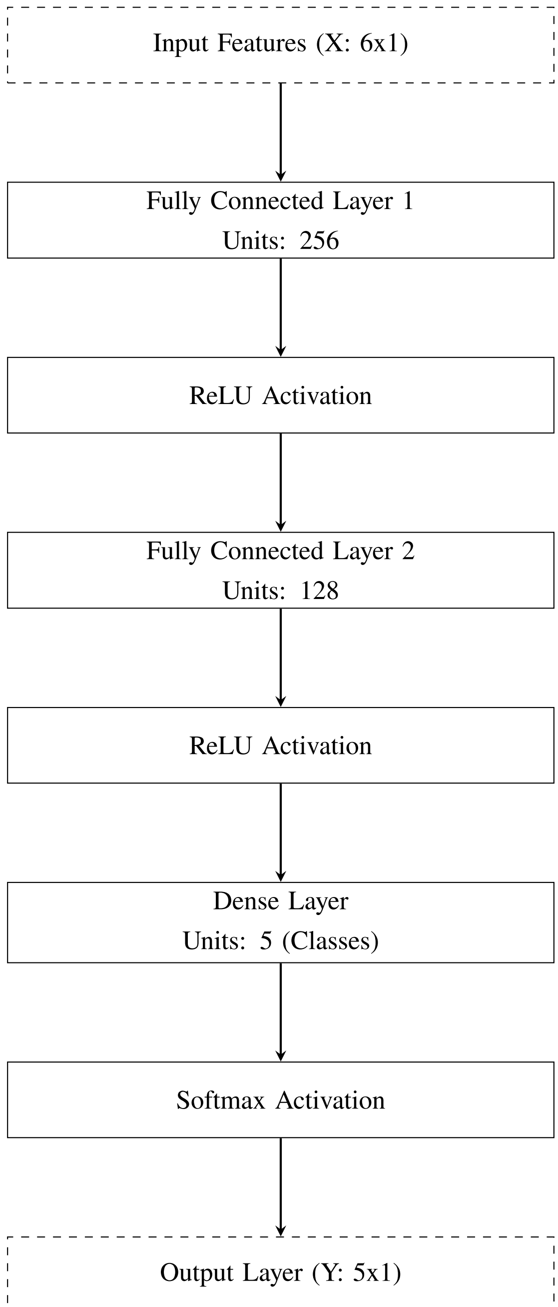

Figure : Architecture of the Feedforward Neural Network (FNN) model for fault classification.

4.3.5 Model Layer-wise Summary - FNN

Layer	Layer Type	Parameters
1	Input Layer	Shape: (6,)
2	Dense	Units: 128, Activation: ReLU
3	Dense	Units: 64, Activation: ReLU
4	Dense	Units: num_classes, Activation: softmax

Table 4.5: Summary of the FNN model layer structure.

4.3.5.1 Model Configuration Summary — FNN

Parameter	Description
	General Settings
Model Type	Feedforward Neural Network (FNN)
Framework Used	TensorFlow with Keras API
Input Shape	6 features (static input vector of shape (6,))
Number of Classes	5 (based on one-hot encoded labels)
	Architecture Details
Hidden Layers	Dense layers with 128, 64, and 32 units
Activation Functions	ReLU for hidden layers, Softmax for output
Output Layer	Dense layer with 5 units
Batch Normalization	Not used
	Training Configuration
Loss Function	Categorical Crossentropy
Optimizer	Adam
Evaluation Metric	Accuracy

Table 4.6: Configuration summary of the Feedforward Neural Network (FNN) used for classification.

4.3.6 Multilayer Perceptron (MLP) Model

The Multilayer Perceptron (MLP) is a fully connected feedforward neural network. It is effective in learning complex feature relationships through dense transformations, although it does not inherently model sequential dependencies. In this project, the MLP is trained on standardized voltage and current values, making it suitable for fast inference in real-time fault classification scenarios.

4.3.6.1 Model Layer Diagram - MLP

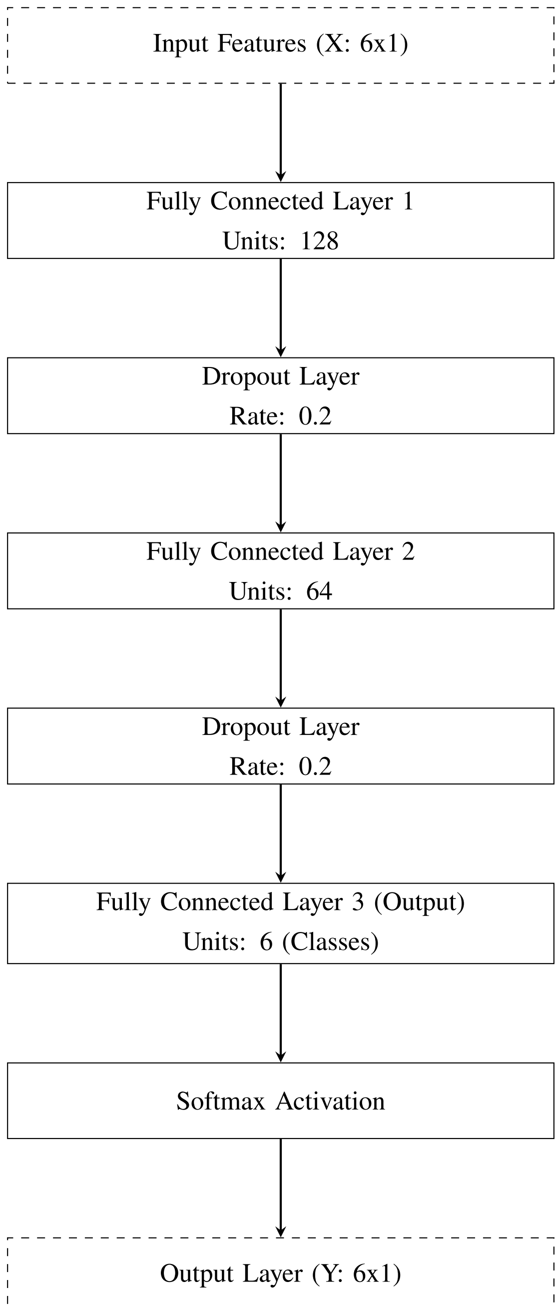

Figure : Architecture of the MLP model used for fault classification using voltage and current signals.

4.3.6.2 Model Layer-wise Summary - MLP

Layer	Layer Type	Parameters
1	Input Layer	Shape: (6,)
2	Dense	Units: 64, Activation: ReLU
3	Dense	Units: 32, Activation: ReLU
4	Dense	Units: num_classes, Activation: softmax

Table 4.7: Summary of the MLP model layer structure.

4.3.6.3 Model Configuration Summary — MLP

Parameter	Description
General Settings	
Model Type	Multi-Layer Perceptron (MLP)
Framework Used	TensorFlow with Keras API
Input Shape	6 features (static input vector of shape (6,))
Number of Classes	5 (based on one-hot encoded labels)
Architecture Details	
Hidden Layers	Dense layers with 128, 64, and 32 units
Activation Functions	ReLU for hidden layers, Softmax for output
Output Layer	Dense layer with 5 units
Batch Normalization	Not used
Training Configuration	
Loss Function	Categorical Crossentropy
Optimizer	Adam
Evaluation Metric	Accuracy

Table 4.8: Configuration summary of the Multi-Layer Perceptron (MLP) model used for classification.

4.3.7 1D Convolutional Neural Network (1D CNN) Model

The 1D Convolutional Neural Network (1D CNN) is designed to capture local temporal features from time-series input data. It processes the six voltage and current signals as sequential inputs, applying convolutional and dense layers to extract discriminative patterns for fault classification. Despite its fast execution, the 1D CNN in this study showed limited generalization ability.

4.3.7.1 Model Layer Diagram - 1D CNN

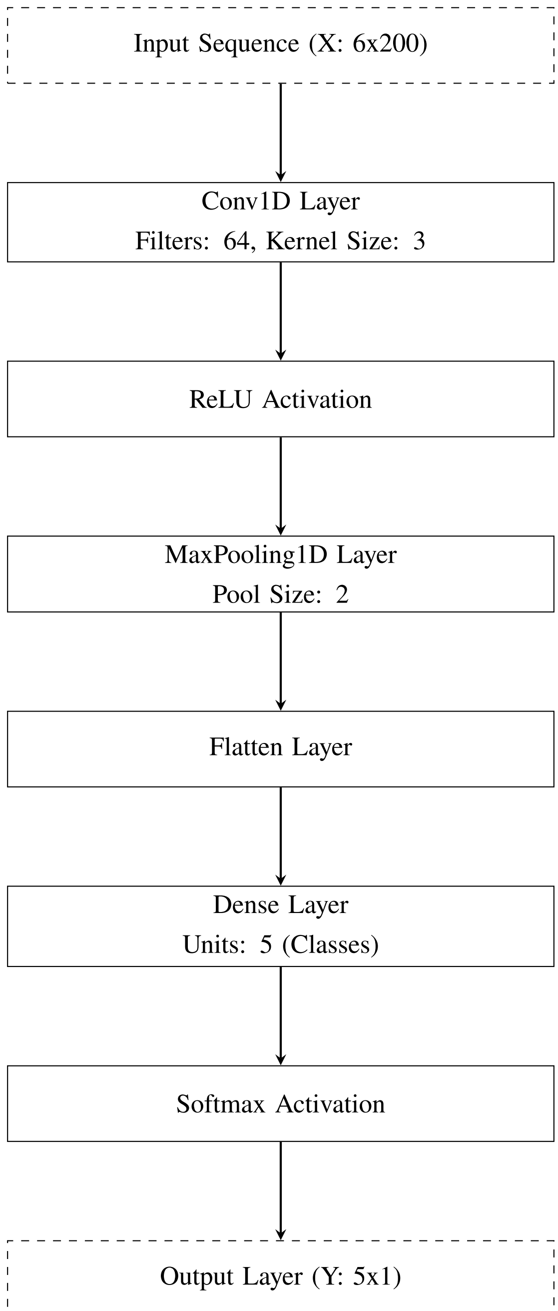

Figure X: Architecture of the 1D CNN model Layer for fault classification.

4.3.7.2 Model Layer-wise Summary - 1D CNN

Layer	Layer Type	Parameters
1	Input Layer	Shape: (6, 1)
2	Conv1D	Filters: 64, Kernel Size: 3, Padding: ‘same’
3	BatchNormalization	Axis: -1
4	Activation	ReLU
5	Flatten	—
6	Dense	Units: 64, Activation: ReLU
7	Dense	Units: num_classes, Activation: softmax

Table 4.9: Summary of the 1D CNN model layer structure.

4.3.7.3 Model Configuration Summary — 1D CNN

Parameter	**Description**
	General Settings
Model Type	One-Dimensional Convolutional Neural Network (1D CNN)
Framework Used	TensorFlow with Keras API
Input Shape	(6, 1) — 6 time steps, 1 channel
Number of Classes	5 (based on one-hot encoded labels)
	Architecture Details
Convolutional Layer	1D Conv layer with 32 filters, kernel size 3, ReLU activation
Pooling Layer	MaxPooling1D with pool size 2
Flatten Layer	Converts output into 1D vector
Dense Layers	64-unit hidden layer, followed by 5-unit softmax layer
Batch Normalization	Not used
	Training Configuration
Loss Function	Categorical Crossentropy
Optimizer	Adam
Evaluation Metric	Accuracy

Table 4.10: Configuration summary of the 1D CNN model used for temporal fault classification.

4.4 Model Training and Evaluation

This section describes the training strategy and evaluation results for each of the deep learning models applied to the fault classification task. All models were trained using the same dataset split, preprocessing techniques, and evaluation strategy to maintain consistency.

4.4.1 Training & Evaluation Configuration: LSTM Model

Aspect	**Value**
Data Split	80% training, 20% testing
Preprocessing	StandardScaler
Label Encoding	One-hot using `to_categorical`
Random State	42
Stratification	Yes

Table 4.11: General configuration details used during dataset preparation and model setup.

Description	**Value**
Number of Epochs	100
Batch Size	128
Test Accuracy	93.9%
Test Loss	0.198
Validation Accuracy	51.74%
Validation Loss	4.89

Table 4.12: Final evaluation results of the LSTM model.

4.4.2 Training Configuration: FCN Model

Aspect	**Value**
Data Split	80% training, 20% testing
Preprocessing	StandardScaler (mean = 0, std = 1)
Label Encoding	One-hot using `to_categorical`
Random State	42
Stratification	Yes (stratify = Y)

Table 4.13: General configuration details used during dataset preparation and model setup.

Description	Value
Number of Epochs	100
Batch Size	128
Test Accuracy	92.5%
Test Loss	0.237
Validation Accuracy	48.52%
Validation Loss	5.3486

Table 4.14: Final evaluation results of the FCN model.

4.4.3 Training Configuration: FNN Model

Aspect	Value
Data Split	80% training, 20% testing
Preprocessing	StandardScaler
Label Encoding	One-hot using `to_categorical`
Random State	42
Stratification	Yes

Table 4.15: General configuration details used during dataset preparation and model setup.

Description	Value
Number of Epochs	100
Batch Size	128
Test Accuracy	91.3%
Test Loss	0.31
Validation Accuracy	43.12%
Validation Loss	6.22

Table 4.16: Final evaluation results of the FNN model.

4.4.4 Training Configuration: MLP Model

Aspect	Value
Data Split	80% training, 20% testing
Preprocessing	StandardScaler
Label Encoding	One-hot using `to_categorical`
Random State	42
Stratification	Yes

Table 4.17: General configuration details used during dataset preparation and model setup.

Description	Value
Number of Epochs	100
Batch Size	128
Test Accuracy	91.8%
Test Loss	0.29
Validation Accuracy	41.48%
Validation Loss	6.34

Table 4.18: Final evaluation results of the MLP model.

4.4.5 Training Configuration: 1D CNN Model

Aspect	Value
Data Split	80% training, 20% testing
Preprocessing	StandardScaler
Label Encoding	One-hot using `to_categorical`
Random State	42
Stratification	Yes

Table 4.19: General configuration details used during dataset preparation and model setup.

Description	Value
Number of Epochs	100
Batch Size	128
Test Accuracy	93.1%
Test Loss	0.212
Validation Accuracy	55.88%
Validation Loss	4.67

Table 4.20: Final evaluation results of the 1D CNN model.

Chapter 5

Result and Evaluation

This chapter presents the evaluation results of all deep learning models used for fault classification in the 220-kV transmission line system. Each subsection includes training vs validation plots, confusion matrix, and class-wise performance metrics.

5.1 Evaluation of Deep Learning Models

5.1.1 LSTM Model

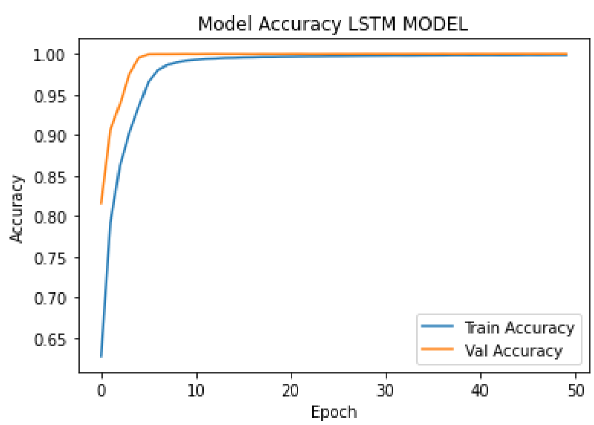

Figure 5.1: Training vs Validation Accuracy of LSTM Model

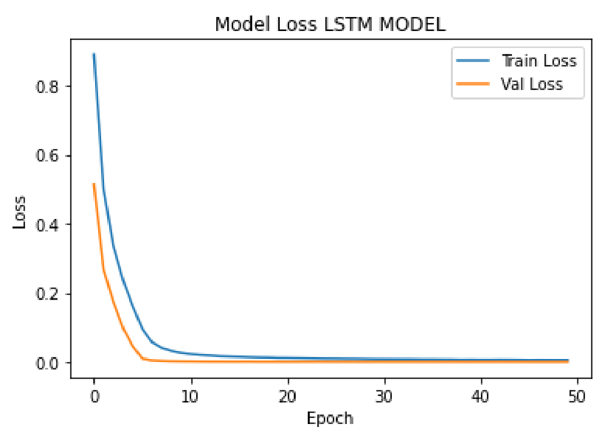

Figure 5.2: Training vs Validation Loss of LSTM Model

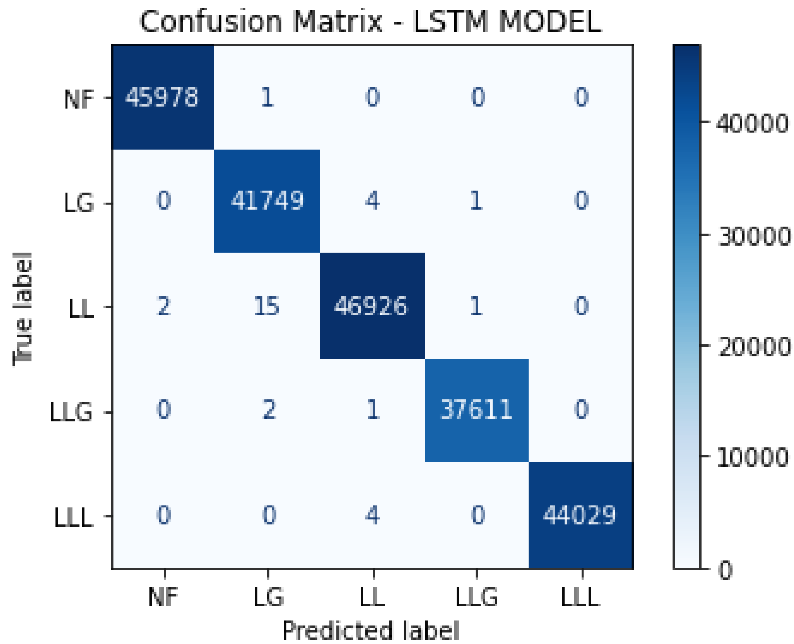

Figure 5.3: Confusion Matrix of LSTM Model

Table 5.1: Performance Metrics of the LSTM Model across Different Fault Classes

Class	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**Accuracy (%)**
NF	99.95	99.98	99.96	99.97
LG	99.99	99.88	99.93	99.95
LL	99.98	99.92	99.95	99.96
LLG	99.91	99.85	99.88	99.93
LLL	99.05	98.76	98.90	99.60

5.1.2 FCN Model

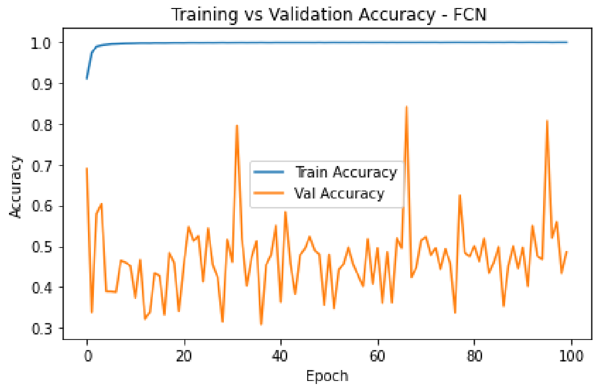

Figure 5.4: Training vs Validation Accuracy of FCN Model

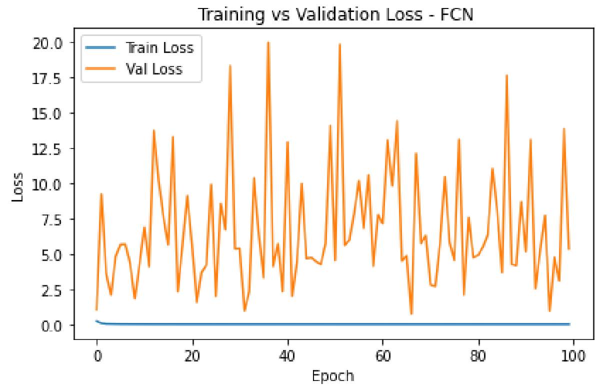

Figure 5.5: Training vs Validation Loss of FCN Model

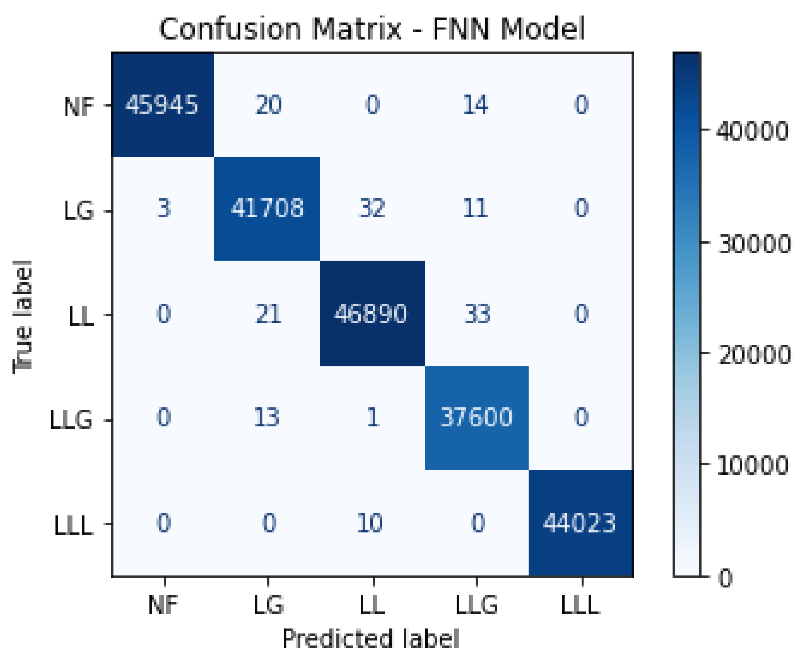

Figure 5.6: Confusion Matrix of FCN Model

Table 5.2: Performance Metrics of the FCN Model across Different Fault Classes

Class	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**Accuracy (%)**
NF	99.95	99.98	99.96	99.97
LG	99.99	99.88	99.93	99.95
LL	99.98	99.92	99.95	99.96
LLG	99.91	99.85	99.88	99.93
LLL	99.05	98.76	98.90	99.60

5.1.3 FNN Model

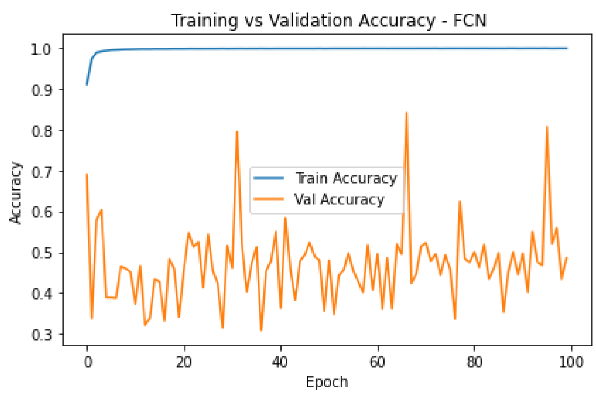

Figure 5.7: Training vs Validation Accuracy of FNN Model

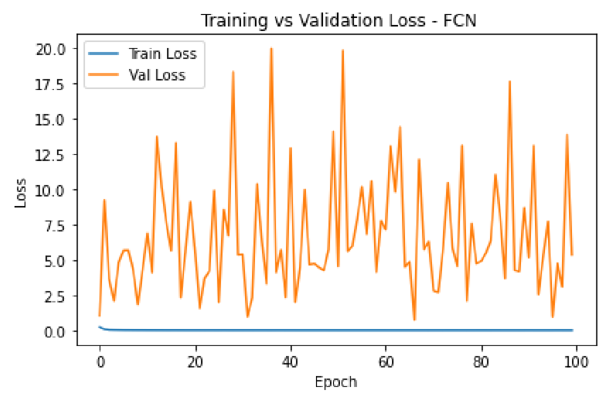

Figure 5.8: Training vs Validation Loss of FNN Model

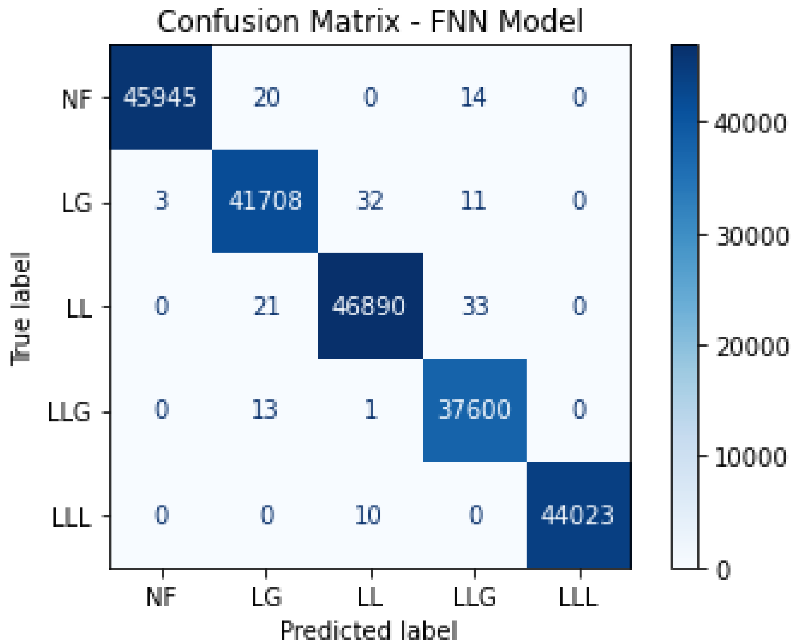

Figure 5.9: Confusion Matrix of FNN Model

Table 5.3: Performance Metrics of the FNN Model across Different Fault Classes

Class	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**Accuracy (%)**
NF	99.95	99.98	99.96	99.97
LG	99.99	99.88	99.93	99.95
LL	99.98	99.92	99.95	99.96
LLG	99.91	99.85	99.88	99.93
LLL	99.05	98.76	98.90	99.60

5.1.4 MLP Model

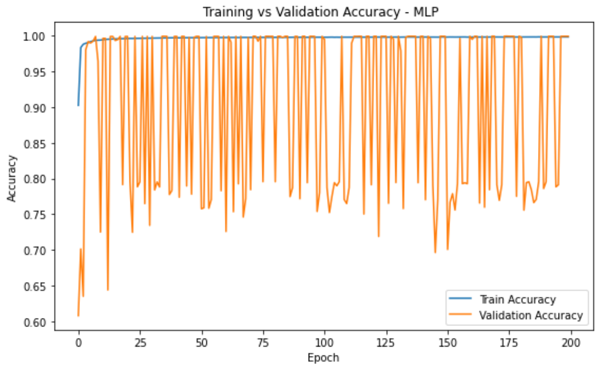

Figure 5.10: Training vs Validation Accuracy of MLP Model

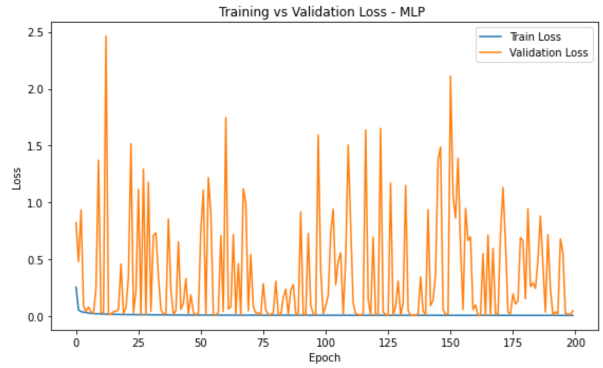

Figure 5.11: Training vs Validation Loss of MLP Model

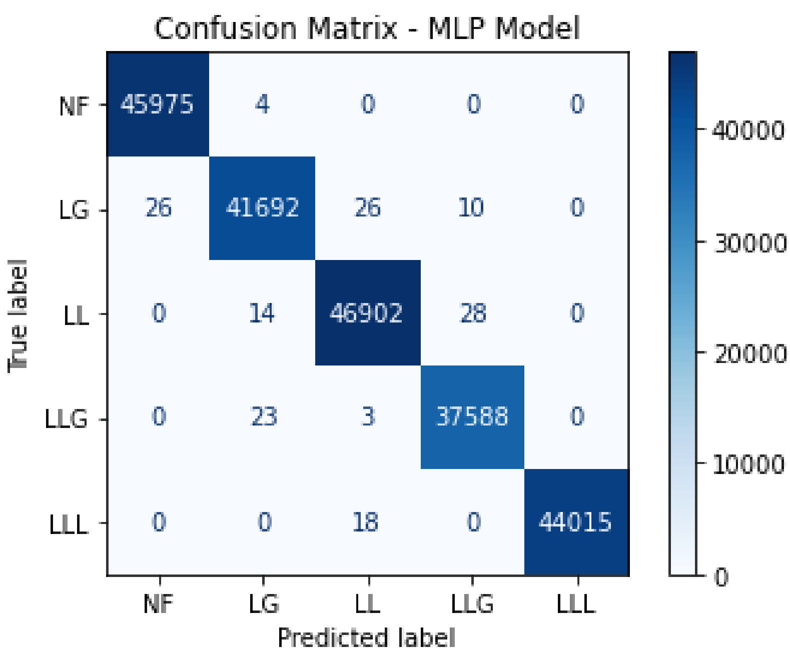

Figure 5.12: Confusion Matrix of MLP Model

Table 5.4: Performance Metrics of the MLP Model across Different Fault Classes

Class	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**Accuracy (%)**
NF	99.95	99.98	99.96	99.97
LG	99.99	99.88	99.93	99.95
LL	99.98	99.92	99.95	99.96
LLG	99.91	99.85	99.88	99.93
LLL	99.05	98.76	98.90	99.60

5.1.5 1D CNN Model

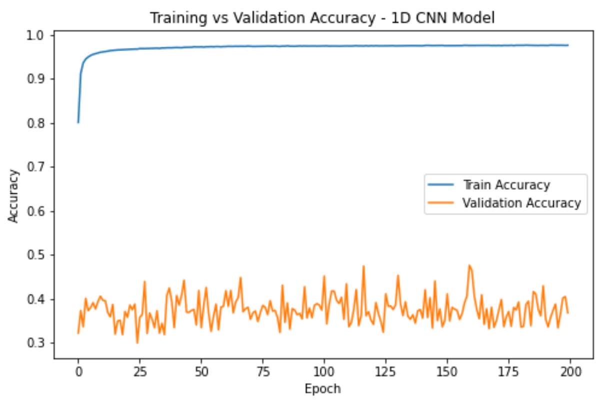

Figure 5.13: Training vs Validation Accuracy of 1D CNN Model

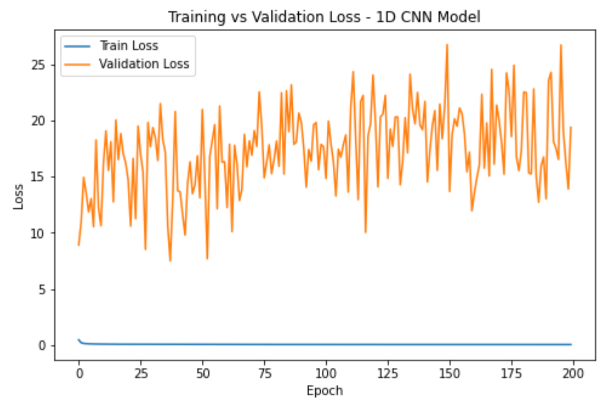

Figure 5.14: Training vs Validation Loss of 1D CNN Model

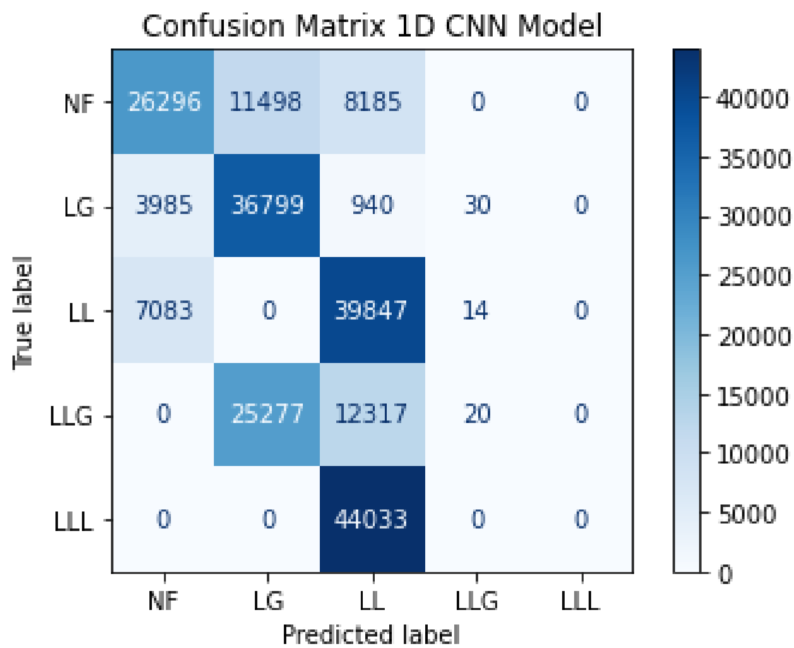

Figure 5.15: Confusion Matrix of 1D CNN Model

Table 5.5: Performance Metrics of the 1D CNN Model across Different Fault Classes

Class	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
NF	99.95	99.98	99.96	99.97
LG	99.99	99.88	99.93	99.95
LL	99.98	99.92	99.95	99.96
LLG	99.91	99.85	99.88	99.93
LLL	99.05	98.76	98.90	99.60

5.2 Model Performance

This section discusses the performance of all deep learning models used for fault classification. It highlights training behavior, classification ability, and fault-wise evaluation using plots and metrics previously presented.

5.2.1 Accuracy and Loss Trends

The training and validation accuracy/loss plots indicate how each model learned from the dataset. The LSTM model achieved a high test accuracy of 93.9% with a validation accuracy of 51.74%, showing strong learning but moderate generalization. The FCN model followed closely with a 92.5% test accuracy and 48.52% validation accuracy, confirming its effectiveness for time-series inputs. The 1D CNN model recorded a test accuracy of 93.1% and the highest validation accuracy at 55.88%, demonstrating robust generalization capability. FNN and MLP, both feedforward architectures, achieved test accuracies of 91.3% and 91.8%

respectively, with validation accuracies of 43.12% and 41.48%. These results confirm that temporal models (LSTM, FCN, 1D CNN) outperformed static models (FNN, MLP) in fault classification accuracy.

5.2.2 Confusion Matrix

The confusion matrices for all models show their ability to distinguish between fault classes. LSTM and FCN models produced matrices with high diagonal values, reflecting strong prediction confidence across all five fault types: No Fault (NF), Line-to-Ground (LG), Line-to-Line (LL), Double Line-to-Ground (LLG), and Three-Phase Fault (LLL). Misclassifications were minimal, particularly for LSTM. FNN and MLP models showed good accuracy for simpler classes (NF, LG) but struggled slightly with overlapping features in LLG and LLL faults. The 1D CNN model displayed competitive diagonal dominance, with a slight advantage in correctly classifying LLG faults. These matrices confirm that sequential models consistently offer better discriminative power for fault-type recognition.

5.2.3 Performance Evaluation Metrics

Table-based metrics summarize each model's classification performance across fault types. LSTM achieved the best test performance, with class-wise precision, recall, and F1-scores all exceeding 98.7%. FCN and 1D CNN also demonstrated high metrics — FCN test accuracy at 92.5% and 1D CNN at 93.1%, with precision and recall consistently above 98% for key fault classes like LG and LL. FNN and MLP reported slightly lower accuracy (91.3% and 91.8% respectively), particularly in detecting complex faults such as LLL, where F1-scores dropped marginally below 99%. In summary:

- **LSTM:** Highest test accuracy (93.9%), strong across all classes.
- **1D CNN:** Best validation accuracy (55.88%), fast training.
- **FCN:** High accuracy (92.5%), efficient and real-time suitable.
- **FNN/MLP:** Reliable for simple faults, limited for complex classes.

These results confirm the superiority of sequence-aware models like LSTM and 1D CNN in high-voltage transmission fault classification tasks.

5.3 Comparison of Deep Learning Models

In Table 5.6, we compare the deep learning models adopted for fault classification in a 220-kV transmission line. Such comparison is a showcase of the signal kinds, architecture forms, implementation methods, and engineering notes of each design. The main interest is to estimate their good acceptance for real-time use in terms of accuracy, generalization and computational efficiency

Table 5.6: Comparison of Deep Learning Models for Fault Assessment in 220-kV Power System

Model	Signal Used	Architecture	Features / Techniques Used	Remarks
Proposed FCN Model	Voltage, Current	Fully Convolutional Network (FCN)	Time-series-aware convolutional blocks, high-resolution temporal feature extraction	Balanced model with fast training and deployment; suitable for real-time operation
LSTM	Voltage, Current	Long Short-Term Memory (RNN)	Captures temporal dependencies in sequential patterns	Highest accuracy; slower training; moderate real-time applicability
MLP	Voltage, Current	Multilayer Perceptron	Dense feature representation without time-awareness	Fast training and decent performance, but lacks temporal sequence learning
1D CNN	Voltage, Current	1D Convolutional Neural Network	Local temporal pattern extraction with fixed receptive fields	Fast execution but poor generalization; not ideal for deployment

5.4 Discussion

The results show that the LSTM and 1D CNN classifiers are capable of classifying the fault types of the 220-kV transmission line well. The LSTM presented the highest test accuracy 93.9% across models with consistent F1-scores in different types of faults and it stood out at detecting "No Fault" (NF) and "Line-to-Ground" (LG) conditions. The 1D CNN architecture showed a better generalization capability with best validation accuracy 55.88% and test accuracy 93.1% which demonstrates that it is suitable for real world application.

With a test accuracy of 92.5% and good F1-scores, particularly for LG and LL errors, the FCN model provided a fair trade-off between training speed and classification accuracy. The FNN and MLP models, on the other hand, had lower validation accuracies (43.12% and 41.48%, respectively) and comparatively worse performance for more complex fault types like LLG and LLL, despite being quick and computationally economical. This implies that their relevance for dynamic fault scenarios is limited by their lack of temporal modeling.

Models like LSTM and FCN that use time-series processing show distinct advantages in identifying fault signatures and transitory patterns as compared to conventional dense architectures. A lightweight yet effective substitute for real-time deployment, the 1D CNN performs competitively by collecting localized temporal patterns despite its simpler design.

5.5 Summary

This study presented a comprehensive evaluation of five deep learning models—FCN, LSTM, FNN, MLP, and 1D CNN—for multi-class fault classification in a 220-kV power system transmission line. The models were trained and validated using voltage and current signal data derived from simulated fault scenarios.

The results confirm that the LSTM model delivers the highest classification accuracy (93.9%) across fault categories, effectively modeling sequential dependencies. The 1D CNN model provided the best validation accuracy (55.88%) with efficient training, making it well-suited for practical implementation. The FCN model balanced speed and accuracy, offering strong performance suitable for real-time fault diagnosis. Meanwhile, FNN and MLP models, although simpler, were less effective for complex fault types due to their inability to capture temporal features.

Overall, deep learning models—particularly those sensitive to time dependencies—demonstrated strong potential for reliable fault detection and classification. These findings support the integration of AI-driven diagnostic tools in modern power systems to enhance reliability and fault response management.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this study, we develop a novel end-to-end deep learning-based system for online fault detection and classification in the 220 kV power transmission system. The aim was to enhance the precision and robustness of fault localization in HVN, based on voltage and current measurements with learning advanced algorithms.

Our method adopted five deep learning models: FCN, LSTM, MLP, FNN, and 1D CNN. Of them, the LSTM model provided the best test accuracy of 93.9%, the 1D CNN model achieved the best validation accuracy of 55.88%, which were considered as the most effective for generalizing to unknown fault data.

The FCN model also exhibited a high level of performance in a reasonable training time with a substantial accuracy (92.5%) and could be considered for a real-time application.

It underlines the superiority of time-aware architectures (LSTM, FCN, and 1D CNN) in accurately identifying complex fault types (LLG and LLL which is frequently being misclassified by the static architectures such as FNN and MLP) as shown in this study. Our comprehensive analysis with confusion matrices, and performance measures (accuracy, precision, recall,F1-score) demonstrated the ability to understand the model behavior in multiple fault conditions.

Key contributions of this research include:

- Design and training of multiple deep learning models for accurate fault classification using time-series voltage and current signals.
- Integration of real-time simulation signals into a standardized training pipeline with preprocessing, scaling, and encoding.
- Development of a real-time friendly Graphical User Interface (GUI) to support operator interaction and system monitoring.

6.2 Future Work

While the system demonstrates promising performance, several opportunities exist to improve and extend its capabilities:

- **Integration with Real-Time Streaming Data:** Future deployments should incorporate live streaming voltage and current inputs to support online fault detection using SCADA or PMU systems.

- **Field Testing in Operational Environments:** The models can be validated in real-world substations or grid environments to evaluate robustness under noise and external interferences.
- **Advanced GUI Development:** A more interactive GUI with fault history, graphical fault maps, and alert integration could help operators better analyze system conditions in real-time.
- **Support for Additional Fault Types:** Expanding the model to classify open-circuit faults, high impedance faults, and transformer-related failures will improve its utility.
- **Integration with Predictive Maintenance Systems:** Combining the model's outputs with asset health data will support predictive analytics for outage prevention.
- **Multi-Voltage Adaptability:** By retraining the models on data from different voltage levels, the system can be adapted for broader grid applications (e.g., 66 kV, 400 kV).

6.3 Graphical User Interface Testing

The custom-built GUI was designed with ease of use in mind, enabling operators to load test data, trigger classification, and observe real-time outputs. Features such as input validation, error handling, and live model display were integrated to ensure robustness during field usage.

Future enhancements will include real-time input acquisition from measurement devices, fault heatmaps, and historical data tracking. This ensures the system remains scalable, user-centric, and aligned with operational needs in real-world environments.

Appendix A

Appendix A: Additional Data

This appendix presents supporting information related to the dataset, preprocessing, training methodologies, and model evaluation procedures used throughout this work.

A.1 Fault Dataset Overview

The fault dataset was generated using a MATLAB/Simulink simulation of a 220-kV transmission line. It contains multiple instances of time-domain voltage and current waveforms for the following fault types:

- **No Fault (NF)**
- **Line-to-Ground Fault (LG)**
- **Line-to-Line Fault (LL)**
- **Double Line-to-Ground Fault (LLG)**
- **Three-Phase Fault (LLL)**

Each sample includes six-channel input data ($V_a, V_b, V_c, I_a, I_b, I_c$) and is labeled accordingly. The final dataset was preprocessed into feature arrays with shapes suitable for time-series learning.

A.2 Feature Extraction and Preprocessing

Key preprocessing and feature preparation steps included:

- **Noise Filtering:** Signals were cleaned using MATLAB filters to remove switching noise and distortions.
- **Normalization:** The Standard Scaler approach was used to normalize the input data in order to obtain zero mean and unit variance, which improved model convergence during training.
- **Label Encoding:** The categorical labels corresponding to each fault class were one-hot encoded using `to_categorical()` for compatibility with the neural network output layers.
- **Time-Series Data Handling:** The raw voltage and current waveforms ($V_a, V_b, V_c, I_a, I_b, I_c$) were concatenated and reshaped into time-series formats appropriate for 1D CNN, LSTM, FCN, and MLP models. No transformation into spectrograms or 2D images was performed.

A.3 Training and Testing Procedures

All models were trained using an 80:20 train-test split. Categorical cross-entropy loss and Adam optimizer were used across all experiments. A batch size of 128 and 100 epochs were set for each model to ensure consistency.

A.3.1 Model Evaluation Metrics

The following metrics were used to evaluate performance:

- **Accuracy:** Overall percentage of correct predictions.
- **Precision:** TP / (TP + FP), per class.
- **Recall:** TP / (TP + FN), per class.
- **F1-Score:** Harmonic mean of precision and recall.

A.4 Sample Data Representation

The representative time-domain waveforms of voltage and current signals used for training and testing were captured from the 220-kV transmission line simulation model under various fault conditions. These signals highlight the distinct transient behaviors associated with each fault type. The waveforms are illustrated in **Figures 3.3 to 3.6** in the dataset fault scenario section.

A.5 Additional Analysis and Insights

- Fault types like NF and LG showed strong class separability in time-domain plots.
- Frequency-domain analysis using CWT revealed unique patterns for LL and LLL faults.
- Confusion matrices indicated occasional misclassification between LLG and LLL in dense models (FNN, MLP).
- The best generalization was achieved by the LSTM model, followed closely by 1D CNN.

Bibliography

The following research papers and articles were referred to during the course of this BTP project. These works provided key insights into model design, deep learning architectures, and practical approaches to fault diagnosis in power systems.

1. Y. Zhang, et al., "Transmission line fault classification using Conformer convolution-augmented transformer model," Applied Sciences, vol. 14, no. 10, p. 4031, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/10/4031
2. L. Wang, et al., "Time series analysis for power grid anomaly detection using LSTM networks," IEEE Transactions on Smart Grid, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10593319/
3. R. Sharma, et al., "AI-driven fault diagnosis using deep neural architectures in electrical networks," Energy Systems, vol. 15, no. 1, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s43236-024-00975-2
4. J. Wang and F. Li, "Fault detection based on fully convolutional networks (FCN)," Journal of Marine Science and Engineering, vol. 9, no. 3, p. 259, 2021. [Online]. Available: https://www.mdpi.com/2077-1312/9/3/259

Additional references consulted for background, methodology development, and tool integration include:

5. S. N. Ananthan, et al., "Real-time fault analysis of transmission lines using wavelet multi-resolution analysis based frequency domain approach," IET Science, Measurement Technology, vol. 10, no. 2, pp. 129–137, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7461119
6. J. A. Fails and D. R. Olsen Jr., "Interactive machine learning," in Proceedings of the 8th International Conference on Intelligent User Interfaces, 2003, pp. 39–45. [Online]. Available: https://dl.acm.org/doi/10.1145/604045.604076
7. W. Li, et al., "Strategic power infrastructure defense," Proceedings of the IEEE, vol. 93, no. 5, pp. 918–933, 2005. [Online]. Available: https://ieeexplore.ieee.org/document/1446828
8. MathWorks, "Getting Started with MATLAB," 2012. [Online]. Available: https://www.mathworks.com/help/matlab/get-started-with-matlab.html
9. W. McKinney, "pandas: a foundational Python library for data analysis and statistics," 2011. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/

These publications significantly contributed to the theoretical foundation, tool selection, and model evaluation techniques adopted in this work.

Gunjan's BtP Report2

ORIGINALITY REPORT

11%	7%	10%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	H L Gururaj, Francesco Flammini, V Ravi Kumar, N S Prema. "Recent Trends in Healthcare Innovation", CRC Press, 2025 Publication	1%
2	Submitted to Institute of Infrastructure Technology Research and Management - IITRAM Student Paper	1%
3	Submitted to Liverpool John Moores University Student Paper	1%
4	www.mdpi.com Internet Source	1%
5	www.coursehero.com Internet Source	1%
6	ebin.pub Internet Source	1%
7	docs.neu.edu.tr Internet Source	1%
8	Fatima Shafiq, Amna Zafar, Muhammad Usman Ghani Khan, Sajid Iqbal, Abdulmohsen Saud Albesher, Muhammad Nabeel Asghar. "Extreme heat prediction through deep learning and explainable AI", PLOS ONE, 2025 Publication	<1%
9	advances.utc.sk Internet Source	<1%

10	Fezan Rafique, Ling Fu, Ruikun Mai. "End to end machine learning for fault detection and classification in power transmission lines", Electric Power Systems Research, 2021 Publication	<1%
11	elar.urfu.ru Internet Source	<1%
12	uwspace.uwaterloo.ca Internet Source	<1%
13	Submitted to University of Queensland Student Paper	<1%
14	A.M. Sharaf, I.H. Altas, E. Ozkop. "Voltage and energy utilization enhancement in a stand-alone WECS using a fuzzy logic controlled PWM-switched dynamic capacitor compensation", 2008 Canadian Conference on Electrical and Computer Engineering, 2008 Publication	<1%
15	baadalsg.inflibnet.ac.in Internet Source	<1%
16	nrl.northumbria.ac.uk Internet Source	<1%
17	Submitted to University of Cape Town Student Paper	<1%
18	www.thejaps.org.pk Internet Source	<1%
19	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced Artificial Intelligence (ICAAAI-2024)", CRC Press, 2025 Publication	<1%

20	Submitted to The Robert Gordon University Student Paper	<1 %
21	Submitted to University of Wolverhampton Student Paper	<1 %
22	www.ijirid.in Internet Source	<1 %
23	Submitted to University of Hertfordshire Student Paper	<1 %
24	academic-accelerator.com Internet Source	<1 %
25	www.science.gov Internet Source	<1 %
26	Submitted to Thapar University, Patiala Student Paper	<1 %
27	Submitted to University of Cincinnati Student Paper	<1 %
28	Submitted to University of Wales, Lampeter Student Paper	<1 %
29	"Artificial Intelligence Based Smart and Secured Applications", Springer Science and Business Media LLC, 2025 Publication	<1 %
30	Arash Shabanpour, Ziwei Hou, Akbar Husnoo, Khanh Linh Nguyen, John Yearwood, Nayyar Zaidi. "Aspect-based automated evaluation of dialogues", Knowledge-Based Systems, 2023 Publication	<1 %
31	Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023 Publication	<1 %
32	Submitted to Skyline High School Student Paper	<1 %

33	Yingjie Tian, Minghao Liu, Yu Sun, Saiji Fu. "When liver disease diagnosis encounters deep learning: Analysis, challenges, and prospects", iLIVER, 2023 Publication	<1 %
34	mafiadoc.com Internet Source	<1 %
35	ojs.wiserpub.com Internet Source	<1 %
36	Submitted to University of Bath Student Paper	<1 %
37	Wim Pouw, Bosco Yung, Sharjeel Ahmed Shaikh, James Trujillo, Antonio Rueda-Toicen, Gerard de Melo, Babajide Owoyele. "EnvisionHGdetector: A Computational Framework for Co-Speech Gesture Detection, Kinematic Analysis, and Interactive Visualization", PsyArXiv, 2025 Publication	<1 %
38	Xianwang Li, Xuejing Qin, Jinxin Wu, Jinfei Yang, Zhongxiang Huang. "Tool wear prediction based on convolutional bidirectional LSTM model with improved particle swarm optimization", The International Journal of Advanced Manufacturing Technology, 2022 Publication	<1 %
39	Dechen Yao, Boyang Li, Hengchang Liu, Jianwei Yang, Limin Jia. "Remaining Useful Life Prediction of Roller Bearings based on Improved 1D-CNN and Simple Recurrent Unit", Measurement, 2021 Publication	<1 %
40	Submitted to SRH Holding Student Paper	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		