Artificial Intelligence and Edge Computing

Bachelor of Engineering

(Civil Engineering)

Submitted By

Rakshit Verma (2310996777)

Ansh Nagi(2310996779)

Bhupinder(2310996780)

Shashank (2310996783)

Semester 4th

Department Of Civil Engineering, CUIET-CE

Chitkara University, Punjab



# Electrical Faults Analysis & Classification

## 1. Introduction

Electrical faults are disturbances in power systems that can lead to **voltage instability, equipment damage, power outages, and even fire hazards**. These faults arise due to various factors such as **overloading, insulation breakdown, weather conditions, and equipment failure**. **Quick and accurate fault detection** is critical to maintaining system stability and preventing widespread damage.

This project focuses on developing a **Machine Learning (ML)-based fault classification model** to **automatically identify and categorize different types of electrical faults** based on key system parameters like **voltage, current, power factor, frequency, and phase angle**.

**Types of Electrical Faults Considered in This Project**

Electrical faults can be classified into **symmetrical faults** (affecting all phases equally) and **unsymmetrical faults** (affecting one or two phases). The following fault types are analyzed in this project:

**1. Line-to-Ground (LG) Fault**

**2. Line-to-Line (LL) Fault**

**3. Double Line-to-Ground (LLG) Fault**

**4. Three-Phase (LLL) Fault**

**5. Three-Phase-to-Ground (LLLG) Fault**

## 2. Dataset Description

This project uses two datasets:  
• Class Data (`classData.csv`): Used for training a fault classification model.  
• Detection Data (`detect\_dataset.csv`): Used for testing and validation.

## 3. System Architecture

The fault classification system follows a structured pipeline from data collection to fault classification. The key tools and frameworks used in this project include:

• Programming Language: Python

• Libraries Used: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn

• Machine Learning Models Tested:

- Random Forest Classifier (Best Performing Model: 99.63% Accuracy)

- Support Vector Machine (SVM)

- Logistic Regression

- K-Nearest Neighbors (KNN)

## 4. Methodology

This project employs various machine learning techniques to classify electrical faults.

• Data Preprocessing: Handled missing values, normalized numerical data, and selected key features.

• Training Data: 80% of `classData.csv` was used for training.

• Testing Data: 20% of `classData.csv` was used for validation.

• Model Selection: Various ML models were tested, including Random Forest, SVM, Logistic Regression, and KNN.

• Hyperparameter Tuning: Grid Search was applied to optimize Random Forest parameters.

• Evaluation Metrics: Accuracy, Precision, Recall, and F1-score were used to measure performance.

## 5. Results

|  |  |
| --- | --- |
| Metric | Value |
| Accuracy (Dataset 1) | 99.63% |
| Accuracy (Dataset 2) | 87.67% |
| Precision | 99.73% |

## 6. Live Prediction Example

The trained Random Forest model can now be used for real-time fault classification. When given new electrical parameters, it predicts the fault type with high accuracy.

Example Prediction Output:

• Voltage: 220V  
• Current: 15A  
• Power Factor: 0.95  
• Frequency: 50Hz  
• Phase Angle: 30°

→ Model Prediction: Line-to-Ground Fault (LG Fault)

## 7. Model Flow Diagram

The fault classification system follows this structured workflow:

1. Data Input – Collecting raw electrical fault data.

2. Preprocessing – Cleaning and normalizing the dataset.

3. Feature Engineering – Selecting and transforming important attributes.

4. Model Training – Training the machine learning model (Random Forest Classifier).

5. Fault Classification – Predicting the fault type from new input data.

## 8. Future Scope & Enhancements

• Integration with IoT-based smart grids for real-time monitoring.

• Enhancing model performance using Deep Learning techniques.

• Expanding dataset with real-world electrical fault data for better generalization.

• Implementing an automated alert system for fault detection and quick resolution.

## 9. References

1. Dataset Files: `classData.csv`, `detect\_dataset.csv` - Used for training and testing models.

2. Jupyter Notebook: `ElectricalFaultAnalysis.ipynb` - Contains implementation details, model training, and evaluations.

## 10. Reference Link