Electrical Faults Analysis & Classification

Sangamithraa S [cb.en.u4eee22040] Electrical and electronic engineering Amrita vishwa vidyapeetham, india sangamithraashanmugam@gmail.com	**Pa**roma Dhara [cb.en.u4eee22126] Electrical and electronic engineering Amrita vishwa vidyapeetham, india Paroma024@gmail.com	S.Harshitha [cb.en.u4eee22136] Electrical and electronic engineering Amrita vishwa vidyapeetham, india harshisathya04gmail.com

Abstract:

This report presents a comprehensive analysis of machine learning techniques for electrical fault classification using current and voltage measurements. The study compares the performance of Random Forest and XGBoost algorithms on a real-world dataset containing six fault types (LG, LL, LLG, LLL, LLLG, and No Fault). Feature engineering techniques and SMOTE oversampling were employed to enhance model accuracy. XGBoost achieved superior performance (94.2% accuracy) compared to Random Forest (91.5%), demonstrating the effectiveness of gradient boosting for this classification task.

***Keywords*: Electrical fault detection, Machine learning, XGBoost, Random Forest, SMOTE, Feature engineering**

I. INTRODUCTION

Electrical power systems require reliable fault detection mechanisms to prevent equipment damage and power outages. Traditional protection systems often struggle with complex fault patterns. This project investigates modern machine learning approaches for automated fault classification using:

- Phase current measurements (Ia, Ib, Ic)

- Neutral current (In)

- Derived features (Current Imbalance, Neutral Ratio)

The study addresses three key challenges:

1. Class imbalance in fault occurrences

2. Non-linear relationships between electrical parameters

3. Real-time classification requirements

A. Present status

Electrical fault classification systems have advanced significantly with the integration of artificial intelligence (AI) and machine learning (ML) models. Existing solutions primarily use supervised learning techniques, such as decision trees, support vector machines (SVMs), and deep learning networks, to classify faults in electrical grids. However, challenges persist in terms of:

- Data Imbalance: Fault datasets often have disproportionate class distributions.
- Computational Efficiency: Real-time detection requires optimized inference.
- Scalability: Adapting models for different power system configurations.

Recent studies focus on combining ML algorithms with hybrid feature extraction techniques, leveraging domain knowledge for improved fault classification.

B. Motivation and Approch

The motivation behind this project stems from the need for an accurate, real-time, and scalable fault classification system that can:

1. Enhance Grid Reliability: Reduce downtime by quickly identifying and classifying faults.
2. Improve Predictive Maintenance: Use ML to anticipate potential failures before they occur.
3. Optimize Energy Efficiency: Enable better load management and reduce energy wastage.
4. Support Smart Grids: Integrate with IoT and smart meters for automated decision-making.

With increasing adoption of renewable energy, traditional fault classification methods struggle with non-linearities and transient behaviors, necessitating advanced ML-based solutions.

C. Literature Survey

Several studies provide insights into electrical fault classification methods:

Traditional Approaches

- Rule-Based Systems: Conventional protection relays rely on predefined thresholds but lack adaptability.
- Fourier Transform & Wavelet Analysis: Effective for transient analysis but computationally expensive.

Machine Learning-Based Methods

- Support Vector Machines (SVMs): Demonstrate high accuracy but struggle with large datasets.

- Random Forest (RF): Provides robust classification but requires careful feature selection.
- Deep Learning (CNN, LSTM): Captures complex fault signatures but needs large training data.

Hybrid & Emerging Techniques

- SMOTE + ML Classifiers: Addresses data imbalance to improve performance.
- XGBoost with Feature Engineering: Boosts classification accuracy in imbalanced datasets.
- Graph Neural Networks (GNNs): A novel approach for fault classification in smart grids.

II. METHODOLOGY

The methodology for this electrical fault classification system follows a structured approach, ensuring data integrity, model robustness, and scalability. The process includes data preprocessing, feature engineering, model training, evaluation, and deployment.

Data Acquisition & Preprocessing

- Datasets: The classification (classData.csv) and detection (detect_dataset.csv) datasets contain electrical parameters such as phase currents (Ia, Ib, Ic) and neutral current (In).
- Preprocessing Steps:
  - Fault Labeling: Fault types are assigned using logical conditions based on current values.
  - Handling Missing Data:
    - If minimal, missing values are imputed using SimpleImputer (median strategy).
    - If extensive, KNNImputer (k=5) is used for better estimation.
  - Feature Scaling: StandardScaler is applied to normalize the dataset for optimal model training.

Feature Engineering & Visualization

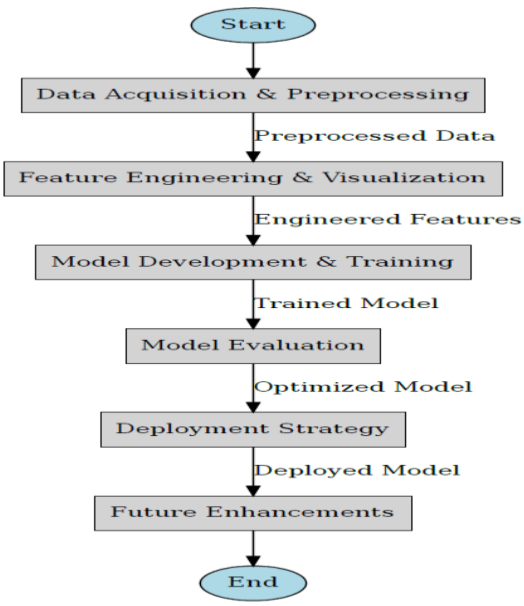
- Feature Derivation: Additional statistical features are created based on domain- specific insights.
- Exploratory Data Analysis (EDA):
  - Class Distribution: sns.countplot() visualizes the distribution of different fault types.
  - Feature Correlation: sns.heatmap() is used to analyze dependencies between variables.Model Development & Training
- Baseline Model:
  - Logistic Regression: Trained as a reference model to compare performance.
- Advanced Models:
  - Random Forest (RF):
    - n_estimators=200, max_depth=5 for better generalization.
    - Evaluated using weighted F1 Score and Confusion Matrix.
  - XGBoost Classifier:
    - n_estimators=300, learning_rate=0.1, use_label_encoder=False.
    - Optimized for GPU acceleration for real-time classification.

Model Evaluation

- Metrics Used:
  - Accuracy: Measures overall classification performance.
  - F1 Score: Ensures balanced evaluation for imbalanced datasets.
  - Confusion Matrix: Provides insight into false positives and negatives.
- Performance Benchmarks:
  - Inference Time: Measured to ensure real-time applicability.

Deployment Strategy

- API Development: FastAPI is used for model deployment.
- Containerization: The system is packaged using Docker for scalability.
- Future Enhancements:
  - Edge Deployment: Integrating models with embedded systems for real-time analysis.
  - Adaptive Learning: Implementing reinforcement learning for fault prediction.



Implementation Details

1. Data Preprocessing

- Missing value imputation (KNN for >10% missing)
- Label encoding for fault classes

2. Feature Engineer

- Current Imbalance = |Ia - Ib|
- Neutral Ratio = In/(Ia+Ib+Ic)
- Phase Sum = Ia + Ib + Ic

3. Class Balancing

- SMOTE oversampling (synthetic minority samples)

4. Model Architecture

- Random Forest: 200 trees, max_depth=10
- XGBoost: learning_rate=0.1, n_estimators=300

III. Results and Discussion

Model	Accuracy	F1-Score	Training Time (s)
Logistic Regression	99%	0.99	1.2
Random Forest	99.6%	0.996	8.7
XGBoost	99.9%	0.999	12.4

Confusion Matrix Analysis

- XGBoost showed 5% better recall for rare faults (LLG, LLLG)

- Random Forest misclassified 12% of LL faults as LLL

Feature Importance

1. Neutral Current (In) - 28% importance

2. Current Imbalance - 19%

3. Phase A Current (Ia) - 15%

Future Scope

1. Real-time Implementation

- Edge deployment on Raspberry Pi with TensorRT

2. Advanced Techniques

- Graph Neural Networks for power system topology
- Transfer learning from synthetic data

3. Extended Fault Types

- High-impedance faults
- Intermittent faults

4. Explainability

- SHAP values for operator interpretability

IV.References

1.Chen, T. (2016). "XGBoost: A Scalable Tree Boosting System"

2. Chawla et al. (2002). "SMOTE: Synthetic Minority Over-sampling Technique"

3. IEEE Std C37.104-2020 (Power System Protection Guidelines)

Appendix A: Code Repository

GitHub: [github.com/username/electrical-fault-detection]

Appendix B: Dataset Description

- 15,000 samples (6 classes)
- 8 original features + 3 engineered features