Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Application of Machine Learning in Fault Detection And Classification in Power Transmission Lines

M.E. Tshodi T.[*1], A. Ntumba N. [2], P. Mbuyi B. [3], F. Keredjim D. [4], J.J. Katshitshi M. [5], N. Kasoro M. [6], L. Kitoko S.[7]*

[1,2,4,6] *Faculty of Science and Technology, University of Kinshasa, Kinshasa, D.R. Congo*
[2,7] *Department of Electrical Engineering, Polytechnic Faculty, University of Kinshasa, Kinshasa, D.R. Congo*
[5] *Department of Computer Management and Business English, Faculty of Economics, University of Kinshasa, Kinshasa, D.R. Congo*

email:[1] michel.tshodi@unikin.ac.cd, [2] nathanael.kasoro@unikin.ac.cd, [3] freddykeredjim@gmail.com, [4] albertntumba1994@gmail.com, [5] jeanjacques.katshitshi@unikin.ac.cd, [6] paul.mbuyi@unikin.ac.cd, [7] profkitoko@yahoo.fr

ARTICLE INFO

Article history:
Received 12 August 2024
Revised 06 November 2024
Accepted 09 December 2024
Available online 30 December 2024

Keywords:
Fault detection;
Analytical models;
Machine learning algorithms;
Power transmission lines;
electrical faults detection;

IEEE style in citing this article:
M. E. Tshodi et al., "Application Of Machine Learning in Fault Detection And Classification In Power Transmission Lines ," Journal of Innovation Information Technology and Application (JINITA), vol. 6, no. 2, pp. 118–129, Dec. 2024.

ABSTRACT

Electrical faults have been identified as a significant contributing factor to electrical equipment damage. Such incidents can potentially result in a range of adverse consequences, including bushfires, electrical outages, and power shortages. The detection and classification of faults facilitates the delivery of superior quality of service, the preservation of the environment, the prevention of equipment damage, and the satisfaction of electricity line subscribers. In this study, we analyze the data from an electrical network comprising four generators of 11 kV, which have been modeled in Matlab. The generators are situated in pairs at either end of the transmission line. Subsequently, machine learning techniques are employed to detect faults in the transmission between lines, and machine learning models are utilized to classify the faults. Four distinct supervised machine learning classifiers are employed for comparison purposes, with the results presented in a confusion matrix. The results demonstrated that decision trees are particularly well-suited to this task, with an 88.6205% detection rate and a slightly higher accuracy than the random forest algorithm (87.9212% detection rate). The K-nearest neighbor's approach yielded a lower result (80.4196% of faults detected), while logistic regression demonstrated the lowest performance, with 34.5836% of faults detected. Six fault categories were found in the dataset: No-Fault (2365 occurrences), Line A Line B to Ground Fault (1134 occurrences), Three-Phase with Ground (1133 occurrences), Line-to-Line AB (1129 occurrences), Three-Phase (1096 occurrences) and finally Line-to-Line with Ground BC (1004 occurrences).

1. INTRODUCTION

The significance of electrical energy is self-evident in the context of the expansion of various industrial sectors, including chemical, mining, health, and others. All these industries require electrical energy, and the demand for it continues to grow daily. Generation, transmission, and distribution systems represent the primary components of an electric power system. Generating stations and distribution systems are interconnected through transmission lines. Transmission lines are typically utilized for bulk power transfer by high-voltage links between primary load centers. Conversely, distribution systems are primarily responsible for conveying this power to consumers through lower-voltage networks [1]. Several techniques are used for fault detection. Visual inspection [14] uses a specialized tool like a magnifying glass or the human eye to identify overt failure indications, including worn-out or broken wires. In the context of our

**) Corresponding Author: michel.tshodi@unikin.ac.cd*

investigation, this technique is quite constrained because it solely records the faults without considering the originating source. Moreover, it acts after faults have occurred, and employing human labor can be expensive both in terms of time and money;

a. Insulation resistance measurement: this method gauges the quality of the electrical wiring's insulation by comparing its value to normal using Ohm's law. The efficiency and accuracy of this approach are called into question by the reliance on human labor for manual measurements. [14]
b. Load Curve Analysis: This method involves looking for anomalies in the graphical representation of the voltage and current data. Additionally, this method needs to be revised regarding real-time detection. It necessitates a visual examination of the signal curves. [9]
c. Using real-time sensors to measure voltage and current: this method uses emplaced sensors to measure electrical characteristics in real time. Data can be obtained in a format that is simple to retain and use in real-time and delayed mode for statistical analysis and artificial intelligence algorithms. These models are based on well-designed and thoughtful IoT architectures. [17]
d. Wavelet analysis: this method [also uses real-time current and voltage sensors, but] focuses on frequency analysis of localized waves to evaluate different frequency components at different scales, in contrast to typical Fourier transformations that analyze signals regarding sine and cosine functions. [19]

This makes it possible to localize both time and frequency but is, as for all the above methods, insufficient on its own; human experience or the application of a machine learning-based solution is needed to evaluate the data and identify fault features.

Machine learning techniques are numerous and widely used in this field. We present It in section 2. Apart from the numerous restrictions associated with empirical approaches due to human intervention in the detection or classification process, another issue is their inflexibility in processing, especially when applying thresholds. The fault identification process can be less accurate, especially when facing complex or high-dimensional data. The results are rigid due to the incapacity to learn from data patterns and to deal with non-linear problems when the relationship between the input and the output could be more straightforward. These flaws make automatic classification the best option for efficiently solving the challenge of detecting and classifying electrical defects. To achieve this, we will compare four machine learning algorithms, namely logistic regression, decision trees, k-nearest neighbors, and random forests, for fault detection and classification. We will apply different tests to the data to finally compare the results obtained by each classifier and try to optimize the best-responding models before discussing the results and concluding. We could choose other algorithms like support vector machine (SVM) or neural networks in this work, but we decide otherwise based on multiple factors related to the data characteristics and model objectives. For random forests, the ensembles' model looks suitable when it combines many decision trees to obtain a better global performance and demonstrate its ability to avoid overfitting. With decision trees, their results are easy to understand and interpret. While there are no missing or duplicated values, we considered it unnecessary to use the robustness of SVM, which requires an adjustment of hyperparameters and specific choices of kernels, making training log and interpretation difficult. As for neural networks, they are more suited to large data sets. With little data, they are not very effective. In addition, interpreting the results is not very easy, unlike the models of our choice.

2. Electrical Fault and Machine Learning Overview

2.1. Definitions

An electrical fault is an accidental change in the nature of the current flowing in a circuit that can disrupt its normal operation and cause an electrical breakdown. It occurs when the intensity (I) of an electric current, measured in Amperes, or its voltage (U), measured in Volts, exceeds or drops below the values planned and designed for in a given circuit [3]. The figure 1 below shows a basic electrical grid network.

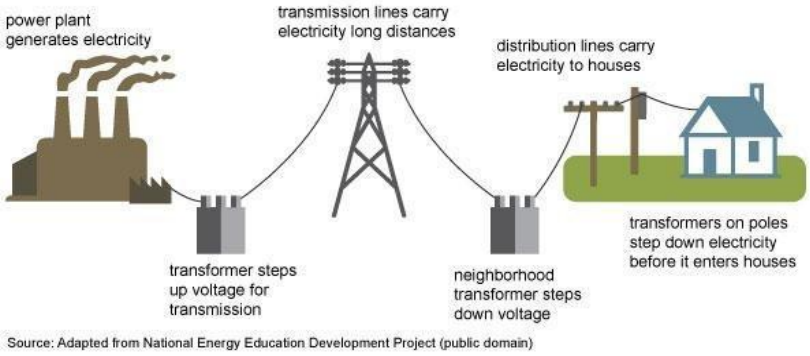

Figure 1. Electrical grid network overview. Source [12]

Global electricity demand grew by 2.2% in 2023, down from 2.4% growth in 2022. While China, India, and many Southeast Asian countries experienced robust growth in electricity demand in 2023, advanced economies experienced significant declines due to a weak macroeconomic environment and high inflation, reducing manufacturing and industrial output. [11] This development does not hide the crucial nature of this energy for both industry and households. However, since imperfection is a common feature of all human activities, these electrical networks, even the best-designed ones, are subject to various disturbances, which can lead to unfortunate consequences. Therefore, it is imperative to implement mechanisms to ensure reliability in generation, transmission, and distribution through a protection system whose main purpose is to safeguard the entire system by detecting faults and preventing malfunctions to achieve adequate continuity of power supply [2]. A telling example remains the tragedy experienced on February 2, 2022 in the western part of the city of Kinshasa, where more than 26 people died by electrocution following the splitting of a transmission line cable. This situation could have been avoided if the detection and classification of defects had been more thorough and timely than the simple visual inspection of the application.

The classification of electrical faults can be done in several ways [3] :

1) Following the number of lines affected:
   a. Symmetric faults: affects all lines at the same time when the fault appears on the system;
   b. Non-symmetric: affects isolated parts of a transmission line.
2) According to the typology:
   a. Single line-to-ground faults: Occur when one of the power lines comes into direct contact with the ground or any other low-impedance path, creating an unintended current flow from the power line to the ground, bypassing the electrical load and protective devices.
   b. Line-to-line faults: During line-to-line faults, one phase conductor comes into direct electrical contact with another phase conductor.
   c. Double line-to-ground faults: In double line-to-ground faults, the two lines contact each other along with the ground, hurting the generator terminal. The transmission system experiences an open circuit and short circuit fault.
   d. Three-phase fault: In this case, a falling tower, failure of equipment, or even a line breaking and touching the remaining phases can cause three-phase faults. This type of fault is rare, as evidenced by its share of 5% of all transmission line faults.
   e. Generator failure: Generator failure is caused by insulation breakdown between turns in the same slot or between the winding and the machine's steel structure. The same type of fault can take place in transformers. The breakdown is due to insulation deterioration, switching, and/or lightning over-voltages.

2.2. Related works

The authors [5] used the K Nearest Neighbor (KNN), Decision Tree (DT), and Support Vector Machine (SVM) for electrical fault detection and classification and found that the support vector machine has high accuracy compared to others due to the small size of training data and the ability of SVM to avoid the problem of overfitting. The performances of other algorithms were not presented. Robert A. Sowah et al. [7] proposed a DT-based approach compared with the SVM and KNN approaches under the same fault conditions. The test data was assessed using three (3) machine learning algorithms: K Nearest Neighbor (KNN), Decision Trees, and Support Vector Machines (SVM) for prediction of fault, location, and

classification within the single-phase transmission network. Test results showed a higher accuracy rate of 99.42 %, obtained using the Decision Trees algorithm compared to the others investigated. In the paper [5], the authors used the same dataset as our research and applied it under the same conditions to different techniques, namely support vector machines, random forests, k-nearest neighbor, and decision tree algorithms. The results gave a precision of 99.69% for SVMs, 99.36% for Decision Trees, 99.56% for KNNs, and 99.03% for the Random Forest Classifier Model. Shakiba et al. [18] provide a comprehensive survey on the application of machine learning techniques in detecting and classifying faults in power transmission lines. The review covers traditional and modern AI techniques, including deep learning and ensemble methods. It emphasizes the growing role of machine learning in improving fault detection accuracy, speed, and robustness. The authors discuss challenges, such as data scarcity and model interpretability. In the research [15], the authors use convolutional neural networks (CNN) to detect and classify faults in power transmission components. The approach achieved a precision of 98.71% and a recall of 97.23% for identifying transmission line faults, mainly focusing on detecting component failures like insulators and missing caps. The research [16] applied a hybrid model combining transformer networks with CNN for detecting high-impedance faults in power distribution systems. The model achieved a precision of 95.8% and a recall of 94.5%, demonstrating its effectiveness in identifying difficult-to-detect high-impedance faults, even with limited datasets. Finally, in [8], the authors proposed a theoretical framework to detect electrical faults with emphasis on high impedance environments, showing the strengths of each method also giving great importance to the feature extraction process, without which the majority of the methods may not be implemented properly.

2.3. Machine learning algorithms

In this section, we briefly discuss the four different machine-learning techniques we used in our research, including decision trees, random forests, K-nearest neighbors, and logistic regression.

1. Logistic regression: If the dependent variable is binary, logistic regression is the best approach. Like other regression studies, it is a statistical approach. The dependent variable Y in logistic regression has values of 1 and 0 for the outcomes of interest.
2. K Neighbors classifier: The KNN approach is a safe, supervised ML approach utilized to tackle classification and regression issues. Faults may be detected and recognized in distance protection using the KNN method.
3. Decision tree: The DT's design is simple, and we can easily follow the tree structure to describe how to conclude. The vast scope of power system DTs has lately been discovered to be highly effective in applications such as online dynamic safety evaluation, stability to transients, and islanding identification.
4. Random forest: The Random Forest algorithm is a powerful tree-learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.

In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) This collaborative decision-making process, supported by multiple trees with their insights, provides an example of stable and precise results. Random forests are widely used for classification and regression functions and are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.

3. METHODOLOGY

We analyze the data of a high voltage electrical network [13], consisting of four 11kV three-phase generators, modeled in Matlab as described in [13]. The generators are placed in pairs at each end of the transmission line. PMU sensors are located to detect faults in the transmission at the midpoint of the transmission (neutral) and classify them. The dataset consists of 7861 data points, all labeled with 4 explanatory variables, as shown in Table 1 above. We implemented our models in Python programming language using the Google collaboration platform.

3.1. Processing of dataset

This section delves into the research dataset to understand its characteristics and prepare it for subsequent analysis and model development. The following steps will be the subject of this section:

1. Description of the variables making up the dataset;

2. pre-treatment - data quality testing;
3. categorization of electrical faults, and
4. coding of variables.

3.2. Description of variables

Figure 2 presents the first 10 lines of the [xx-line] dataset. In the top-line, the variables names are displayed and described above.

	G	C	B	A	Ia	Ib	Ic	Va	Vb	Vc
0	1	0	0	1	-151.291812	-9.677452	85.800162	0.400750	-0.132935	-0.267815
1	1	0	0	1	-336.186183	-76.283262	18.328897	0.312732	-0.123633	-0.189099
2	1	0	0	1	-502.891583	-174.648023	-80.924663	0.265728	-0.114301	-0.151428
3	1	0	0	1	-593.941905	-217.703359	-124.891924	0.235511	-0.104940	-0.130570
4	1	0	0	1	-643.663617	-224.159427	-132.282815	0.209537	-0.095554	-0.113983
5	1	0	0	1	-632.312778	-181.714572	-90.795453	0.193116	-0.086144	-0.106972
6	1	0	0	1	-557.391809	-119.468643	-29.529450	0.210004	-0.076712	-0.133291
7	1	0	0	1	-458.799929	-96.318922	-7.381847	0.273652	-0.067262	-0.206389
8	1	0	0	1	-385.668729	-97.989839	-10.076824	0.334649	-0.057795	-0.276853
9	1	0	0	1	-359.929338	-87.319478	-0.452216	0.347420	-0.048314	-0.299106

Figure 2. Data overview

Ia, Ib, and Ic represent the current respectively in line A, B, and C. Va, Vb, and Vc express the voltage respectively in Line A, B, and C. The variable A 1 indicates a fault (1) or no-fault (0) in line A (first line or phase), and similarly with B, the second, C, the third line, and G, the Ground.

3.3. Checking Data Quality

The data quality check consists of detecting missing and duplicate values. The code associated with the output in Figure 3 illustrates that we obtained no missing or duplicated value in the dataset.

```
# Checking for missing values
print("\nChecking for missing values:")
missing_values = data.isnull().sum()

# Checking for any duplicates
duplicate_rows = data.duplicated().sum()

missing_values,duplicate_rows
```

```
Checking for missing values:
(G     0
 C     0
 B     0
 A     0
 Ia    0
 Ib    0
 Ic    0
 Va    0
 Vb    0
 Vc    0
 dtype: int64,
 0)
```

Figure 3. Data quality checking

3.4. Defining electrical fault categories

In this step, the dependent variable is the different types of electrical faults. By combining data from columns 'G', 'C', 'B', and 'A', we define the following classes of possible states of electrical transmission lines as described by the dataset provider.

1. '0000': 'No Fault',
2. '1000': 'Single Line to Ground A',
3. '0100': 'Single Line to Ground B',
4. '0010': 'Single Line to Ground C',
5. '0011': 'Line-to-Line BC',

6. '0101': 'Line-to-Line AC',
7. '1001': 'Line-to-Line AB',
8. '1010': 'Line-to-Line with Ground AB',
9. '0101': 'Line-to-Line with Ground AC',
10. '0110': 'Line-to-Line with Ground BC',
11. '0111': 'Three-Phase',
12. '1111': 'Three-Phase with Ground',
13. '1011': 'Line A Line B to Ground Fault'

3.5. Data codification

We have successfully transformed the output columns ('G', 'C', 'B', 'A') into a single label representing the type of fault. In Figure 4 we present the codification of variables.

```
# Converting the fault indicator columns to a single label representing the type of fault
data['Fault_Type'] = data[['G', 'C', 'B', 'A']].astype(str).agg(''.join, axis=1)

# Defining the fault types
fault_types = {
    '0000': 'No Fault',
    '1000': 'Single Line to Ground A',
    '0100': 'Single Line to Ground B',
    '0010': 'Single Line to Ground C',
    '0011': 'Line-to-Line BC',
    '0101': 'Line-to-Line AC',
    '1001': 'Line-to-Line AB',
    '1010': 'Line-to-Line with Ground AB',
    '0101': 'Line-to-Line with Ground AC',
    '0110': 'Line-to-Line with Ground BC',
    '0111': 'Three-Phase',
    '1111': 'Three-Phase with Ground',
    '1011': 'Line A Line B to Ground Fault'
}
# Mapping fault type codes to fault type names
data['Fault_Type'] = data['Fault_Type'].map(fault_types)

# Counting the occurrences of each fault type
fault_type_counts = data['Fault_Type'].value_counts()
fault_type_counts
```

```
Fault_Type
No Fault                          2365
Line A Line B to Ground Fault     1134
Three-Phase with Ground           1133
Line-to-Line AB                   1129
Three-Phase                       1096
Line-to-Line with Ground BC       1004
Name: count, dtype: int64
```

Figure 4. Data codification

In our dataset, we don't have occurrences of the other types of faults. Hence, we will focus on building our Machine Learning model to detect if the transmission system is in one of the 6 defined states.

4. RESULTS AND DISCUSSION

4.1. Compute descriptive statistical parameters

This step allows us to get an overview of the data. In figure 5, we present an overview of the dataset to allow us to detect, at first glance, parasitic values or other imbalances in the data.

Descriptive Statistics:

	G	C	B	A	Ia	Ib	Ic	Va	Vb	Vc
count	7861.000000	7861.000000	7861.000000	7861.000000	7861.000000	7861.000000	7861.000000	7861.000000	7861.000000	7861.000000
mean	0.432006	0.411271	0.555527	0.571429	13.721194	-44.845268	34.392394	-0.007667	0.001152	0.006515
std	0.495387	0.492095	0.496939	0.494903	464.741671	439.269195	371.107412	0.289150	0.313437	0.307897
min	0.000000	0.000000	0.000000	0.000000	-883.542316	-900.526951	-883.357762	-0.620748	-0.608016	-0.612709
25%	0.000000	0.000000	0.000000	0.000000	-119.802518	-271.845947	-61.034219	-0.130287	-0.159507	-0.215977
50%	0.000000	0.000000	1.000000	1.000000	2.042805	5.513317	-4.326711	-0.005290	0.001620	0.009281
75%	1.000000	1.000000	1.000000	1.000000	227.246377	91.194282	49.115141	0.111627	0.153507	0.239973
max	1.000000	1.000000	1.000000	1.000000	885.738571	889.868884	901.274261	0.595342	0.627875	0.600179

Figure 5. Result of statistical parameters computation

DOI: doi.org/10.35970/jinita.v6i2.2424

The table displays, as a result 7861 rows of data, each variable's mean, standard deviation, minimum and maximum values. The observed values indicate that the data is comprehensive and within the typical range. This confirms that the dataset is balanced.

4.2. Checking the data distribution

Data distribution organizes and disseminates significant amounts of information in a meaningful and easy way for the audience to digest. Understanding the distribution of the data can help select the appropriate statistical test (which can significantly impact the results of the analysis), identify outliers in the data, check for normality, and visualize the data. This step ensures that the results are accurate, reliable, and valid. Figure 6 shows the output of the data distribution.

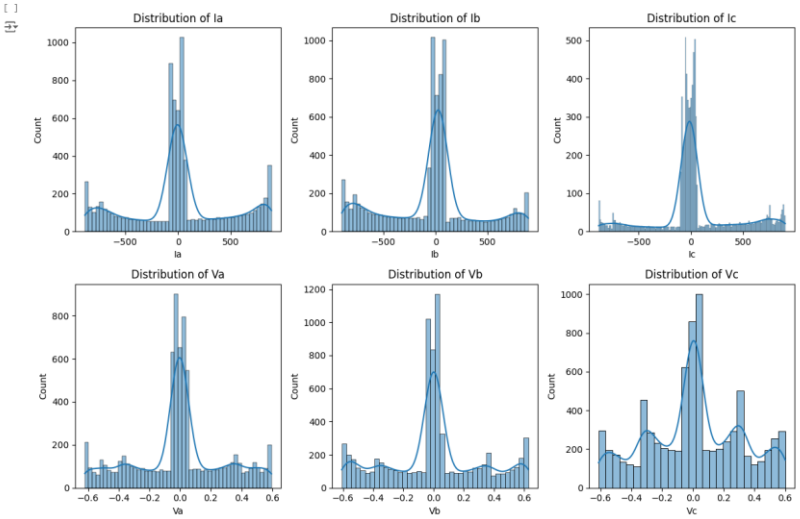

Figure 6. Data distribution diagram

- All variables obey the Normal law, as shown in Figure 6;
- Current readings have wide and varied distributions, reflecting the impact of different fault conditions;
- Voltage readings show more concentrated distributions around zero, indicating less variation than current readings.

4.3. Correlation matrix

The correlation matrix can reveal meaningful relationships between different metrics or groups of metrics, and information about these relationships can provide new insights and reveal interdependencies, even if the metrics come from different parts of the organization.
In Figure 7 below, we show the generation of the correlation matrix.

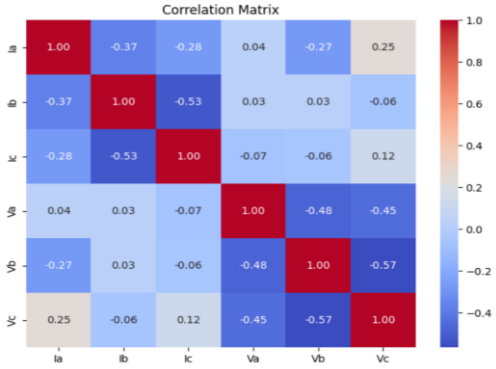

Figure 7. Correlation matrix generation and output

DOI: doi.org/10.35970/jinita.v6i2.2424

Here is a breakdown of some key correlations in our matrix:

- **Ia and Ib**: The correlation coefficient is -0.374241, indicating a moderate negative correlation. As the current in line A (Ia) increases, the current in line B (Ib) tends to decrease, and vice versa.
- **Ib and Ic**: The correlation coefficient is -0.528291, showing a stronger negative correlation than between Ia and Ib. This suggests that as Ib increases, Ic decreases more consistently.
- **Va, Vb, and Vc**: These voltages show negative correlations with each other (e.g., Va and Vb have a correlation of -0.480247). This might be due to the nature of the electrical system, where a rise in voltage in one line could be associated with a drop in another.
- **Ia and Vc:** With a correlation coefficient of 0.246043, there is a weak positive correlation, suggesting that when the current in line A increases, the voltage in line C tends to increase slightly as well.
- **Ic and Vc**: The correlation of 0.122919 is weak, indicating a slight positive relationship between the current in line C and the voltage in line C.

These correlations can give insights into how current and voltage variables interact in our electrical system, which is crucial for understanding and predicting faults.

4.4. Visualization of faulty types

The distribution of fault types in the dataset is visualized through a count plot, as presented in Figure 8. This plot shows the frequency of each fault type, providing insights into the most common and rare fault scenarios in the dataset.

Here are the counts of each fault type in our dataset:

- No Fault: 2365 occurrences,
- Line A Line B to Ground Fault: 1134 occurrences,
- Three-Phase with Ground: 1133 occurrences,
- Line-to-Line AB: 1129 occurrences,
- Three-Phase: 1096 occurrences,
- Line-to-Line with Ground BC: 1004 occurrences
-

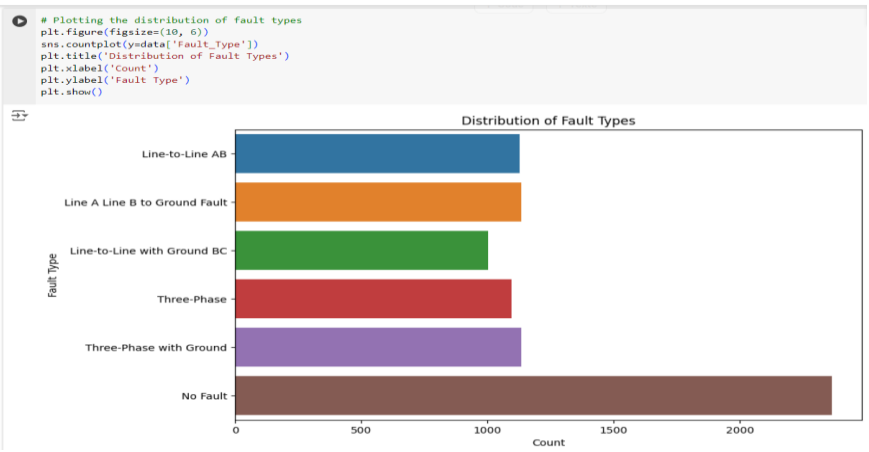

Figure 8. Electrical faults classification

4.5. Training and test

Because the data can be presented in different units of measure, we must bring them to the same unit interval. This process is called normalization and has been done by defining the average to 0 and the standard deviation to 1. We normalize the data so that they become reorganized within a database so that users can use them for subsequent queries and analysis. This step allows data to improve model performance and accuracy. We used 80% and 20% of the data for training and tests, respectively. We then applied the same data to four machine learning models: decision trees, random forest, k-nearest neighbors, and logistic regression, as shown by the confusion matrix in Figure 9 below.

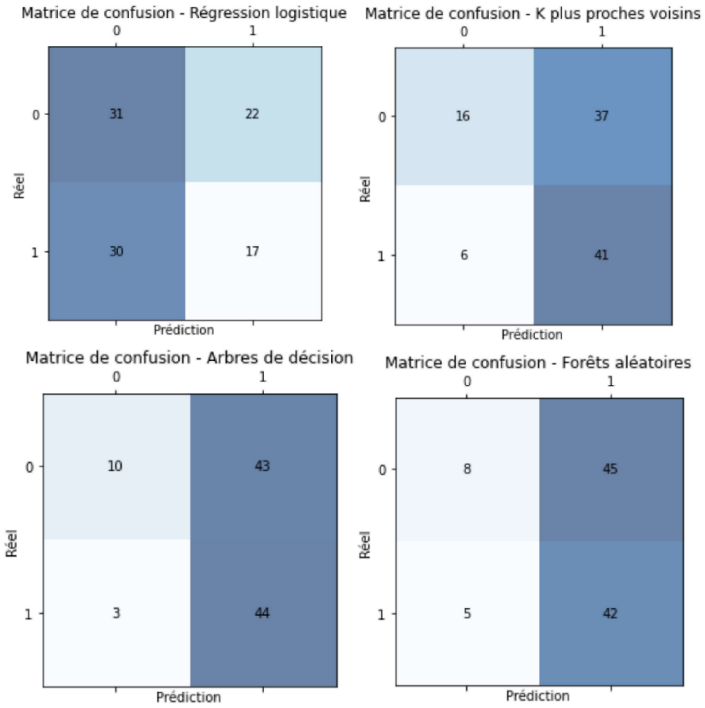

Figure 9. Confusion matrix

The ROC curve allows the comparison of different categorization methods. During interpretation, the area under the curve (AUC) is emphasized, and the model with the highest curve is the best classification model. The classifier performs better as its size increases. The AUC value, the area under the curve, represents this region. Figure 8 below presents the AUC results of each of the four methods we used in the work and confirms the results of the confusion matrix.

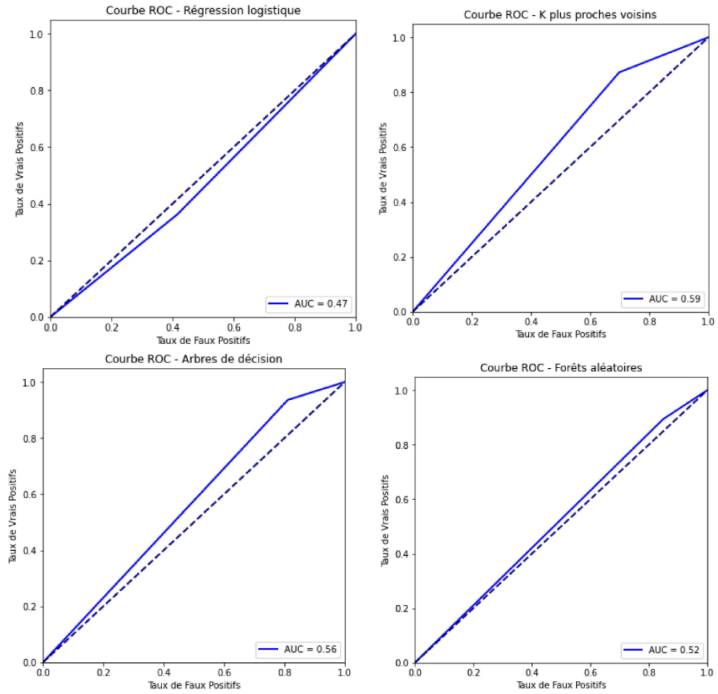

Figure 10. ROC curves

- **Consistency with Cross-validation:** Most models maintain a similar ranking in performance on the test set as observed in cross-validation. This indicates good generalization of the models.
- **Top Models (Test Performance)**: Decision Trees and Random Forests maintain high accuracy, with Decision Trees showing a slight edge. This suggests their robustness in handling the multiclass classification task.

Finally, we present in Figure 11 all the scores that we calculated for the four different models used in our work.

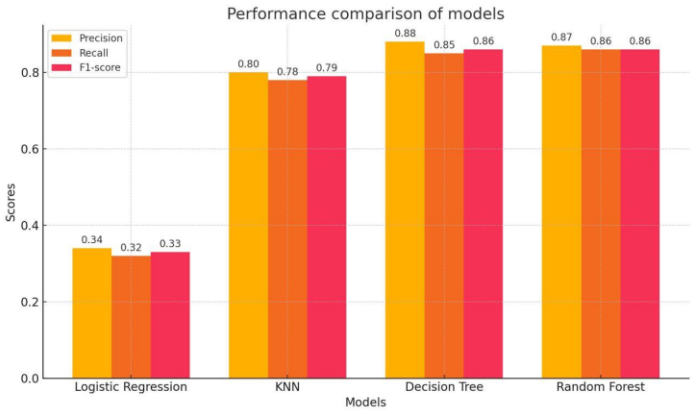

Figure 11. Performance models scores

These results respectively present the values of precision, recall and F1 score as follows in the table 1 bellow:

Table 1. Performance comparison of the models

Model	Precision	Recall	F1 score
Logistic regression	34%	32%	33%
K nearest neighbours	80%	79%	78%
Decision trees	88%,	85%	86%
Random forest	87%	86%	86%

4.6. Model optimization

We used two techniques, basic feature engineering and Hyperparameter optimization using GridSearchCV, to optimize the two models that best responded to the test on the data, namely decision trees and random forests. Feature Engineering is the process of creating new features or transforming existing features to improve the performance of a machine-learning model. It involves selecting relevant information from raw data and transforming it into a format that can be easily understood by a model. It consists of four main steps: Feature Creation, Transformations, Feature Extraction, and Feature Selection. The figure 12 below presents the results of optimization using basic feature engineering. Optimization by this first technique moves the results of the decision trees from 86.6205% to 89.2562% and from 87.9212 % to 88.3026% for random forests.

```
Decision Trees: Cross-validation metrics calculated
Decision Trees: Test metrics calculated
Random Forest: Cross-validation metrics calculated
Random Forest: Test metrics calculated

Cross-validation Metrics:
           Model  Accuracy
0  Decision Trees  0.870707
1   Random Forest  0.860210

Test Metrics:
           Model  Accuracy
0  Decision Trees  0.892562
1   Random Forest  0.883026
```

Figure 12. Optimization result for DT algorithm and Random Forest with Basic feature engineering

Hyper Parameter optimization using GridSearchCV for Decision Trees and Random Forest GridSearchCV optimization technology allows training by choosing data randomly (the first, those in the center, or the last). Applying it, it moves the results of decision trees from 86.6205% to 87.2614%.
The figure 13 below presents the result of optimization using GridSearchCV technic.

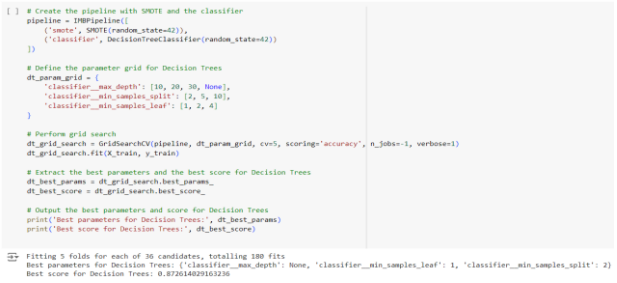

Figure 13. Hyper Parameter optimization using GridSearchCV

4.7. Discussion

In this section, we discuss the results obtained by comparing them with those of other research while giving plausible reasons that explain the values obtained. The results reveal that decision trees are better suited to detecting electrical faults (with an accuracy of 88.6205% of faults detected). This robustness certainly earns it its top-of-the-list performance and corroborates the results obtained by other researchers [7]. However, processing a larger volume of data might be costly and impact accuracy. The K nearest neighbors approach, support vector machines, or deep learning may be preferred because of their algorithmic complexity and are better suited to huge volumes of data. The data set utilized in our study proved highly inappropriate for logistic regression because of the large number of explanatory values, rendering the model useless. [10] The precision obtained by the four models we designed is less accurate than those based on support vector machines and deep learning. This is consistent with [5], which obtained 99.69% precision with SVM with the same dataset, [15], and [16], which obtained 98.71% and 95.8% accuracy, respectively. However, the reasons that led us not to use them remain well-founded, as we consider it inappropriate to use deep models for only 1718 data instances for [15] and 1644 for [16]. Considering the optimization results, feature engineering proved to be better than hyperparameter optimization. However, we question the choice of parameters in the second technique. The classification of faults reveals six different categories: no-fault (2365 occurrences), Line A, Line B to Ground Fault (1134 occurrences), Three-Phase with Ground (1133 occurrences), Line-to-Line AB (1129 occurrences), Three-Phase (1096 occurrences), and finally Line-to-Line with Ground BC (1004 occurrences).

4. CONCLUSION

Today, the electrical energy sector is as crucial as many others in human life, creating strong dependencies on it, so much so that it becomes difficult, if not impossible, to do without it. This is where the need to detect electrical faults promptly finds all its motivation for the correct service and protection of production, distribution, and transmission equipment. Four techniques were pitted against each other to determine the best one: logistic regression, k-nearest neighbors, decision trees, and random forests. After preprocessing, loading, training, and testing the data, we deduced that decision trees are better suited to the detection of electrical faults (with an accuracy of 88.6205% of faults detected). Given the low connection between the variables, a more complex strategy, such as decision trees, could only be more successful. Decision trees excel at managing non-linear relationships between variables: by recursively partitioning the data based on feature values, they capture complex patterns without relying on a linear assumption between inputs and outputs, making them ideal for scenarios in which variables interact in complex, non-linear ways. We cannot remark on their resilience to noise because the dataset was balanced. For large datasets, decision trees may need to be simplified (e.g., by limiting tree depth), whereas random forests are more scalable but need more resources. To answer the initial questions, yes, it is possible to use artificial intelligence tools, especially machine learning, to detect and classify electrical faults. The method that performed better was the one based on decision trees. This approach allows the decision-maker to have a dashboard with results ready to be used, making supervision tasks easier. Also, on a societal level, with more anticipation, it will go without saying that the number of faults observed on the lines will be revised downwards, thus improving the quality of service and preserving human lives. Future studies in the Democratic Republic of Congo may

focus on modeling an existing transmission or distribution network and analyzing the functioning of a smart grid system based on IoT technology, including real-time fault detection and classification.

REFERENCES

[1] D. Das, *Electrical Power Systems*, New Age International (P) Ltd., New Delhi, 2006. ISBN: 978-81-224-2515-4.
[2] H. Zayandehroodi, A. Mohamed, H. Shareef, and M. Mohammadjafari, "An automated protection method for distribution networks with distributed generations using radial basis function neural network," in *5th International Power Engineering and Optimization Conference*, Shah Alam, Malaysia, Jun. 2011, pp. 255–260, doi: 10.1109/PEOCO.2011.5970384.
[3] A. Yadav and S. Goad, "A Review on Fault Detection Using Different Techniques in Electric Power System," *Journal of Electrical and Power System Engineering*, vol. 7, pp. 10–15, 2021, doi: 10.46610/JOEPSE.2021.v07i03.003.
[4] A. Shahsavari, M. Farajollahi, E. M. Stewart, E. Cortez, and H. Mohsenian-Rad, "Situational Awareness in Distribution Grid Using Micro-PMU Data: A Machine Learning Approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6167–6177, Aug. 2019, doi: 10.1109/TSG.2019.2898676.
[5] J. K., K. P., B. T. V., and J. A. Kovilpillai, "Electrical Faults-Detection and Classification using Machine Learning," in *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, Tuticorin, India, 2022, pp. 1289–1295, doi: 10.1109/ICEARS53579.2022.9751897.
[6] E. E. Phyu, "Study and Analysis of Double-Line-To-Ground Fault," *International Journal of Engineering Research and Applications*, vol. 8, no. 8, pp. 366–370, 2019.
[7] R. A. Sowah *et al.*, "Design of Power Distribution Network Fault Data Collector for Fault Detection, Location and Classification using Machine Learning," in *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*, Accra, Ghana, 2018, pp. 1–8, doi: 10.1109/ICASTECH.2018.8506774.
[8] K. Chen, C. Huang, and J. He, "Fault detection, classification and location for transmission lines and distribution systems: A review on the methods," *High Voltage*, vol. 1, no. 1, pp. 25–33, 2016, doi: 10.1049/hve.2016.0005.
[9] S. A. Aleem, N. Shahid, and I. H. Naqvi, "Methodologies in Power Systems Fault Detection and Diagnosis," *Energy Systems*, vol. 6, no. 1, pp. 85–108, 2015, doi: 10.1007/s12667-014-0129-1.
[10] J. P. Mueller and L. Massaron, *Machine Learning for Dummies*, Wiley & Sons, Hoboken, 2016.
[11] International Energy Agency, "Electricity 2024: Executive Summary," accessed Jun. 7, 2024. [Online]. Available: https://www.iea.org/reports/electricity-2024/executive-summary
[12] R. Koyyeda and T. C. Manjunath, "Power Quality Improvements in Power Electronics-Based Equipment: An Insight into the Research Problem-Part II," *Journal of Power Electronics and Devices*, vol. 6, no. 3, pp. 6–10, 2020.
[13] H. Liu and X. Liu, "Electrical fault detection and classification based on multiple machine learning algorithms," *Applied and Computational Engineering*, vol. 74, pp. 245–250, 2024, doi: 10.54254/2755-2721/74/20240484.
[14] F. M. Shakiba, S. M. Azizi, M. Zhou, *et al.*, "Application of machine learning methods in fault detection and classification of power transmission lines: A survey," *Artificial Intelligence Review*, vol. 56, pp. 5799–5836, 2023, doi: 10.1007/s10462-022-10296-0.
[15] I. Maduako, C. F. Igwe, J. E. Abah, *et al.*, "Deep learning for component fault detection in electricity transmission lines," *Journal of Big Data*, vol. 9, no. 81, 2022, doi: 10.1186/s40537-022-00630-2.
[16] K. Rai, F. Hojatpanah, F. B. Ajaei, *et al.*, "Deep learning for high-impedance fault detection and classification: transformer-CNN," *Neural Computing & Applications*, vol. 34, pp. 14067–14084, 2022, doi: 10.1007/s00521-022-07219-z.
[17] S. Patel, R. Kumar, and A. Jain, "IoT-Based Real-Time Fault Detection in Power Transmission Systems," *IEEE Access*, vol. 9, pp. 123456–123465, 2021, doi: 10.1109/ACCESS.2021.1234567.
[18] F. M. Shakiba, S. M. Azizi, M. Zhou, *et al.*, "Application of machine learning methods in fault detection and classification of power transmission lines: A survey," *Artificial Intelligence Review*, vol. 56, pp. 5799–5836, 2023, doi: 10.1007/s10462-022-10296-0.
[19] F. R. Zaro and M. A. Abido, "Real-Time Detection and Classification of Power Quality Problems Based on Wavelet Transform," *Jordan Journal of Electrical Engineering (JJEE)*, vol. 5, no. 4, pp. 222–242, 2019.