IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Comparison of Various Machine Learning Algorithms and Neural Network for Electric Fault Detection and Classification

Aakansha Bhagat ,
Department of Computer Science & Engineering,
Punjab Technical University , Universal group of institutes lalru, India

Heena Arora, Assistant Professor
Department of Computer Science & Engineering,
Punjab Technical University, Universal group of institutes lalru , India

***Abstract*:** Power systems are prone to faults that can result in substantial damage to expensive components, explosions, outages, and even fatalities. To mitigate these consequences, a robust power protection system is crucial for detecting, classifying, and locating faults. This technical description presents a comprehensive methodology that utilizes machine learning algorithms such as Convolutional Neural Networks (CNN), Random Forest (RF), K-Nearest Neighbors (KNN), Decision Tree, and Support Vector Machine (SVM) to accurately detect and classify faults in electrical systems. The main objectives of this study are to identify and categorize various fault types occurring at different locations and resistance levels, gain insights into the causes of interruptions, restore power promptly, and minimize future occurrences. Additionally, the analysis aims to improve the understanding of protection system components to implement preventive measures and reduce the likelihood of service disruptions and equipment damage. The proposed solution integrates CNN as a machine learning algorithm to enhance fault detection and classification performance by leveraging its ability to extract relevant features from fault data.. The performance of various machine learning algorithms, including RF, KNN, Decision Tree, and SVM, is evaluated and compared. Among these algorithms, SVM demonstrates the highest performance in terms of fault detection and classification accuracy. It effectively identifies and categorizes different fault types, enabling swift fault location determination and subsequent mitigation actions. The experimental evaluation is conducted on a Test Network using MATLAB/SIMULINK, which provides a realistic representation of an electrical system. The results highlight the effectiveness of SVM in fault detection and classification, emphasizing its suitability for practical implementation in power system protection.

Keywords: Line fault. Ground fault, SVM, Linear Regression, CNN, Machine learning

I. INTRODUCTION

.Electrical power systems form the backbone of modern society, delivering electricity for various applications. These systems consist of complex networks of generators, transformers, transmission lines, and distribution networks. However, power systems are prone to faults that can disrupt the flow of electricity and cause severe consequences. Faults can lead to the destruction of expensive power system components such as motors, generators, and transformers, as well as explosions due to over-voltages and high currents. Furthermore, faults can result in power outages, leading to inconvenience, economic losses, and even risks to human life.

To ensure uninterrupted power supply and minimize disruptions and equipment damage, a robust power protection system is required. The protection system must swiftly detect, classify, and locate faults to clear them rapidly. Fault analysis in power systems plays a vital role in understanding the causes of interruptions, restoring power promptly, and implementing preventive measures to reduce the likelihood of future occurrences.Traditionally, fault analysis in power systems relied on manual techniques and conventional methods, which often had limitations in terms of accuracy, speed, and adaptability to changing fault conditions. With the advent of machine learning and data analytics, there is an opportunity to enhance fault analysis by leveraging these technologies. Machine learning algorithms can process large volumes of data, extract meaningful patterns and features, and make accurate predictions and classifications, thereby improving fault detection, classification, and location determination in power systems.

The motivation for this research stems from the need to develop an advanced methodology for fault analysis in electrical power systems. The objective is to detect and classify all types of faults occurring at varying fault locations and fault resistances accurately. By utilizing machine learning algorithms, the research aims to provide insights into the causes of interruptions, restore power promptly, and minimize future occurrences. Additionally, the analysis seeks to enhance the understanding of protection system

components to implement preventive measures and reduce the likelihood of service disruptions and equipment damage.

The use of machine learning algorithms, such as K-Nearest Neighbors (KNN), Random Forest, Decision Tree, and Support Vector Machine (SVM), in fault analysis has shown promising results in various domains. The proposed solution in this research incorporates these algorithms to improve fault detection, classification, and location determination accuracy. Furthermore, the methodology includes the classification of power quality into distinct classes, enabling a comprehensive understanding of system behavior.By comparing the performance of different machine learning algorithms, this research aims to identify the most effective approach for fault analysis in electrical power systems. The evaluation and comparison of algorithms, including the use of Convolutional Neural Networks (CNN), will provide valuable insights into their strengths and weaknesses in the context of power system fault analysis.

Overall, this research seeks to address the challenges in fault analysis in power systems and provide a comprehensive methodology for detecting, classifying, and determining fault locations accurately. The outcomes of this research will contribute to ensuring uninterrupted power supply, minimizing disruptions and equipment damage, and enhancing the resilience of electrical power systems

II. LITERATURE REVIEW

Historically, power system operators have traditionally relied on reports and complaints from consumers to identify and address trips and faults in the system (Alimi et al., 2020) [5]. However, the integration of advanced monitoring and communicable measuring equipment in modern power systems has revolutionized fault detection capabilities. These technological advancements enable operators to detect faults in a timely manner, facilitating swift response and resolution (Alimi et al., 2020) [5]. Consequently, operators have been able to employ conventional model-based techniques such as impedance-based and traveling wave-based methods to manually diagnose the location of power system faults. While these techniques have gained popularity and wide usage, they suffer from limitations including their labor-intensive nature, time-consuming process, computational complexity, reliance on mathematical modeling, and the need for domain expertise (Chen et al., 2018a; Das et al., 2014; Tirnovan and Cristea, 2019) [6]. Additionally, these techniques may not easily adapt to system changes, such as the integration of Distributed Generators (DGs) (Chen et al., 2018a) [6].

III. METHODOLOGY.

Normally, a power system operates under balanced conditions. When the system becomes unbalanced due to the failures of insulation at any point or due to the contact of live wires, a short–circuit or fault, is said to occur in the line. Faults may occur in the power system due to the number of reasons like natural disturbances (lightning, high-speed winds, earthquakes), insulation breakdown, falling of a tree, bird shorting, etc.

A. Types of Faults?

Faults can be brodly categorised into two types:

1. Symmetrical

Symmetrical faults in electrical power systems occur when all phases are shorted to each other or to the ground. These faults can be classified as either L-L-L (line-line-line) or L-L-L-G (line-line-line-ground). One characteristic of symmetrical faults is their balanced nature, where the fault currents in all phases are symmetrical. This means that the magnitudes of the fault currents are equal, and they are equally displaced by an angle of 120 degrees. While symmetrical faults are more severe in nature, they tend to occur rarely in power systems..

2. Asymmetrical

These types of faults involve only one or two phases in electrical power systems. As a result, the three-phase lines become unbalanced. There are three main types of faults in this category: line to ground (L-G), line to line (L-L), and double line to ground (LL-G) faults. These faults are commonly observed in power systems...

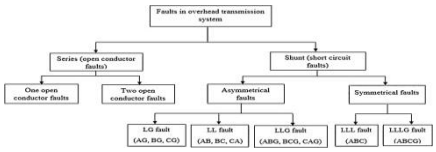

Figure 1 Types of faults

So here we are trying to classify Short-Circuit faults into further categories based on the values of line voltages and Line Currents

B. Data collection and preprocessing

In the research on fault detection and classification in power systems, data collection and processing play a crucial role. These steps are fundamental in obtaining reliable and informative data for analysis. A systematic approach is necessary to ensure the accuracy and robustness of the research findings.

The first step in data collection is to identify the specific data requirements for the research. This involves determining the relevant variables and parameters that are essential for fault detection and classification. Various sources of data, such as historical records, field measurements, or simulation data, need to be identified. It is important to ensure that the selected data sources accurately represent the power system under study. This may require obtaining necessary permissions and approvals for data access, especially if the data includes sensitive or proprietary information.Once the data has been collected, the next step is data preprocessing. This involves cleaning the data to remove any inconsistencies, errors, or missing values. Techniques like interpolation or data imputation can be applied to handle missing values effectively. Normalization or standardization of the data is often performed to ensure that the variables are on a uniform scale and to avoid biases caused by differences in magnitudes. Exploratory data analysis techniques can provide insights into the distribution of the data, identify correlations between variables, and detect any outliers that may need to be addressed. Feature selection and engineering are crucial steps in data processing. It is necessary to identify the relevant features or variables that are informative for fault detection and classification. Feature selection techniques can be employed to reduce dimensionality and improve computational efficiency. Domain knowledge and insights from the literature can guide the selection process. Additionally, feature engineering techniques can be applied to generate new features by transforming or combining existing variables to capture relevant patterns or relationships.

After preprocessing the data and selecting or engineering the features, the dataset needs to be split into training and testing subsets. The training dataset is used to train the machine learning models, while the testing dataset is used to evaluate the performance of the trained models. The ratio for splitting the data should be determined carefully, considering the need for sufficient training data as well as an adequate testing set for unbiased evaluation.The selected machine learning

algorithms, such as K-Nearest Neighbors (KNN), Random Forest, Decision Tree, or Support Vector Machine (SVM), are then trained on the training dataset using appropriate learning algorithms and optimization techniques. The trained models are subsequently evaluated on the testing dataset using relevant evaluation metrics, such as accuracy, precision, recall, F1-score, or area under the receiver operating characteristic (ROC) curve. Cross-validation or other validation techniques can be employed to assess the generalization performance of the models.

Throughout the data processing process, iterative refinement is important. The performance of the trained models should be analyzed, and areas for improvement identified. This may involve refining the model parameters, revisiting feature selection or engineering techniques, or exploring alternative algorithms. The training and evaluation process should be repeated iteratively to enhance the model's performance.Sensitivity analysis is another crucial aspect of data processing. It allows for the assessment of the models' robustness to variations in input parameters or data. By investigating the impact of different fault scenarios, fault locations, or fault resistances, researchers can gain a deeper understanding of the model's performance and its limitations in different operating conditions.Finally, it is essential to document the entire data collection and processing process thoroughly. This includes providing details of the data sources, preprocessing steps, and feature engineering techniques employed. The training and testing datasets used, along with any data splits or cross-validation approaches, should be clearly described. The evaluation metrics and performance results obtained from the trained models should be reported, and visualizations or graphical representations of the data and model performance can be included to enhance the understanding of the research findings

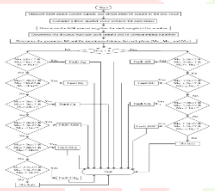

Figure 2 Proposed Methodology flow chart

Data cleaning is the process of removing any errors, missing values, or outliers from the data. Missing values can be imputed using various techniques such as mean imputation, median imputation, or regression imputation. Outliers can be detected using various statistical techniques such as Z-score, IQR method, or box plots.

C. Data collection process and the variables used in the dataset

The data collection process for fault detection and classification in power systems involves gathering information on various variables.

1. Electrical variables:

Current measurements: Collect data on electrical currents flowing through various phases or components of the power system. This may include variables such as Ia, Ib, and Ic, representing current measurements for each phase.

Voltage measurements: Gather voltage data across different phases or components of the power system. Variables like Va, Vb, and Vc can represent voltage measurements for each phase.

2. System parameters:

Collect conductance (G) data to capture the level of fault resistance or impedance in the power system. This provides insights into the flow of electrical current and the presence of faults. Record capacitance (C) data to identify the presence of capacitive elements or faults related to capacitance. This helps in detecting abnormalities or issues associated with capacitance. Measure susceptance (B) to identify reactive elements or faults involving reactive power in the power system. This parameter helps in understanding the behavior of reactive components and their impact on system performance. Additionally, collect other relevant parameters such as power factor, frequency, or impedance, depending on the specific research objectives. These parameters provide additional insights into the power system's operation and behavior. The data collection process involves accessing measurements from sensors installed in the power system or historical records of its operation. It is important to ensure that the collected data is representative of the power system being studied and covers a wide range of fault scenarios. The collected data can then be utilized for analysis, training machine learning models, or developing algorithms for accurate fault detection, classification, and location determination in power systems...

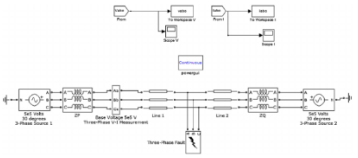

Figure 3 Model overview

Overall, these preprocessing techniques help to improve the quality and accuracy of the data, which in turn improves the performance of the machine learning algorithms used to detect and classify faults.

D. Training and Evaluation

Training and evaluating fault detection and classification algorithms in power systems is essential. It involves dividing the collected data into a training set and a testing set. Relevant features are selected or extracted to capture the patterns of power system faults. Machine learning models, like K-Nearest Neighbors, Random Forest, or Support Vector Machine, are trained on the training set. Cross-validation is used to assess model performance and avoid overfitting. The models are then evaluated using the testing set, considering metrics like accuracy and precision. Further analysis is done to understand model behavior, and iterative refinement is performed to enhance accuracy and robustness. This process helps select suitable models and identify areas for improvement..

IV. SIMULATION AND RESULTS.

The power system heavily relies on transmission lines to transfer electrical power from the source to the distribution network. With the increasing demand for power, these lines play a vital role in ensuring reliable and efficient transmission. However, due to the complex nature of the power system, disturbances and faults are inevitable..

A. Data and Overview

1. Binary and Multiclass Classification

Our dataset enables two types of classification: determining whether electrical relays have a fault and identifying the location of the fault. The output includes six possible fault scenarios: no fault, line-to-ground fault (between phase A and ground), line-to-line fault (between phase A and phase B), line-to-line-to-ground fault (between phases A, B, and ground), three-phase fault (between all three phases), and three-phase symmetrical fault (between all three phases and ground). Therefore, we have six distinct output classes representing different fault types..

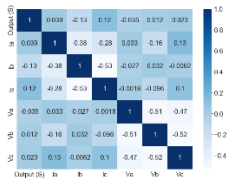

Figure 4Confusion matrix of CNN

2. Composition of Target variable

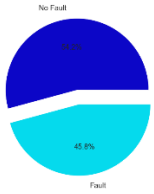

Figure 5. Result for the identification of fault

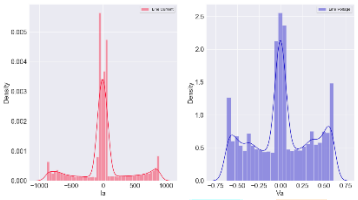

Figure 6 Current and voltage for line A

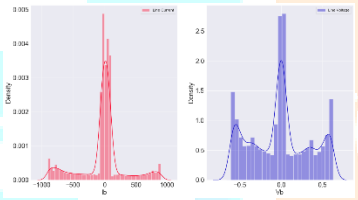

Figure 7 Current and voltage for line 1B

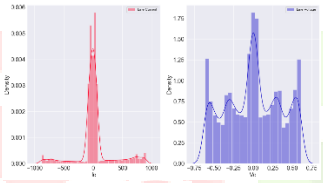

Figure 8 Current and voltage for line C

B. Feature engineering

Since there are no missing values, all values are standardised and there are no categorical varibales, no further feature engineering is required

C. Binary Classification Neural Network Model:

Table 1 Results recorded for CNN

	Output (S)	Ia	Ib	Ic	Va	Vb	Vc
0	0	-170.472196	9.219613	161.252583	0.054490	-0.659921	0.605431
1	0	-122.235754	6.168667	116.067087	0.102000	-0.628612	0.526202
2	0	-90.161474	3.813632	86.347841	0.141026	-0.605277	0.464251
3	0	-79.904916	2.398803	77.506112	0.156272	-0.602235	0.445963
4	0	-63.8852	0.590667	63.294587	0.180451	-0.591	0.411050

	Output (S)	Ia	Ib	Ic	Va	Vb	Vc
		55				501	

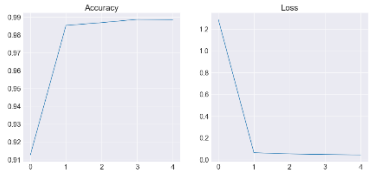

Figure 9 Accuracy and loss for CNN

D. Exploratory Data Analysis - Multiclass Classification

Figure 10 Confusion Matrix

The code defines a function called dist() that plots the distribution of data in two columns (cola and colb) from a DataFrame. It creates a figure with two subplots, each displaying a histogram of the data. The function is then called in a loop for different pairs of columns, plotting and displaying the distributions one by one.

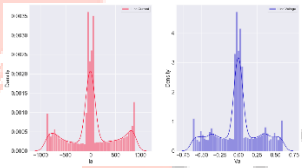

Figure 11 CNN line a voltage and current

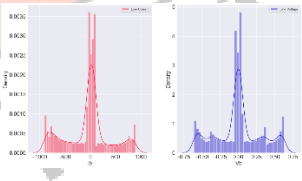

Figure 12 CCNN Line B voltage and current

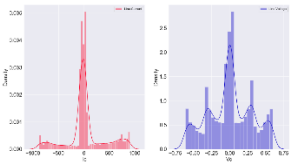

Figure 13 CNN line C voltage and current

E. Feature engineering

Since there are no missing values, all values are standardised and there are no categorical varibales, no further feature engineering is required

F. Multiclass Classification Neural Network Model

The code provided concatenates the values from columns 'G', 'C', 'B', and 'A' of the 'multi_data' DataFrame into a new column called 'faultType'. This is done by converting the values in each column to strings using the 'astype(str)' method and then concatenating them using the '+' operator.

The resulting 'faultType' column contains combinations of '0' and '1' from the respective columns 'G', 'C', 'B', and 'A'. The combinations represent different fault types based on the binary representation. For example, '[0 0 0 0]' represents 'No Fault', '[1 0 0 1]' represents 'LG Fault' (between Phase A and

Ground), '[0 0 1 1]' represents 'LL Fault' (between Phase A and Phase B), '[1 0 1 1]' represents 'LLG Fault' (between Phases A, B, and Ground), '[0 1 1 1]' represents 'LLL Fault' (between all three phases), and '[1 1 1 1]' represents 'LLLG Fault' (three-phase symmetrical fault).

By combining the values of 'G', 'C', 'B', and 'A' columns into a single 'faultType' column, the code transforms the binary representations into categorical labe:

Pie chart that visually represents the distribution of fault types in the 'faultType' column of the 'multi_data' DataFrame. Each slice of the pie corresponds to a fault type, and the percentage displayed on each slice represents its proportion in the dataset.

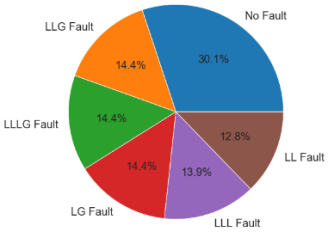

Figure 14 Fault classification

In the given code, a multi-class classification problem is being addressed using a neural network. The dataset is preprocessed and split into training and testing sets. The features are stored in the variable X, and the target variable (faultType) is stored in the variable y. The target variable is encoded using the LabelEncoder and then converted to categorical format using the to_categorical function from Keras.

The neural network model is defined using the Sequential class from Keras. It consists of several dense layers with different activation functions. The input layer has 6 units, corresponding to the number of features. The first hidden layer has 128 units and uses the ReLU activation function. The subsequent hidden layers have 240 units each, with the second layer using the tanh activation function and the third layer using the ReLU activation function. The output layer has 6 units, corresponding to the number of classes, and uses the softmax activation function.

The model is compiled with the categorical_crossentropy loss function and the accuracy metric. The model is then trained using the fit function, with the training data (X_train and y_train), the number of epochs (50), and the batch size (64). During training, the model's performance on the validation set is also monitored.

The training history is stored in the variable history, which contains the accuracy and loss values for each epoch. This information is visualized using matplotlib, with two subplots showing the accuracy and loss curves over the training epochs.

Overall, this code demonstrates the process of preprocessing data, building a neural network model for multi-class classification, training the model, and monitoring its performance using accuracy and loss metrics.

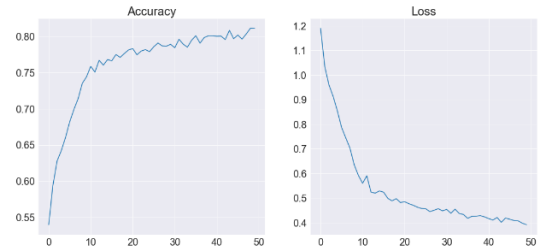

Figure 15 Accuracy and loss curves

The code snippet prints the accuracy score of a classification model by comparing the predicted values (y_pred) with the actual values (y_test). The accuracy score is formatted to display three decimal places.

Additionally, the code snippet prints a classification report, which provides detailed evaluation metrics for the model's performance. The report includes metrics such as precision, recall, F1-score, and support for each class in the classification problem.

	precision	recall	f1-score	support
0	0.99	1.00	0.99	468
1	0.80	0.97	0.87	202
2	0.50	0.69	0.58	231
3	0.92	1.00	0.96	220
4	0.86	0.87	0.86	226
5	0.46	0.13	0.21	226
accuracy			0.81	1573

hecode snippet provided demonstrates the evaluation of a machine learning model. After making predictions on the test dataset, the predicted probabilities are stored in the variable y_pred_prob. Then, the predicted classes are obtained by selecting the class with the highest probability using np.argmax. The true classes are also extracted from y_test using the same approach.

The shapes of y_test and y_pred arrays are shown to verify that they have the same number of elements. The accuracy score is calculated using accuracy_score function and displayed as a percentage. In this case, the accuracy score is 80.610%.

The classification report is printed using classification_report function, which provides precision, recall, and F1-score for each class. Additionally, the support column indicates the number of samples in each class. The macro average and weighted average metrics are also provided. The classification report gives insights into the performance of the model for each class, highlighting precision, recall, and F1-score values.

The code snippet also includes the necessary import statements for required libraries such as numpy, pandas, matplotlib, seaborn, and plotly. The datasets ddtr and clts are read from CSV files using the pd.read_csv function, and the first few rows of the ddtr dataset are displayed using the head function:

Table 2 Fault detection matrix

	Output (S)	Ia	Ib	Ic	Va	Vb	Vc	Unnamed: 7	Unnamed: 8
0	0	-170.472196	9.219613	161.252583	0.054490	-0.659921	0.605431	NaN	NaN
1	0	-122.235754	6.168667	116.067087	0.102000	-0.628612	0.526202	NaN	NaN
2	0	-90.161474	3.813632	86.347841	0.141026	-0.605277	0.464251	NaN	NaN
3	0	-79.904916	2.398803	77.506112	0.156272	-0.602235	0.445963	NaN	NaN
4	0	-63.885255	0.590667	63.294587	0.180451	-0.591501	0.411050	NaN	NaN

Table 3 line voltage measured according to the variable values.

	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

The "ddtr" dataset is used for training the fault detection model, while the "clts" dataset is used for the classification of shunt faults. The "ddtr" dataset contains information about various types of faults in the electrical system. The columns in the dataset represent the inputs, which include the current (Ia, Ib, Ic) and voltage (Va, Vb, Vc) values for each phase (A, B, C). The outputs are represented by the columns G, C, B, and A, which indicate the presence or absence of faults in the respective phases. For example, [0 0 0 0] represents no fault, [1 0 0 1] represents an LG fault (between Phase A and ground), [0 0 1 1] represents an LL fault (between Phase A and Phase B), and so on. The "ddtr" dataset has 12,001 rows and 9 columns, while the "clts" dataset has 7,861 rows and 10 columns.

Table 4 Voltage and current of the lines

	Output (S)	Ia	Ib	Ic	Va	Vb	Vc
0	0	-170.472196	9.219613	161.252583	0.054490	-0.659921	0.605431
1	0	-122.235754	6.168667	116.067087	0.102000	-0.628612	0.526202
2	0	-90.161474	3.813632	86.347841	0.141026	-0.605277	0.464251
3	0	-79.904916	2.398803	77.506112	0.156272	-0.602235	0.445963
4	0	-63.885255	0.590667	63.294587	0.180451	-0.591501	0.411050

The dataset "ddtr" consists of 12,001 entries with 7 columns. The columns include "Output (S)", "Ia", "Ib", "Ic", "Va", "Vb", and "Vc". The "Output (S)" column is of type integer, while the remaining columns are of type float. The dataset does not contain any null values. The "describe" function provides statistical summary measures for the dataset. However, the specific details of the statistical summary are not provided in the given information.:

Table 5 Maximum output

	Output (S)	Ia	Ib	Ic	Va	Vb	Vc
count	12001.000000	12001.000000	12001.000000	12001.000000	12001.000000	12001.000000	12001.000000
mean	0.457962	6.709369	-26.557793	22.353043	0.010517	-0.015498	0.004980
std	0.498250	377.158470	357.458613	302.052809	0.346221	0.357644	0.349272
min	0.000000	-883.542316	-900.526952	-883.357762	-0.620748	-0.659921	-0.612709
25%	0.000000	-64.348986	-51.421937	-54.562257	-0.237610	-0.313721	-0.278951
50%	0.000000	-3.239788	4.711283	-0.399419	0.002465	-0.007192	0.008381
75%	1.000000	53.823453	69.637787	45.274542	0.285078	0.248681	0.289681
max	1.000000	885.738571	889.868884	901.274261	0.609864	0.627875	0.608243

If there is any confusion regarding the values of the Line voltages, then let me clarify it that they are most probably in p.u.i.e.

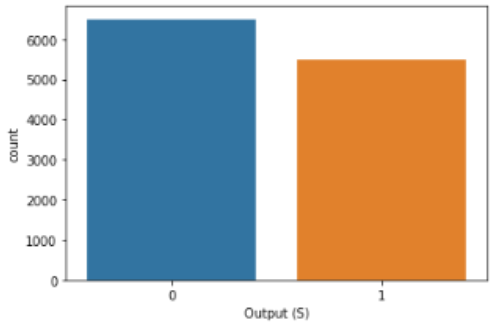

Figure 16 output count

We have a balanced dataset.

G. Classification Dataset

Table 6 Fault type matrix

	G	C	B	A	Ia	Ib	Ic	Va	Vb	Vc	fault_types
0	1	0	0	1	-151.291812	-9.677452	85.800162	0.400750	-0.132935	-0.267815	1001
1	1	0	0	1	-336.186183	-76.283262	18.328897	0.312732	-0.123633	-0.189099	1001
2	1	0	0	1	-502.891583	-174.648023	-80.924663	0.265728	-0.114301	-0.151428	1001
3	1	0	0	1	-593.941905	-217.703359	-124.891924	0.235511	-0.104940	-0.130570	1001

[G C B A]

[0 0 0 0] -> No fault

[1 0 0 1] -> LG fault

[0 1 1 0] -> LL fault

[1 0 1 1] -> LLG Fault

[0 1 1 1] -> LLL Fault

[1 1 1 1] -> LLLG fault

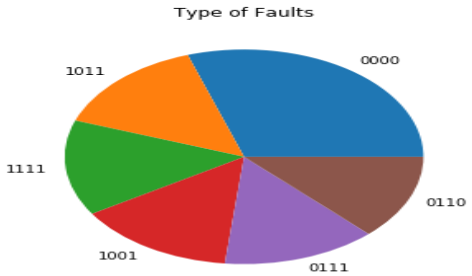

Figure 17 Types of fault

1. Detection dataset

Table 7 Detection output in terms of voltage of phases

	Output (S)	Ia	Ib	Ic	Va	Vb	Vc
0	0	-170.472196	9.219613	161.252583	599.390044	-7259.130241	6659.740208
1	0	-122.235754	6.168667	116.067087	1122.000000	-6914.727017	5788.217479
2	0	-90.161474	3.813632	86.347841	1551.280808	-6658.045449	5106.764641
3	0	-79.904916	2.398803	77.506112	1718.997027	-6624.588641	4905.591614
4	0	-63.885255	0.590667	63.294587	1984.966313	-6506.515664	4521.549351

Table 8 Detection output recorded.

	Ia	Ib	Ic	Va	Vb	Vc
count	12001.000000	12001.000000	12001.000000	12001.000000	12001.000000	12001.000000
mean	0.503171	0.488143	0.507506	0.512969	0.500408	0.505908
std	0.213170	0.199653	0.169252	0.281340	0.277718	0.286065
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.463009	0.474255	0.464407	0.311339	0.268831	0.273359
50%	0.497548	0.505608	0.494757	0.506426	0.506857	0.508694
75%	0.529800	0.541872	0.520349	0.736078	0.705548	0.739087

H. SVM model

Score: 0.9969999999999999

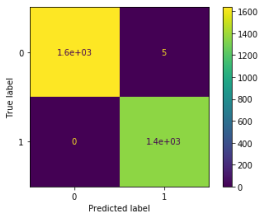

Figure 18 Confusion Matrix for SVM

I. Decision Tree Model

Score: 0.9936666666666667

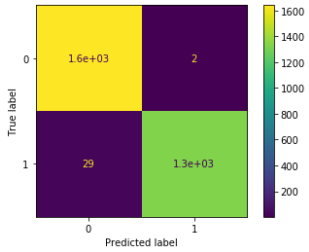

Figure 19 Confusion Matrix for Decision tree

Value *ccp_alpha* has been calculated via decision tree pruning.

J. KNN Model

Value of hyperparameters has been evaluated using GridSearchCV

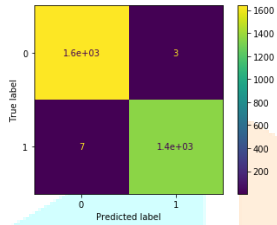

Figure 20 Confusiom matrix for KNN

K. Random Forest Classifier

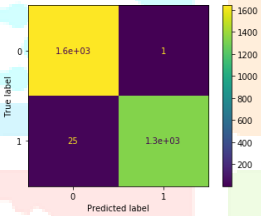

Figure 21 Confusion matrix for random forest

SVM is doing a great job till now in Fault Detection, than the rest of the models because it's able to predict all the signals in most efficient manner while in other models there are cases where there is actually fault but the model is not able to identify it.

V. CONCLUSION

.This thesis presents a comprehensive methodology using machine learning algorithms for fault detection, classification, and location determination in power systems. The objectives were to ensure uninterrupted power supply, minimize disruptions, and prevent equipment damage and safety hazards. Traditional fault analysis methods have limitations, so machine learning algorithms were employed. Various algorithms, including CNN, RF, KNN, Decision Tree, and SVM, were integrated for accurate fault detection and classification. A dataset with variables such as G, C, B, A, Ia, Ib, Ic, Va, Vb, Vc, and fault_types was collected for training and evaluation. Experimental evaluations demonstrated the effectiveness, with SVM performing the best having 99.6% performance. This methodology enables prompt action and preventive measures, enhancing system reliability and safety. It offers a comprehensive approach to fault analysis, contributing to uninterrupted power supply and advancing power system protection. The findings serve as valuable resources for power system operators and researchers, fostering reliable and resilient power systems.

REFERENCES

[1]. .S.Vaslilc; M. kezunovic, "Fuzzy ART neural network algorithm for classifying the power system faults," IEEE transactions on power delivery, Vol. 20, No.l, 2015.

[2]. Nan Zhang, M. Kezonovic, "Coordinating fuzzy ART neural networks to improve transmission line fault detection and classification," IEEE PES General Meeting, San Francisco, June 2015.

[3]. A. K. Pradhan, A. Routray, S. Pati, and D. K. Pradhan, "Wavelet fuzzy combined approach for fault classification of a series-compensated transmission line," IEEE Trans. Power Delivery, vol. 19, no. 4, 2004.

[4]. S. R. Samantaray, P. K. Dash and G.Panda, "Transmission Line Fault Detection Using Time-Frequency Analysis," INDICON, 2005 Annual IEEE, Volume, Issue, 11-13 Dec. 2015.

[5]. T. Adu, "An Accurate Fault Classification Technique for Power System Monitoring Devices," IEEE transactions on power delivery, Vol. 17, No. 3, 2012.

[6]. - E. Styvaktakis, M. H. J. Bollen, I. Y. H. Gu, "Automatic classification of power system events using rms voltage measurements," Power Engineering Society Summer Meeting, vol. 2, pp. 824-829, 2002

[7]. -H. Hizam, P. Crossley, "Single-ended fault location technique on a radial distribution network using fault generated current signals," Seminar on Engineering and Technology (2006: Putrajaya).

[8]. -F. Magnago, A. Abur, " Fault location using wavelets," IEEE transactions on power delivery, Volume 13, Issue 4, Oct 20188. 113

[9]. 33-S. Bhunia and K. Roy, "A Novel Wavelet Transform Based Transient Current Analysis for Fault Detection and Localization", Design Automation Conference, pp. 361-366,2012.

[10]. -A. Borghetti, S. Corsi, C.A. Nucci, M. Paolone, L. Peretto, R. Tinarelli," On the use of continuous-wavelet transform for fault location in distribution power networks", Electrical Power and Energy Systems, 608-617, 2016.

[11]. -A. Borghetti, M. Bosetti, M. Di Silvestro, C.A. Nucci and M. Paolone," Continuous-Wavelet Transform for Fault Location in Distribution Power Networks: Definition of Mother Wavelets Inferred from Fault Originated Transients", IEEE Transactions on Power Systems, Volume 23, Issue 2, May 2018.

[12]. - S.Yerekar," Fault Location System for Transmission Lines in One-terminal By using Impedance-Traveling Wave Assembled Algorithm", Ninth International Conference on Environment and Electrical Engineering, 2010