Bangladesh University of Engineering and Technology

Department of Electrical and Electronic Engineering

Course No: EEE-306 **Course Title: Power System I Laboratory**

Project Title

Fault Detection and Classification in Electrical Power Transmission System Using Artificial Neural Network and Machine Learning

Submitted To:

Munia Ferdoushi	Shoilie Chakma
Assistant Professor Department of EEE BUET	Lecturer Department of EEE BUET

Submitted By:

M. Fahmin Rahman	1806001
Shomen Kundu	1806002
Sheikh Iftekhar Ahmed	1806011
Auronno Ovid Hussain	1806012
Raihan Mahmud Chowdhury	1806014
Group: 01 Section: A1 Level: 03, Term: 01	

Abstract:
The goal of our project is to detect electrical fault location on electrical power transmission line and to classify the fault using artificial neural network. The instantaneous values of the three phase currents and voltages measured from one side of the transmission line has been used to produce datasets for training our models. We have also used several machine learning algorithms to make a comparison between the performances of the different models. The comparison shows that artificial neural network performs satisfactorily with good percentiles.
For generating data, we have simulated a small power system with one transmission line in Simulink, MATLAB. We introduced various types of faults at different locations on the transmission line for some electrical cycles and used the instantaneous data to produce datasets.

Motivation:
The electrical power system consists of so many different complex dynamic and interacting elements, which are always prone to disturbance or an electrical fault. The use of high-capacity electrical generating power plants and concept of grid, i.e., synchronized electrical power plants and geographical displaced grids, required fault detection and operation of protection equipment in minimum possible time so that the power system can remain in stable condition. The faults on electrical power system transmission lines are supposed to be first detected and then be classified correctly and should be cleared in least possible time. The protection system used for a transmission line can also be used to initiate the other relays to protect the power system from outages. A good fault detection system provides an effective, reliable, fast and secure way of a relaying operation. The application of a pattern recognition technique could be useful in discriminating the faulty and healthy electrical power system. It also enables us to differentiate among three phases which phase of a three-phase power system is experiencing a fault.
Several pattern recognition techniques have been used to detect fault in transmission lines. Some of them are fuzzy logic, adaptive neuro-fuzzy interface system (ANFIS), unconventional synchronized method, wavelet transformation etc. Some of these methods are either complex or provide low performance. As far as ANNs are considered they exhibit excellent qualities such as normalization and generalization capability, immunity to noise, robustness and fault tolerance. Therefore, the declaration of fault made by ANN-based fault detection method should not be affected seriously by variations in various power system parameters.

Methodology:

In this part, we describe the planning of our developed system, Simulink model development and data generation.

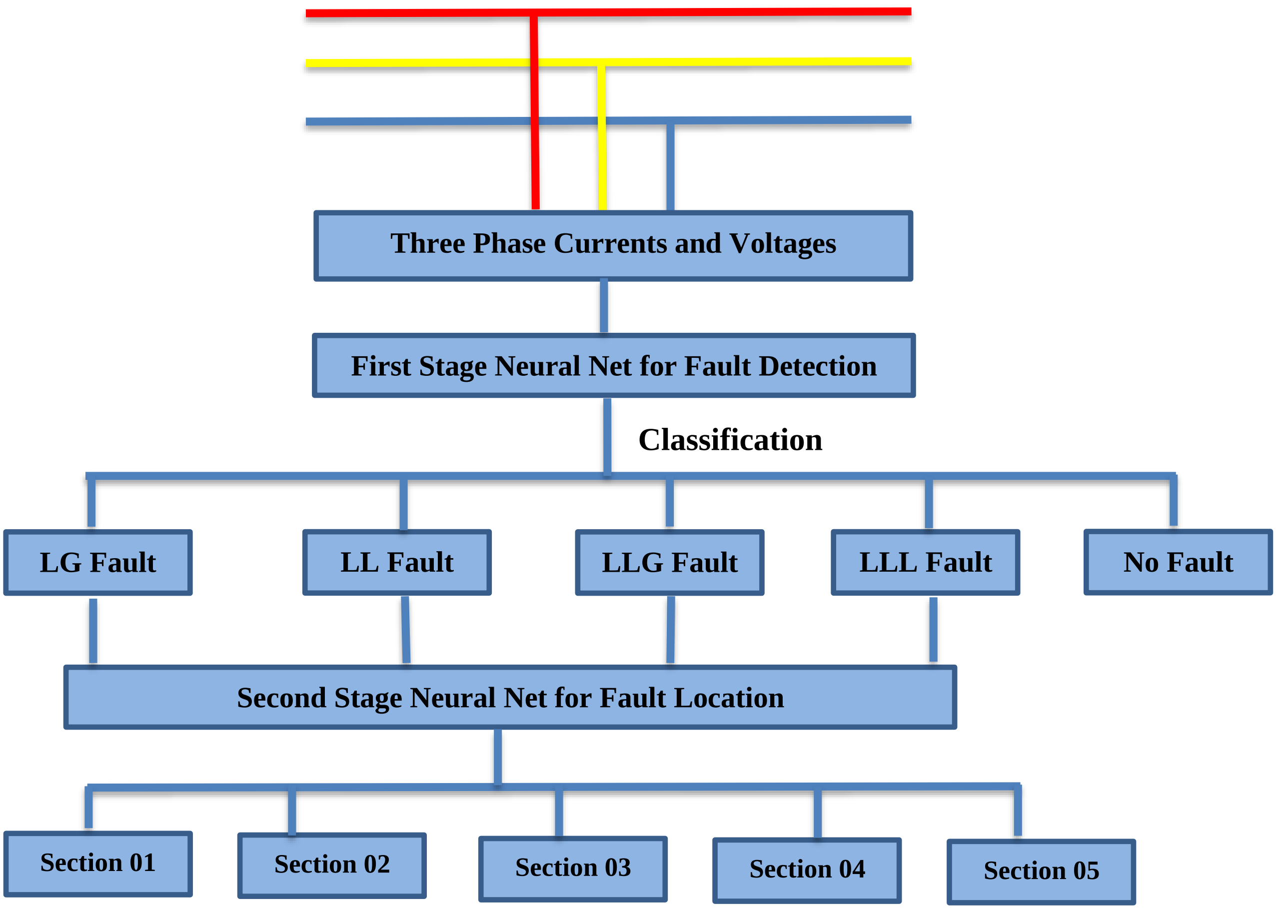
System Planning:

The generation of dataset greatly depends on how our system will work. Our system takes the instantaneous currents and voltages of the three phases as input. From this, it decides whether there is fault on the line and where the fault is. So, the first part is fault detection. This is done by classifying the fault into five classes, namely LG (line to ground), LL (line to line), LLG (double line to ground), LLL (three phase fault) and No fault. This classification is done by the first stage neural network. If this stage indicates that there is fault on the line, then the data will be processed by the second stage for fault location detection.

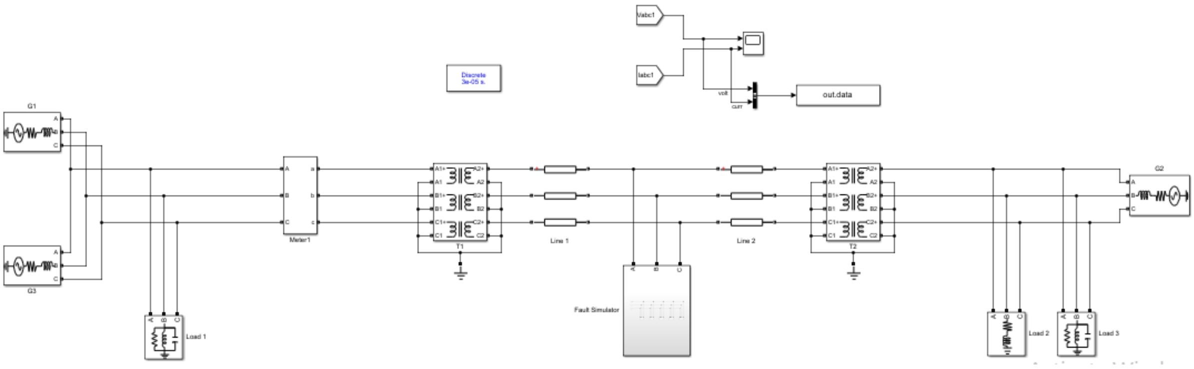
For fault location detection, our system will tell in which section of the transmission line the fault has occurred. We have transmission line length of 400km. We have divided the line in 5 zones.

Section 01	Section 02	Section 03	Section 04	Section 05
0-80km	80-160km	160-240km	240-320km	320-400km

The second stage neural network will tell us the section where the fault occurred. The overall system in a flowchart:



Simulink Model:
The Simulink Model for data generation:



The model power system consists of 3 11kV generators, 3 static loads, one 11kV/132kV transformer, one 132kV/11kV transformer, one 400km long transmission line, one voltage-current measuring meter and a fault simulator. The system element parameters are given below:

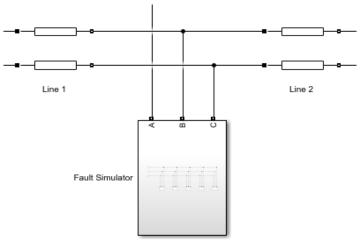
Generator Details			
Name	**Power Generation**	**R**	**L**
G1 (11kV)	50MW	12ohm	16.58mH
G2 (11kV)	50MW	8.4ohm	16.58mH
G3 (11kV)	50MW	12ohm	16.58mH

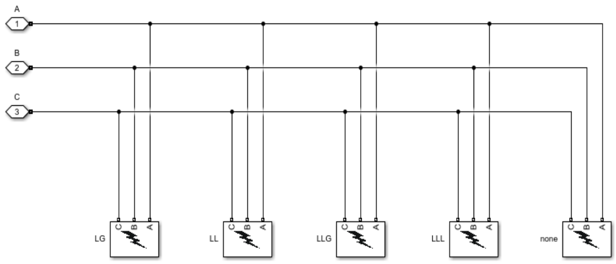
Transmission Line (132kV) Details	
Zero Sequence Resistance	0.3 ohms/km
Zero Sequence Inductance	3.638mH/km
Zero Sequence Capacitance	6.19nF/km
Positive Sequence Resistance	0.035 ohms/km
Positive Sequence Inductance	1.349mH/km
Positive Sequence Capacitance	8.67nF/km

Static Load Details		
Name	Real Power	Reactive Power
Load 1	10MW	70kVAR
Load 2	10MW	6.5MVAR
Load 3	1MW	600kVAR

Transformer Details		
	T1 (100MVA)	T2 (100MVA)
LT side (11kV) resistance	0.002pu	0.002pu
LT side inductance	0.05pu	0.05pu
HT (132kV) side resistance	0.002pu	0.002pu
HT side inductance	0.05pu	0.05pu
Magnetizing Branch Resistance	200pu	200pu
Magnetizing Branch Inductance	200pu	200pu

Inside the Fault Simulator Block





This block introduces fault to the transmission line by swithcing action. When switched on, the blocks indide the simulator shorts the desired phase(s) and ground. The fault remains on for a specified amount of time.

Data generation:

For training our two neural networks, two datasets are required. The first stage neural network is for detecting whether there is fault in the transmission line. This is done by classifying the input currents and voltages in either 1 of 5 target classes: LG, LL, LLG, LLL and No fault. If the data is classified into one of the 4 fault classes, then the system concludes that there is fault. Then the data is passed to the second stage network. For producing the classification dataset, we introduced the stated types of faults at 2 locations in each 80km section. We simulated the model and took instantaneous values of 3 phase currents and voltages to produce the dataset. For ease in classification, we one-hot encoded the classes. A glimpse of the 1st stage dataset is given below. This dataset has a total of 333340 entries.

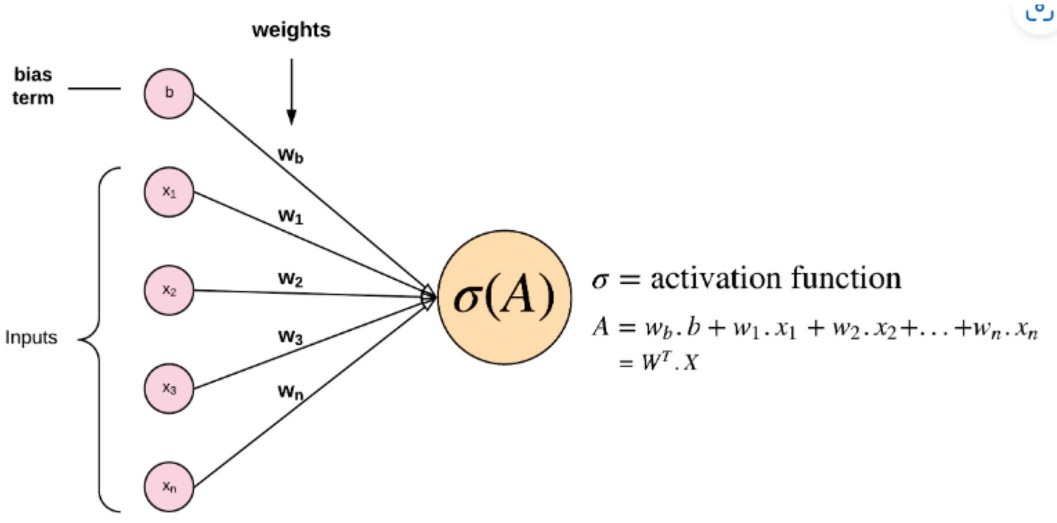
LG	LL	LLG	LLL	None	Ia	Ib	Ic	Va	Vb	Vc
0	1	0	0	0	-297.382	1115.206	-822.731	3481.807	702.9385	-4184.75
0	0	1	0	0	2.460112	-628.744	-671.573	5414.18	1035.661	-6449.84
0	0	1	0	0	-230.092	1146.082	-224.837	-5002.43	-1163.1	6165.534
1	0	0	0	0	1030.465	382.9196	-216.731	-1695.15	6581.06	-4885.91
0	0	0	0	1	52.18412	307.0898	-358.91	7975.352	1130.87	-9106.22
0	0	0	1	0	779.5489	-1206.25	428.3445	1049.187	1449.532	-2498.72
0	0	0	1	0	294.8466	839.5535	-1132.78	-3088.66	1020.666	2067.995
0	0	1	0	0	361.4975	-995.049	1097.689	2526.251	-223.907	-2302.34

For the 2nd stage neural net, another dataset is required. The 2nd stage neural net indicates the section where the fault has occurred. For this, the dataset needs to be built with only fault condition data. The target class is the section number where the fault has occurred. As this is a classification problem too, we one-hot encoded the section classes. A glimpse of the sectioning dataset is given below. This dataset has 186670 entries.

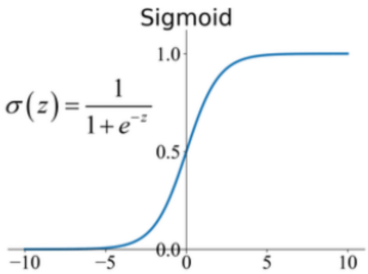
Distance	Zone1	Zone2	Zone3	Zone4	Zone5	Ia	Ib	Ic	Va	Vb	Vc
30	1	0	0	0	0	1297.102	-404.425	-890.265	-656.138	1022.583	-366.445
110	0	1	0	0	0	-246.756	-971.692	1220.093	1487.626	-387.899	-1099.73
220	0	0	1	0	0	-381.798	1254.476	-877.959	-334.899	533.7132	-198.814
30	1	0	0	0	0	109.9385	-943.341	-296.538	5502.002	977.88	-6479.88
60	1	0	0	0	0	-343.795	1203.928	-1049.83	-2013.05	-116.054	2129.1
140	0	1	0	0	0	-207.972	788.5721	-586.449	5251.845	1425.319	-6677.16
110	0	1	0	0	0	1294.055	-596.529	-695.758	-746.163	1543.595	-797.432
110	0	1	0	0	0	-568.857	-716.059	1287.15	1543.01	-763.95	-779.06
110	0	1	0	0	0	1289.972	-741.688	-545.888	-558.104	1524.768	-966.664
110	0	1	0	0	0	1070.231	-1162.43	94.61556	209.2839	1220.035	-1429.32

Artificial Neural Network for Fault Classification and Location Zone Detection

Artificial neural networks (**ANNs**), usually simply called **neural networks** (**NNs**) or **neural nets**, are computing systems inspired by the biological neural networks that constitute animal brains. An ANN is based on a collection of connected units or nodes called artificial neurons, which closely model the neurons in a biological brain. A single neuron is:



Here, activation function is a non-linear function. Some useful activation functions are: Sigmoid, ReLU and Softmax. Their corresponding input-output characteristics are:



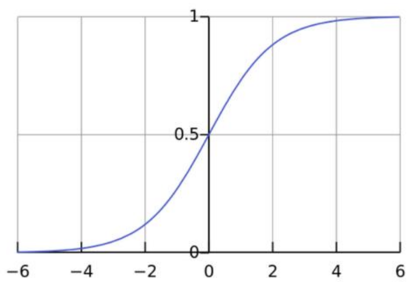
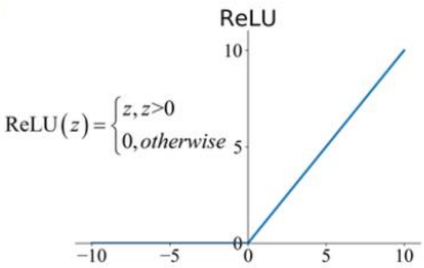
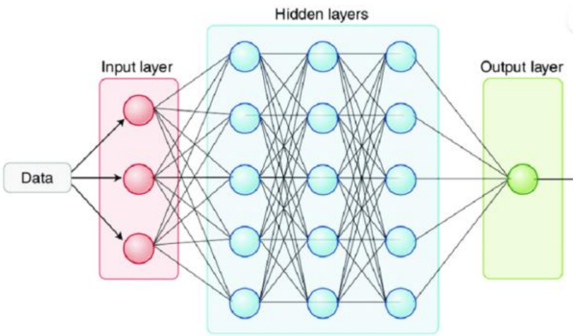


Fig: Softmax



Neural networks typically organize their neurons into **layers**. When we collect linear units having a common set of inputs, we get a **dense** layer. Through a deep stack of layers, a neural network can transform its inputs in more and more complex ways. In a well-trained neural network, each layer is a transformation getting us a little bit closer to a solution.

A typical Structure of a deep neural Network:



In our problem, we have instantaneous values of three-phase voltages and currents. So, for each sample we have 6 input features. And no of neurons in the output layer depends on the number of target class for each sample. We designed two deep neural Networks, one for Classification and another for Fault Zone Detection.

```
model1 = keras.Sequential([
    layers.Dense(50, activation='relu', input_shape=[6]),
    layers.BatchNormalization(),
    layers.Dense(150, activation='relu'),
    layers.Dense(100, activation='relu'),
    layers.Dense(60, activation='relu'),
    layers.Dense(5, activation = 'sigmoid')
])
```

Fig: layer structure for Fault Classification

We have used BatchNormalization layer after the first hidden layer which basically Normalizes the data in the same scale. The used activation functions are ReLU and Sigmoid.

```
model1 = keras.Sequential([
    layers.Dense(100, activation='relu', input_shape=[6]),
    layers.BatchNormalization(),
    layers.Dense(150, activation='relu'),
    layers.Dense(100, activation='relu'),
    layers.Dense(50, activation='relu'),
    layers.Dense(5, activation = 'softmax')
])
```

Fig: layer structure for Distance Zone Detection

The used activation functions are ReLU and Softmax. These two models are constructed by several trials monitoring the best accuracy.

At first, we fitted the Fault Classification data to our constructed Dense Neural Network through 200 epochs. In each epoch, the batch size was $1/5^{th}$ of the whole train dataset. The accuracy is based on the **‘categorical cross entropy’**. We can see that after almost 125 iterations, the model perfectly fits to the training dataset. Also, the accuracy of the validation dataset is also near to the best optimum value after almost 125 iterations.

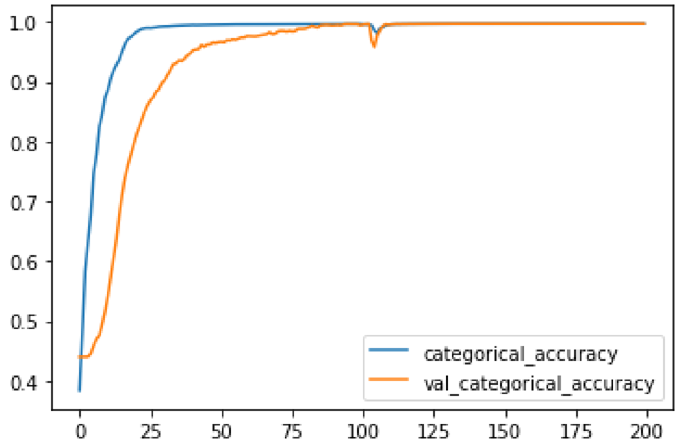


Fig: Accuracy Plot for Fault Classification on Train and Validation (Test) Dataset

Then, we fitted the Fault Zone Classification data to our constructed 2nd Dense Neural Network through 200 epochs. Here in each epoch, the batch size was 1/4th of the whole train dataset. Again, the accuracy is based on the **'categorical cross entropy'**. Here, we can see that the accuracy of the Validation Data (Test Data) is not converging consistently over the 200 epochs.

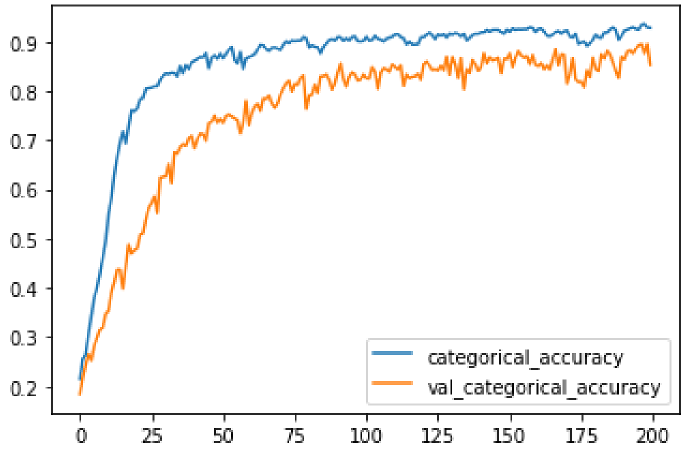


Fig: Accuracy Plot for Fault Classification on Train and Validation (Test) Dataset

In each model, we applied Model Check Point for tracking the best model with highest accuracy and saved them. Then we predicted the result upon the Test Dataset through the Saved Best models for each type of Classification.

Type of Classification	**Accuracy**
Fault Classification	99.4567108657826 %
Fault Zone Detection	89.4851877645042 %

Machine Learning Model Evaluation for Electrical Fault Classification and Location Zone Detection:

The ML Models that have been evaluated for the Electrical Fault Classification and Location Zone Detection are:

- XGBClassifier
- DecisionTreeClassifier
- RandomForestClassifier
- KNeighborsClassifier

XGBClassifier

The XGBoost stands for eXtreme Gradient Boosting, which is a boosting algorithm based on gradient boosted decision trees algorithm. XGBoost applies a better regularization technique to reduce overfitting, and it is one of the differences from the gradient boosting.
The 'xgboost' is an open-source library that provides machine learning algorithms under the gradient boosting methods. The xgboost.XGBClassifier is a scikit-learn API compatible class for classification.

DecisionTreeClassifier

Decision tree classifiers are supervised machine learning models. This means that they use prelabelled data in order to train an algorithm that can be used to make a prediction. Decision trees can also be used for regression problems. Decision tree classifiers work like flowcharts. Each node of a decision tree represents a decision point that splits into two leaf nodes. Each of these nodes represents the outcome of the decision and each of the decisions can also turn into decision nodes. Eventually, the different decisions will lead to a final classification.
The diagram below demonstrates how decision trees work to make decisions. The top node is called the root node. Each of the decision points are called decision nodes. The final decision point is referred to as a leaf node.

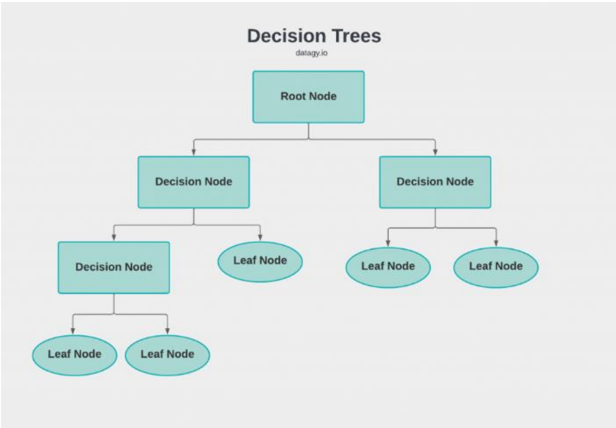


Fig: A visual representation of Decision Trees

RandomForestClassifier

Random forest is one of the most popular tree-based supervised learning algorithms. It is also the most flexible and easy to use. The algorithm can be used to solve both classification and regression problems. Random forest tends to combine hundreds of decision trees and then trains each decision tree on a different sample of the observations. The final predictions of the random forest are made by averaging the predictions of each individual tree. The benefits of random forests are numerous. The individual decision trees tend to overfit to the training data but random forest can mitigate that issue by averaging the prediction results from different trees. This gives random forests a higher predictive accuracy than a single decision tree. The random forest algorithm can also help you to find features that are important in your dataset. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

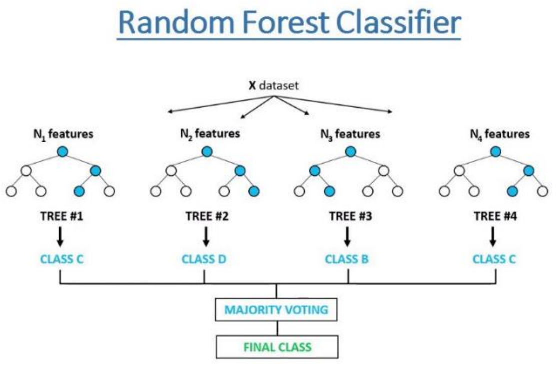


Fig: A visual representation of Random Forest Classifier

KNN Algorithm:

The k-nearest neighbors' algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another. For classification problems, a class label is assigned based on a majority vote—i.e., the label that is most frequently represented around a given data point is used.

Fig: Graphical demonstration of KNN algorithm

Cosine Distance – This distance metric is used mainly to calculate similarity between two vectors. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in the same direction. It is often used to measure document similarity in text analysis. When used with KNN this distance gives us a new perspective to a business problem and lets us find some hidden information in the data which we didn't see using the above two distance matrices.
It is also used in text analytics to find similarities between two documents by the number of times a particular set of words appear in it.

Formula for cosine distance is:

$$cos\theta = \frac{\boldsymbol{a}.\boldsymbol{b}}{||\boldsymbol{a}||||\boldsymbol{b}||}$$

Using this formula, we will get a value which tells us about the similarity between the two vectors and 1- $cos\theta$ will give us their cosine distance.

Using this distance, we get values between 0 and 1, where 0 means the vectors are 100% similar to each other and 1 means they are not similar at all.

Number of Neighbor:

In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. When K=1, then the algorithm is known as the nearest neighbor algorithm.

Machine Learning Model Selection:

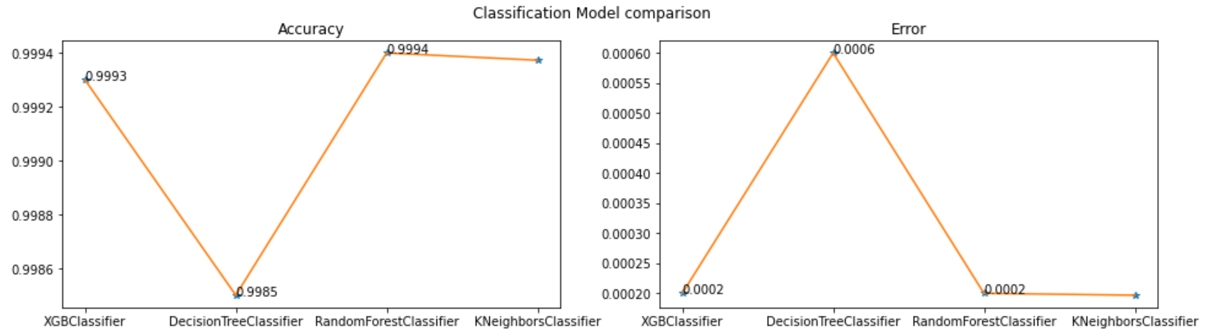


Fig: Model Accuracy and Error Comparison for Classification based on the Generated Data

All the four models have performed quite well and all of them have almost 99% accuracy and ignorable error.

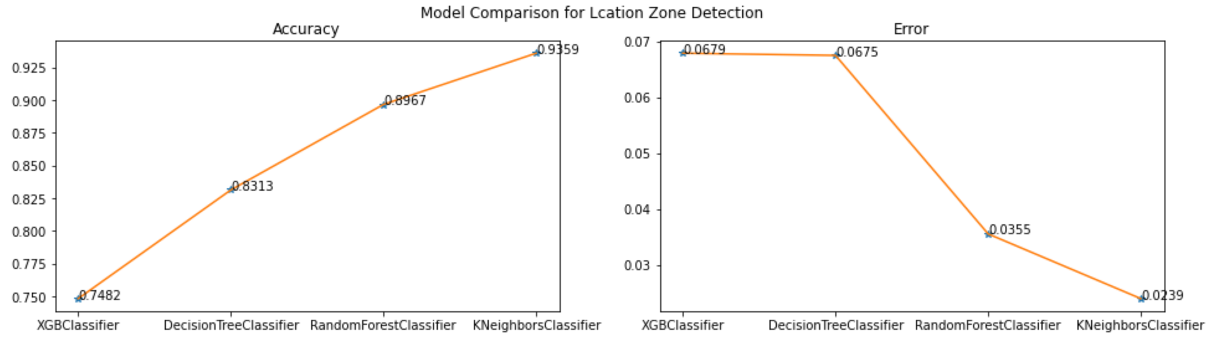


Fig: Model Accuracy and Error Comparison for Location Zone Detection on the Generated Data

We observe that ‘KNeighborsClassifier’ has the highest accuracy and lowest error among the four models in case of Location Zone Detection.

As ‘KNeighborsClassifier’ has performed satisfactory in both fault classification and location zone detection, ‘KNeighborsClassifier’ is the appropriate selection for the task.

KNN shows 99.8818% accuracy for fault classification and for zone classification KNN shows 95.2839% accuracy. The combined accuracy is 95.2426%.

First few matches of classification fault by KNN algorithm:

Actual class: LL	Predicted class: LL
Actual class: NO FAULT	Predicted class: NO FAULT
Actual class: LLG	Predicted class: LLG
Actual class: LG	Predicted class: LG
Actual class: NO FAULT	Predicted class: NO FAULT
Actual class: LL	Predicted class: LL
Actual class: LLG	Predicted class: LLG
Actual class: LLG	Predicted class: LLG
Actual class: NO FAULT	Predicted class: NO FAULT
Actual class: LG	Predicted class: LG

First few matches of zone classification:

Actual zone: 0 km - 80 km	Predicted zone: 0 km - 80 km
Actual zone: 160 km - 240 km	Predicted zone: 160 km - 240 km

Actual zone: 240 km - 320 km	Predicted zone: 240 km - 320 km
Actual zone: 80 km - 160 km	Predicted zone: 80 km - 160 km
Actual zone: 240 km - 320 km	Predicted zone: 240 km - 320 km
Actual zone: 160 km - 240 km	Predicted zone: 160 km - 240 km
Actual zone: 320 km - 400 km	Predicted zone: 320 km - 400 km
Actual zone: 240 km - 320 km	Predicted zone: 240 km - 320 km
Actual zone: 80 km - 160 km	Predicted zone: 80 km - 160 km
Actual zone: 240 km - 320 km	Predicted zone: 240 km - 320 km

References:

1. Jamil, M., Sharma, S. K., & Singh, R. (2015, July 9). *Fault detection and classification in Electrical Power Transmission System using artificial neural network - springerplus*. SpringerOpen. Retrieved August 26, 2022, from https://springerplus.springeropen.com/articles/10.1186/s40064-015-1080-x#:~:text=section%20is%20conclusion.-,Artificial%20neural%20network,form%20can%20be%20dealt%20with.

2. Huang, Nantian & Qi, Jiajin & Li, Fuqing & Yang, Dongfeng & Cai, Guowei & Huang, Guilin & Zheng, Jian & Li, Zhenxin. (2017). *Short-Circuit Fault Detection and Classification Using Empirical Wavelet Transform and Local Energy for Electric Transmission Line*. Sensors. 17. 2133. 10.3390/s17092133. https://www.researchgate.net/publication/319905026_Short-Circuit_Fault_Detection_and_Classification_Using_Empirical_Wavelet_Transform_and_Local_Energy_for_Electric_Transmission_Line