ANALYSIS AND COMPARISON OF MACHINE LEARNING APPROACHES FOR TRANSMISSION LINE FAULT PREDICTION IN POWER SYSTEMS

1Meera Viswavandya, 2Shashwat Patel, 3Kaushik Sahoo
1Head of Department, Electrical Engineering, College of Engineering and Technology, Bhubaneswar, India
2,3B.Tech. Student, Electrical Engineering, College of Engineering and Technology, Bhubaneswar, India
Email: {1mviswavandya@cet.edu.in, 2shashwatpatel5@gmail.com, 3kaushiksahoo54@gmail.com}

Abstract
The transmission lines suffer from various faults subjected to numerous natural as well as manmade causes. This paper presents a proposed MATLAB-SIMULINK model for generation of such random disturbances. The output of the system is input to another python-based model in order to detect and predict the exact nature of disturbances using various artificial neural networks with their respective accuracy scores. This paper provides a brief comparison between Decision Tree Classifier, Random Forest Classifier, Support Vector Machines, K-Nearest Neighbors and Multi-Layer Perceptron methodologies for detection of line to ground fault, as an example in this model-based approach.

Key Words – Transmission Line Faults, K-Nearest Neighbors, Multi-Layer Perceptron, Support Vector Machines, Decision Tree, Random Forest

1. Introduction

We live in an era of ever-increasing power demand. Nowadays, every power utility is working hard with immense efforts to reduce the consequences of power failure and to reduce system downtime, keeping in mind that every transmission line has its own operating limits. Faults within a transmission line should be cleared as soon as possible to increase the overall reliability of the system [1-2].

Faults may occur in the transmission line for different reasons. Each type of fault has different phase angles, magnitude, and intensity at the sink point [3-4]. The sink point of fault may result in increase in the magnitude of phase current or decrease in the magnitude of phase voltage. Intensity of the fault depends on the type of fault occurring at that point e.g. Line to ground (L-G), Line to line (L-L), Double line to ground (L-L-G) or three-phase fault (L-L-L). Among all these faults, the most frequently occurring that is 70% of all faults are line to ground (L-G) fault [5-8]. When a particular kind of fault involves disturbances in all the three phases is termed as symmetrical fault, and another configuration of fault involving faults in one or two phases is termed as unsymmetrical fault. The need of the hour is to classify all kinds of fault in real-time to restore uninterrupted supply within the minimum possible time thereby increasing the reliability of the overall power system. In reality, the transmission line system consists of thousands of interconnected buses and protective equipment that makes the conventional study unsuitable for fault detection and classification accurately in real time. The conventional study includes applications of traditional distance relay as a parameter of study, which may introduce additional errors into the system. Fault classification is generally done by comparing the matrix values of current and voltage in a healthy phase with the help of fault time matrix values, requiring high computational power and software examining efficiency. [9-12]

It takes unnecessary time in classification, and decision-making leading to decreased reliability of the overall system.

In the present scenario, utilities and customers need high reliability of power systems. Hence, the system needs to be error-free, efficient, and able to take various autonomous decisions in case of a critical situation. This paper, introduces various machine learning approaches such as K-Nearest neighbors, Multi-layer perceptron, Support vector machine, and Decision tree classifier for the classification and predictive analysis of the transmission line faults using...
the dataset matrices generated during normal and faulted condition. The output of this paper produces an accuracy

score of the above-mentioned algorithms, compares among all the proposed Python-based models and concludes the best method for analysis and prediction of line to ground fault.

2. Machine Learning Techniques

Machine learning enables computers to make smart decisions without being explicitly programmed. It enables computers to predict a certain output based on some experience data sets. A machine may learn based on certain mapping function (supervised learning) or some clustering algorithms (unsupervised learning). Some machine learning algorithms also revolve around decision-making algorithms such as Decision-Tree Classifier and Random Forest Classifier. A decision tree classifier predicts the value of responses by learning decision rules that are derived from certain feature points. This paper provides a brief comparison between various supervised algorithms for predicting the line to ground fault. The methodology opted is supervised learning techniques that includes K-Nearest neighbors, Multi-layer perceptron, Support vector machine and Decision tree classifier. A supervised machine-learning algorithm requires optimizing datasets with clear-cut learning patterns to perform with a good accuracy score and to obtain fast processing capabilities.

1.1 Dataset Filtering

Transmission line fault simulation is performed using the MATLAB-SIMULINK platform. Datasets generated is exported to MATLAB workspace from SIMULINK, consisting of specific labels and specific features in RMS values of volts and amperes. Feature points consists of 3 sets of voltage and current \( V_a, V_b, V_c \) and \( I_a, I_b, I_c \).

![Fig. 1: Snapshot of Dataset](image-url)

1.2 Predictive Algorithm

Supervised machine learning algorithms uses various learning patterns to feature sets of RMS values of voltage and current. This paper implements optimizing feature sets to strengthen the predictive ability of four algorithms namely KNN, SVM, Decision tree classifiers, and MLP. It also provides a brief comparison among these algorithms based on the Root-Mean-Square (RMS) error, and accuracy score obtained upon experimentation on LG fault.

1.2.1. K-Nearest Neighbors (KNN)

KNN is a non-parametric and lazy learning tool used for regression and classification of predictive problems. K’ in KNN is the number of nearest neighbors to include in the majority voting process for the similarity measure. The algorithm is based on the feature similarity process choosing the right value of ‘K’ by parameter tuning that is very important for improved accuracy. In this paper, the K-nearest neighbor works within a python module K-Neighbors Classifier. This classifier works as a clustering algorithm that map the distance between various feature sets. ‘K’ value is varied between the limits i.e. 1 and 25.

1.2.2. Support Vector Machines (SVM)

Usually, it is much easier to classify patterns that are linearly separable, that is a hyperplane separating the classes can be formulated so that the patterns belonging to a particular class lie in a distinct side of the hyperplane. But if the patterns are not linearly separable, the classification task becomes much more difficult. The SVM is capable of classifying both linearly and non-linearly separable patterns. A hyperplane is formulated using an instance object, which fits the dataset according to the classes. It revolves around the idea of finding a hyperplane that best separate features into different domains. The point closest to the hyper-plane is called support vectors and the distance of the vectors from the hyper-plane is called the margins. The SVM seeks to draw an optimal hyperplane between the classes that maximize the margin of separation between the classes, so that the number of misclassified classes is reduced. In this paper, Radial Basis Function (RBF) is used as a non-linear kernel function for the SVM model. SVM works fine with both linear and non-linear kernel functions using sklearn. SVM module runs on anaconda-python IDE.

1.2.3. Multi-Layer Perceptron (MLP)

MLP is a kind of supervised learning technique principally working with backpropagation algorithm. MLP neural networks use a gradient descent approach to update their
iterative weights in a feed-forward neural network, so that after training and testing, the MLP captures the inherent characteristics of the training data and can act as a non-linear model of the actual system, in this case, a fault classifier. In this paper, MLP is used to separate non-linearly separable data using a non-linear activation function using sklearn.neural_network running on anaconda-python IDE.

1.2.4. Decision Tree (DT)

In this method, a supervised and non-parametric method is used to classify feature sets and is based on a decision tree rule traversing to multiple nodes. In this paper, the decision tree is imported using sklearn.tree module in anaconda-python IDE for experimentation.

1.2.5. Random Forest (RF)

RF is a supervised learning technique that consists of multiple decision trees with the same nodes but every node leads to a different leaf node. Random Forest in general is a bunch of decision trees with an average of all trees as their output. Here, the Random Forest Classifier is implemented using sklearn.ensemble.Random Forest module in anaconda-python IDE.

![Fig. 2: Supervised Learning Model](image)

Among all the power system equipment, transmission line is most exposed to environment. Hence, the transmission line is more prone to faults compared to any other equipment that affects its stability and operating limits. The parameters of the transmission line that vary during the fault conditions are voltage, current, and impedance value of the line. At the safe operating limits, transmission line carries the rated voltage and current. As a fault occurs in the line the value of voltage and current deviate from their nominal values. These values follow a specific pattern depicting the fault nature when compared to the standard operating waveforms of voltage and current.

Some major causes of faults include open-circuit fault and short circuit fault. Open-circuit fault results from the uneven breakage of the conductors or false opening of circuit breakers. Short-circuit fault occurs due to the physical breakage of a transmission line or due to the loss of insulation on the line or due to improper installation.

Over-loading is also a catalytic factor, which leads to insulation breakdown at an early stage. In this paper, the line to ground fault is taken as an experimenting factor for the predictive models mostly because the majority of the faults occurring in transmission lines are line to ground in nature. The physical damage to the conductor may be due to natural reasons, which results in the contact of one of the three phases with the ground.

Further, sections of this paper consist of simulation of transmission line using MATLAB-SIMULINK in normal conditions as well as in line to ground fault conditions to generate specific datasets in CSV file format. Dataset acts as an experience feature sets for the respective predictive algorithms to generate an accuracy score and Root-Mean-Square error value.

3. System Modelling for Fault Datasheet Generation

MATLAB environment has been used here for the purpose of simulation modelling. The component libraries have been provided by Simscape Electrical.

Here a Simulink model is designed with all the simulink blocks, which constitute a transmission line of a power system model. A 400 KV transmission model is used to develop the neural network models. The system consists of a generator of 11 KV located at the source end, an inductive load at the other end and a 3 phase fault simulator block with a view to simulating faults is placed in between the transmission line. A 200 Km overhead transmission line simulation block is used here.
The model shown in Fig. 3 is modeled using MATLAB (R2017b). This model is used with a view to obtaining the datasets for training and testing purpose. In this model the three phase V-I measurement block use is for measuring different samples of voltage and current. The 1:1 division of the transmission lines as line 1 & line 2 having 100 Km long each. The 3-phase fault simulator simulates different types of faults that are line to ground fault, line-to-line fault, and three-phase fault in the power system model.

The waveform in Fig. 4 is the output of the simulation of the power system model in no fault condition. The occurrences of single line to ground fault take place when one of the phases of the three-phase line gets short with the ground. At the time of occurrence of the fault, the impedance need not be zero but a very minute value in accordance with the line impedance.

In a faulty line the current rises remarkably then the normal current. The waveform in Fig. 5 shows the remarkable rise of current in the occurrence of single line to ground fault in the transmission line.

| Ia   | Ib   | Ic   | Va   | Vb   | Vc   | Fault State |
|------|------|------|------|------|------|-------------|
| 1254 | 1254 | 1254 | 1254 | 1254 | 1254 | 1254        |

The numeric quantities of the 3 phase voltages Va, Vb, Vc and currents Ia, Ib, Ic are fed after having been generated in both the normal and the faulty condition. Then the data is tabulated and exported as a CSV file from workspace.

A snapshot of a CSV sheet having the data in the normal and faulty condition is shown in Fig. 6 and Fig. 7. The training and testing datasets are given as zero signifies healthy network and one signifies faulty network. The data is subsequently fed into the machine-learning algorithm for training.

3. Machine learning Algorithm Design and Accuracy Count

The main objective of this paper is to thrive a machine learning based autonomous self-learning system that has the
capability of self-acquisition of knowledge in real time with a little supervision.

In this paper, the evaluation of different algorithms is done by the accuracy score and mean squared error, which is mostly used having multi-labels, and the result is measured in percentage.

Here the accuracy can be represented as

\[
\text{accuracy} (y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}} 1(\hat{y}_i = y_i)
\]

The accuracy is found by dividing the number of matches by the number of samples.

From the given list of y predict and y true, for sample index value of ‘i’ is compared to find matches. Based upon the number of matches accuracy is calculated.

Here the root mean square error can be represented as

\[
\text{RMSE}(X,Y) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2}
\]

Root mean square error measures the average magnitude of the error as a square root of the average squared differences between prediction and actual observations.

3.1 Implementation of Decision Tree Classifier

A sequence of test cases and different conditions is being organized in a tree structure in the decision tree classifier model and the classification takes place based on decision rules.

Fig. 8: Predicted (a) and Testing (b) Labels of Decision Tree

The Fig. 8 shows the predicted and testing labels of decision tree classifier plotted which a non-parametric method of supervised learning is.

Prediction points for datasets in red line and all testing data points represented in blue line. ‘0’ level points indicate training and testing data sets for normal operating condition. ‘1’ indicates training and testing datasets for fault conditions. The training dataset is fed into decision tree classifier and the testing dataset was predicted by the classifier with accuracy up to 86.17%.

3.2 Implementation of Support Vector Machines

Support Vector Machines (SVM) is a supervised learning algorithm, which fits the data in accordance to the classes after finding a hyperplane and does a distinct classification of data points.
The Fig. 9 shows the predicted and testing labels of support vector machine plotted after separation of data points in different classes by a hyperplane.

Prediction points for datasets in red line and all testing data points represented in blue line. ‘0’ level points indicate training and testing data sets for normal operating condition. ‘1’ indicates training and testing datasets for fault conditions. The training dataset is fed into support vector machine classifier and the testing dataset was predicted by the classifier with accuracy up to 75.94%.

3.3 Implementation of K Nearest Neighbor

K Nearest Neighbor is a supervised lazy and non-parametric learning algorithm use for predictive problems classification having a class membership as its output, which uses the distance for classification.

The Fig. 10 shows the predicted and testing labels of K Nearest Neighbor plotted by assignment of weights for the contributions of the neighbors, where the nearest neighbors has more contribution.

Prediction points for datasets in red line and all testing data points represented in blue line. ‘0’ level points indicate training and testing data sets for normal operating condition. ‘1’ indicates training and testing datasets for fault conditions. The training dataset is fed into K Nearest Neighbor classifier and the testing dataset was predicted by the classifier with accuracy up to 88.89%.

3.4 Implementation of Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) provides a mapping which is nonlinear in midst of an input and an output vector and uses a nonlinear activation function. It employs a supervised learning technique called backpropagation for training purpose.
The Fig. 11 shows the predicted and testing labels of Multi-Layer Perceptron plotted by utilizing nonlinear activation function and backpropagation for training. Prediction points for datasets in red line and all testing data points represented in blue line. ‘0’ level points indicate training and testing data sets for normal operating condition. ‘1’ indicates training and testing datasets for fault conditions. The training dataset is fed into Multi-Layer Perceptron classifier and the testing dataset was predicted by the classifier with accuracy up to 78.53%.

3.5 Implementation of Random Forest Classifier

Random forest Classifier is also a supervised learning algorithm. It creates many decision trees, takes the prediction value from each of them, and among them selects the best result by voting.

The Fig. 12 shows the predicted and testing labels of Random forest Classifier plotted by getting the mean prediction of each of the trees.

Prediction points for datasets in red line and all testing data points represented in blue line. ‘0’ level points indicate training and testing data sets for normal operating condition. ‘1’ indicates training and testing datasets for fault conditions. The training dataset is fed into Random forest Classifier and the testing dataset was predicted by the classifier with accuracy up to 85.55%.

4. Analysis of Results

The five algorithms, namely, Decision Tree Classifier, Support Vector Machines classifier, K Nearest Neighbors Classifier, Multi-Layer Perceptron and Random Forest Classifier were implemented to the whole dataset by splitting it into training and testing part. The comparison is done based on the accuracy score where K Nearest Neighbors gave the best accuracy, which is close to 89 percent, whereas Support Vector Machines did not perform well producing accuracy close to 76 percent.
### Table 1
Comparison of different Machine Learning Algorithm

| SI No. | Algorithm                          | Accuracy (in percentage) | Root Mean Square Error |
|--------|------------------------------------|--------------------------|------------------------|
| 1      | Decision Tree Classifier           | 86.1693                  | 0.1383                 |
| 2      | Support Vector Machine             | 75.9362                  | 0.2406                 |
| 3      | K-Nearest Neighbor                 | 88.8916                  | 0.1111                 |
| 4      | Multi-Layer Perceptron             | 78.5345                  | 0.2146                 |
| 5      | Random Forest Classifier           | 85.5478                  | 0.1445                 |

5. Conclusion

This paper provides a predictive model for the detection of faults in transmission lines. This predictive model uses phase currents as input to the system of neural network. The outcome of this predictive model provides a suitable algorithm for the designing of a protective stratagem for transmission line based on the machine-learning algorithm. Our method being reliable and feasible, modelling of transmission line can be done. Support vector machines are supposed to perform well in small feature-sets, but this is not always true. In cases where the dataset is not separable by a single curve, SVM will perform worse than other neural networks and by feeding more data, MLP will perform naturally better than SVMs. From this, it is concluded that dataset is not perfect curve separable, and forms small clusters in feature space where KNN often gives results in clustered data as in this case.

References

[1] Eisa Bashier M Tayeb 2013, “Neural network approach to fault classification for high speed protective relaying” American Journal of engineering research (AJER) volume 02, pp 69-75.

[2] P. P. Pattanaik and C. K. Panigrahi, “Stability and fault analysis in a power network considering IEEE 14 bus system,” 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, pp. 1134-1138, doi: 10.1109/ICISC.2018.8398981, 2018.

[3] S. Saha, M. Aldeen, C.P.Tan, “Fault detection in transmission networks of power systems,” Science Direct Electrical Power and Energy Systems 33, pp 887–900, 2011.

[4] H. Singh, M.S. Sachdev, T.S. Sidhu “Design, Implementation and Testing of an Artificial Neural Network Based Fault Direction Discriminator for protecting Transmission Lines,” IEEE Transactions on Power Delivery, Vol. 10, No. 2, 1995, pp 697-706.

[5] Abhijit A Dutta, A.K.Naidu & M.M.Rao 2011 “Intelligent control for locating fault in transmission lines” International Journal of Instrumentation, Control & Automation (IJICA) ISSN: 2231-1890 volume 1, Issue-2.

[6] Thomas Dalstein, Brend Kulicke 1995, IEEE Transaction on Power Delivery, volume 10, Issue-2, pp 1002-1011.

[7] Rajveer Singh 2012, “Fault detection of electric power transmission line by using neural network”, Volume-02, Issue-12.

[8] S.N.Sivanandam, S.Sumathi, S.N.Deepa “Introduction of Neural Network Using MATLAB 6.0” TMH Pbs.

[9] Ibrahim Farahat, Dept. of Electrical & computer Science engineering, Concordia University, Canada.

[10] An Adaptive k-Nearest Neighbor Algorithm, by Shiliang Sun, Rongqing Huang S. Websper, R. W. Dunn, R. K. Aggarwal, A. T. Johns and A. Bennett: Feature extraction methods for neural network-based transmission line fault discrimination.

[11] F.Zahra, B. Jeyasurya, J. E. Quaicoe: High-speed Transmission Line relaying using artificial neural networks.

[12] A.P.Vaidya, Prasad A. Venikar: ANN Based Distance Protection of Long Transmission Lines by Considering the Effect of Fault Resistance.