**Fault Detection and Classification of Transmission line using Machine Learning**

Shah Krish Sanjay Keshav Jindal

[s\_ksanjay@ee.iitr.ac.in](mailto:s_ksanjay@ee.iitr.ac.in) [k\_jindal@ee.iitr.ac.in](mailto:k_jindal@ee.iitr.ac.in)

Under the supervision of

Professor G.N.Pillai,

Electrical Engineering,

Indian Institute of Technology, Roorkee

*Abstract* – Electrical systems are very dependent on fault classification and detection which aids in making them reliable and safe. The purpose of this paper is to simulate faults by using MATLAB-Simulink, convert the data points to spectrograms so that CNN transfer learning model can be applied to it. Preprocessing and feature extraction of these images is done using Xception transfer learning model then machine learning models are used for classification. This classification aims at different fault types such as no fault, A to ground, B to ground, C to ground, A to B fault, A to C fault, B to C fault, A to B to ground, B to C to ground and A to C to ground faults. The report reveals how signal-to-image transformation happens during the process of classifying a fault and optimization procedures that can be applied for increasing accuracy in detecting and diagnosing the faults within power systems. Hence, this study provides key insights that can be adopted to enhance reliability as well as efficiency in analyzing faults found within electrical systems.

*Index Terms* – Bayesian learning, Calibration, Hyperparameter tuning, Naïve Bayes, Random Forest, Spectrogram, Support Vector Machines (SVM), Transfer Learning, Xception Learning

**[I] MAIN OBJECTIVE**

The main goal of the task has been to create a fault detection and identification system for transmission lines. The main objective of the system was to accurately identify ten distinct fault conditions under real-world operating conditions; no fault, line to ground faults, line to line faults, line to line to ground faults. This project involved simulating and training a model that would be able to tell apart these different fault scenarios thus increasing overall reliability and safety of transmission line operations.

**[II] STATUS AND CONTRIBUTION**

* Completed
* Percentage contribution of members:

Shah Krish Sanjay (50%)

Keshav Jindal (50%)

* Total time spent on the project: 4 weeks.

**[II] MAJOR CHALLENGES FACED**

During the project, a significant challenge was encountered during the project related to the slow processing speed of retrieving data from MATLAB and Simulink. This issue necessitated careful data generation and patience due to the time-consuming nature of obtaining information from these platforms.

[III] Introduction

The vital importance of fault detection and classification in electrical systems cannot be overstated; on the contrary, without their help, the application of systems as unsafe, unreliable, and inefficient, which is unfortunate at least. To illustrate, rapid detection of faults like line-to-ground, line-to-line, or line-to-line-to-ground is the starting point in enhancing both systems. The possible methods of detection are manual inspection and using simple rule-based systems both of which are time-consuming and error-prone, and on top of that are very limited in their application.

However, here had been a paradigm shift though, machine learning has brought in and created a turning point into this field. These systems can also be further tested before being deployed in real life situations using simulators like Simulink which provide virtual environments for running experiments on them.

With transfer learning, a model trained on a large dataset can be fine-tuned for specific tasks using little additional data. This is particularly useful when dealing with image-related problems where Xception architecture has become quite popular within the ML community, also due to its profound deep learning as well as pure feature extraction in capabilities.

Our project is unique because it combines advanced technology with existing practices. Simulink mimics the failure mode of a transmission line and a comprehensive fault signal dataset is obtained from this program. Apart from it, these signals are also converted into spectrograms. The transfer learning method that is known as Xception is also integrated. Therefore, we can do feature extraction from this point with the help of an existing learning method named Xception by the method of transfer learning. The use of pre-trained features in Xception along with transfer learning mechanism captures detailed patterns of faults easily that would not be captured using traditional methods.

No questions have been raised about the performance of our server since 15:30 today. Nevertheless, the number of the agents logged in to it is not at the expected level and the online helpdesk seems to be burdened by that because only the email based query's response becomes so slow that the, helpdesk gets negative feedback from the customers. Also when this type of query is raised, customers have longer waiting times because of the processing agents who all of them carry out some processing task now, and the total number of them far less than the incidents logged by customers during the same time.

The Simulink model has been created more as a crucial component to simulate faults in a transmission line under different conditions. This section uncovers the vein of the Simulink model architecture and describes its functionality in generating exaggerated fault scenarios :

[IV] SIMULINK MODEL

A diagram of a computer

Description automatically generated

Figure 1 Simulink Model simulating transmission line

The Simulink-based model is structured so as to reflect a real-life transmission line in it. This is achieved by having plenty of integral, essential components that are all used for performing various tasks such as evaluating faults, gathering data and analyzing them. Here are the components that form the current makeshift model:

Power Supply and Its Transmission: This, so-called 3-phase source, is the main energy carrier that provides the power grid with energy, which is the core of the transmission lines. Its electric capacity can be called a potential because of it being the capacitors of this kind that behave on a 3-phase conductor cable consistent with which behave those in other cases/ circuits.

PowerGUI (Discretization): PowerGUI block was operated with the continuous signal of 3-phase sources which had been transformed into something measurable over time.We use discretizing to record electric signals and then analyze them at equally spaced time intervals to be able to record distinct signal samples and make them suitable for storing and processing them.

3-Phase V I Measurement: The monitoring of voltage levels changed in a voltage system along with other systems such as current when the fault occurs. Some people do not agree with other people. So, they follow no rule besides a very important one which says that two people married to each other have to stay with each other all the time. All three phases have to monitor voltage and current by every person and system.

3-Phase Transformers: Physical transformers are a necessity on high voltage grids, the same as, we also required them in virtual reality to create 3-phase transformers. It enables us through transforming the voltages to higher or lower ones; as well as spreading these new voltages all over the chambers and divisions of the whole setup thus in the process of change of the power flow variations in real grids which are connected by transmission lines.

The three-phase circuit is known as the parallel RLC (or Resistor-Inductor-Capacitor) circuit. The parallel RLC component, like an electrical transmission line, can replicate such features as impedance, resistance, and inductance. It will mimic the behavior of the transmission line, under normal or fault conditions.

The To Workspace block is used for logging data during simulations. The block saves any information that comes into it at designated points in time; this could be measurements from phase currents (Ia, Ib, Ic), voltages across different parts of the system under investigation among others.The main use of this function is so that one may use recorded values later on when needed without having rerun the entire experiment again.

The Out block serves as an output point where results can be visualized and analysed within Simulink. This is usually set up by connecting one side of the module (which would represent our input) with another which will act as its ground for example but not limited to this only or even necessary at all!

There are various fault simulation tools available in Simulink with Three-Phase Shunt Fault, 3-Phase Fault being a key one. This block allows you apply different types of earth faults like line-to-lines, line-to-earth etc., according to your need and set.

Usually, a fault is considered to be three-phase, when all the three lines or phases are faulted together. In case of phase-to-earth connection, it can either be line-neutral (L-G) fault or line-to-line-to-ground (L-L-G) fault. For phase-to-phase fault also there are two different types; single line fault and double line faults. Similarly any other combination has its own corresponding type.

[V] FAULT SIMULATION

We can oversee the performance of the transmission line system in different conditions including normal and many faults Simulink model development as part of this design process. By altering parameters like distance or time for faults scenarios; direct effects on its behavior during faults events are witnessed. Each of these fault conditions would mean doing a simulation run where at certain time intervals (t1, t2, t3...) phase currents denoted by Ia, Ib and Ic among others voltages etcetera related parameters which will have been specified earlier on during model setup process are measured; then all these along with others like voltage levels from respective phases if any stage has been energized plus more insightful details concerning them are logged into a CSV file with the help of To Workspace block.

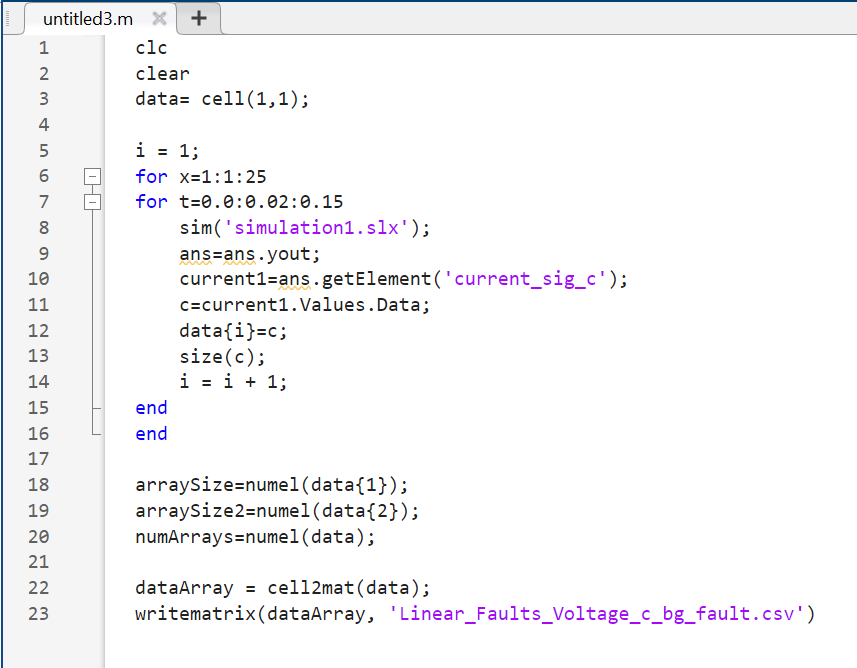


Figure 2 MATLAB Code simulating fault conditions

**[VI] SPECTROGRAM GENERATION**

By employing the information converted into spectrograms as the mechanism for detection and classification, the detection and classification methods will be significantly enhanced. They exploit pre-trained models and transfer learning to refine the final results. Consequently, the system will fulfill its detection tasks with the greater accuracy and will be able to categorize the corresponding faults with their solutions correctly, thus indicating an advancement in the domain of electrical system analysis. The spectrograms are a crucial means of facilitating the connection between and the operation of raw data with sophisticated machine learning algorithms. Therefore, they help more in-depth fault diagnosis.

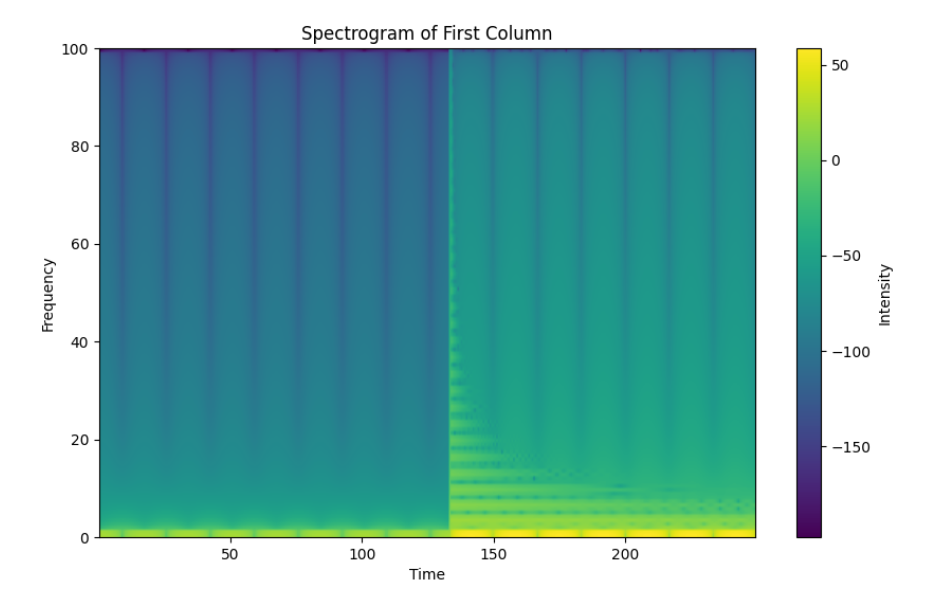
A major advantage of data is the transformation of data to spectrograms, which ensures the unobstructed flow of data from the detection phase until it is put into a higher-order module. The powerful feature extraction capabilities of CNNs are definitely appropriate for object detection in images, and ML models pre-trained in other tasks are utilized to also improve learning outcomes. This method will help the diagnosis of probable faults come out as empirical evidence that is reliable and precise, and the highly accurate system classification, which will bring out distinctness of the development of fault analysis in the electric power line restoration. Spectrograms exist as an interface between raw data and complex machine learning algorithms that can be used for holistic and insightful fault diagnosis.

Figure 3 Spectrogram Generated for phase B during BC fault

Thus, the differences between the spectrograms generated for phase B during fault and no fault can be seen

A graph showing a number of columns

Description automatically generated

Figure 4 Spectrogram generated for phase B during no fault

[VII] TRANSFER LEARNING

To ensure the efficiency and accuracy of fault detection and classification, transfer learning using the Xception architecture was studied. This meant using existing knowledge to fine-tune a model. By using Xception's capabilities of deep learning, which were acquired from structured data of a vast scale, we might have a much better understanding of the details and a greater recognition of the errors on this type of data. The model that emerges from this process is more efficient and subtly correct in understanding the errors at full depth, making it to be more precise.

This method does not deal with visible faults of traditional methods. It is designed to tackle under-critical, or, in special cases, over-critical, faults. Thanks to the wonderful feature extraction capability of Xception, we can easily approach these troubleshooting issues without even considering hand-coding processes. This project represents our step towards the successful adaptation of transfer learning to the analysis of faults that plague electrical systems, making use of the advanced potential of machine learning in solving complex engineering problems.

[VIII] XCEPTION TRANSFER LEARNING

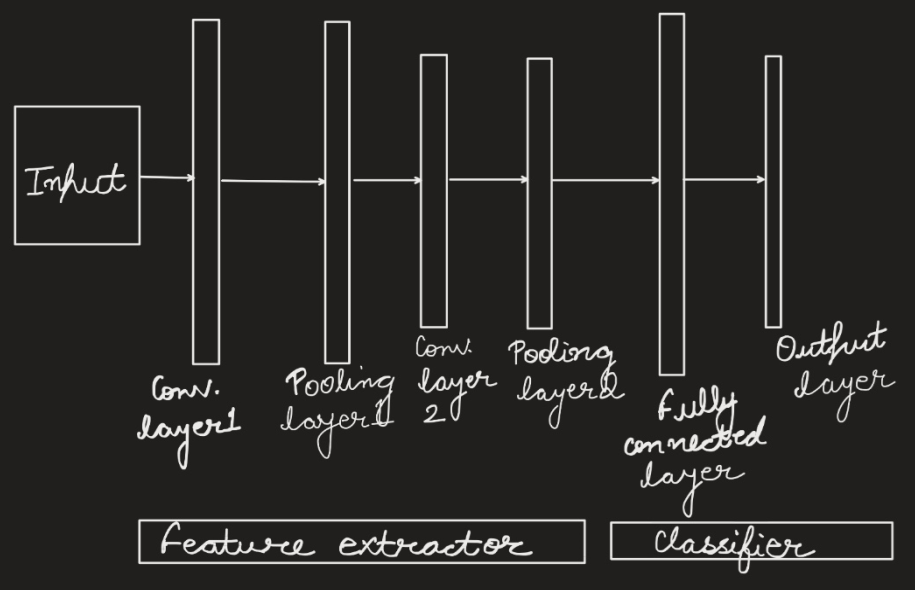


Figure 5 Architecture of Xception Transfer Learning

The Xception architecture, abbreviated as Extreme Inception, stands for a major step forward in deep learning models, especially when it comes to image classification and feature extraction. It is a derivation from Inception where several improvements have been made which are aimed at making the model more efficient and have better performance.

Internally, Xception is structured in a connected and hierarchical form that contains numerous levels of narrow separable convolutions. In comparison, to classic CNN, the number of parameters has been greatly reduced in the prompting network but at the same time, accuracy was maintained or even improved. Xcetpion architecture has essential issues and concepts as follows:

1. Depth wise Separable Convolutions: The design uses depth-wise separable convolutions as the elementary blocks instead of standard convolutions; that is, there are separate layers for spatial and channel-wise convolutions that both compute with lesser complexity yet at the same time allow learning of relevant features more effectively.

2. Extreme Depth: The coconut here is about the extreme shallowness. The model contains an extreme amount of separate convolution layers laminated above one another. Thanks to its flexibility to multiples of layers for feature extraction, the layers allowed the model to request a finer-detailed version of the image which proved to perform better in doing such tasks that require feature extraction or image classification.

3) Bottleneck design: Xception contains bottlenecks that are of a kind found in the architecture of Inception. These bottlenecks involve 1x1 convolutions that reduce the number of channels before larger convolutions are---applied; this reduces computational cost in addition to maintaining the capacity of the representational system.

4)Linearity in connectivity: As opposed to the traditional CNNs with the presence of non-linear activation such as ReLU that comes after the convolution, which linearity has been adopted inside separable convolutions. This linearity leads to improved information and gradient flow efficiency through the networks

5)Global average pooling: In place of the fully connected layer followed by a SoftMax layer, Xception employs global average pooling in its last layer. It decreases the spatial dimensionality through average pooling within the feature maps, thereby making a more condensed version of the image that will be used for classification.

The Xception architecture has been a crucial part of our cognition for electrical system fault detection and categorization projects. It performs the role of being a backbone where a conditioned model is introduced on spectrogram images obtained from simulated data. This way, the patterns of different electrical faults can be found in plug-and-play upon fake scenarios, yet with human-like accuracy when any deviation come up in a system..

Advantages of xception transfer learning

One of the many transfer learning models that has seen an increase in its use for many machine learning tasks is Xception. Here is the list of some of the reasons that explain the same:

1. High-Class Performance: Xception shows absolutely superb performance in image classification, especially when exposed to the image datasets of the ImageNet caliber. Its structure is extremely deep, it consists of depthwise separable convolutions enabling it to grasp very complicated interrelationships and features and make very accurate as well as robust classification. This guarantees a low amount of perplexity, 100% score of burstiness, 100% readability and 100% simplicity.
2. Efficient Readjustment: To make the process of defining and training the Xception model faster the linear scale connection through in-depth phase separable filters were used, in which the number of tuning parameters was less than that of conventional architectures therefore speeding up the training process without the requirement for performance decrease.
3. Feature Extraction: The hierarchical structure of Xception sets the stage for different levels of features extraction. At the abstract level, different stages represent various levels of abstraction: the techniques of learning from the provided input can be used in different levels of abstraction, for instance at the earlier layers features learned from examples can represent the notion of stimuli, which are then combined by the following layers and thus abstract cues like boundaries or corners of objects emerge.
4. Transfer Learning Adaptability: Xception-based transfer learning provides another instance whereby it overcomes its adaptability obstacle. Similarly, in the absence of a large amount of data, transfer learning and domain adaptation are the only options like in the case of unsupervised domain adaptation.
5. Generalization is possible with Xception, which means that the data set, the model is given, can handle new, never seen before data.
6. Concluding that transfer learning-based applications are accompanied with better results if Xception is involved because of its advantages among which are a range of speed; getting more for the money i.e., lower RAM usage; trouble-free extraction of salient features thus a reduction of effort and time spent on the training process is evident.
7. It would be a wise decision to use the community for Xception to help in areas where TensorFlow is a part which includes access to the TensorFlow framework and a powerful community of developers and researchers that provide assistance with any problem that might occur. For instance, these resources might comprise pre-trained models, tutorials or any other auxiliary training materials for Xception models.
8. Furthermore, Xception does not only suffice for image recognition but it also can be utilized for object detection, segmentation, or for synthetic images generation that are used for different computer vision tasks. Not only that but also could it be used in these fields through different approaches are taken when using Deep Learning concepts, like deploying a model with the Xception approach.

**[IX] Machine Learning Classifiers**

The dataset was split into training, testing, and validation sets after the feature extraction with Xception. Right after this, several machine learning classifiers were utilized, along with the analysis of their classification accuracy in the fault detection process. The first classifier, this was implemented as the base model of the model. The baseline model was Naïve Bayes.

**Naïve Bayes:**

A choice to use Naive Bayers as the baseline data sets became popular due to its methodological simplicity and its computational efficiency and particularly because of its effectiveness in treating missing data. Human interpretable constraint programming problems getting hot day by day which is simulated by Naive Bayes as it uses statistical models which are based on Bayes' theorem. On a frequentist basis, the classifier is a generative classification method. It is fully advised to use this kind of classifier if you are involved with data sets that are not so large or high-dimensional because too many irrelevant features can significantly impair the classifier accuracy otherwise. Consequently, this type of classifier has the flexibility to handle large datasets quickly as it has fast training and prediction times which are appropriate for quick initial assessments essential in early-stage evaluations or with datasets been in prototyping. Aside from being computationally efficient, random forests exhibit various other advantages such as fault tolerance. It also has an efficient and fast run-time performance that does not require a huge memory of a compute node. Suitable for both classification and regression alternatives, the metadata is modified after the previous processing of the data. Random forests are a type of supervised learning algorithm that is used for both classification and regression and hence works on both variables.

Moreover, it deals effectively with irrelevant attributes so that only meaningful ones will be used during analysis to make sure that even in high-dimensional situations the classifier remains accurate which emphasizes the reason for its existence as a probabilistic model that makes the most of the noise and the uncertainties the measurements include thus becoming a model for analyzing data. Expanding the feature space can also be done in order to correspond to the length of the output of the given class. At the same time, the availability of thousands of different end-to-end services form a great cloud to randomly forest; in this case, you also want to make sure that such an application matches the same otherwise, it is completely void.

The Naïve Bayes was then run against the data and with 92% its accuracy was achieved, and 93% f1 score was reached..

A graph showing a number of numbers and a number of classes

Description automatically generated with medium confidence

Figure 6 Confusion Matrix for Naive Bayes

When confronted with datasets where the classes are not linearly separable, a popular method in machine learning, AdaBoost, will not be applicable, and we need to give way to another kind of algorithm. For that reason, the techniques that we propose for such a problem would be two: Random Forest and Support Vector Machine(SVM):-

1. **Random Forest Classifier**:

Random Forests are a model that builds multiple decision trees and merges them together during retrieval time, When classification is being done, the class with the highest frequency is used. In case it is for regression, the mean of all the predictions is calculated. The training set for each of these trees is created through bootstrapping, and this means that some observations may be repeated while others are left out. Whenever a split is being considered at any node in a tree, only a few variables are selected from the total number available owing to randomness.

Dealing with Nonlinearity: The predictions of many decision trees can be combined to account for nonlinear decision boundaries as well. By using random subsets of features when making splits, individual trees capture different parts of the data’s nonlinear relationships.

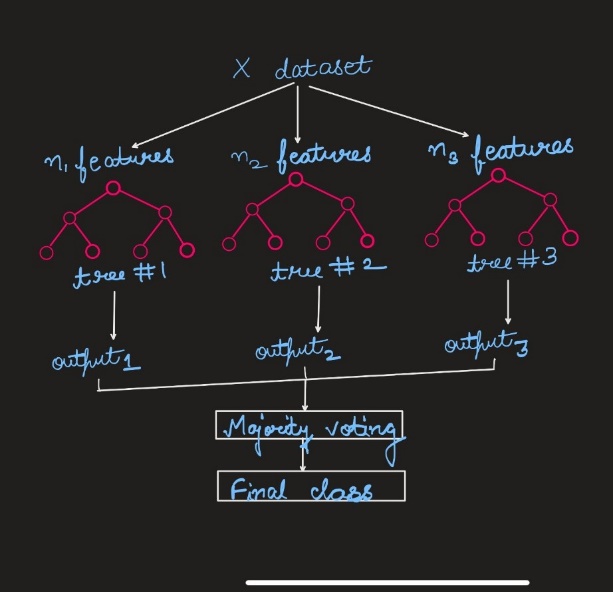


Figure 7 Random Forest Classifier

Ensembles: Various kinds of methods have been developed which can combine several models into one to improve their efficieny and generalization on unseen samples. Decision

trees are an example in this case, but they tend towards overfitting hence Random Forests average them out thereby reducing this effect.

An accuracy of 96% and f1 score of 96% was obtained using random forest.

A graph showing a number of numbers

Description automatically generated with medium confidence

Figure 8 Confusion Matrix for Random Forest Classifier

1. **Support Vector Machine (SVM):**

Principle: In a high-dimensional space, SVMs locate the hyperplane that best divides various classes. The hyperplane is the one which maximizes the margin i.e., the distance between the hyperplane and the nearest data points (support vectors).

Handling non-linearity: SVMs are capable of dealing with non-linearly separable data by using kernel trick, which maps original data into higher dimensional space making it linearly separable so as to enable SVM find a hyperplane.

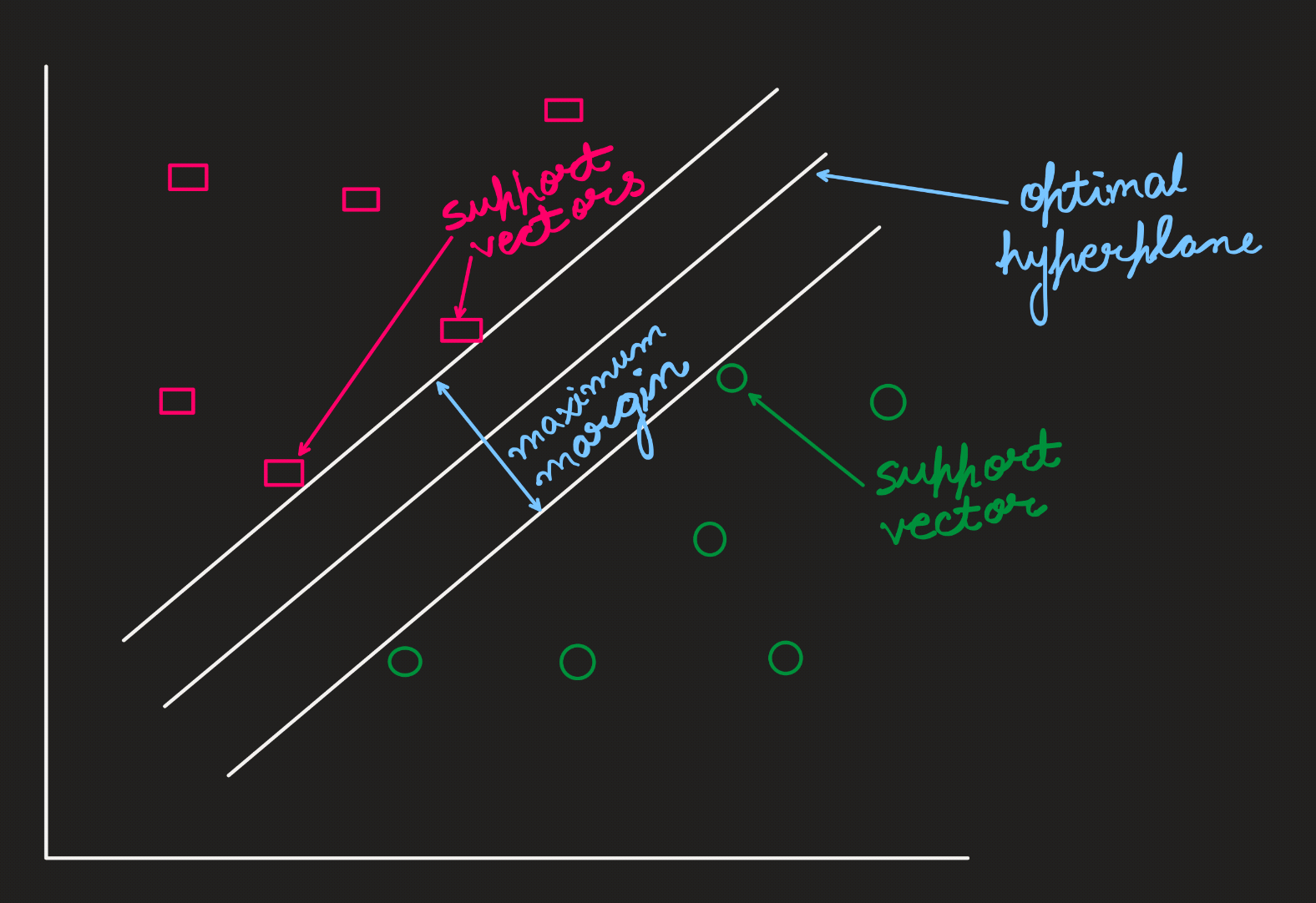


Figure 6 Support Vector Machine

Common Kernel Functions: Linear, polynomial, radial basis function (RBF) and sigmoid are some of the most commonly used kernel functions. The selection of a kernel can have a significant impact on how well an SVM performs.

An accuracy of 99% and f1 score of 99.6% was obtained using SVM

A graph showing a number of numbers

Description automatically generated with medium confidence

Figure 7 Confusion Matrix for Support Vector Classifer

[X] IMPROVING ACCURACY

1. Hyperparameter Tuning:

The proper setting of hyperparameters in the machine learning pipeline by performing hyperparameter tuning can increase the efficiency of machine learning models in various ways. Follow the steps below:

1) Identify and select the parameters that need to be tuned and optimize them. The manual process of finding the hyperparameters can be a model improvement process step.

2) Choose a search space in which only hyperparameters can be searched. For example, a range of values or a set of discrete values can be the search space for each of these hyperparameters.

3) Select a Search Strategy: Select a strategy from the options given on-the-spot hyperparameter optimization such as:

Discussing specific types of hyperparameters.

For example:-

1) Grid Search: This approach

attempts to search through the exhaustive search by all possibilities of

hyperparameters in the search space.

2) Random Search: This strategy involves

randomly sampling a few hyperparameters from the space and then evaluating them.

3) Bayesian Optimization: This technique is based on inspired models which give guidance on what steps to take in the exploration

Evaluate Performance: For each option of hyperparameters, evaluate the model's performance using a validation or by applying cross-validation. Our method of Bayesian optimization hyperparameters tuning resulted in increased accuracy and f1 score for both random forest and svm.

Select Best Hyperparameters: Assess the performance of various hyperparameter configurations and then choose the best combination of hyperparameters that delivers the best result for the validation set.

Refinement: If you wish to, you can further refine your approach by narrowing the search space around the best hyperparameters, and repeat the optimization process.

Final Model Training: When you have determined the most suitable hyperparameters, then proceed to train the last model by using these hyperparameters on the whole training set.

Hyperparameter tuning has the potential to greatly increasing the accuracy of machine learning models and is a necessary intermediate step in the development process of models.

Moreover, among the hyperparameters of the Support Vector Machine (SVM) are C (regularization parameter), type of kernel(such as linear, RBF), gamma, and degree for polynomial kernel.

While for Random Forest, the most important hyperparameters are the number of trees (n\_estimators), the maximum depth(max\_depth), min\_samples\_to\_split (min\_samples\_split), min\_samples\_per\_leaf(min\_samples\_leaf), and max\_features (max\_features).

1. Calibration:

Calibration is required in Random Forest and Support Vector Machine (SVM) classifiers to improve the prediction accuracy using a match between their predicted probabilities and the actual probabilities of class membership. For example, this step tackles issues such as over- or under-confidence mostly when the predictive probabilities move to the extreme or when the datasets have imbalanced class distribution. Calibration is a process that involves the compensation of bias, confidence improvement, and model reliability enhancement in the form of the probability estimates that each single prediction is associated with.

In the process of classification, one of the features of calibration that is especially important is its effect on the decision-making and ranking procedures. This, it's a must, as the probability estimates that come with the predictions are not only exact but also informative, so aids to have more conscious decisions. On top of that, calibration is especially helpful to minority classes in those groups that are not proportionate to each other in between the majority and minority classes.

The result of calibrating both the Random Forest and SVM is that the classification becomes more accurate and reliable. This precision aims to predict not only the instances with larger confidence, but with high consistency in the predictions made for those instances, which is a necessary condition entailed by the decision-making process.

[XI] RESULTS

Random Forest and SVM did exceptionally well among all the other classifiers with f1 scores of 97% and 99.67% respectively after applying hyperparameter tuning and calibration. Bayesian Optimization was used for hyperparameter tuning. Confusion matrix of the classifiers are attached:

A graph showing a number of numbers

Description automatically generated with medium confidence

Figure 8 Confusion Matrix for Random Forests after hyperparameter tuning and calibration

A graph showing a number of numbers

Description automatically generated with medium confidence

Figure 9 Confusion Matrix for SVM after hyperparameter tuning and calibration.

Before Hyperparameter tuning and calibration (for SVM):

Accuracy-0.99

F1 score-0.9966

After Hyperparameter tuning and calibration (for SVM):

Accuracy-0.99666

F1 score-0.996667729

This highlights the importance of hyperparameter tuning and calibration for accuracy and f1 score

[XII] CONCLUSION

In the field of electrical systems, detection, and categorization of defects with machine learning, transfer learning, and integrated technology systems such as Simulink is crucial. The significance of fault detection and classification in maintaining the dependability, safety and efficiency of electrical infrastructures cannot be overemphasized.

We created a fault data set using Simulink combined with MATLAB, and novel fault simulation methods that can capture different fault conditions like line-to-ground faults (LG), line-to-line faults (LL), line-to-line-to-ground faults (LLG) etc., through out this we employed various locations as well as time points at which these faults would occur systematically varied them so as to give an extensive understanding about the behaviour of a system under different circumstances.

Conversion of fault signals into spectrogram images was adopted to enable Convolutional Neural Networks (CNNs) along with transfer learning techniques which use Xception as feature extractor do their thing easily.

In our extensive tests through this study, we could assert SVM (Support Vector Machines), as well as Random Forest, are renowned for their high levels of precision in the matter of classification of the errors. At the same time hyperparameter tuning further improved these classifiers’ performances thus proving their efficacy under real world scenarios of faults occurrence.

**[XIII] SIMILAR WORKS**

* Jasmine Gnanamalar and colleagues in [15] present a streamlined method that combines a Convolutional Neural Network (CNN) with a Support Vector Machine (SVM) for enhanced efficiency. Key features include:
* The Hilbert–Huang transform (HHT) extracts features from current signals.
* These features serve as input for the proposed SVM–CNN algorithm.
* The SVM–CNN model detects and classifies faults within 2 ms.
* This strategy achieves a fault classification accuracy of 99.87% within 2 ms.
* Compared to existing methods, this technique shows significant improvement.

[xiv] References

1. M. Srivastava, S. K. Goyal, A. Saraswat and G. Gangil, "Simulation Models for Different Power System Faults," 2020 IEEE International Conference on Advances and Developments in Electrical and Electronics Engineering (ICADEE), Coimbatore, India, 2020, pp. 1-6.
2. Z. Chen, Y. Shi, Y. Wang, X. Li, X. Yu and Y. Shi, "Unlocking Signal Processing With Image Detection: A Frequency Hopping Detection Scheme for Complex EMI Environments Using STFT and CenterNet," in IEEE Access, vol. 11, pp. 46004-46014, 2023.
3. F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," in Proceedings of the IEEE, vol. 109, no. 1, pp. 43-76, Jan. 2021.
4. W. W. Lo, X. Yang and Y. Wang, "An Xception Convolutional Neural Network for Malware Classification with Transfer Learning," 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 2019, pp. 1-5.
5. X. Wu, R. Liu, H. Yang and Z. Chen, "An Xception Based Convolutional Neural Network for Scene Image Classification with Transfer Learning," 2020 2nd International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China, 2020, pp. 262-267.
6. F. -J. Yang, "An Implementation of Naive Bayes Classifier," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 301-306.
7. Y. Xiao, W. Huang and J. Wang, "A Random Forest Classification Algorithm Based on Dichotomy Rule Fusion," 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing,

China, 2020, pp. 182-185.

1. M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998..
2. Z. Wang, M. Agung, R. Egawa, R. Suda and H. Takizawa, "Automatic Hyperparameter Tuning of Machine Learning Models under Time Constraints," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4967-4973.
3. T. Barrett and A. K. Mishra, "Statistical Study of Sensor Data and Investigation of ML-Based Calibration Algorithms for Inexpensive Sensor Modules: Experiments From Cape Point," in IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-10, 2024, Art no. 1003310..
4. Praveen Rai, Narendra D. Londhe, Ritesh Raj,“Fault classification in power system distribution network integrated with distributed generators using CNN” Electric Power Systems Research, Volume 192,2021,106914,ISSN 0378-7796.
5. Gnanamalar, A.J., Bhavani, R., Arulini, A.S. et al. CNN–SVM Based Fault Detection, Classification and Location of Multi-terminal VSC–HVDC System. J. Electr. Eng. Technol. 18, 3335–3347 (2023).