



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
CENTRAL CAMPUS, PULCHOWK**

THESIS NO.: 079/MSPSE/023

Adaptive Hybrid AI Model for the Detection of the Power System Faults

by

Suresh Prajapati

**A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN POWER SYSTEM ENGINEERING**

**DEPARTMENT OF ELECTRICAL ENGINEERING
LALITPUR, NEPAL**

APRIL, 2025

COPYRIGHT ©

The authors have agreed that the library, Department of Electrical Engineering, Pulchowk Campus, Institute of Engineering may make this report freely available for inspection. Moreover, the authors have agreed that permission for extensive copying of this thesis for the scholarly purpose may be granted by the professor(s) who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electrical Engineering, Pulchowk Campus and Institute of Engineering in any use of the material of this project report. Copying or publication or the use of this report for financial gain without the approval of the Department of Electrical Engineering, Pulchowk Campus, Institute of Engineering and author's written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head of Department
Department of Electrical Engineering
Institute of Engineering, Central Campus, Pulchowk, Lalitpur, Nepal



Accredited by University Grants
Commission (UGC) Nepal 2020

त्रिभुवन विश्वविद्यालय
TRIBHUVAN UNIVERSITY
इन्जिनियरिङ्ग अध्ययन संस्थान
INSTITUTE OF ENGINEERING
पुल्चोक क्याम्पस
PULCHOWK CAMPUS

DEPARTMENT OF ELECTRICAL ENGINEERING

Pulchowk, Lalitpur

Institute of Engineering
Electrical Engineering Department
Pulchowk Campus

CERTIFICATE OF APPROVAL

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a thesis entitled " **Adaptive Hybrid AI Model for the Detection of the Power System Faults** " submitted by **Suresh Prajapati** in partial fulfilment of the requirements for the degree of **Master of Science in Power System Engineering**.

Asst. Prof. Dr. Kamal Chapagain

Department of Electrical and Electronics Engineering

Kathmandu University

(External Examiner)

Asst. Prof. Dr. Bishal Silwal

Program Coordinator

MSc. in Power System Engineering

Pulchowk Campus, Lalitpur

(Supervisor)

Assoc. Prof. Dr. Basanta Kumar Gautam

Head of Department

Department of Electrical Engineering

Pulchowk Campus, Lalitpur

April 2025

ACKNOWLEDGEMENT

I would like to extend our heartfelt gratitude to our supervisor, **Assist. Prof. Dr. Bishal Silwal**, for his invaluable guidance, insightful suggestions and continuous encouragement throughout this project. His support and collaboration were instrumental in bringing this work to completion.

I also express our sincere appreciation to the **Department of Electrical Engineering** for granting us the opportunity to undertake my thesis work in my chosen field of interest.

Again, my special thanks go to **Assist. Prof. Dr. Bishal Silwal**, MSc in Power System Engineering Coordinator, for his support and guidance.

Lastly, I am deeply grateful to our friends and seniors for their unwavering assistance and encouragement from the very beginning of this project to its successful completion, whether directly or indirectly.

ABSTRACT

Accurate fault detection in power systems is essential to ensure both the reliability and efficiency of the electrical network. However, physically inspecting faults is expensive and requires numerous sensors for accurate diagnosis. To overcome this, ML algorithm can be deployed in power system fault identification work. Since traditional machine learning and artificial neural network (ANN) models often encounter limitations in accuracy, this literature introduces an efficient, cost-effective fault detection model capable of accurately identifying both the type and location of faults. Rather than relying solely on conventional machine learning or neural network models, this work employs a hybrid model that combines multiple AI models with advanced optimization techniques. These optimization techniques are used to weigh the AI models and fine-tune their hyperparameters to achieve improved accuracy. As most works based on machine learning are only applied to the static power system, this work presents an adaptive fault detection model which can detect fault type and location in a dynamic power system instantly with minimum error. In this proposed approach, three distinct machine learning algorithms are integrated with the Particle Swarm Optimization (PSO) technique. Power system failures are simulated using MATLAB/SIMULINK, while the machine learning model is developed with the Python programming language. This AI Model is trained with fault data without any pre-computation. In contrast to many existing methods, this approach does not require data conditioning or feature extraction. Instead, the entire AI model is trained and developed using raw bus data, such as voltage and current. The result demonstrates that the developed AI model can detect fault location and fault type. This model is designed to work across various topologies, systems, and data collections, demonstrating its robustness and adaptability. This makes it a promising approach for the monitoring and maintenance of real-world power systems. Finally, the research gap and future direction are discussed.

TABLE OF CONTENTS

COPYRIGHT ©.....	i
ACKNOWLEDGEMENT	iii
ABSTRACT.....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES	ix
LIST OF TABLES.....	x
LIST OF ABBREVIATION.....	xi
CHAPTER 1: INTRODUCTION	1
1.1 Background.....	1
1.2 PROBLEM STATEMENT.....	2
1.3 OBJECTIVES.....	2
1.4 Scope.....	3
CHAPTER 2: LITERATURE REVIEW.....	4
2.1 Literature Research Gap	5
CHAPTER 3: THEORETICAL BACKGROUND	7
3.1 Faults in Power System.....	7
3.2 Machine Learning.....	8
3.2.1 Adaptive Machine Learning	9
3.3 ML Algorithms	10
3.3.1 Decision Tree Classifier.....	10
3.3.2 Random Forest Classifier.....	11

3.3.3	Support Vector Machine	12
3.3.4	Gradient Boosting	12
3.3.5	Hyper parameter.....	13
3.3.6	Ensemble.....	13
3.3.7	Over Fitting and Under Fitting	14
3.3.8	Performance Metrics for ML Algorithms	14
3.3.9	Cross Validation.....	16
3.3.10	Root Mean Squared Error	17
3.4	PSO	18
3.4.1	Pseudocode for PSO	18
3.5	Load Flow Studies	19
3.5.1	Bus Classification	20
3.5.2	Power Flow Calculation Methods.....	21
3.5.3	Load Flow Equation:.....	22
3.5.4	Algorithm for NRLF.....	23
3.6	Contingency Anaylsis	26
CHAPTER 4: METHODOLOGIES		28
4.1	Proposed System.....	28
4.2	Data Collection	29
4.3	Model Development using Simulation	30
4.3.1	Load Flow Analysis	30
4.3.2	Contingency Analysis	30
4.3.3	Fault Analysis	31
4.4	Steps for Data Generation and Model Development	32

4.4.1	Dataset Generation.....	32
4.4.2	Hybrid AI Model Development.....	32
4.4.3	Adaptive ML Model development.....	34
4.5	TOOLS AND SOFTWARE	35
4.6	Final Adaptive Hybrid AI Model.....	36
CHAPTER 5: RESULT AND DISCUSSION.....		39
5.1	Load Flow Result:.....	39
5.2	Post Contingency Analysis report:.....	39
5.3	Fault Analysis Result:	40
5.4	PSO Optimization Result:.....	41
5.5	Model Testing:	41
5.5.1	Fault Type Identification.....	42
5.5.2	Fault Location Identification.....	42
5.5.3	Fault Distance Identification.....	43
5.5.4	Adaptive Model Testing Result	44
5.5.5	Discussion on Result.....	44
CHAPTER 6: CONCLUSION		46
CHAPTER 7: FUTURE WORKS		47
CHAPTER 8: REFERENCES		48
Appendix A: Simulation Model of INPS System.....		53
Appendix B: INPS TL Data.....		54
Appendix C: Bus Data		57
Appendix D: Generation Data		58
Appendix E: Substation Data.....		59

Appendix F: Simulation Result.....	60
Appendix G: Load Flow Result	63
Appendix H: Dataset sample	66
Appendix I: Publication	67
Appendix L: Plagiarism Test Report	77

LIST OF FIGURES

Figure 3-1: Different types of faults in the Power System.	7
Figure 3-2 : ML Algorithm.....	8
Figure 3-3: Difference between Static and Adaptive ML.....	9
Figure 3-4: Random Forest.....	11
Figure 3-5: Ensemble.....	14
Figure 3-6: overfitting, underfitting and good fitting.....	15
Figure 3-7: Confusion Matrix.....	15
Figure 3-8: K Fold Cross Validation.....	17
Figure 3-9: Working of PSO.....	18
Figure 3-10: PSO illustration.....	18
Figure 4-1: Proposed INPS System.....	29
Figure 4-2: Workflow of the Hybrid AI model.....	34
Figure 5-1: Confusion Matrix for Fault type Identification.....	42
Figure 5-2: Confusion Matrix for Fault Location Identification.....	43
Figure 5-3: Confusion Matrix for Fault Distance Identification.....	44
Figure 5-4: Accuracy vs Time Curve.....	45

LIST OF TABLES

Table 1: Buses Types in Load Flow	21
Table 2: Post Contingency Analysis report	39
Table 3: Performance Metrics for Fault Type Identification	42
Table 4: Performance Metrics for Fault Location Identification	43
Table 5: Performance Metrics for Fault Distance Identification	43

LIST OF ABBREVIATION

MATLAB	Matrix Laboratory
Sklearn	Scikit Learn
ML	Machine Learning
SVM	Support Vector Machine
RF	Random Forrest
DT	Decision Tree
IEEE	Institute of Electrical and Electronics Engineers
KNN	K-Nearest Neighbor
AI	Artificial Intelligence
CV	Cross Validation
RMSE	Root Mean Squared Error
INPS	Integrated Nepal's Power System
MLP	Multilayer Perceptron
GB	Gradient Boosting
NB	Naïve Bayers
PSO	Particle Swarm Optimization
GA	Genetic Algorithm

DER	Distributed Energy Resources
NEA	Nepal Electricity Authority
GOD	Grid Operational Department
TL	Transmission Line Data
NRLF	Newton Raphson Load Flow
GSLF	Gauss Seidel Load Flow
FDLF	Fast Decouple Load Flow
PI	Performance Index
CPU	Central Processing Unit
GPU	Graphics Processing Unit

CHAPTER 1: INTRODUCTION

1.1 Background

In the current era, every social class relies heavily on electric power for daily functioning, leading to a significant increase in dependence on power systems. This growing reliance has driven rapid advancements in the power sector over the past few decades, particularly in the transmission and distribution domains [1]. However, transmission and distribution lines are often exposed to harsh environmental conditions such as rain, lightning, strong winds, snow, floods, and other external disturbances like birds or human-related accidents. These factors can lead to faults that result in power outages. When faults occur in a power system, they compromise its reliability from the customer's perspective. To ensure a continuous power supply, utility providers must detect and classify faults efficiently. Accurate fault detection and identification enable quicker restoration of service. Manually locating faults is typically expensive, slow, and challenging, making computational approaches a more viable solution. The increasing availability of computing power, combined with lower hardware costs and miniaturization, has led to the growing adoption of advanced fault classification techniques, such as machine learning, to enhance the speed and accuracy of outage response.

Alongside traditional energy generation systems, smaller on-site power generation units, known as DERs, have become increasingly prominent. These systems have opened new avenues for investment, particularly in technologies like microgrids [2]. A microgrid is a localized, self-sufficient group of energy sources that can be remotely controlled and operated either in connection with the main grid or independently in islanded mode. It relies on specialized controllers to manage energy dispatch effectively. However, as the share of renewable energy in the grid increases, the overall system experiences a decline in rotating inertia. This reduction can lead to instability and, in severe cases such as major faults or catastrophic events, may trigger widespread blackouts. As a result, it is crucial to implement timely fault-clearing

mechanisms and take preventive measures to minimize the risk of extensive outages or cascading failures.

1.2 PROBLEM STATEMENT

Physically checking for the fault is proven to be costly and many physical sensing devices are required for fault diagnosis. Also due to integration of different renewable energy sources which lack inertia causes power system prone to faults [3]. It is exceedingly difficult for a utility to detect the fault type and location using voltage and current waveforms. As a conventional fault identification scheme, there is a method using relay. So, ML algorithm is introduced which can diagnosis fault from Past Data and classify them without additional devices. Also, there are many Electrical companies that have sufficient bus data which may help to train ML Algorithms. As in [4] different ML algorithms have different limitations and benefits, so hybrid techniques i.e. combining ML algorithms with Optimization techniques like GA or PSO can be integrated for better accuracy [5]. And finally, many presented algorithms are developed for static power system network that do not change frequently, which require adaptive models to work for real time data [6].

1.3 OBJECTIVES

1. The main objective of this work is to develop an Adaptive Hybrid AI Model (i.e. PSO based Ensemble ML Model) to detect the fault in the power system.
2. To achieve the main objective, this research aims to:
 - I. perform a simulation of a Power grid system and to study their waveforms for the faulty and non-faulty scenarios
 - II. To create a dataset from the simulation of the system for diverse types of faults and train ML algorithms with the datasets generated
 - III. To use the PSO Technique to ensemble the ML algorithms and develop the final Hybrid model

- IV. To test the final model with different metrics for fault classification and fault location identification.
- V. To test the model for different structural changes in the system to make the system adaptive

1.4 Scope

1. Load Flow analysis of the proposed power grid model and perform contingency analysis.
2. Perform a fault analysis and study the current voltage waveform for different fault cases.
3. Dataset is created from the current and voltage waveform for ML Model Training (i.e. dataset doesn't consist of other electrical parameters except three phase current and voltage)
4. Hyperparameters of ML models are tuned using PSO Optimization Technique; also, ML Models are weighted using PSO Optimization Technique for ensemble Model.
5. Dataset is trained using Ensemble Model without any data preprocessing or feature selection and finally model is tested using different performance metrics.
6. Adaptivity of the model is also tested using different structural changes in Power grid system.

CHAPTER 2: LITERATURE REVIEW

Numerous fault detection techniques have been introduced in recent years and extensively documented in power system literature. These methods aim to enhance the accuracy, speed, and reliability of identifying faults within electrical networks, contributing to more efficient system monitoring and faster restoration times [7] [8]. In power systems, protective relays detect faults by sensing large fault currents and triggering isolation to maintain system stability [9]. Most modern distributed generators (DGs) use inverter technology, which contributes minimally to power system inertia and balance. [10]. This poses a challenge, as the low current may not be enough to activate relays, putting inverter-based modern distribution systems at risk. Frequency Disturbance Recorders, Optimally Located Phasor Measurement Units and different other equipment have been reported to identify faults in transmission line [11]. Also, Phase angle measurement has been used to identify line outages in power network [12]. Communication-assisted directional OC relays and the differential localization scheme has been applied to perform Hierarchical protection for fault identification occurred in Power system [6]. In [13], A dual setting directional over-current relay-based communication assisted scheme is presented. A three-stage approach based on real time data for protection of power system is proposed in [14]. Adaptive fault protection techniques based on quadrature and zero sequence components have been successfully applied in [15]. Hardware-based solutions are often not scalable and can be impractical due to high financial costs. As a result, a range of advanced, intelligent, computer-based prediction schemes have been developed to deal with the challenges of fault identification and analysis. These software-driven approaches offer more flexibility, cost-effectiveness, and scalability for efficiently scanning and diagnosing faults in power systems [4].

Different computational intelligence and data-based schemes have been analyzed in the past few years. Using factors like accuracy, cost, fault location identification time, flexibility and adaptability, the performance of above-mentioned schemes can be evaluated. The higher accuracies can be gained by using different

methods like performing islanding operation [2]. Mainly, hybrid methods with suitable limit settings and signal processing schemes have been reported in [16] [17] [18] [19]. Machine learning techniques enable computers or systems to recognize patterns from input data, making them appropriate for detecting faults in power systems. For example, algorithms like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) are commonly used to detect faults within the electrical grid [20] [21]. The fault detection in the power network shall be sufficiently fast either using any fault detection methods. Accuracy and fault detection time, both should be considered while proposing any model. According to IEEE 1547 standards, fault detection time shall be within 2 seconds for better performance of model [22].

2.1 Literature Research Gap

Most literature has used a relay-based protection scheme, which needs a huge amount of fault current to identify the fault situation. With that, to identify the faults in the power network the recommended schemes should be within time constraints. It has been noted that many fault detection schemes are developed for static power networks, and when system changes occur, such as adding or removing a line or introducing new buses, most of the presented techniques may not perform well. Therefore, the system needs to be adaptive [23]. The fault detection model must be budget friendly, which is lacking in the above-mentioned literature. In other methods, signal processing or feature selection ideas are required to train algorithms which will slow down the training process. In this literature raw datasets will be utilized (input voltage and current signals) for predictions. And finally, most ML based literatures deployed ML Algorithm with its default hyper parameters which may not give accurate result. For the above Problem Hyper Parameter tuning can be performed.

In this work different basic ML Models (also known as weak learners) are combined. PSO, a strong optimization technique, will be applied to combine the models in this proposed method. Optimized weights are calculated using PSO and set to this Basic ML Models to make a strong predictor together. Again, PSO is applied to tune these ML Models to increase accuracy of this optimized Algorithm. Each weak ML model's limitation will be covered by another weak learner by adopting this scheme.

The ensemble technique is what drives the effectiveness of the proposed method. The core concept involves determining the optimized weights for each weak learner and aggregating them using soft voting during testing. The ensemble Model will be continuously trained and tested for streaming data or real-time data to make the model adaptive. By using this adaptive technique, model can boost accuracy and performance while minimally increasing training time. This study presents a highly effective, cost-efficient and adaptive fault detection model capable of identifying fault types and locations in real time with remarkable accuracy. Instead of relying solely on traditional machine learning or neural network models, the proposed approach employs a hybrid methodology. This hybrid model integrates multiple AI algorithms with robust optimization techniques, where the optimization process assigns weights to the AI models to enhance accuracy.

CHAPTER 3: THEORETICAL BACKGROUND

The fundamental theories of power system faults, contingency analysis and load flow analysis are presented in the below sections. Also, basic overview of Machine Learning, their types, performances metrics etc. are discussed.

3.1 Faults in Power System

Since the beginning of power system networks, faults have been the leading cause of power cuts, resulting in considerable economic and societal consequences. In power networks, faults can occur due to various reasons, including equipment aging, animal interference, accidents, and adverse weather. Notably, faults account for approximately 80% of power interruptions in distribution systems. These faults are generally classified into two types: open circuit (series) faults and short circuit (shunt) faults. [24]. Further types of faults in power systems are classified as in Figure 3-1 [4]

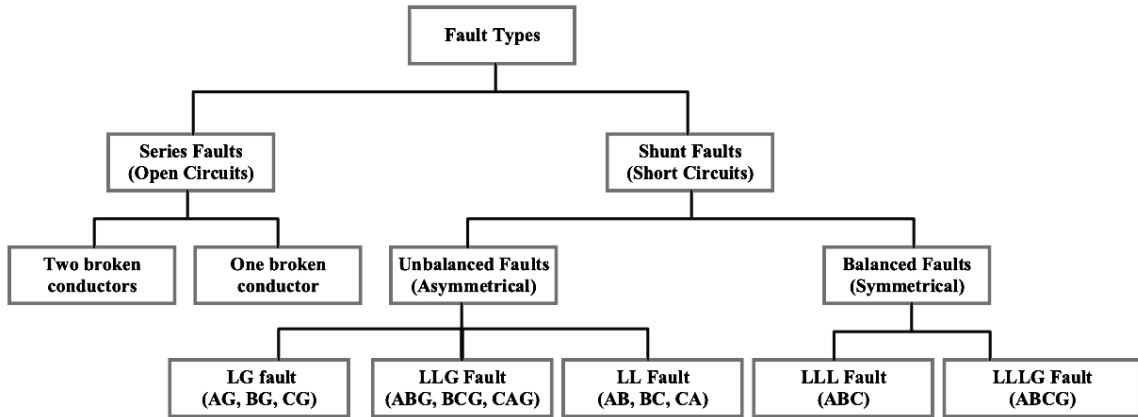


Figure 3-1: Different types of faults in the Power System.

Expressions for Current for different types of faults are specified in equations below for fault in k^{th} bus. Equations (1), (2) and (3) are expressions of phasor current for LG fault, LL fault and LLG fault respectively.

$$I_f = \frac{3V_f}{Z_{kk1} + Z_{kk2} + Z_{kk0} + 3Z_f} \quad (1)$$

$$I_f = \frac{j\sqrt{3}V_f}{Z_{kk1} + Z_{kk2} + Z_f} \quad (2)$$

$$I_f = \frac{3V_f Z_{kk2}}{Z_{kk1}(Z_{kk2} + Z'_{kk0}) + Z_{kk2}Z'_{kk0}} \quad (3)$$

where, $Z'_{kk0} = Z_{kk0} + 3Z_f$

In the above expressions, I_f is fault current, V_f is pre-fault voltage at k^{th} bus and Z_f is fault Impedance. Z_{kk0} , Z_{kk1} and Z_{kk2} are zero sequence component, positive sequence component and negative sequence component of kk^{th} element of Z-Bus.

3.2 Machine Learning

Machine Learning, a branch of **Artificial Intelligence**, allows systems to automatically learn and enhance their performance through experience, without the need for explicit programming. It aims on creating algorithms competent of analyzing data, recognizing patterns, and making decisions or predictions based on the provided input [25]. ML systems use algorithms to analyze substantial amounts of data, study from the insights and then make decisions. The more data a system is trained to, the better it gets at performing tasks.

Machine Learning algorithms, for example, can be trained to perform specific tasks, such as identifying flowers in a photograph, predicting housing prices, or making chess moves.

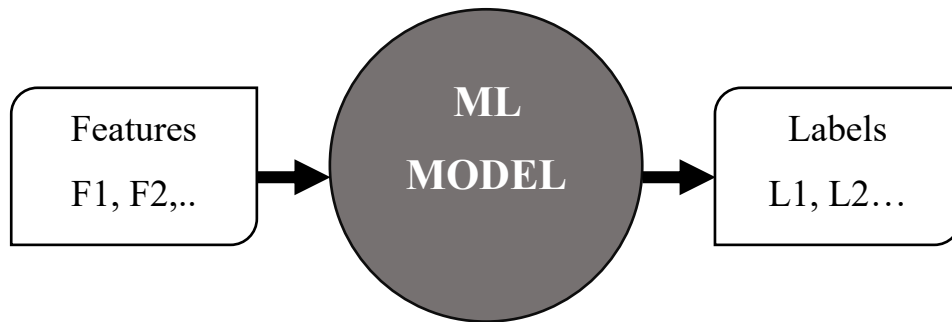


Figure 3-2 : ML Algorithm

The block diagram of the ML Algorithm is shown in Figure 3-2 Which includes Features as input and labels as output. Features are the input variables (also called independent variables) used by a machine learning model to make predictions. They represent measurable characteristics or attributes of data. Features can be numerical (e.g., current, voltage) or categorical (e.g., sex, state). A label (also called the dependent variable or target variable) is the output that the model is trying to predict. Labels are the ground truth that the model learns from during training. Features and labels are data provided to the ML model for training model. After training the ML Model we can get desirable output for given input or features.

3.2.1 Adaptive Machine Learning

Adaptive Machine Learning are the systems and algorithms capable of dynamically updating and improving their performance as new data becomes available or as the environment changes [26]. Unlike traditional or static machine learning models that are trained once and then deployed, adaptive systems are designed to continually learn, adjust and optimize without requiring complete retraining from scratch as shown in Figure 3-3. Key Characteristics of Adaptive Machine Learning are:

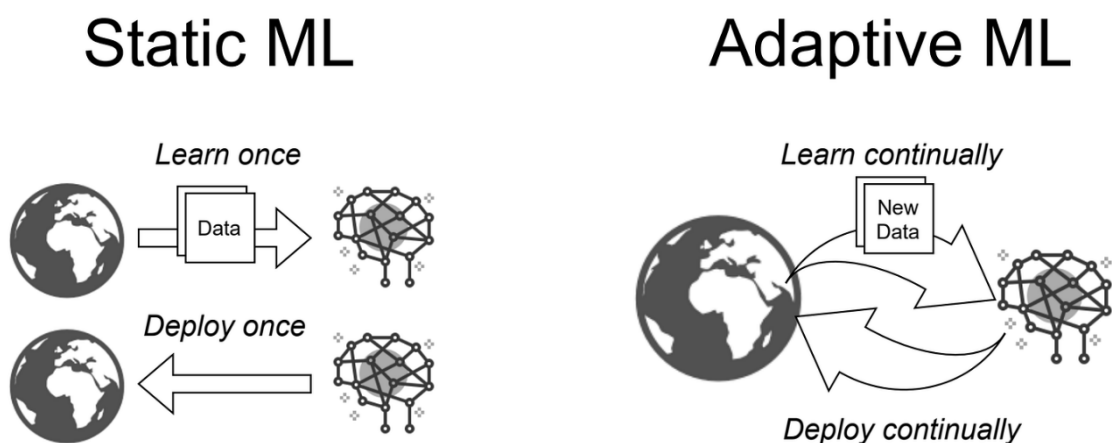


Figure 3-3: Difference between Static and Adaptive ML

1. **Continuous Learning:** The model updates itself incrementally as new data comes in.
2. **Real-Time Adaptation:** Capable of adjusting immediately or near real-time to changing conditions.

3. **Context Awareness:** Adjusts its behaviour based on changes in the environment or underlying data patterns.
4. **Minimal Human Intervention:** Designed to automate the learning process as much as possible.

3.3 ML Algorithms

3.3.1 Decision Tree Classifier

The Decision Tree (DT) Classifier, commonly used for classification problems, is a supervised machine learning algorithm [25]. It functions by splitting the dataset into smaller subsets according to feature values, creating a tree-like structure where each node represents a feature, and each branch signifies a decision rule. **Key Concepts of Decision Trees are:**

Nodes:

1. **Root Node:** it is a starting point of the decision tree containing the entire dataset.
2. **Internal Nodes:** It represents decisions or splits based on features.
3. **Leaf Nodes:** It represents final class labels or outcomes.

Splitting Criterion: Decision trees employ metrics to identify the most suitable feature for splitting.

1. **Gini Impurity:** Measures the likelihood of incorrect classification at a node.
2. **Entropy/Information Gain:** Assesses the decrease in uncertainty following a split.
3. **Variance Reduction** (for regression trees): Measures how much variance is reduced by a split.

Stopping Criteria:

1. Maximum tree depth.
2. Minimum samples per leaf node.

3. Minimum samples are required to split a node.
4. No significant improvement in split metrics.

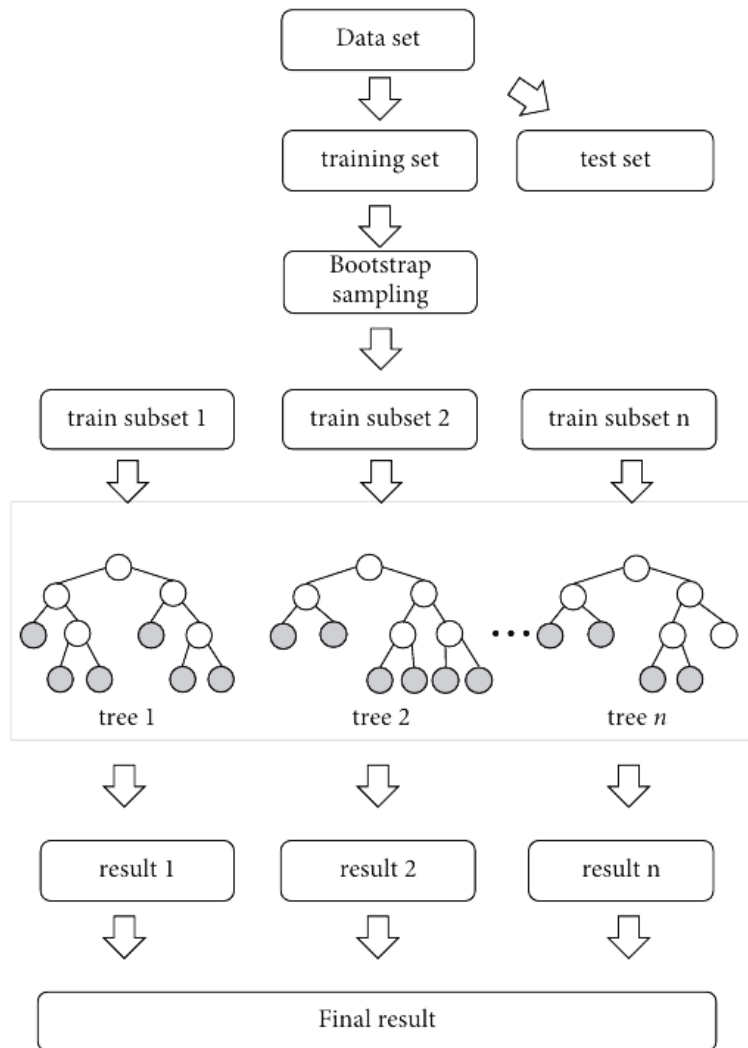


Figure 3-4: Random Forest

3.3.2 Random Forest Classifier

Random Forest is a robust and commonly utilized machine learning algorithm that falls under the ensemble learning category [27]. It is designed to improve prediction accuracy and control overfitting by aggregating the outputs of many decision trees. Rather than relying on a single tree, Random Forest constructs a collection (or "forest") of decision trees during execution.

Each tree in the forest is built using a method known as bootstrap aggregating, or bagging, which involves randomly selecting a subset of the training data with replacement. Moreover, when dividing nodes in each tree, only a randomly chosen subset of the features is taken into account. This introduces diversity among the trees and helps ensure that they are not all learning the same patterns from the data.

One of the key strengths of Random Forest is its resilience to overfitting, especially when compared to a single decision tree. It also handles high-dimensional data well, manages missing values, and provides estimates of feature importance, making it an important tool for feature selection and interpretation. Workflow of RF Algorithm is as shown in Figure 3-4.

3.3.3 Support Vector Machine

Support Vector Machine, a supervised ML algorithm, which is used for classification and regression tasks, though it is mostly used for classification [25].

The core concept of SVM is to identify the optimal boundary (or hyperplane) that separates data points from different classes with the widest possible margin. The points that are nearest to this boundary are known as support vectors, and they play a key role in determining the hyperplane's position and direction. In a 2D space, the decision boundary is a line; in higher dimensions, it becomes a plane or hyperplane. SVM chooses the boundary that maximizes the distance (margin) between the classes. When the data cannot be separated linearly, SVM applies a method known as the kernel trick, which transforms the data into a higher-dimensional space where a clear separation becomes possible.

3.3.4 Gradient Boosting

GB is an ML algorithm used for regression and classifying problems. It builds the model in an iterative way by summing weak learners (general decision trees) into building up a strong learner. The approach is to develop new models on an iterative basis to correct the errors of previous models. Work of Gradient Boosting is as Below.

1. Initialize the model with simple prediction, most often the target means.

2. Calculate residuals (errors) of the model.
3. Train a weak learner (a shallow decision tree, for example) to predict these residuals.
4. Update the model by incorporating the predictions of the new learner, scaled by a learning rate.
5. This process is repeated until a stopping condition is met, such as reaching a predetermined number of iterations or observing minimal improvement.

3.3.5 Hyper parameter

Hyper parameters in ML Algorithms are parameters that control the learning process and model actions but are not derived from the data itself. These are defined prior to the start of training and can significantly influence the model's performance. Some Hyper Parameters include:

1. **Learning Rate:** Determines how much the model updates its weight during training.
2. **Number of Epochs:** It is the frequency or number of times the model gets to see the entire dataset during training.
3. **Kernel Type:** In SVM it decides the kind of transformation to the input space.
4. **C Parameter:** In SVM, it is used to control the influence of outliers. A low value of C allows more outliers by creating a wider margin, while a high value of C restricts outliers by enforcing a narrower margin and stricter separation.

Number of estimators: In RF it is the number of decision trees in the forest

3.3.6 Ensemble

Ensemble refers to a method where multiple models (often called "base learners" or "weak learners") are combined to achieve better predictive performance than any individual model. The idea is that combining the predictions of different models can reduce the risk of overfitting and improve generalization. Some Commonly

used ensemble models are bagging, Boosting, Stacking, Voting, Blending etc. The Ensemble Model is as shown in Figure 3-5.

Bagging, short for Bootstrap Aggregating, is an ensemble learning method aimed at enhancing the stability and accuracy of machine learning models, especially

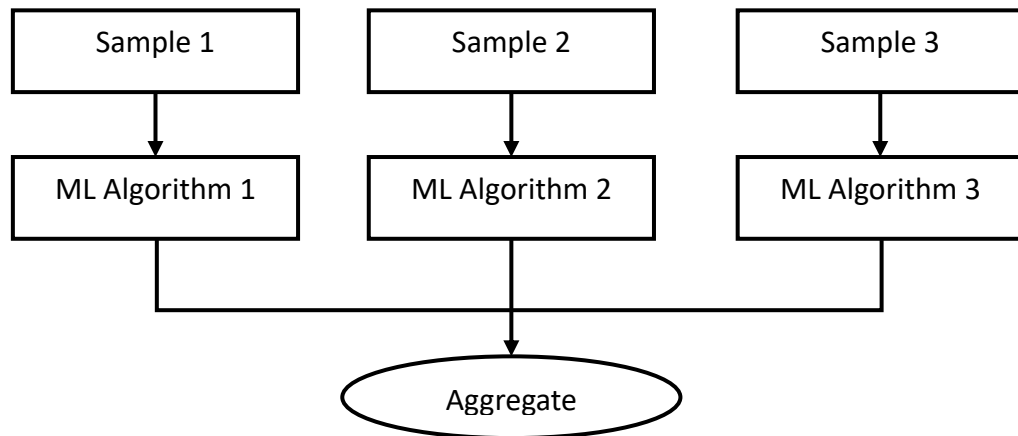


Figure 3-5: Ensemble

for algorithms prone to overfitting (e.g., decision trees). It combines predictions from multiple versions of the same base model, each trained on a slightly different dataset.

3.3.7 Over Fitting and Under Fitting

Overfitting happens when a model captures not just the true patterns in the training data but also the noise and irrelevant details, resulting in high accuracy on the training set but poor performance on unseen data. In contrast, underfitting arises when the model is too simplistic to grasp the underlying structure of the data, leading to poor performance on both training and test sets. It results in high errors on both training and test sets. A model achieves good fitting when it captures the underlying patterns in the data without memorizing noise. It performs well on both training and test data, ensuring good generalization [28]. The curves presenting overfitting, underfitting and good fitting are as shown in Figure 3-6.

3.3.8 Performance Metrics for ML Algorithms

Performance metrics help evaluate how well a machine learning model is performing. The selection of performance metric depends on the type of problem:

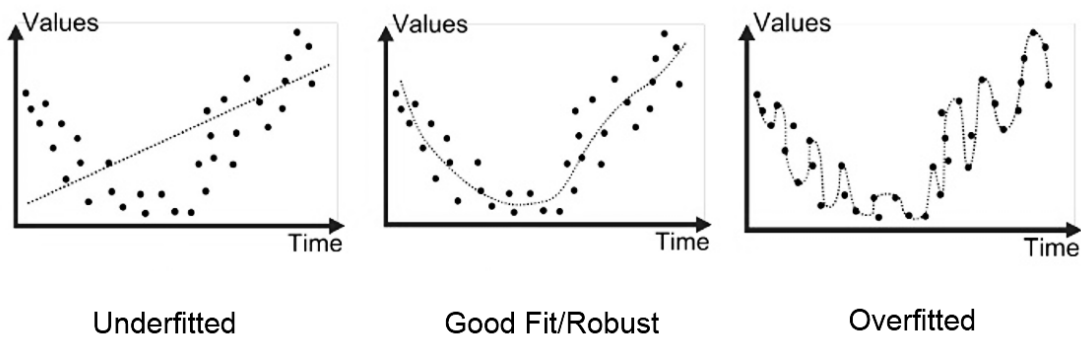


Figure 3-6: overfitting, underfitting and good fitting

classification, regression, or clustering. To calculate various classification performance metrics **confusion matrix** is used. A confusion matrix is a tabular representation used to analyze the performance of a classifying problem. It compares the actual (true) labels with the predicted labels and consists of four key components as below:

1. True Positives (TP): positive cases which are correctly predicted.
2. True Negatives (TN): negative cases which are correctly predicted.
3. False Positives (FP) (Type I error): Negative cases incorrectly predicted as positive
4. False Negatives (FN) (Type II error): Positive cases incorrectly predicted as negative (Type II error)

The Confusion matrix with their components is also illustrated in Figure 3-7. From the Confusion matrix different performance metrics (accuracy, precision, f1score and Recall) are derived as below.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 3-7: Confusion Matrix

1. **Accuracy:** Accuracy evaluates the overall effectiveness of a model by calculating the proportion of correctly predicted instances—both positive and negative—out of the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2. **Precision:** Precision calculates the proportion of correctly predicted positive cases among all instances predicted as positive, emphasizing the accuracy of the model's positive predictions.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3. **Recall:** Recall, also referred to as Sensitivity or the True Positive Rate, measures the proportion of correctly predicted positive cases out of all actual positive cases in the dataset.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4. **F1 Score:** F1 score is a harmonic means of recall and precision. It is used in the case where there is imbalance dataset.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

3.3.9 Cross Validation

Cross-validation (CV) is a statistical technique used in machine learning and data analysis to assess a model's performance. It works by splitting the dataset into multiple subsets, where the model is trained on some subsets and validated on the remaining ones. This approach helps evaluate how well the model generalizes to new, unseen data and aids in reducing the risk of overfitting.

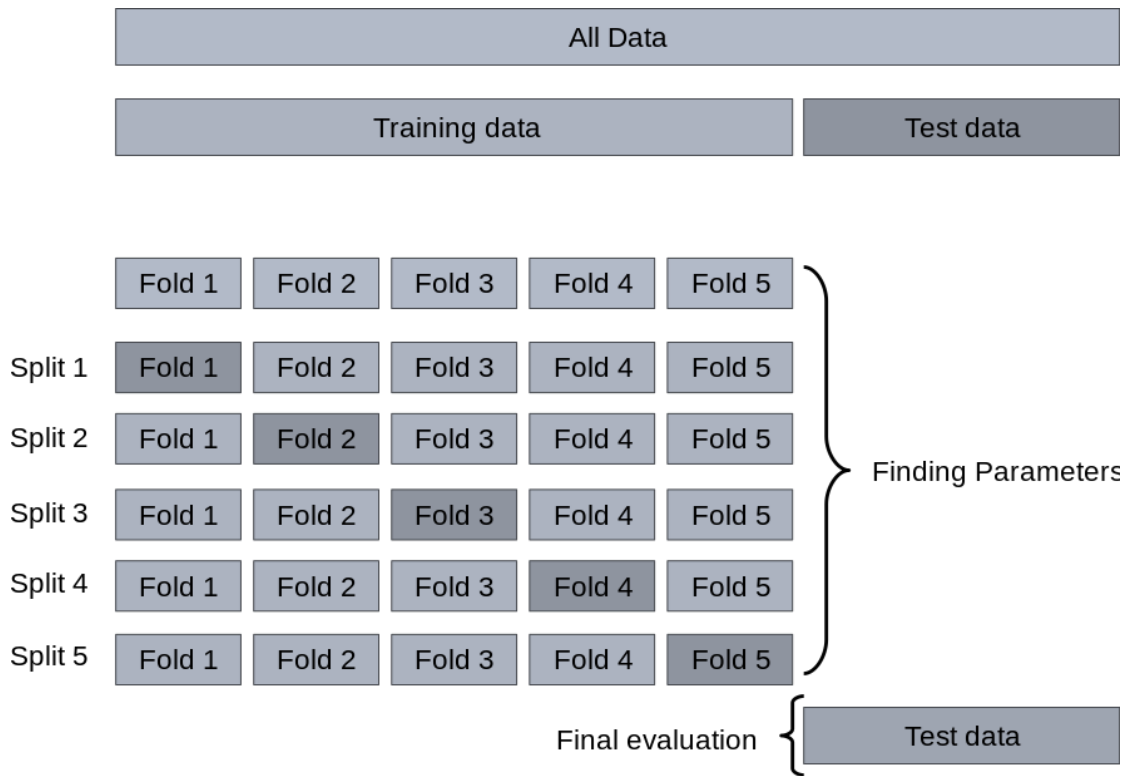


Figure 3-8: K Fold Cross Validation

Commonly used CV Techniques is K-Fold CV which is as shown in Figure 3-8. The dataset is divided into k equally sized folds. In each iteration, the model is trained on k-1 folds and evaluated on the remaining fold. This procedure is repeated k times, ensuring each fold serves as the test set once. The overall performance is then calculated as the average of the results from all iterations.

3.3.10 Root Mean Squared Error

RMSE is a widely used measure to estimate the error of a model in ML problems. It is the square root of the *mean* of squared differences between predicted and actual values. Mathematically,

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (1)$$

where:

y_i is the actual value,

\bar{y}_i is the predicted value,
 n is the total number of samples.

3.4 PSO

Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the collective behavior observed in bird flocking or fish schooling. As In Each bird from of Swarm following a bird (particle) which is near to the food (target) as shown in Figure 3-10. It is also known as Population based stochastic algorithm. A stochastic process is one that involves a random probability distribution or pattern, which can be

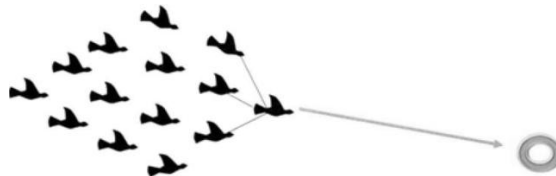


Figure 3-10: PSO illustration.

analyzed using statistical methods but cannot be predicted with complete certainty. It

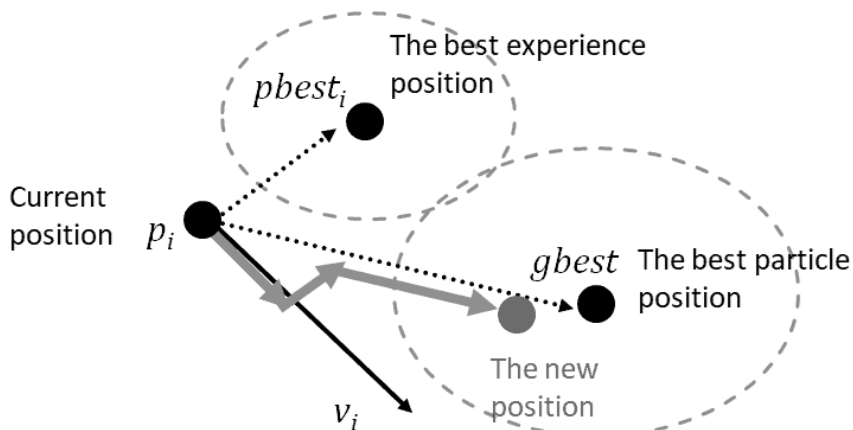


Figure 3-9: Working of PSO

has some similar characteristics to GA like Random Population, Fitness value and Update population steps. Genetic operator (crossover and mutation) is not used in PSO and Each particle has its own velocity. Each have $pBest$ and nearest have $gBest$, each will try to have $gBest$ updating velocity and weights.

3.4.1 Pseudocode for PSO

The step for each particle is shown below:

1. initialize particles.
for $i = 1$ to n ($n \rightarrow$ no. of iteration)
 2. for each particle:
 - a. Calculate fitness value (FV)
 - b. If $FV > pBest_{old}$ (Personal Best):
 - i. $pBest_{new} = FV$
 3. Select the particle with the best FV of all particles as the $gBest$ (Global Best)
 4. For every Particle:
 - a. Particle Velocity Calculation

$$v = v + c_1 \times rand(0,1) \times (pBest - pp)$$

$$+ c_2 \times rand(0,1) \times (gBest - pp)$$

$pp \rightarrow$ present position
 $c_i \rightarrow$ learning variable
 - b. Update particle position.
 5. $pp = pp + v$
- Is $error < \epsilon$

The Pseudocode or working of PSO can also be illustrated as in Figure 3-9.

3.5 Load Flow Studies

Load flow analysis is a widely used computational method in the operation, planning and control of recent power systems. Before implementing any changes—such as adding interconnections, new loads, generators, or transmission lines—utilities rely on power flow studies to assess the impact on the network. The primary goal of a load flow study is to determine the voltage levels at each bus when specific real power loads and generators are added. With this data, engineers can calculate both real and reactive power flows across individual branches, as well as assess the reactive power output of generators, ensuring optimal system performance [29].

Power system equations are generally non-linear and solving them requires iterative techniques. These methods work by updating values from previous iterations until the solution meets a predefined tolerance. Both direct and indirect iterative methods are available for solving these equations. Direct methods, though quicker in convergence, are better suited for smaller networks. On the other hand, indirect methods

are more appropriate for large-scale power grids. Among the various iterative techniques, the Newton-Raphson and Gauss-Seidel methods are the most used. They differ in terms of memory requirements, implementation complexity, convergence speed, and suitability for different types of problems. Of these, the Newton-Raphson method is especially favored for large-scale analyses due to its computational efficiency, reliable results, and rapid convergence [29].

3.5.1 Bus Classification

A bus is a key element in a power system where various components—such as generators, transmission lines, substation loads, and other electrical equipment—are connected. In the circumstances of power system operation and control, each bus is associated with four main variables: active power (P), reactive power (Q), magnitude of voltage ($|V|$), and power angle (δ). In a load flow analysis, two of these four variables are given, while the other two are determined by solving the non-linear load flow equations until convergence is achieved. Based on which variables are specified, and which are calculated, power system buses are classified into three distinct types.

1. **Swing bus (reference bus):** The Swing Bus (also called the slack bus) serves as the reference point for the entire power system. At this bus, both the voltage magnitude and the voltage angle are specified, typically set to 1.0 per unit and 0 degrees respectively. The purpose of the slack bus is to balance the total active and reactive power in the system. Since power flow calculations cannot perfectly predict losses and system mismatches, the slack bus compensates by adjusting its generated power (both active and reactive) to ensure that the total generation equals the total demand plus losses. There is only one slack bus in any given system, and it is usually a large generator that can handle fluctuations [29].
2. **PV bus (Generator bus):** The PV Bus, also known as the Generator Bus, is where the active power generation and the voltage magnitude are specified. However, the reactive power and voltage angle at this bus are unknown and need to be calculated. This type of bus typically represents generator terminals where the real power output is controlled by the turbine input and the voltage magnitude is regulated

through excitation systems. The generator adjusts its reactive power output to maintain the specified voltage while supplying the required active power [29].

3. **PQ bus (Load Bus):** The PQ Bus, or Load Bus, is the most common type in a power system. At a PQ bus, the active and reactive power demands are known, while the magnitude of voltage and angle are unknown and determined through load flow analysis. These buses represent load centers or substations where electricity is consumed. The primary objective at a PQ bus is to find the bus voltage that satisfies the given power demand using system equations [29].

Together, these three bus types form the basis of load flow analysis, allowing engineers to model and analyze the steady-state operation of electrical power networks. Table 1 below shows the bus classification.

Table 1: Buses Types in Load Flow

SN	Types of Bus	Variables	
		Known variables	Unknown Variables
1	Slack Bus	$ V , \delta$	P, Q
2	PV Bus	$ V , P$	δ, Q
3	PQ Bus	P, Q	P, Q

3.5.2 Power Flow Calculation Methods

In power system network, load flow (or power flow) studies are essential for determining the voltage magnitudes and angles at different buses under steady-state conditions. To solve the nonlinear load flow equations, several numerical methods are used. The most common ones are the Newton-Raphson method, Gauss-Seidel method, and Fast Decoupled Load Flow method.

1. **Gauss Seidel Power Flow:** The Gauss-Seidel load flow method is an iterative technique based on solving nonlinear algebraic equations one at a time, updating the solution after each step. It is relatively simple to implement and is particularly useful for smaller systems. In this method, the voltage at each bus is updated sequentially using the latest available values, which can help accelerate convergence. However, the convergence is generally slow and can become unstable

for large or ill-conditioned systems. Despite this, Gauss-Seidel can be a good starting point for understanding load flow because of its simplicity [29].

2. **Newton-Raphson Power Flow:** The Newton-Raphson method is a more advanced and powerful technique that uses Taylor series expansion to linearize the nonlinear load flow equations around an initial guess. It then solves the resulting linear equations using matrix operations to find corrections to the bus voltages. This process is repeated iteratively until convergence. The Newton-Raphson method is known for its fast and quadratic convergence, especially for large power systems. However, it requires more memory and computational resources, as it involves forming and solving large Jacobian matrices at each iteration. It's widely used in practical applications due to its accuracy and robustness [29].
3. **Fast Decoupled Power Flow:** The Fast Decoupled power flow method is a simplification of the Newton-Raphson method, taking advantage of the fact that in power systems, real power (P) is primarily dependent on voltage angles (δ), and reactive power (Q) is mostly dependent on voltage magnitudes ($|V|$). By decoupling these dependencies, the Jacobian matrix becomes constant or changes very little during iterations, significantly reducing the computational burden. This makes FDLF much faster and more memory-efficient, especially for large systems. While it is less precise than the full Newton-Raphson method, it offers faster convergence and is commonly used in real-time applications where computational speed is essential [29].

In summary, Gauss-Seidel is simple but slow, Newton-Raphson is accurate and robust but computationally heavy, and Fast Decoupled Load Flow strikes a good balance with speed and reasonable accuracy for large-scale systems.

3.5.3 Load Flow Equation:

Load flow analysis is a numerical computational method used in steady-state or static power system conditions to determine the voltage magnitude and load angle at each bus in a power system. In addition to these primary results, it also provides information on real and reactive power flows and transmission losses. The analysis requires input data including generation details, substation bus information,

transmission line parameters, and transformer specifications. The process begins with constructing the bus admittance matrix (Y-bus), which is derived from the transmission line and transformer data [29].

$$P_i = V_i \sum_{k=1}^n Y_{ik} V_k \cos(\theta_{ik} + \delta_k - \delta_i) \quad (1)$$

$$Q_i = -V_i \sum_{k=1}^n Y_{ik} V_k \sin(\theta_{ik} + \delta_k - \delta_i) \quad (2)$$

Where,

$P_i \rightarrow$ Active Power Injected at Bus i

$Q_i \rightarrow$ Reactive Power Injected at Bus i

$V_i \angle \delta_i \rightarrow$ Voltage of Bus i

$Y_{ik} \angle \theta_{ik} \rightarrow$ i, k^{th} element of Y Bus Matrix

3.5.4 Algorithm for NRLF

By using NRLF there are two specific methods for analyzing load flow equations. The first method is by using Polar coordinates and other using rectangular coordinates. In the Load flow studies, the polar coordinates are used to solve the power system network in MALAB. [29]. The Newton-Raphson method can be applied using the following steps:

- Read the system parameters and specified/scheduled variables
 - a. bus admittance matrix (Y_{bus})
 - b. number of buses (n)
 - c. magnitude of specified voltage (V) and load angle (δ) of slack bus.
 - d. injected real and imaginary power (P and Q) at load buses (their values are taken as negative because loads absorb power instead of injecting)

e. injected real power (P), magnitude of voltage (V), Q_{min} and Q_{max} of generator buses.

- Assume suitable V for load buses and δ for both load and generator buses
- Start iteration count, $j=0$
- Take 1st bus as reference bus, so that 2nd to m^{th} buses are generator bus and $(m+1)^{th}$ to n^{th} buses are *load bus*
- Calculate active power (P_{calc}) for generator and load buses and calculate reactive power (Q_{calc}) for only load buses:

$$P_{i(calc)} = V_i \sum_{k=1}^n Y_{ik} V_k \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$Q_{i(calc)} = -V_i \sum_{k=1}^n Y_{ik} V_k \sin(\theta_{ik} + \delta_k - \delta_i)$$

- Calculate error vector:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \vdots \\ \Delta P_n \\ \Delta Q_{m+1} \\ \Delta Q_{m+2} \\ \vdots \\ \Delta Q_n \end{bmatrix} = \begin{bmatrix} P_{2(spec)} - P_{2(calc)} \\ P_{3(spec)} - P_{3(calc)} \\ \vdots \\ P_{n(spec)} - P_{n(calc)} \\ Q_{m+1(spec)} - Q_{m+1(calc)} \\ Q_{m+2(spec)} - Q_{m+2(calc)} \\ \vdots \\ Q_{n(spec)} - Q_{n(calc)} \end{bmatrix}$$

- Calculate Jacobian matrix (J)

$$\begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \cdots & \frac{\partial P_2}{\partial \delta_n} & \frac{\partial P_2}{\partial V_{m+1}} & \frac{\partial P_2}{\partial V_{m+2}} & \cdots & \frac{\partial P_2}{\partial V_n} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \cdots & \frac{\partial P_3}{\partial \delta_n} & \frac{\partial P_3}{\partial V_{m+1}} & \frac{\partial P_3}{\partial V_{m+2}} & \cdots & \frac{\partial P_3}{\partial V_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial P_n}{\partial \delta_2} & \frac{\partial P_n}{\partial \delta_3} & \cdots & \frac{\partial P_n}{\partial \delta_n} & \frac{\partial P_n}{\partial V_{m+1}} & \frac{\partial P_n}{\partial V_{m+2}} & \cdots & \frac{\partial P_n}{\partial V_n} \\ \frac{\partial Q_{m+1}}{\partial \delta_2} & \frac{\partial Q_{m+1}}{\partial \delta_3} & \cdots & \frac{\partial Q_{m+1}}{\partial \delta_n} & \frac{\partial Q_{m+1}}{\partial V_{m+1}} & \frac{\partial Q_{m+1}}{\partial V_{m+2}} & \cdots & \frac{\partial Q_{m+1}}{\partial V_n} \\ \frac{\partial Q_{m+2}}{\partial \delta_2} & \frac{\partial Q_{m+2}}{\partial \delta_3} & \cdots & \frac{\partial Q_{m+2}}{\partial \delta_n} & \frac{\partial Q_{m+2}}{\partial V_{m+1}} & \frac{\partial Q_{m+2}}{\partial V_{m+2}} & \cdots & \frac{\partial Q_{m+2}}{\partial V_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial Q_n}{\partial \delta_2} & \frac{\partial Q_n}{\partial \delta_3} & \cdots & \frac{\partial Q_n}{\partial \delta_n} & \frac{\partial Q_n}{\partial V_{m+1}} & \frac{\partial Q_n}{\partial V_{m+2}} & \cdots & \frac{\partial Q_n}{\partial V_n} \end{bmatrix}$$

Size of Jacobian Matrix: $(2N_l + N_g) \times (2N_l + N_g)$

- Compute correction vector:

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \vdots \\ \Delta\delta_n \\ \Delta V_{m+1} \\ \Delta V_{m+2} \\ \vdots \\ \Delta V_n \end{bmatrix} = J^{-1} \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \vdots \\ \Delta P_n \\ \Delta Q_{m+1} \\ \Delta Q_{m+2} \\ \vdots \\ \Delta Q_n \end{bmatrix}$$

- Compute updated value of δ and V :

$$\delta_i(j) = \delta_i(j-1) + \Delta\delta_i(j)$$

$$V_i(j) = V_i(j-1) + \Delta V_i(j)$$

- Compute Q_i for generator bus:

$$Q_i = -V_i \sum_{k=1}^n Y_{ik} V_k \sin(\theta_{ik} + \delta_k - \delta_i)$$

$$\text{Is } Q_i^{max} \geq Q_i \geq Q_i^{min}?$$

YES: go to step 11

NO: IF minimum limit is violated then $Q_i = Q_i^{min}$ if maximum limit is violated $Q_i = Q_i^{max}$, treated this bus i as new load bus; go to step 11

- Check for convergence:
Is $\Delta P_i(j) \leq \varepsilon$ and $\Delta Q_i(j) \leq \varepsilon$?

YES: go to step 12

NO: $j=j+1$, go to step 4

- Compute P_i and Q_i of the slack bus:

$$P_i = V_i \sum_{k=1}^n Y_{ik} V_k \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$Q_i = -V_i \sum_{k=1}^n Y_{ik} V_k \sin(\theta_{ik} + \delta_k - \delta_i)$$

- Calculate power loss:

$$S_{loss(ik)} = (V_i - V_k) I_{ik}^*$$

$$I_{ik} = (V_i - V_k)y_{ik} + V_i y_{i0}$$

3.6 Contingency Analysis

Contingency in a power system is the possible loss of one or more elements of the system (e.g., transmission lines, generators, transformers) and it represents a hypothetical but credible event that could affect system stability, reliability, or operation [29]. There are two types of contingencies: (i) N-1 Contingency and (ii) N-2 or Higher Contingency. N-1 Contingency assumes the loss of a single component (most common scenario). The power network must be able to bear any single failure without violating operational limits. N-2 or Higher Contingency assumes the simultaneous loss of two or more components. It is Less common but analyzed for critical systems or in planning stages. The purpose of Contingency Analysis are to Ensure system reliability and security, to avoid cascading failures or blackouts and to determine corrective actions (e.g., load shedding, generator redispatch). Steps to perform Contingency analysis are as follows:

1. Identify Contingency Scenarios like Transmission line outages, Generator trips or Transformer failures.
2. Perform Base Case Power Flow i.e. solve the load flow for the current system condition without any failures. This is a reference for comparing contingency cases.
3. Simulate each contingency case. For example, in N-1 analysis, Remove the component (line, generator, etc.) from the system model and compute the power flow for each case.
4. Check the post-contingency results for each case like Voltage violations (outside acceptable limits, e.g., 0.95–1.05 p.u.), line overloads (thermal limits exceeded), Generator overloads etc.
5. Rank Contingencies based on severity or criticality.
6. Determine Corrective Actions (if needed).

After the post-contingency results contingencies are ranked using Performance Index (PI). A simple form of PI is MW ranking Method which is expressed as:

$$PI = \sum_{j=1}^L W_j \left[\frac{P_j}{P_{jmax}} \right]^n$$

Where L is number of Lines, P_j and P_{jmax} are MW flow and MW capacity of line j respectively, W_j is weightage factor and n is a suitable index nearly equal to 1.

CHAPTER 4: METHODOLOGIES

4.1 Proposed System

For this thesis work, the section from INPS System was taken for simulation. From INPS, the Grid Section under Pokhara Grid and Butwal Grid Division were selected, which consists of 31 Buses which include generating stations, substations, transmission Lines, consumer loads and industrial loads. The line data and bus data for this system were collected from GOD under NEA. Also, real-time power flow data can be collected from the Power System Operation Department of NEA for testing the Model Developed at the end. The Proposed INPS System is as shown in Figure 4-1. The brief description of the proposed Grid systems according to NEA Transmission Directorate 2081 are as follows.

Pokhara Grid Division is in Pokhara. The major duty of this division is to supervise, maintain and operate 132 kV and 220kV substations and transmission lines in Gandaki province. This division has ten substations namely at Damauli, Pokhara, Lekhnath, Markhichwok, Syangja, Dana, Kushma, Kritipur, New Modi and Lahachwok. The transmission lines under this division include 132 kV transmission lines Bharatpur - Pokhara - Lekhnath – Damauli - Modi Khola section, Lekhnath - Kaligandaki section, Kushma- Lowermodi section, Kritipur-Udipur section and Middle Marsyangdi- Markichowk - Marsyangdi section and 220 kV line from Dana substation to Kushma substation.

Butwal Grid Division is located at Yogikuti, Butwal. This division supervises, operates and maintains ten substations, namely at Kawasoti, Sunawal, Bardghat, Butwal, New Butwal, Chanauta, Gandak, Motipur, Mainahiya and Sandhikharka. This division is also accountable for regular maintenance and checking Line failure of 132 kV and higher voltage TL in Nawalparasi, Rupandehi, Kapilvastu, Arghakhachi and Palpa districts.

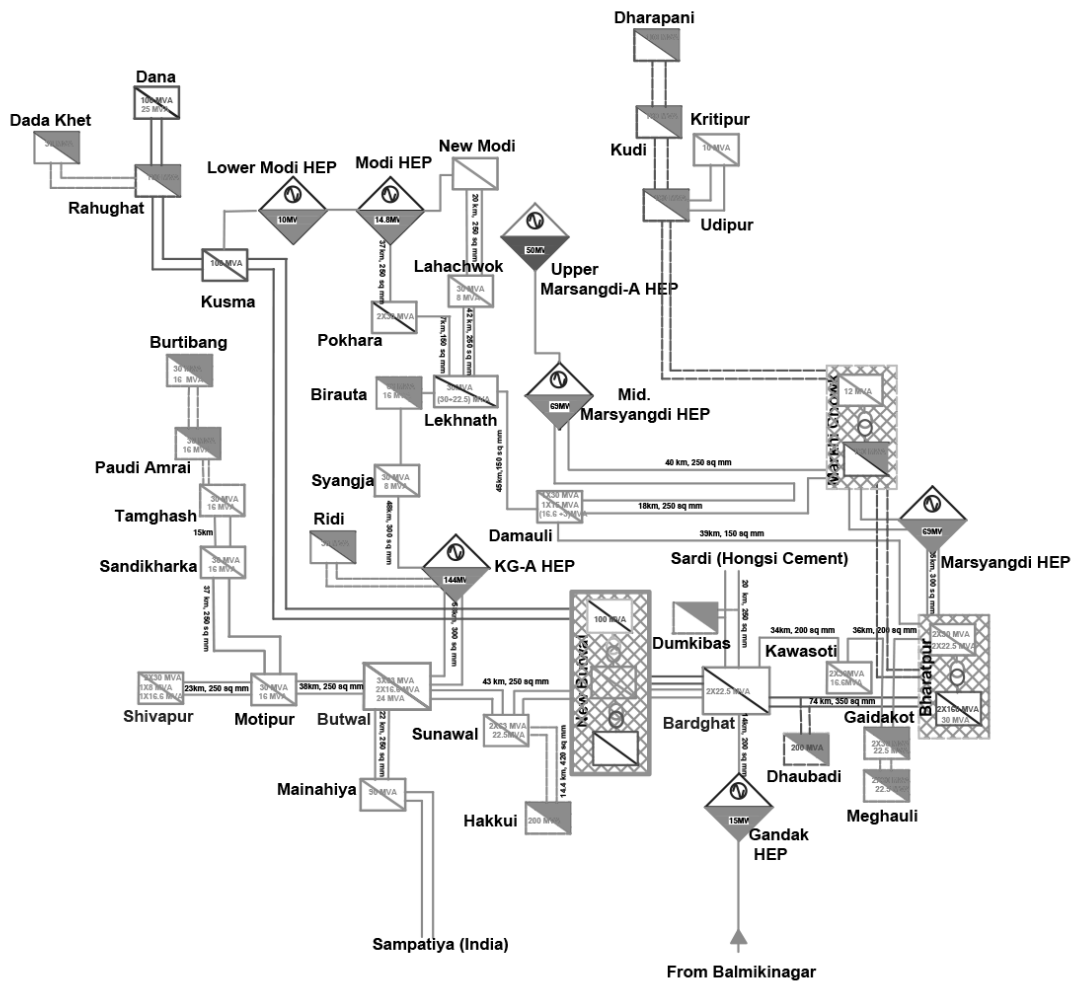


Figure 4-1: Proposed INPS System

4.2 Data Collection

Data collection involves compiling essential data and information for a specific model to assess the intended outcomes. Ensuring data accuracy is crucial for maintaining research validity. In this study, relevant data were obtained from various departments of the NEA. The required data for the study can be outlined as follows.

1. **Generation Data:** Generation Data with its installed Capacity are mentioned in Annex D as per NEA Generation Directorate 2081.
2. **Substation Data:** Substation Data with its Voltage level and Capacity are mentioned in Annex E as per NEA Transmission Directorate 2081.

3. **Bus Data:** Bus Data are taken from GOD, NEA of a time instant 2025/01/26 11:05 AM which are mentioned in Annex C. It consists of Generation and Load Data. It includes Active Power and Reactive Power etc.
4. **Transmission Line Data:** TL parameters are calculated as per NEA Transmission Directorate 2081 which are mentioned in Annex B. It consists of Substation Name, Length, Types of Conductors, Conductor Configuration. The Required Parameters for simulation are Resistance, Line Reactance or Inductance, Shunt Admittance or Capacitance per unit Length. Also Positive, Negative and Zero Sequence components of TL are required for Fault Analysis.

4.3 Model Development using Simulation

4.3.1 Load Flow Analysis

First, an INPS system was modelled in MATLAB/Simulink as per Data mentioned in 4.2. MATLAB/ Simulink Model is mentioned in Annex A. After setting Bus Data and Line Data Load Flow Analysis was performed. For Load Flow analysis *powergui* load flow analyzer tool in MATLAB/ Simulink was used with PQ tolerance $1e-4$, maximum iteration 50. NRLF Method was used for Load Flow of the Systems. Load Flow results are Discussed in 5.1.

4.3.2 Contingency Analysis

After the desired load flow result, an N-1 Contingency analysis was performed. For Contingency analysis *panda power* library was used, which is python library used in power system. After post-contingency PI index was calculated. MW raking method was used in this work for ranking the contingencies. Mathematical formula for MW raking method is mentioned in 3.6. Maximum power flow (P_{max}) is equivalent to thermal capacity of conductor. The report of post contingency analysis is Discussed in 5.2. In this analysis only line outages are taken.

4.3.3 Fault Analysis

From contingency analysis, most sensitive lines were detected. Fault analysis was also performed on the line with highest rank. Faults were injected to study the current voltage characteristics. The sampling time step for the simulation was taken as 50 μ s. Voltage and current data were generated for analysis. Conditions Taken for Fault analysis is as follows.

1. For TL Fault Type identification: Below Mentioned Fault types are taken on the Bus 24 for given period.
 - a. Normal (0 to 0.2 seconds)
 - b. LG (0.2 to 0.4 seconds)
 - c. LL (0.4 to 0.6 seconds)
 - d. LLG (0.6 to 0.8 seconds)
 - e. LLLG (0.8 to 1 seconds)
2. For TL Fault Location identification: LLLG Fault is taken at below Mentioned Buses for given period.
 - a. Normal (0 to 0.2 seconds)
 - b. Bus 2 (0.2 to 0.4 seconds)
 - c. Bus 22 (0.4 to 0.6 seconds)
 - d. Bus 24 (0.6 to 0.8 seconds)
 - e. Bus 28 (0.8 to 1 seconds)
3. For TL Fault Distance identification: LLLG Fault is taken at below Mentioned Distance from Bus 1 to 24 for 0.2 seconds each.
 - a. Normal (0 to 0.2 seconds)
 - b. 0 km (0.2 to 0.4 seconds)
 - c. 39 km (0.4 to 0.6 seconds)
 - d. 77 km (0.6 to 0.8 seconds)
 - e. 116 km (0.8 to 1 seconds)

Current and Voltage waveform were recorded at Swing Bus (Bus 1) for all above mentioned Condition. So, the effect on swing bus was analyzed for different faults condition.

4.4 Steps for Data Generation and Model Development

4.4.1 Dataset Generation

Current and Voltage waveforms at Bus 1 were converted to Dataset using Scope Data Logging. The datasets were generated from the current and voltage waveform for several types of faults and normal conditions analyzed as a XLS file Format. Since the time step was $50\mu\text{s}$ and simulation time was 1 second, there were 20,000 recorded data for each condition. Also, the dataset consists of three phase voltage and three phase current, so the dataset size is about $20,000*6$. A sample of Dataset generated for fault type identification is in Annex H.

The Generated Data sets are required for training and testing the machine learning algorithms. The gathered data sets were split into training and testing in the ratio of 4:1 i.e. the training set was about 80% of the total dataset and remaining for kept for testing. K fold CV is also performed for result analysis.

4.4.2 Hybrid AI Model Development

Three different ML Algorithms were used to train the model. Algorithms mostly used are SVM, DT, RF, KNN, NB and MLP [4], due to their better performance in fault diagnosis. In this work also these Algorithms were chosen according to their better performance. Also, Different algorithm should be chosen with different working principle so that error in prediction by one Algorithm can be predicted right by another algorithm. Hyper Parameter of each ML algorithm were tuned by using PSO module using RMSE as fitness function for ensemble. Only one hyper parameter of each ML Algorithm was taken for tuning keeping the other parameter as default. For Each ML Algorithm significant one Hyper Parameter was taken which will affect more on result. And again, the weights of ML algorithm for the ensemble process were calculated using PSO technique. Ensemble with soft voting scheme is used with bagging technique. Three ML algorithms were used in the ensemble process with PSO optimized weights. After this process, to test the performance of the model results after training process, CV was performed with the dataset. While performing this process, K-fold Cross

Validation, the most effective CV method, is applied with CV value of 5. CV=5 will choose 80% of data for testing and remaining 20% of data for testing purpose.

The generalized ensemble model with soft voting approach according to [30] is shown as follows:

$$\text{Ensemble Probability for class } C (P_c) = \frac{\sum_1^M w_M p_{MC}}{\sum_1^M w_M}$$

$$\text{Final predicted class } (y) = \arg \max(P_c)$$

For Hybrid Model the optimized weight is given by PSO as follows:

$$\text{PSO}(ml_1, ml_2, \dots, ml_M) = \{w_1, w_2, \dots, w_M\}$$

Where,

$C \rightarrow$ the number of classes or labels

$M \rightarrow$ the number of classifiers or weak Learners

$w_M \rightarrow$ the weight for classifier M

$p_{MC} \rightarrow$ the predicted probability of class C by classifier M

$ml_i \rightarrow$ classifier or weak Learner

In the scope of this project, C denotes the class targets/labels which are fault location, fault type or fault distance. PSO computes the optimized set of weights for ensemble to obtain higher accuracy. The workflow is as shown in Figure 4-2. as discussed in this section.

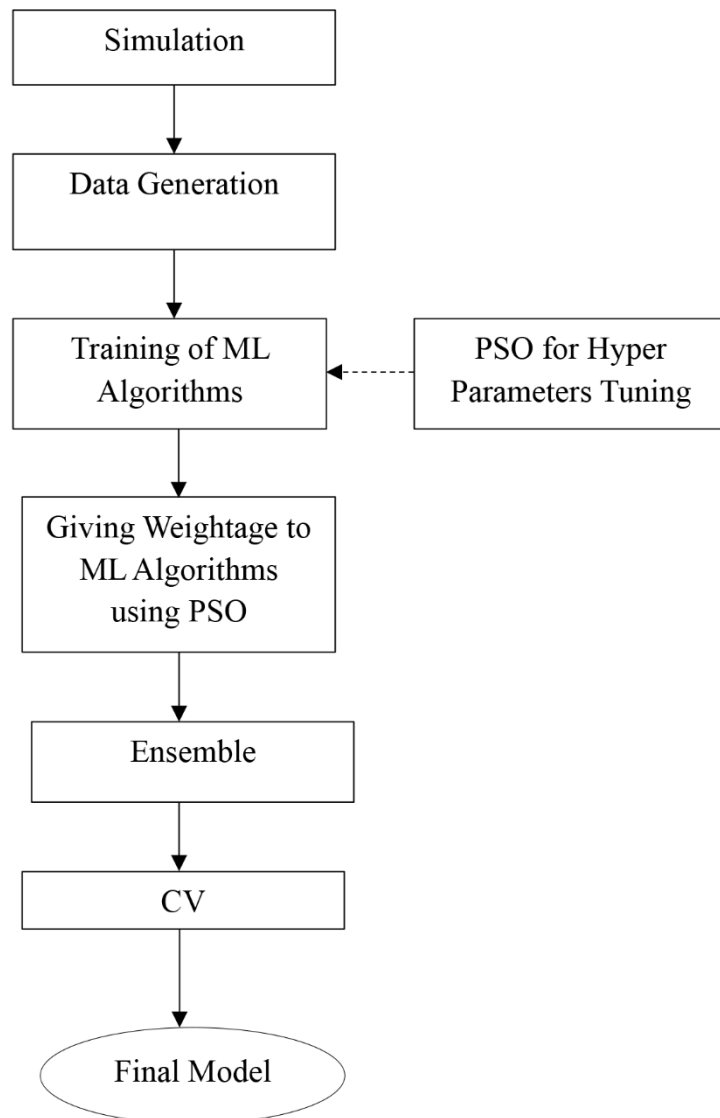


Figure 4-2: Workflow of the Hybrid AI model

4.4.3 Adaptive ML Model development

The above model only works for static power system network, i.e. constant load and fixed network which will not work for real-time dynamic power systems. For the above problem adaptive ML model should be deployed to train the model with streaming data from dynamic power system model. For the adaptive ML Model deployment streaming data or real time data are required for training, i.e. simulation and ML model training shall be done at the same time. This can be done by MALTAB-

Python Co-simulation in which MATLAB Simulink will continuously simulate power system network and in python, training of ML model can be done. For this *matlab.engine* library is used to stream simulation data from MATLAB Simulink file to python environment. Another important library used in Python is *river* library which is used for online ML learning i.e. real time ML learning. Again, the same INPS system is taken for this adaptivity test. The setup for execution of model is as follows:

1. For making network dynamic Active Load at bus 22 and 27 was randomized between 0 to 100 MW.
2. For randomization, sample time was taken for 0.2 seconds, i.e. the load will change randomly at every 0.2 seconds.
3. To make a structural change, the line between bus 5 and 11 was disconnected and connected at every 4 second time interval.
4. LLLG fault was taken at bus 24 after every 2 seconds for duration of 2 seconds.
5. Simulation time step was taken as 0.001sec i.e. at every 0.001sec of simulation a new data is fed to ML Model.
6. In this Hoeffding Tree Classifier ML algorithm was taken. The Hoeffding Tree Classifier (also called Very Fast Decision Tree) is a type of decision tree designed for streaming data.
7. The model was tested and trained at every inclusion of new streamed data. By applying this the developed model becomes adaptive, effective and flexible in case of any physical or parameter changes, which are very frequent and normal characteristics of the power system network.

4.5 TOOLS AND SOFTWARE

Simulink, a MATLAB-based graphical programming environment for modelling, from **MathWorks** were used for simulation and Load flow analysis of the Proposed Power Grid Model. The simulation results were recorded to tabulated data using **Scope Data Logging**. This data was then exported to **Microsoft Excel** and saved in .XLSX File.

For ML Model Development and optimization **Python** programming Language was used. The Modules or Libraries used for the programming are as follows:

1. *NumPy* provides support for large, multi-dimensional arrays and matrices, along with a vast library of high-level mathematical functions designed to efficiently perform operations on these arrays.
2. *Scikit-learn (sklearn)* offers a wide range of classification, regression, and clustering algorithms, such as support vector machines, random forests, gradient boosting, k-means, and DBSCAN. It is built to work seamlessly with the Python ecosystem.
3. *Keras* is an open-source library that offers a user-friendly Python interface for building and training artificial neural networks.
4. *Pandas* is a Python library designed for data manipulation and analysis, providing powerful data structures like Data Frames.
5. *PySwarms* is a flexible and extensible Python toolkit for implementing Particle Swarm Optimization (PSO) algorithms in research and development.
6. *Matplotlib* is a versatile Python library used to create static, animated, and interactive visualizations.
7. *Seaborn* is a Python visualization library built on top of Matplotlib, offering a high-level interface for crafting attractive and informative statistical plots.
8. *Pandapower* is a robust open-source Python library developed on top of Pandas and Matplotlib, tailored for power system modeling, analysis, and simulation.
9. *MATLAB.engine* for Python lets you call MATLAB functions and execute MATLAB scripts from within a Python environment.

4.6 Final Adaptive Hybrid AI Model

As a step Discussed in 4.4 Model was developed using Python Code. The Summary of Code is as follows:

1. **Hyper Parameter Tuning**

- 80% of data were taken for training and the remaining were taken for testing randomly
 - Voltage and Current were taken as Features and Fault Location, Fault Distance and Fault type were as Target/Label for ML Model
 - Scaling of data was done for better performance
 - ML Algorithms Taken:
 - i. RF
 - ii. SVM
 - iii. GB
 - Hyper Parameters chosen for tuning keeping other parameters constant:
 - i. no. of estimators for RF
 - ii. C parameter for SVM
 - iii. Learning Rate for GB
 - Limit taken for Hyper parameters:
 - i. no. of estimators: (50,200)
 - ii. C parameter: (0.1,10)
 - iii. Learning Rate: (.05,2)
 - PSO parameters (Randomly Chosen):
 - i. $c1=0.5$
 - ii. $c2=0.5$
 - iii. $w=0.9$
 - iv. no. of Particles=20
 - v. iteration no.=20
 - Soft Voting Classifier was taken in ensemble
 - Fitness function for PSO was taken as RMSE of ensemble algorithm
 - Objective of PSO technique was to minimize RMSE
 - From This we get the optimum Hyper Parameters
- 2. Calculating Optimized Weights for Soft Voting.**
- Limit of weights for all was taken between 1 to 5
 - Fitness Function was again chosen as RMSE and PSO parameters were set as above steps.

- From this the optimum weights for all ML Algorithms were received.

3. Main Model

- Optimum Hyper Parameters and Weights were taken and ML Model was trained
- Finally, K Fold Cross Validation with CV=5 was performed
- Also, Performance Metrics (Confusion Matrix, Accuracy, Recall and Precision) were analyzed

4. Adaptive ML Model

- MATLAB Simulink model was loaded in python using *matlab.engine* Library.
- For streaming data, training and testing was applied using Hoeffding Tree Classifier Algorithm.
- Scaling of data was done for better performance
- Accuracy metric was used for testing models
- Data were fetched using scope data logging
- The model simulation time was taken for 8 seconds
- The features taken here are three phase current, status of line between buses 5 and 11, load at bus 22 and load at 27 (i.e. there are 5 features)
- Since step time was taken 0.001s final dataset size will 8000*5.
- Here the label or target is faulty condition i.e. for this work 'normal' or 'faulty'.
- Streaming data were Trained in ML models one by one. Also, the model was tested with new data each time i.e., training and testing will be done 8000 times if simulation is performed for 8 seconds. The accuracy metrics was also updated at each testing process.
- Finally, accuracy vs time curve was plotted

CHAPTER 5: RESULT AND DISCUSSION

5.1 Load Flow Result:

First NRLF was Performed for the proposed system with Data mentioned in 4.2. Load Flow Result are mentioned in Annex G which consists of Bus voltage, Power Angle Active Power (P) and Reactive Power (Q) at each Bus. The Load flow Converged in 7 iterations with total PQ generation of 492.39 MW and 114.71 MVAR. The System got total loss of 17.88 MW and 81.18 MVAR. The Voltage at Swing Bus was Kept 1.06 p.u for the following Load Flow Analysis. The Generation Value got for Swing Bus (Kali Gandaki A) is 111.69 MW and 12.14 MVAR after Load Flow Analysis, which is quite close with Real Data 112 MW and 8.7 MVAR at time instant 2025/01/26 11:05 AM. Further the Load Flow Result was applied to the Simulink Model for Contingency and Fault Analysis.

5.2 Post Contingency Analysis report:

As Discussed in 4.3.2 post contingency analysis was performed and report is illustrated as in Table 2. Ranking had been done according to PI index. Also No. of overloaded lines has been shown in the table during the line outages i.e. for line outage between bus 24 and 1 the number of overloaded lines is 4 and PI index is 47.83. During the line outage case between line 24 and 23 the load flow didn't converge due to isolation of some sections from swing bus. According to the PI ranking fault analysis has been performed.

Table 2: Post Contingency Analysis report

Rank	PI (%)	Overloaded lines	From bus	To bus
1	47.83	4	24	1
2	25.82	1	11	5
3	23.05	0	3	4
4	23.02	0	15	14
5	22.84	0	24	25

6	22.77	1	31	26
7	22.72	1	5	6
8	22.66	0	17	15
9	22.48	1	31	27
10	22.38	0	6	10
11	22.28	0	2	3
12	22.15	0	13	12
13	22.12	0	10	4
14	22.07	0	1	2
15	22.03	0	14	13
16	21.92	0	12	3
17	21.90	0	19	28
18	21.90	0	16	19
19	21.90	0	29	28
20	21.86	0	18	3
21	21.62	0	4	6
22	21.42	0	14	18
23	21.12	1	11	30
24	21.06	0	21	20
25	21.06	0	23	21
26	20.82	0	8	10
27	20.62	0	4	11
28	20.36	0	7	8
29	20.18	0	30	31
30	20.05	0	10	9
31	19.66	0	23	22
32	19.56	0	27	24
33	NAN	0	24	23

5.3 Fault Analysis Result:

The Waveforms of the simulation result are mentioned in Annex F. It consists of voltage and current for different fault types, different fault location and different fault distance. The Voltage and current waveform mentioned are of Swing Bus (Kali Gandaki A). The Fault at Bus 2 had seen more impact on Swing Bus than faults at other buses from the given waveform. Similarly, magnitude of fault current had been decreasing order with increasing the distance from the swing Bus. The Above waveform was converted to dataset for further ML Algorithm.

5.4 PSO Optimization Result:

Results obtained by PSO Hyper Parameters Tuning are as follows:

1. number of estimators for RF=150
2. C parameter for SVM=7
3. Learning Rate for GB=0.2

Higher Number of estimators in RF can require higher memory and cause overfitting and lower number of estimators can cause opposite. Thus, the optimum value may have been chosen according to its impact on model. Similarly, C parameter value in SVM may have been chosen in the same sense. And higher Learning Rate in GB can train the model fast but may cause overfitting which also requires optimum value for better results. After the optimum Hyperparameters the weights shall be given to the ML algorithms according to their performance. The weights for different ML Algorithms after optimizing from PSO are:

1. for RF=4
2. for SVM=3
3. for GB=1.5

The study found that the RF classifier outperformed the others, while the GB algorithm demonstrated the least effective performance. So, PSO had Given weightage to them according to their performance i.e. RF had been given more weightage than other Algorithms.

5.5 Model Testing:

Finally, by using the above hyper parameters and weights ensemble machine learning Model with Soft Voting was Formed. Accuracy, Precision, Recall, Confusion Matrix and CV Mean score were analyzed for checking model performance and compared with weak learners (RF, GB and SVM) with their default hyper parameters as follows:

5.5.1 Fault Type Identification

The performance of Hybrid ensemble model was compared with weak learners as shown in Table 3. Here Hybrid model performed better than other ML Models. Also, for hybrid model accuracy with CV was found to be 95.81% and Confusion Matrix is as shown in Figure 5-1: Confusion Matrix for Fault type Identification.

Table 3: Performance Metrics for Fault Type Identification

Performance Metrics	Hybrid Model	RF	GB	SVM
Accuracy (%)	97.68	96.73	91.10	79.46
Recall (%)	97.68	96.65	91.10	79.46
Precision (%)	97.73	96.68	91.58	82.36

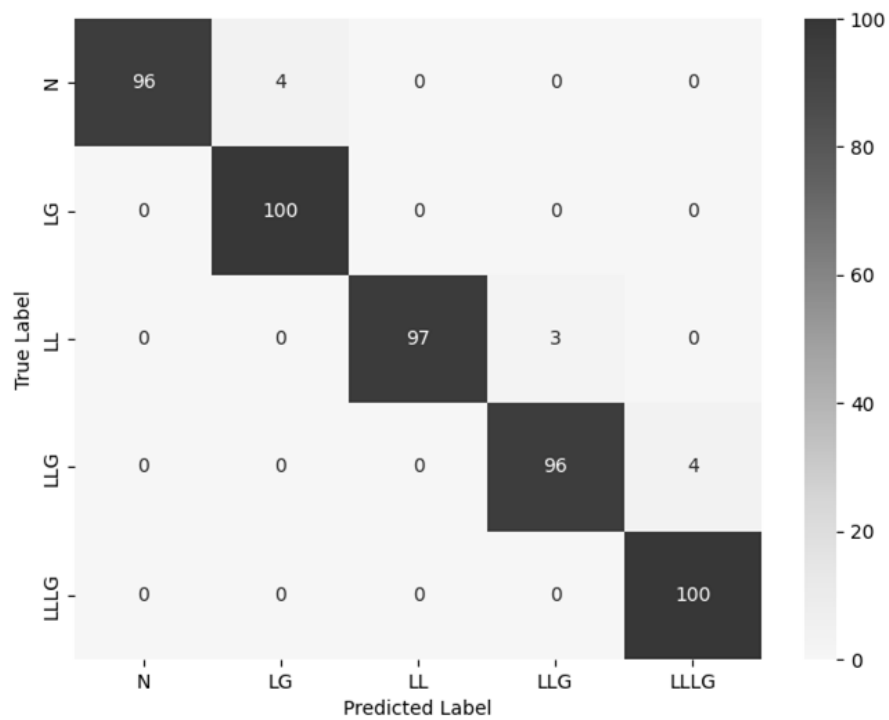


Figure 5-1: Confusion Matrix for Fault type Identification

5.5.2 Fault Location Identification

Similarly, the performance of Hybrid ensemble model was compared with weak learners as shown in Table 4. Also, for hybrid model accuracy with CV was found to be 94.88% and Confusion Matrix is as shown in Figure 5-2.

Table 4: Performance Metrics for Fault Location Identification

Performance Metrics	Hybrid Model	RF	GB	SVM
Accuracy (%)	97.60	96.75	96.00	78.96
Recall (%)	97.60	96.75	96.00	78.96
Precision (%)	97.61	96.76	96.09	85.68

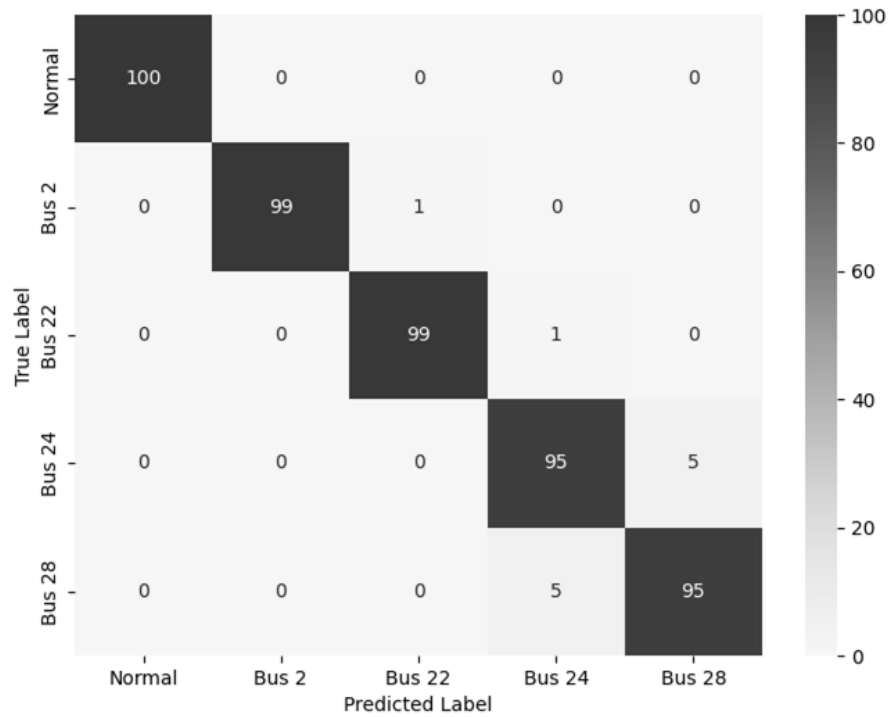


Figure 5-2: Confusion Matrix for Fault Location Identification

5.5.3 Fault Distance Identification

Similarly, the performance of Hybrid ensemble model was compared with weak learners as shown in Table 5: Performance Metrics for Fault Distance Identification. Also, for hybrid model accuracy with CV was found to be 95.24% and Confusion Matrix is as shown in Figure 5-3.

Table 5: Performance Metrics for Fault Distance Identification

Performance Metrics	Hybrid Model	RF	GB	SVM
Accuracy (%)	98.28	96.35	72.31	83.45

Recall (%)	98.28	96.35	91.10	83.45
Precision (%)	98.29	96.37	75.80	83.98

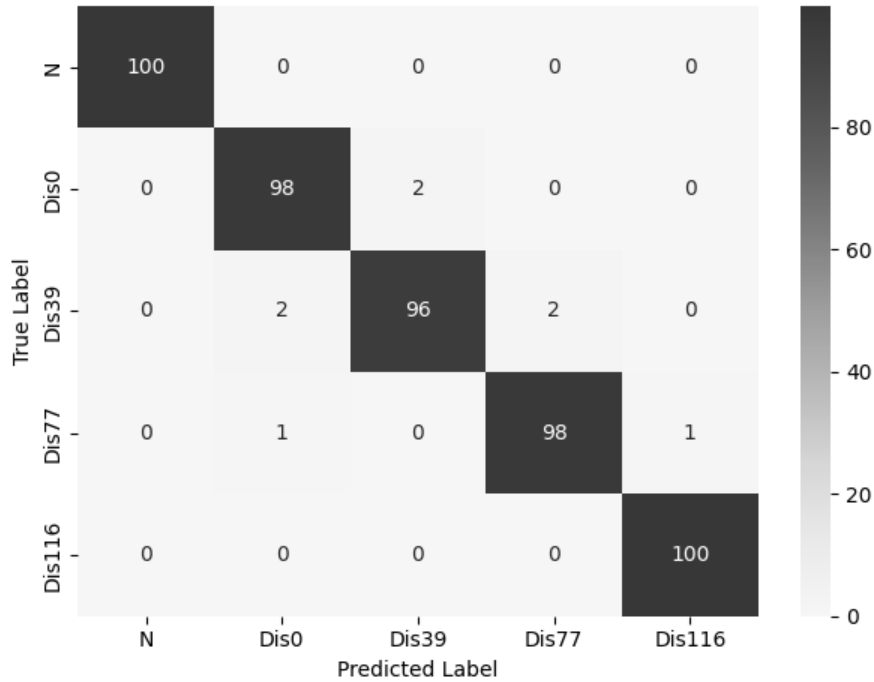


Figure 5-3: Confusion Matrix for Fault Distance Identification

5.5.4 Adaptive Model Testing Result

As Discussed in 4.6 Adaptive ML model was executed for a duration of 8 seconds. For every 1/100 second accuracy of model was recorded. The accuracy vs time curve is plotted as shown in Figure 5-4: Accuracy vs Time Curve. The Final accuracy at the end of 8 second was found 99.5 %.

5.5.5 Discussion on Result

Accuracy, precision and recall were found to be above 97%, which infers that the ML model has performed well. Challenges were aroused during distinguishing fault data in each transition or switching time, impacting the classification accuracy of this ML Model. From the Confusion matrix it was seen that there is slight confusion in identifying fault location between bus 24 and 28. In comparison with other weak learners, the Hybrid AI model has performed very well. Among other ML algorithms,

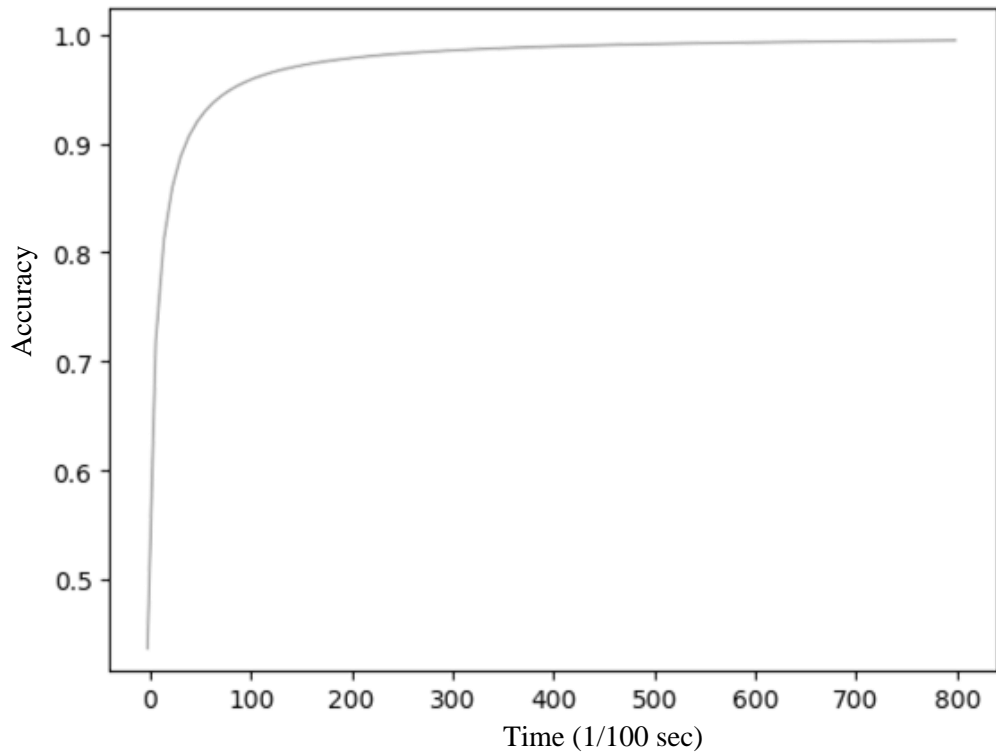


Figure 5-4: Accuracy vs Time Curve

RF performed close to the hybrid model with good accuracy since it is also one type of ensemble model. The training time and testing time of this model was 21.3 seconds and 0.6 seconds respectively. Since dataset size was 20,000*6 and training data was taken about 80%, 16,000 data were trained in 21.3 seconds. Similarly, 4,000 data were tested or predicted in 0.6 seconds with average accuracy of about 97%.

For adaptive ML model, final accuracy was found to be 99.5%. The simulation time was set for 8 seconds and the total time taken to execute this model was 67 seconds. Since at every time step of 0.001 second training and testing of model was done total running time of adaptive model took more than a minute.

CHAPTER 6: CONCLUSION

This paper presented the study of Adaptive Hybrid AI model consisting of ML algorithms and Optimization technique for classifying power system faults that overcome the limitations of traditional machine learning approaches. A brief theoretical overview of the different ML algorithms and optimization technique have been introduced. The use of raw voltage and current data enhances the model's practicality for realistic applications, while its validation across various system topologies confirms its adaptability. Voltage and current signals were obtained from the INPS sub system and analyzed under both fault and no-fault conditions. Datasets were developed for the analysis of Fault identification. These datasets were used to train and evaluate the classifiers, with performance metrics such as CV-Mean score, Accuracy, Recall, Precision and Confusion Matrix. Performances metrices were found to be greater than 97% which is rather good accuracy for detecting fault location. According to IEEE 1547 standards, it allows 2 seconds of delay [22] to identify any types of faults; in this work it took as low as 0.1 seconds for detecting fault. This means the fault detection time was approximately 5% of the IEEE 1547 standard. It is significantly faster and more efficient for power system studies. The results finally highlight the model's strength for real-time fault detection in dynamic environments, offering a budget-friendly and efficient solution for power system monitoring and maintenance.

CHAPTER 7: FUTURE WORKS

Future studies can incorporate deeper higher-level models, which, being more computationally expensive and requiring larger datasets, can possibly result in higher classification accuracy. Other methods like Deep Learning, Reinforcement learning and transfer learning can be deployed in this type of work. Machine learning models designed in literature are trained from simulated data. ML can also be trained using real data for better real-world applications. In this work, PSO is not applied in adaptive model due to speed limitation. Computational speed can be increased using GPUs or Higher speed CPUs. Finally, in this work all ML models are deployed as classification models which can also be used as regression model.

CHAPTER 8: REFERENCES

- [1] S. V, "Comparative Analysis of Machine Learning Algorithms for the Classification of Power System Faults," in *Coimbatore*, Santhosh V, 2019.
- [2] L. Yavuz, A. Soran, A. Onen, X. Li and S. M. Muyeen, " Adaptive Fault Detection Scheme Using an Optimized Self-healing Ensemble Machine Learning Algorithm," *CSEE JOURNAL OF POWER AND ENERGY SYSTEMS*, vol. 8, no. 4, pp. 1145-1156, 2022.
- [3] G. M. Njoka, L. Mogaka and A. Wangai, "Impact of variable renewable energy sources on the power system frequency stability and system inertia," *Energy Reports*, vol. 12, pp. 4983-4997, 2024.
- [4] M. M. ZABEN, MUHAMMEDY.WORKU and M. HASSAN, "Machine Learning Methods for Fault Diagnosis in AC Microgrids: A Systematic Review," *IEEE Access*, vol. 12, pp. 20260-20298, 2024.
- [5] D. Zheng, C. Qin and a. Liu, "Adaptive Particle Swarm Optimization Algorithm Ensemble Model Applied to Classification of Unbalanced Data," *Hindawi*, vol. 2021, no. 7589756, p. 13, 2021.
- [6] L. Che, M. E. Khodayar and M. Shahidehpour, "Adaptive protection system for microgrids," *IEEE Electrification Magazine*, vol. 2, no. 1, p. 66–80, 2014.
- [7] D. S. Pillai and N. Rajasekar, "A comprehensive review on protection challenges and fault diagnosis in PV systems," *Renewable and Sustainable Energy Reviews*, vol. 91, p. 18–40, 2018.

- [8] D. S. Pillai and N. Rajasekar, "Metaheuristic algorithms for PV parameter identification: A comprehensive review with an application to threshold setting for fault detection in PV systems," *Renewable and Sustainable Energy Reviews*, vol. 82, p. 3503–3525, 2018.
- [9] S.-H. Kim, G.-W. Choi and J.-F. Moon, "A Study on the Fault Analysis of the LVDC Using PSCAD/EMTDC," *Korean Institute of Electrical Engineers P*, vol. 65, no. 3, pp. 219-223, 2016.
- [10] J. Keller and B. Kroposki, "Understanding fault characteristics of inverter-based distributed energy resources," 2010.
- [11] H. G. Jiang, J. Zhang, W. Z. Gao and Z. P. Wu, "Fault detection identification, and location in smart grid based on data-driven computational methods," *IEEE Transactions on Smart Grid*, vol. 5, no. 6, p. 2947–2956, 2014.
- [12] J. E. Tate and T. J. Overbye, "Line outage detection using phasor angle measurements," *IEEE Transactions on Power Systems*, vol. 23, no. 4, p. 1644–1652, 2008.
- [13] H. M. Sharaf, H. H. Zeineldin and E. El-Saadany, "Protection coordination for microgrids with grid-connected and islanded capabilities using communication assisted dual setting directional overcurrent relays," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 143-151, 2018.
- [14] S. Teimourzadeh, F. Aminifar, M. Davarpanah and M. Shahidehpour, "Adaptive protection for preserving microgrid security," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 592-600, 2019.
- [15] M. Singh and P. Basak, "Adaptive protection methodology in microgrid for fault location and nature detection using q_0 components of

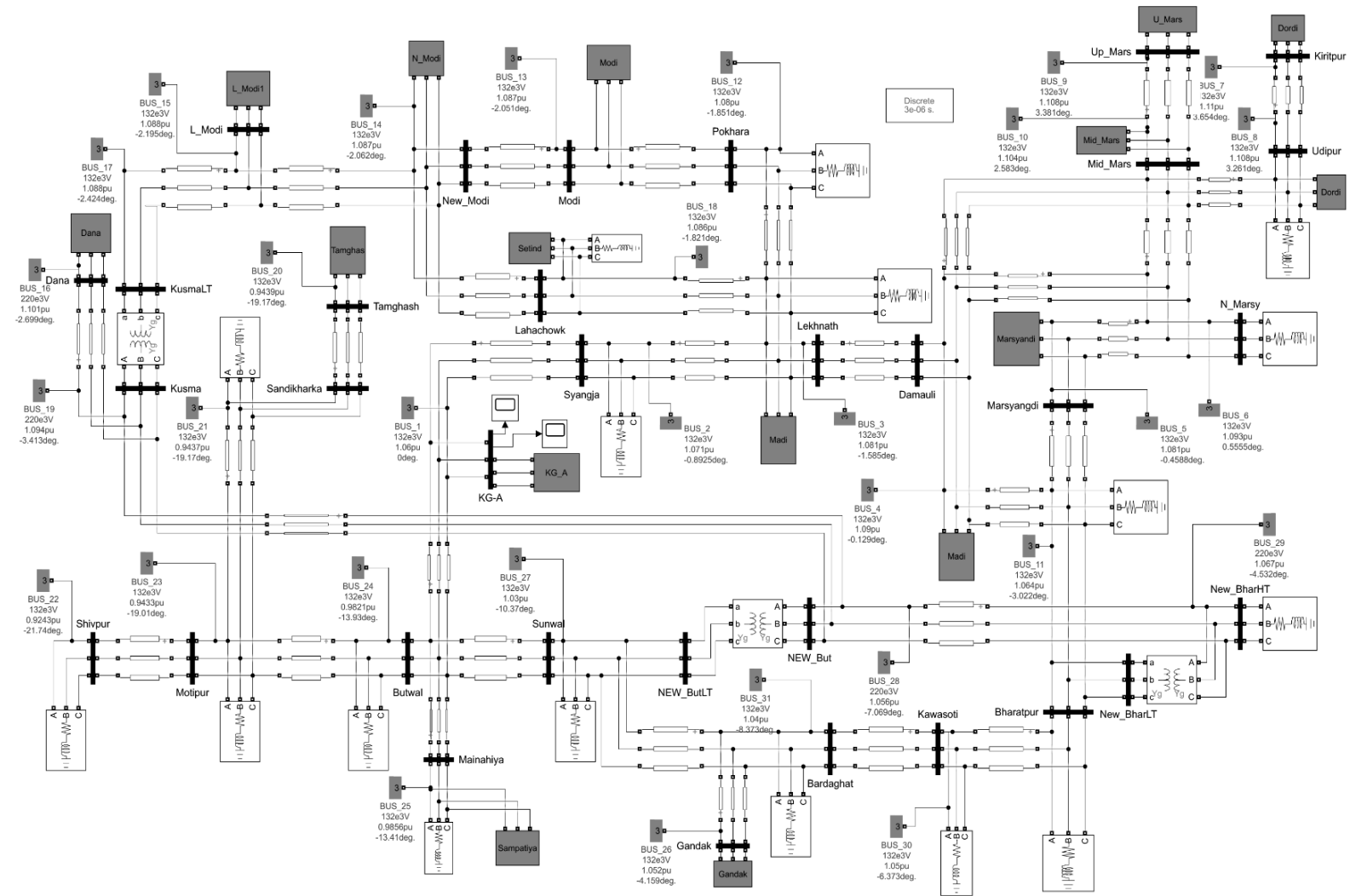
fault current," *IET Gener., Transmiss. Distrib.*, vol. 13, no. 6, pp. 760-769, 2019.

- [16] A. A. Aburommam and M. B. I. Reaz, "A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems," *Information Sciences*, vol. 414, pp. 225-246, 2017.
- [17] S. R. Mohanty, P. K. Ray, N. Kishor and B. K. Panigrahi, "Classification of disturbances in hybrid DG system using modular PNN and SVM," *International Journal of Electrical Power & Energy Systems*, vol. 44, no. 1, p. 764–777, 2013.
- [18] Z. L. Gaing, "Wavelet-based neural network for power disturbance recognition and classification," *IEEE Transactions on Power Delivery*, vol. 19, no. 4, p. 1560–1568, 2004.
- [19] M. S. ElNozahy, E. F. El-Saadany and M. M. A. Salama, "A robust wavelet-ANN based technique for islanding detection," *IEEE Power and Energy Society General Meeting*, pp. 1-8, 2011.
- [20] L. S. Moulin, A. P. A. D. Silva, M. A. El-Sharkawi and R. J. Marks, "Support vector machines for transient stability analysis of large-scale power systems," *IEEE Transactions on Power Systems*, vol. 19, no. 2, p. 815–825, 2004.
- [21] D. Zhang, X. Han and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE Journal of Power and Energy Systems*, vol. 4, no. 3, p. 362–370, 2018.
- [22] "IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems," *IEEE Standard*, vol. 2003, p. 1547–2003.

- [23] P. Sharma, S. Banerjee, S. Tandel, R. Aguiar, R. Amorim and D. Pinheiro, "Enhancing network management frameworks with SDN-like control," *IEEE International Symposium on Integrated Network Management*, p. 688–691, 2013.
- [24] Y. G. Paithankar and S. Bhide, *Fundamentals of Power System Protection*, PHI Learning Pvt. Ltd., 2022.
- [25] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
- [26] P. Romano, M. Casimiro and G. A. M. Garlan, "Self-adaptive Machine Learning Systems: Research Challenges and Opportunities," *Software Architecture*, pp. 133-155, 2022.
- [27] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5-32, 2001.
- [28] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009.
- [29] J. J. Grainger and W. D. Stevenson, *Power System Analysis*, New York: McGraw-Hill, 1994.
- [30] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press, 2012.
- [31] L. L. Grigsby, *Power System Stability and Control*, 2017.
- [32] A. Esmailian and M. Kezunovic, "Prevention of power grid blackouts using intentional islanding scheme," *IEEE Transactions on Industry Applications*, vol. 53, no. 1, p. 622–629, 2017.

- [33] S. Raza, H. Mokhlis, H. Arof, J. A. Laghari and L. Wang, "Application of signal processing techniques for islanding detection of distributed generation in distribution network: A review," *Energy Conversion and Management*, vol. 96, p. 613–624, 2015.
- [34] S. Kar, S. R. Samantaray and M. D. Zadeh, "Data-mining model based intelligent differential microgrid protection scheme," *IEEE Systems Journal*, vol. 11, no. 2, p. 1161–1169, 2017.
- [35] J. P. Singh and N. Bouguila, "Proportional data clustering using K means algorithm: A comparison of different distances," *IEEE International Conference on Industrial Technology*, pp. 1048-1052, 2017.
- [36] H. Shayeghi, B. Sobhani, E. Shahryari and A. Akbarimajd, "Optimal neuro-fuzzy based islanding detection method for Distributed Generation," *Neurocomputing*, vol. 177, p. 478–488, 2016.
- [37] C. M. Bishop and N. M. Nasrabadi, "Pattern Recognition and Machine Learning," *USA: Springer*, vol. 4, 2006.
- [38] NEA, "Transmission/Project Management Directorate," NEA, Kathmandu, 2024.

Appendix A: Simulation Model of INPS System



Appendix B: INPS TL Data

Line No.	Bus (From)	Bus (To)	Voltage (kV)	Length (km)	Conductor	Circuit type	I_ka
1	19	28	220	176	ACCC Drake	Double	0.901
2	16	19	220	79.6	ACCC Drake	Double	0.901
3	29	28	220	148	BISON	Double	0.591
4	17	15	132	6.2	Bear	Single	0.591
5	15	14	132	5	Bear	Single	0.591
6	14	13	132	1	Bear	Single	0.591
7	13	12	132	37	Bear	Single	0.591
8	8	10	132	22	Bear	Single	0.591
9	31	27	132	43	Bear	Double	0.591
10	27	24	132	43	Bear	Double	0.591
11	24	23	132	38	Bear	Double	0.591
12	23	22	132	23	Bear	Double	0.591
13	23	21	132	37	Bear	Double	0.591
14	21	20	132	15	Bear	Double	0.591
15	24	25	132	22	Bear	Double	0.591
16	5	6	132	40	Bear	Double	0.591
17	10	9	132	22	Bear	Single	0.591
18	10	4	132	46	Bear	Double	0.591
19	4	6	132	18	Bear	Double	0.591
20	14	18	132	20	Bear	Double	0.591
21	18	3	132	42	Bear	Double	0.591
22	6	10	132	40	Cardinal	Double	0.9
23	7	8	132	20	Cardinal	Double	0.9
24	24	1	132	116	Duck	Double	0.774
25	1	2	132	48	Duck	Double	0.774
26	2	3	132	48	Duck	Double	0.774
27	11	5	132	25	Duck	Double	0.774
28	11	30	132	28	Panther	Single	0.421
29	30	31	132	42	Panther	Single	0.421
30	31	26	132	14	Panther	Double	0.421
31	12	3	132	7	Wolf	Single	0.355
32	3	4	132	45	Wolf	Single	0.355
33	4	11	132	39	Wolf	Single	0.355

Line No.	R	X	B	R0	X0	B0	L	C	L0	C0
1	0.0749	0.3993	2.915E-06	0.2247	1.4774	1.81E-06	1.27E-03	9.278E-09	4.70E-03	5.753E-09
2	0.0749	0.3993	2.915E-06	0.2247	1.4774	1.81E-06	1.27E-03	9.278E-09	4.70E-03	5.753E-09
3	0.0697	0.3978	2.904E-06	0.2091	1.4719	1.8E-06	1.27E-03	9.244E-09	4.69E-03	5.731E-09
4	0.1087	0.3973	2.940E-06	0.3262	1.4699	1.82E-06	1.26E-03	9.358E-09	4.68E-03	5.802E-09
5	0.1087	0.3973	2.940E-06	0.3262	1.4699	1.82E-06	1.26E-03	9.358E-09	4.68E-03	5.802E-09
6	0.1080	0.3973	2.940E-06	0.3241	1.4699	1.82E-06	1.26E-03	9.358E-09	4.68E-03	5.802E-09
7	0.1087	0.3974	2.941E-06	0.3262	1.4705	1.82E-06	1.27E-03	9.362E-09	4.68E-03	5.804E-09
8	0.1087	0.3974	2.941E-06	0.3262	1.4705	1.82E-06	1.27E-03	9.362E-09	4.68E-03	5.804E-09
9	0.1086	0.3973	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
10	0.1086	0.3973	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
11	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
12	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
13	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
14	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
15	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
16	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
17	0.1087	0.3974	2.941E-06	0.3262	1.4705	1.82E-06	1.27E-03	9.362E-09	4.68E-03	5.804E-09
18	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
19	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
20	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
21	0.1086	0.3974	2.940E-06	0.3259	1.4702	1.82E-06	1.26E-03	9.360E-09	4.68E-03	5.803E-09
22	0.0610	0.2962	2.192E-06	0.1830	1.0960	1.36E-06	9.43E-04	6.977E-09	3.49E-03	4.326E-09
23	0.0610	0.2962	2.192E-06	0.1830	1.0960	1.36E-06	9.43E-04	6.977E-09	3.49E-03	4.326E-09
24	0.0949	0.4052	2.998E-06	0.2848	1.4991	1.86E-06	1.29E-03	9.543E-09	4.77E-03	5.917E-09
25	0.0949	0.4052	2.998E-06	0.2848	1.4991	1.86E-06	1.29E-03	9.543E-09	4.77E-03	5.917E-09
26	0.0949	0.4052	2.998E-06	0.2848	1.4991	1.86E-06	1.29E-03	9.543E-09	4.77E-03	5.917E-09
27	0.0949	0.4052	2.998E-06	0.2848	1.4991	1.86E-06	1.29E-03	9.543E-09	4.77E-03	5.917E-09
28	0.1356	0.3861	2.857E-06	0.4068	1.6222	1.77E-06	1.23E-03	9.095E-09	5.16E-03	5.639E-09
29	0.1356	0.3861	2.857E-06	0.4067	1.6222	1.77E-06	1.23E-03	9.095E-09	5.16E-03	5.639E-09
30	0.1355	0.3861	2.857E-06	0.4066	1.6222	1.77E-06	1.23E-03	9.095E-09	5.16E-03	5.639E-09
31	0.1812	0.4220	3.122E-06	0.5436	1.5612	1.94E-06	1.34E-03	9.939E-09	4.97E-03	6.162E-09
32	0.1813	0.4220	3.123E-06	0.5439	1.5614	1.94E-06	1.34E-03	9.940E-09	4.97E-03	6.163E-09
33	0.1813	0.4220	3.123E-06	0.5439	1.5615	1.94E-06	1.34E-03	9.941E-09	4.97E-03	6.163E-09

Where,

I_{ka}: Thermal Capacity of Conductor in (KA)

R: Positive/Negative Sequence Resistance per unit length (Ω/km)

X: Positive/Negative Sequence Reactance per unit length (Ω/km)

B: Positive/Negative Sequence Shunt Admittance per unit length (mho/km)

L: Positive/Negative Sequence Inductance per unit length (H/km)

C: Positive/Negative Sequence Parallel Capacitance per unit length (F/km)

R0: Zero Sequence Resistance per unit length (Ω/km)

X0: Zero Sequence Reactance per unit length (Ω/km)

B0: Zero Sequence Shunt Admittance per unit length (mho/km)

L: Zero Sequence Inductance per unit length (H/km)

C: Zero Sequence Parallel Capacitance per unit length (F/km)

Substation Name and Bus Number:

SN	Substation Name
1.	Kaligandaki-A
2.	Syangja
3.	Lekhnath
4.	Damauli
5.	Marsyandi
6.	New-Marsyangdi
7.	Kirtipur
8.	Udipur
9.	Upper Marsyandi
10.	Mid Marsyandi
11.	Bharatpur
12.	Pokhara
13.	Modi
14.	New Modi
15.	Low Modi
16.	Dana
17.	Kushma
18.	Lahachowk
19.	Kushma
20.	Tamghash
21.	Sandikharka
22.	Shivpur
23.	Motipur
24.	Butwal
25.	Mainahiya
26.	Gandak
27.	Butwal
28.	New Butwal
29.	New Bharatpur
30.	Kawasoti
31.	Bardghat

Appendix C: Bus Data

Time: 2025/01/26 11:05 AM

SS	SS	P	Q
KIRTIPUR	SDORDI	13.1	2.73
	UDORDI	8.6	-0.86
	DORDI	3.3	0.39
UDIPUR	DORDI	6.9	0.00
	LOAD	-2.5	-0.30
MID_MARS	NYADI	8.7	1.03
	MID_MARS	46.6	0.00
UP_MARS	UP_MARS	34.2	1.53
MARSYA	MARSYA	66.7	4.22
	LOAD	-43.3	-5.15
N_MARS	LOAD	-4.2	-0.50
DAMAULI	LOAD	2.3	0.27
BHARATPUR	LOAD	-37.7	-3.78
NEW_BHAR	LOAD	-4.7	-0.47
KAWASOTI	LOAD	-15.1	-1.52
BARDAGHAT	LOAD	-32	-3.21
GANDAK	GANDAK	9	0.90
	RAMNAGAR	52.1	5.23
SUNWAL	LOAD	-59.8	-6.00
BUTWAL	LOAD	-81.3	-8.16
MAINYA	LOAD	-45.5	-4.57
	SAMPATIYA	63.7	6.39
SHIVPUR	LOAD	-81.8	-9.73
MOTIPUR	LOAD	-12.1	-1.44
SANDIKHARKA	LOAD	-2.9	-0.34
TAMGHAS	LOAD	0.2	0.02
KGA	KGA	112	8.70
SYANGJA	LOAD	-2.8	-0.22
LEKHNATH	SMADI	3.2	-0.20
	UMADI	10.8	1.19
	LOAD	-17.4	-1.91
POKHARA	LOAD	-27.4	-3.02
LAHCHOWK	LOAD	-1.2	-0.13
	SETIND	7.7	0.85
N_MOD	N_MOD	4.3	0.73
MODI	MODI	4.9	-0.83
L_MODI	L_MODI	6.6	1.50
	L_MODI1	3.3	0.75
DANA	NILGIRI1	4.1	0.93
	NILGIRI2	9.3	0.00
	MISTRI	8	-2.27
	GHAR	3.1	3.12
KUSMA	LOAD	0	0.00

Appendix D: Generation Data

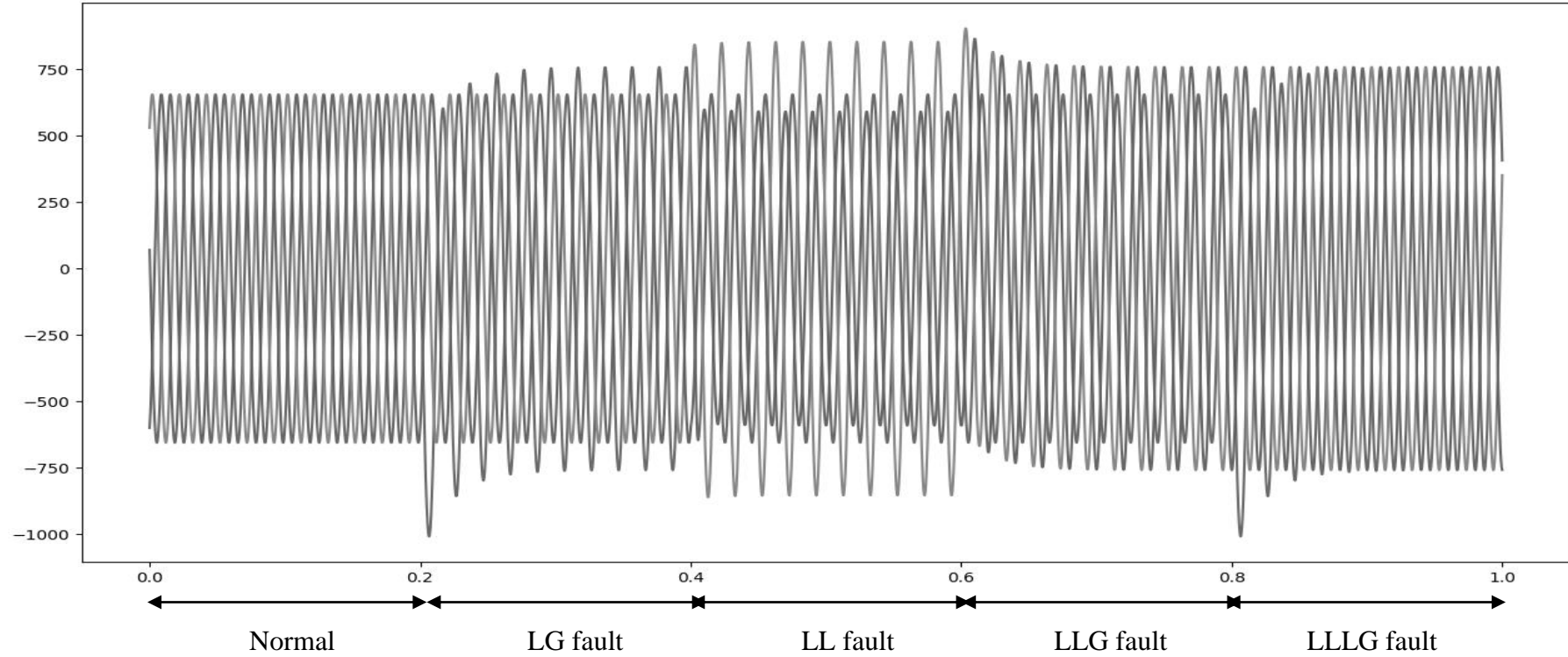
S.N.	Hydropower Unit	Installed Capacity (MW)
1.	Kaligandaki A	144
2.	Dordi Khola	27
3.	Dori 1	12
4.	Upper Dordi	25
5.	Nyadi	30
6.	Mid Marsyandi	70
7.	Marsyangdi	69
8.	Upper Marsyandi	50
9.	Gandak	15
10.	Upper Madi	25
11.	Super Madi	44
12.	Seti	1.5
13.	Modi	14.8
14.	Lower Modi 1	10
15.	Lower Modi 2	20
16.	Nilgiri Khola-1	38
17.	Nilgiri Khola-2	62
18.	Ghar Khola	14
19.	Mistri Khola	42

Appendix E: Substation Data

S.N.	Substation	Voltage (KV)	Capacity (MVA)
1.	Pokhara	132/11	60
2.	New Bharatpur	132/11	22.5
3.	Gandak	132/33	60
4.	Bharatpur	132/11	67.5
5.	Lekhnath	132/33	30
6.	Lekhnath	132/11	22.5
7.	Kawasoti	132/33	60
8.	Kawasoti	33/11	16.6
9.	Syangja	33/11	16.6
10.	Motipur	132/33	30
11.	Motipur	33/11	16.6
12.	Sandhikharka	132/33	30
13.	Damauli	33/11	16.6
14.	Butwal	33/11	57.2

Appendix F: Simulation Result

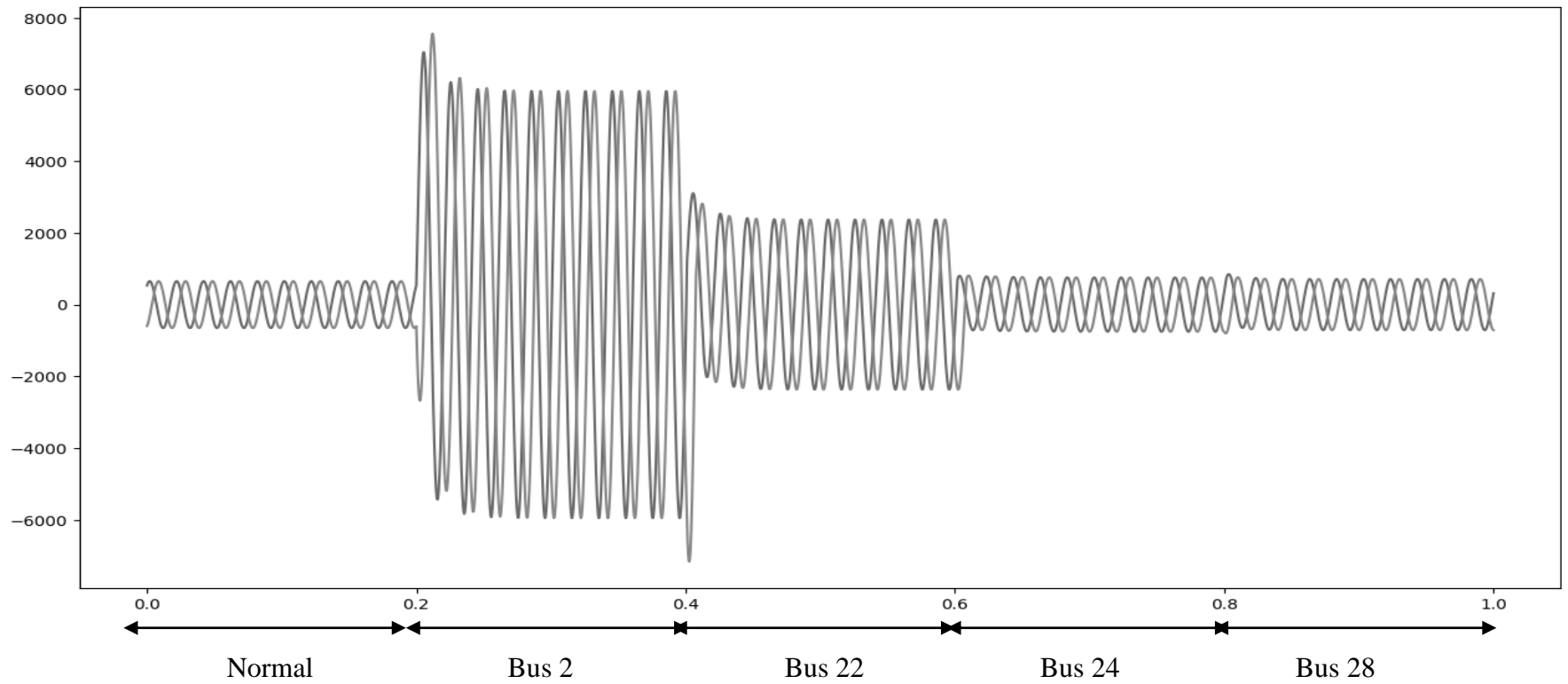
Current waveform for Different Fault Type:



X-axis: time(s)

Y-axis: Current at Bus 1 (A)

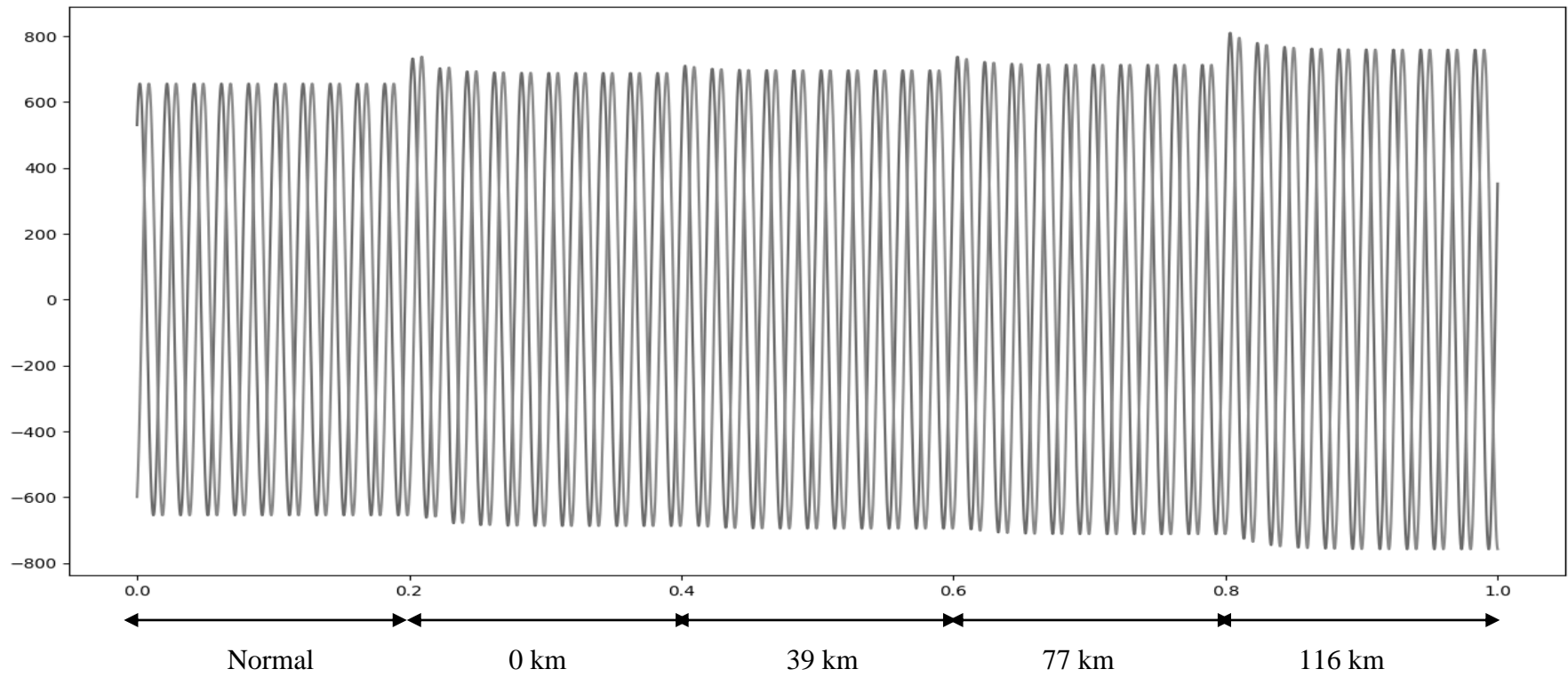
Current waveform for Different Fault Location:



X-axis: time(s)

Y-axis: Current at Bus 1 (A)

Current waveform for Different Fault Distance:



X-axis: time(s)

Y-axis: Current at Bus 1 (A)

Appendix G: Load Flow Result

1 : BUS_1 V= 1.060 pu			17 : BUS_24 V= 0.996 pu/132kV -14.28 deg		
	P(MW)	Q(Mvar)		P(MW)	Q(Mvar)
Generation	111.6935	12.13857	Generation	0	0
PQ Load	0	0	PQ Load	81.29951	6.159805
Z shunt	0.03915	-0.00041	Z shunt	0.069072	-0.00124
BUS_2	11.88552	-1.83671	BUS_1	-94.061	10.38812
BUS_24	99.76878	13.97568	BUS_23	100.2167	12.78559
			BUS_25	-17.9697	-34.6614
2 : BUS_10 V= 1.073 pu/132kV 3.35 deg			BUS_27		
	P(MW)	Q(Mvar)		P(MW)	Q(Mvar)
Generation	55.30003	4.01E-05	18 : BUS_25 V= 1.016 pu/132kV -14.04 deg		
PQ Load	0	0		P(MW)	Q(Mvar)
Z shunt	0.080226	-0.00182	Generation	63.69877	40.00206
BUS_4	54.39203	-0.00852	PQ Load	45.5	4.57
BUS_6	64.06889	2.442052	Z shunt	0.017956	-0.00056
BUS_8	-29.1996	-1.40617	BUS_24	18.18081	35.43262
BUS_9	-34.0415	-1.02551			
			19 : BUS_26 V= 1.031 pu/132kV -3.82 deg		
3 : BUS_11 V= 1.036 pu/132kV -2.61 deg				P(MW)	Q(Mvar)
	P(MW)	Q(Mvar)	Generation	61.1	6.13
Generation	0	0	PQ Load	0.000192	-1E-05
PQ Load	81.00001	8.929998	Z shunt	0.017763	-0.00364
Z shunt	0.323967	0.265381	BUS_31	61.08204	6.133655
BUS_29	46.2821	5.88205			
BUS_30	24.81495	3.502153	20 : BUS_27 V= 1.012 pu/132kV -10.31 deg		
BUS_4	-61.8878	1.592421		P(MW)	Q(Mvar)
BUS_5	-90.5332	-20.172	Generation	0	0
			PQ Load	59.79981	3.999992
4 : BUS_12 V= 1.057 pu/132kV -1.50 deg			Z shunt	0.240299	0.245033
	P(MW)	Q(Mvar)	BUS_24	70.87152	-0.5171
Generation	0	0	BUS_28	-93.1237	-6.8006
PQ Load	27.4	2.02	BUS_31	-37.7879	3.072673
Z shunt	0.038749	-0.00238			
BUS_13	1.640305	-3.59966	21 : BUS_28 V= 1.020 pu/220kV -6.87 deg		
BUS_3	-29.0791	1.582045		P(MW)	Q(Mvar)
			Generation	0	0
5 : BUS_13 V= 1.060 pu/132kV -1.62 deg			PQ Load	0.00021	-8.7E-06
	P(MW)	Q(Mvar)	Z shunt	0.308626	0.249156
Generation	4.9	0.83	BUS_19	-52.7143	-10.8369
PQ Load	3.81E-08	-3.8E-07	BUS_27	93.29374	12.48043
Z shunt	0.032821	-0.00952	BUS_29	-40.8883	-1.89272
BUS_12	-1.63707	3.611465			
BUS_14	6.504253	-2.77194	22 : BUS_29 V= 1.032 pu/220kV -4.18 deg		

			P(MW)	Q(Mvar)	
6 : BUS_14 V= 1.060 pu/132kV -1.62 deg			Generation	0	0
	P(MW)	Q(Mvar)	PQ Load	4.69999	0.470001
Generation	4.3	10.17	Z shunt	0.31759	0.265874
PQ Load	0	0	BUS_11	-46.2492	-4.5883
Z shunt	0.052017	-0.01249	BUS_28	41.23164	3.852422
BUS_13	-6.50398	2.772959			
BUS_15	19.87128	5.405917	23 : BUS_3 V= 1.059 pu/132kV -1.24 deg		
BUS_18	-9.11932	2.003609	P(MW)	Q(Mvar)	
			Generation	14.00001	10.00002
7 : BUS_15 V= 1.059 pu/132kV -1.73 deg			PQ Load	17.4	0
	P(MW)	Q(Mvar)	Z shunt	0.077968	-0.00295
Generation	9.9	2.26	BUS_12	29.13427	-1.45348
PQ Load	5.04E-07	-1.7E-07	BUS_18	2.670592	-1.83028
Z shunt	0.038464	-0.0048	BUS_2	-8.99273	2.285434
BUS_14	-19.8595	-5.36289	BUS_4	-26.2901	11.0013
BUS_17	29.72104	7.627688			
			24 : BUS_30 V= 1.023 pu/132kV -6.13 deg		
8 : BUS_16 V= 1.060 pu/220kV -2.17 deg			P(MW)	Q(Mvar)	
	P(MW)	Q(Mvar)	Generation	0	0
Generation	24.5	8.845836	PQ Load	15.1	1.519999
PQ Load	0	0	Z shunt	0.036224	-0.0031
Z shunt	0.054378	-0.00047	BUS_11	-24.6874	-1.94795
BUS_19	24.44563	8.846302	BUS_31	9.551134	0.431043
			25 : BUS_31 V= 1.019 pu/132kV -8.21 deg		
9 : BUS_17 V= 1.057 pu/132kV -1.93 deg			P(MW)	Q(Mvar)	
	P(MW)	Q(Mvar)	Generation	0	0
Generation	0	0	PQ Load	31.9999	3.209986
PQ Load	5.59E-06	-8.3E-07	Z shunt	0.053495	-0.00509
Z shunt	0.242522	0.265857	BUS_26	-60.6956	-1.42567
BUS_15	-29.6885	-7.50897	BUS_27	38.16478	-1.69568
BUS_19	29.44602	7.243118	BUS_30	-9.52261	-0.08355
			26 : BUS_4 V= 1.060 pu/132kV 0.48 deg		
10 : BUS_18 V= 1.060 pu/132kV -1.40 deg ; Qmin limit			P(MW)	Q(Mvar)	
	P(MW)	Q(Mvar)	Generation	2.300017	-6.68586
Generation	7.7	3.35E-07	PQ Load	0	0
PQ Load	1.2	0.13	Z shunt	0.078265	-0.00173
Z shunt	0.039143	-0.001	BUS_10	-53.6547	2.702513
BUS_14	9.129001	-1.96823	BUS_11	63.33706	1.782212
BUS_3	-2.66814	1.839228	BUS_3	26.62889	-10.2124
			BUS_6	-34.0895	-0.95643
11 : BUS_19 V= 1.052 pu/220kV -2.94 deg					
	P(MW)	Q(Mvar)	27 : BUS_5 V= 1.060 pu/132kV -0.00 deg		
Generation	0	0			
PQ Load	7.4E-06	-2.4E-06			

Z shunt	0.328288	0.264821		P(MW)	Q(Mvar)
BUS_16	-24.3715	-8.4513	Generation	66.70001	27.20888
BUS_17	-29.4296	-6.69411	PQ Load	0	0
BUS_28	53.47282	14.88059	Z shunt	0.039134	-0.00089
			BUS_11	91.62554	24.83451
12 : BUS_2 V= 1.059 pu/132kV -0.70 deg			BUS_6	-24.9647	2.37526
	P(MW)	Q(Mvar)			
Generation	0	0	28 : BUS_6 V= 1.064 pu/132kV 1.19 deg		
PQ Load	2.799993	0.219973		P(MW)	Q(Mvar)
Z shunt	0.039079	-0.00057	Generation	0	0
BUS_1	-11.8519	1.98039	PQ Load	4.199982	0.500002
BUS_3	9.012791	-2.19979	Z shunt	0.059146	-0.00135
			BUS_10	-63.5692	-0.01506
13 : BUS_20 V= 0.964 pu/132kV -19.45 deg			BUS_4	34.20573	1.381227
	P(MW)	Q(Mvar)	BUS_5	25.10436	-1.86482
Generation	0.2	0.02			
PQ Load	3.77E-06	-6.6E-07	29 : BUS_7 V= 1.080 pu/132kV 4.48 deg		
Z shunt	0.016147	-0.00074		P(MW)	Q(Mvar)
BUS_21	0.183849	0.020744	Generation	25	2.26
			PQ Load	1.48E-05	-2.2E-06
14 : BUS_21 V= 0.964 pu/132kV -19.45 deg			Z shunt	0.020309	-0.00052
	P(MW)	Q(Mvar)	BUS_8	24.97968	2.260525
Generation	0	0			
PQ Load	2.899944	1.12E-05	30 : BUS_8 V= 1.078 pu/132kV 4.07 deg		
Z shunt	0.032322	-0.001		P(MW)	Q(Mvar)
BUS_20	-0.18385	-0.02073	Generation	6.899982	4.99E-05
BUS_23	-2.74842	0.021724	PQ Load	2.5	0.3
			Z shunt	0.040439	-0.00116
15 : BUS_22 V= 0.951 pu/132kV -21.96 deg			BUS_10	29.30146	1.778348
	P(MW)	Q(Mvar)	BUS_7	-24.9419	-2.07714
Generation	0	0			
PQ Load	81.79834	0.00029	31 : BUS_9 V= 1.078 pu/132kV 4.19 deg		
Z shunt	0.015744	-0.00047		P(MW)	Q(Mvar)
BUS_23	-81.8141	0.000183	Generation	34.2	1.53
			PQ Load	8.59E-06	-1.8E-06
16 : BUS_23 V= 0.964 pu/132kV -19.28 deg			Z shunt	0.020234	-0.00064
	P(MW)	Q(Mvar)	BUS_10	34.17976	1.530637
Generation	0	0			
PQ Load	12.10001	-0.00014			
Z shunt	0.048592	-0.00104			
BUS_21	2.750603	-0.01375			
BUS_22	82.8763	3.880971			
BUS_24	-97.7755	-3.86604			

Appendix H: Dataset sample

Dataset sample for fault type identification:

va	vb	vc	ia	ib	ic	f_type
98938.38	-197877	98938.38	70.88125	529.1842	-600.065	0
104272.3	-197779	93506.82	50.36897	541.0897	-591.459	0
109503.3	-197486	87982.97	29.8069	552.4614	-582.268	0
114626.3	-196999	82372.3	9.215364	563.2879	-572.503	0
119636.1	-196316	76680.33	-11.3853	573.5587	-562.173	0
124527.9	-195441	70912.69	-31.9748	583.2634	-551.289	0
129296.8	-194372	65075.07	-52.5327	592.3926	-539.86	0
133938	-193111	59173.22	-73.0389	600.9372	-527.898	0
138447.1	-191660	53212.98	-93.473	608.8889	-515.416	0
142819.6	-190020	47200.23	-113.815	616.2396	-502.425	0
147051.1	-188192	41140.89	-134.044	622.9823	-488.938	0
151137.5	-186178	35040.95	-154.142	629.1102	-474.969	0
155074.7	-183981	28906.44	-174.087	634.6173	-460.531	0
158858.9	-181602	22743.39	-193.86	639.4982	-445.638	0
162486.3	-179044	16557.9	-213.442	643.7479	-430.306	0
165953.4	-176309	10356.07	-232.814	647.3625	-414.549	0
169256.7	-173401	4144.018	-251.955	650.3381	-398.383	0
172393	-170321	-2072.12	-270.848	652.6721	-381.824	0
175359.1	-167073	-8286.22	-289.474	654.3619	-364.888	0
178152.2	-163660	-14492.1	-307.814	655.406	-347.592	0
180769.4	-160086	-20683.8	-325.851	655.8034	-329.953	0
183208.3	-156353	-26855	-343.565	655.5535	-311.988	0
185466.3	-152467	-32999.7	-360.941	654.6568	-293.716	0
187541.3	-148430	-39111.8	-377.961	653.114	-275.153	0
189431.3	-144246	-45185.3	-394.607	650.9267	-256.32	0
191134.3	-139920	-51214.3	-410.864	648.0971	-237.233	0
192648.6	-135456	-57192.7	-426.716	644.6279	-217.912	0
193972.9	-130858	-63114.6	-442.146	640.5226	-198.376	0
195105.7	-126131	-68974.3	-457.14	635.7852	-178.645	0
196046	-121280	-74765.9	-471.683	630.4203	-158.737	0
196792.8	-116309	-80483.7	-485.761	624.4334	-138.672	0
197345.4	-111223	-86122.1	-499.359	617.8302	-118.471	0
197703.2	-106028	-91675.5	-512.465	610.6174	-98.1528	0
197865.9	-100727	-97138.4	-525.064	602.802	-77.7378	0
197833.4	-95327.9	-102506	-537.146	594.3917	-57.2461	0
197605.6	-89834.2	-107771	-548.697	585.3949	-36.6978	0
197182.8	-84251.8	-112931	-559.707	575.8204	-16.1134	0
196565.4	-78586.3	-117979	-570.164	565.6776	4.486827	0
195754	-72843.3	-122911	-580.059	554.9766	25.08265	0

Appendix I: Publication

Paper acceptance Notification from Editorial Team

Notifications

×

[IOEGC16] Editor Decision

2025-04-01 11:22 PM

suresh prajapati, Bishal Silwal:

We are pleased to inform you that your manuscript titled "Hybrid AI Model for TL Fault Identification" submitted to 16th IOE Graduate Conference is **Accepted** for presentation in the Conference as well as inclusion in the Peer-Reviewed Proceedings. Please note that inclusion in hard copy proceedings is contingent upon your timely response to further edits, if any, during the publication process.

With Warm Regards,
IOEGC-16 Editorial Team

Hybrid AI Model for TL Fault Identification

Suresh Prajapati ^a, Bishal Silwal ^b

^{a,b}, Department of Electrical Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal

✉ ^a 079mस्पे023.suresh@pcampus.edu.np, ^b bishal.silwal@pcampus.edu.np

Abstract

Accurate fault detection in power systems is crucial for reliability and efficiency. Traditional Machine Learning (ML) and Artificial Neural Network (ANN) models face limitations in accuracy. This literature proposes an effective, cost-effective and fault-detecting model that detects fault type and location with a high accuracy about 95%. Rather than only using traditional machine learning models, it uses a hybrid model which combines some AI models with powerful optimization techniques, in which optimization techniques will weigh and tune the AI models used for better accuracy. In this proposed method, three different ML Algorithms (RF, SVM and GB) are unified with the PSO optimization technique. Power system failures are simulated in MATLAB/SIMULINK and the AI model is developed using the Python programming language. In this feature selection and data tuning are not performed, the whole model is trained and developed with raw bus data like voltage and current etc. It is tested with fault data using performance metrics (RMSE and CV) without any precomputation. The model performs well on different topologies and data sets of the system, validating its robustness and adaptability, making it a promising approach for monitoring and maintenance of real-world power systems.

Keywords

Fault Detection, Hybrid AI Model, PSO Technique, Machine Learning, Random Forest Classifier (RF), Gradient Boosting (GB), Support Vector Machine (SVM), Cross Validation (CV), Root Mean Squared Error (RMSE), TL Fault, Artificial Intelligence, MATLAB/SIMULINK, Python, IEEE 14 Bus

1. Introduction

1.1 Background

All social classes in the modern era require electric power for its functioning, this created an increased dependence on power systems. This dependence resulted in a rapid boom in the field of power systems during the past few decades, leading to drastic growth in the development of the transmission and distribution sector. The distribution and transmission lines are exposed to adverse weather conditions such as rain, lightning strikes, heavy wind, snow, floods, and factors like birds tripping lines, and accidents caused by humans. This results in various faults which turn into a power outage. So, when a fault occurs in the power system architecture, it reduces the reliability of the system for the customers. To provide an uninterrupted power supply, the power utility needs efficient detection and classification of the fault. Proper detection and identification of the fault that occurred will help in the faster restoration of an outage caused by a fault. Physically checking for the fault is proven to be costly, time-consuming, and difficult so computational techniques for finding the fault type can speed up the restoration in the event of a fault occurring. The computing capacity, lower cost, and a reduction in the size of computers have resulted in the wider usage of high-performance solutions for fault classifications like machine learning techniques.

In addition to traditional systems for energy generation,

such smaller on-site power-generating platforms have emerged recently. DER arouses new investment areas, such as Micro-grids. A micro-grid, a localized remotely controllable and self-operating group of energy sources, uses a type of controller that may operate in grid-connected or islanded modes to dispatch energy [1]. More importantly, increasing renewable penetration reduces rotating inertia. This may cause system instability or even a blackout, in case of catastrophic events or grid fault conditions and faults need to be cleared quickly [2] [3]. Therefore, special precautions must be taken to avoid any greater outage and/or cascading blackouts.

1.2 Fault Detection Approaches

Many fault detection techniques have been developed recently and reported in power system literature [4][5]. In the power system, the protective relays require a large fault current to detect the faulty condition [6]. Most of the DGs, nowadays, are equipped with inverter technology which typically contributes at two times of per unit rated current, as a rule of thumb [7]. This is troublesome because the current level may not be sufficient to trigger the relaying action, and it puts the inverter-equipped modern distribution system at risk. Real-time data acquisition devices like Frequency Disturbance Recorders (FDR), optimally placed Phasor Measurement Units (PMUs), and many other equipment are used to detect power system faults [8]. Line outage detection using phasor angle measurements is reported in [9]. Fault detection occurred

in DC and AC micro-grids are reported in [10]. Hardware-based applications seem not scalable, and they are infeasible due to financial problems. Thus, many new intelligent computer-aided prediction techniques have been developed to deal with fault scanning [11]. A dual setting directional over-current relay based communication assisted scheme is proposed in [12]. A three-stage approach based on real time data for protection of micro-grid is introduced in [13]. The authors in [14] presented a quadrature and zero sequence components based adaptive protection scheme.

Above mentioned most of the literature are based on protective relays and other physical devices, which require a large fault current to detect the faulty condition. And in ML Algorithm based literature lacks hyperparameter tuning which are essential for greater accuracy. In other methods, signal processing or feature selection techniques are required to train algorithms which will slow down the training process. In this literature raw datasets will be utilized (voltage and current waveform) for predictions.

1.3 Hybrid AI Model for Fault Detection

As in [11] different ML algorithms have different limitations and benefits, so hybrid techniques i.e. combining ML algorithms with Optimization techniques like GA or PSO can be integrated for better accuracy. In this work, Ensemble Model is developed using three different ML Algorithms. For ensemble, these ML Algorithms are weighted by using PSO technique. Also for Hyper Parameter tuning of the ML Models PSO technique is used.

2. Theoretical Overview

2.1 Faults in Power System

Since the inception of power distribution networks, faults have been the primary cause of power outages, which have had a significant negative impact on the economy and society. In power systems, faults are caused by several factors, such as component deterioration, animal intrusion on electrical equipment, accidents, and weather conditions. In distribution systems, 80% of power disruptions are caused by faults. Faults in the power system are divided into two categories, series faults, and shunt faults [15]. Further types of faults in power systems are classified as in Figure 1 [11]

Expressions for Current for different types of faults are specified in equations below for fault in k^{th} bus. Equation (1),(2) and (3) are expressions of phase wise current for LG fault, LL fault and LLG fault respectively.

$$I_f = \frac{3V_f}{Z_{kk1} + Z_{kk2} + Z_{kk0} + 3Z_f} \quad (1)$$

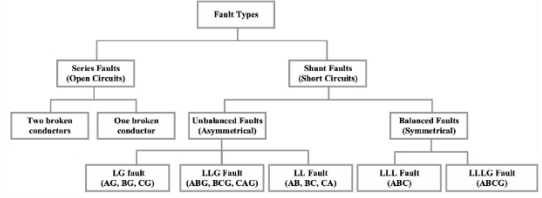


Figure 1: Fault type classification in power system

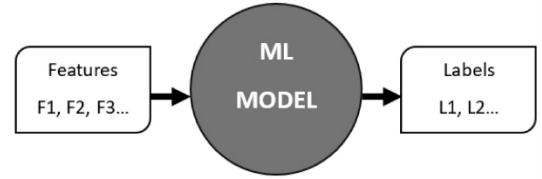


Figure 2: : ML Algorithm

$$I_f = \frac{j\sqrt{3}V_f}{Z_{kk1} + Z_{kk2} + Z_f} \quad (2)$$

$$I_f = \frac{3V_f Z_{kk2}}{Z_{kk1}(Z_{kk2} + Z'_{kk0}) + Z_{kk2}Z'_{kk0}} \quad (3)$$

where,

$$Z'_{kk0} = Z_{kk0} + 3Z_f \quad (4)$$

In the above expressions, I_f is fault current, V_f is pre-fault voltage at k^{th} bus and Z_f is fault impedance. Z_{kk1} and Z_{kk2} are zero sequence, positive sequence and negative sequence of kk^{th} element of Z-Bus.

2.2 Machine Learning

Machine Learning (ML) is a field of artificial intelligence AI that uses algorithms to teach machines to learn and improve from data. ML systems use algorithms to analyze substantial amounts of data, learn from the insights, and then make decisions. The more data a system is exposed to, the better it gets at performing tasks. For Example, ML algorithms can be trained to perform specific tasks, such as identifying flowers in a photograph, predicting housing prices, or making chess moves.

The Structure of the ML Algorithm is shown in Figure 2 Which includes Features as input and output as labels. Features and labels are data provided to the ML model for training it. After training the ML Model we can get desirable output for given input or features.

2.2.1 Random Forest Classifier

Random Forest (RF) classifier is an ensemble learning method for classification which was introduced by Breiman

[16]. The need for a random forest classifier arose due to the inaccuracy and over-fitting found in Decision tree algorithm. This classifier combined the simplicity of the decision tree with flexibility resulting in an improvement of accuracy and variance by creating a set of decision trees resembling a forest. Bootstrap aggregation (bagging) is used to avoid over-fitting, improve stability, and reduce the high variance found in the decision tree. In random forest, only a subset of features is selected at random out of the total and the best-split feature from the subset is used to split each node in a tree, whereas in bagging all features are considered in splitting the node. Each tree must assign a vote, and the number of votes will help diagnose the problem and improve classification accuracy. Using a bootstrapped sample and considering only a subset of variables at each step results in a wide variety of trees. While designing each tree almost one-third of the input data will not enter the training dataset, this data is called out-of-bag data. Since this data is not used for training, this can be used to calculate the estimation of the out-of-bag error which is the proportion of out-of-bag samples that are incorrectly classified. Every tree is allowed to grow to its maximum, no pruning is done and finally, the prediction of the class is done by calculating the majority votes. The random forest has advantages like handling the missing values and maintaining the accuracy for the missing data and it can handle large data sets with higher dimensionality.

2.2.2 Support Vector Machine

Support Vector Machine (SVM) algorithm is a supervised learning model which uses a maximum margin hyperplane for a decision boundary to distinguish between two classes. The main objective of SVM classifiers is to find an optimal hyperplane that maximizes the margin between two classes, this linear way of separation can also be extended to multi-class problems. For cases with points which are not linearly separable using linear classifier results in a large training error. This can be rectified by using a non-linear SVM which the dot products map each data point to a higher dimensional space, but this results in a high computational requirement. The kernel-based functions obtain the dot product without explicitly mapping the points to the higher dimensional space thus saving computational resources. In this work, the kernel used is an RBF kernel. The important parameters for RBF-based SVM are a regularization parameter denoted by C and γ which determines the wrongly classified point on the SVM boundary.

2.2.3 Gradient Boosting

Gradient Boosting (GB) is a machine learning algorithm used for classifying and regression problems. It builds the model in an iterative way by summing weak learners (general decision trees) to build up a strong learner. The approach is to develop new models on an iterative basis to correct the errors of previous models. Work of Gradient Boosting is as Below.

1. Initialize the model with simple prediction, most often the target mean.
2. Calculate residuals (errors) of the model.
3. Train a weak learner (a shallow decision tree, for example) to predict these residuals.
4. Update the model by adding the predictions of the new learner, weighted by a learning rate.
5. Repeat until a stopping criterion (a fixed number of iterations or very small improvement) is reached

2.2.4 Hyper parameters

Hyper parameters in Machine Learning are parameters that control the learning process and model behavior but are not derived from the data itself. They are pre-set before training begins and may have a big impact on the performance of the model. Some Hyper Parameters include:

Learning Rate: Determines how much the model updates its weights during training.

Number of Epochs: The number of times the model gets to see the entire dataset during training.

Kernel Type: In SVM it decides the kind of transformation to the input space.

C Parameter: In SVM it is used for controlling the outliers — low C implies we are allowing more outliers, high C implies we are allowing fewer outliers.

Number of estimators: In RF it is the number of decision trees in the forest

2.2.5 Ensemble

Ensemble is a method where multiple models (typically known as "base learners" or "weak learners") are combined to possess better predictive accuracy than any individual model. The idea is that by combining the predictions of numerous models, the risk of over-fitting will be reduced and generalization will be improved. Some Commonly employed ensemble models are bagging, Boosting, Stacking, Voting, Blending etc. The Ensemble Model is shown as in Figure 3. In the figure three weak learners (ML Models) are being fed by random three sample from same dataset.

In Ensemble multiple models are generally trained parallelly or Sequentially. Parallel training model is known as Bagging whereas sequential training model is known as Boosting. In Figure 3 it shows a simple bagging ensemble model.

Finally, predictions from different weak learners are combined or aggregated as shown in Figure 3 using different technique. These techniques are known as soft

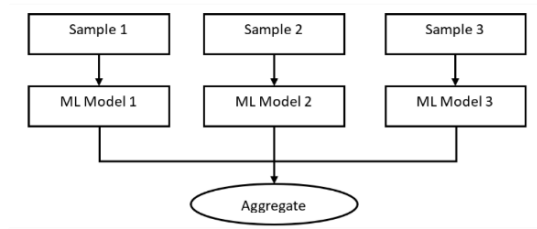


Figure 3: Ensemble

voting and hard voting. Soft voting is based on probability averaging of weak learners prediction. And, hard voting is based on majority voting of weak learners prediction i.e. prediction result with highest count is taken as final prediction result. During soft voting different weak learners can be assigned with different weights to make a weighted final prediction.

2.2.6 Validation or Performance Metrics

In Machine Learning, validation and performance metrics are used to evaluate how well a model generalizes to new, unseen data. They help in comparing different models and fine-tuning hyper parameters. Common validation techniques include train-test split, dividing the dataset into training and test sets (e.g., 80% train, 20% test). Performance Metrics quantify a model’s accuracy and overall effectiveness. Some commonly used validation or performance metrics are Cross Validation (CV), accuracy, recall, precision, F1-score, Root Mean Squared Error, Mean Absolute Error etc.

2.2.7 Cross Validation

Cross Validation (CV) is a performance metric used in machine learning which is statistical method for data analysis to estimate the performance of a model. It involves dividing the dataset into subsets, training the model on some subsets, and verifying it on others. This is performed to estimate the model’s ability to generalize over unseen data and prevent over fitting. One of the most widely used CV Techniques is K-Fold CV which is as shown in Figure 4. The data set is split into k equal-sized folds. The model is trained on k-1 folds and tested on the remaining fold. This is repeated k times, with each fold having been used as a test set once. The final performance measure is the average of all the test scores. Commonly used CV Techniques is K-Fold CV which is as shown in Figure 4. The dataset is split into k equal-sized folds. The model is trained on k-1 folds and tested on the remaining fold. This process is repeated k times, with each fold used as a test set once. The final performance metric is the average of all the test results. In common practice in machine learning and data science, CV = 5(i.e. No. of Fold = 5) is because it strikes a balance between model evaluation and training effectiveness.

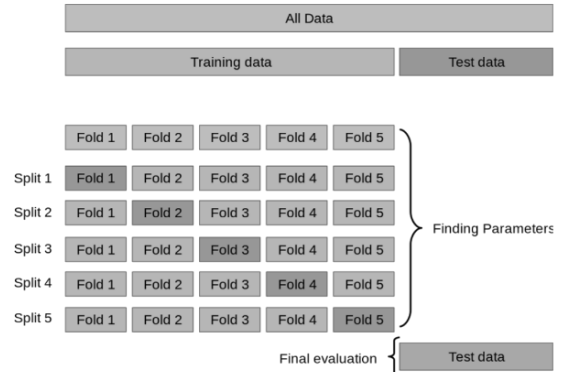


Figure 4: K Fold Cross Validation

2.2.8 Root Mean Squared Error

Root Mean Squared Error (RMSE) is a widely used measure to estimate the error of a model in ML problems. It is the square root of the mean of squared differences between predicted and actual values. Mathematically,

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \tag{5}$$

where:

- y_i is the actual value,
- \hat{y}_i is the predicted value,
- n is the total number of samples.

2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an Optimization technique Inspired by the social behavior of bird flocking or fish schooling. As In Each bird from of Swarm following a bird (particle) which is near to the food (target) as shown in Figure 5. It is also known as Population based stochastic algorithm. Stochastic is a process having a random probability distribution or pattern that may be analyzed statistically but may not be predicted precisely. It has some similar characteristics to GA like Random Population, Fitness value and Update population steps. Genetic operator (crossover and mutation) is not used in PSO and Each particle has its own velocity. Each have and nearest have, each will try to have updating velocity and weights.

2.3.1 Pseudo-code for PSO

The step for each particle is shown below:

1. initialize particle. for $i=1$ to n ($n \rightarrow$ no. of iteration)
2. for each particle:

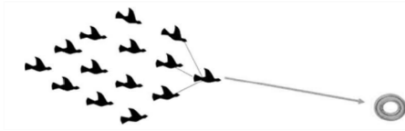


Figure 5: Particle Swarm Optimization

- (a) Calculate fitness value (FV)
- (b) If $FV > pBestold$ (Personal Best):
- (c) $pBestnew = FV$
3. Choose the particle with the best FV of all particles as the gBest (Global Best)
4. For each Particle:
 - (a) Calculate Particle Velocity
 - i. $v = w * v + c1 * rand(0,1) * (pBest - pp) + c2 * rand(0,1) * (gBest - pp)$
 - (b) Update particle position.
 - i. $pp = pp + v$
5. Is $error < tol$

pp → Present Position

ci → Learning Variable

w → Inertia Weight

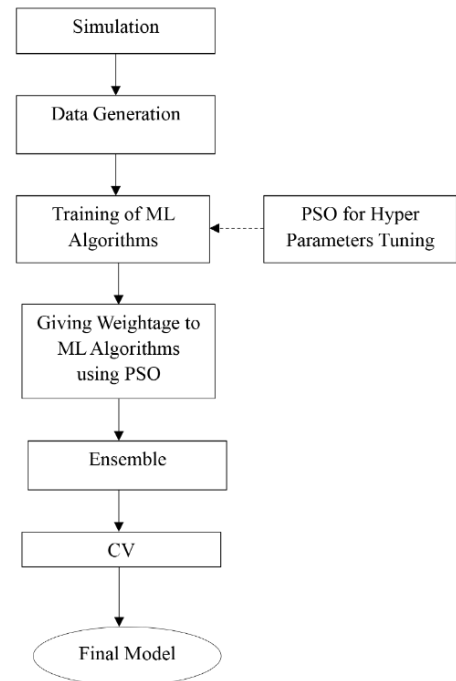


Figure 6: Hybrid AI Model

3. Methodologies

3.1 Approach

The summarized workflow of the hybrid AI model for fault identification is shown in figure 6. Firstly, simulation of the electric grid system is performed, which is studied for both faulty and non-faulty scenarios. Data are generated and classified according to different faulty scenario which are used for training ML Algorithms. Hyper parameters of the ML Algorithms are tuned using the PSO Optimization Technique. This tuned ML Algorithms are used by ensemble to give a weighted result. Weighting to ML Algorithms are again done using PSO so that final fault identification error is minimized. Finally, Cross-Validation performance metric is used to test the validity or accuracy of the Final PSO based Ensemble Model.

3.2 Proposed Power Grid Model

The IEEE-14 bus system is used in this work for the study of fault classification. It has a total of 14 buses, 2 generators, 3 synchronous compensator, five generators in total, and 11 loads. Different types of fault are injected in this to study the response of voltage and current for faulty scenarios. IEEE 14 bus system is as shown in Figure 7.

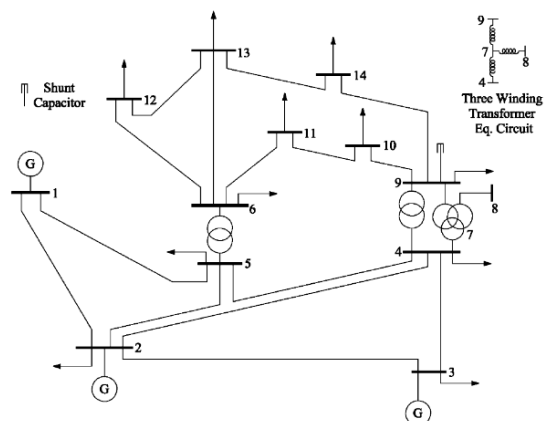


Figure 7: IEEE 14 Bus System

3.3 Simulation and Data Generation

First, IEEE 14 bus system is modeled in MATLAB/Simulink. Load Flow of the System is performed. After the desired load flow result, the faults are injected to study the current voltage characteristics. The datasets are generated from the current voltage waveform for types of faults and normal conditions. The Generated Data sets are required for training and testing the machine learning algorithms.

3.3.1 Simulation and Simulation Results

Simulation is done for a runtime of 1.2 seconds with fault induced on line Between Bus 9 and 14. LG Fault is induced at five different location on line Between Bus 9 and 14. The time interval for a each fault location and normal condition is taken about 0.2 seconds.

3.3.2 Dataset for ML Model

Dataset for IEEE 14 Bus System with 6 different cases (Normal case and fault case at 5 different location) are generated. The datasets are generated from the current, voltage waveform for different cases and saved as a CSV file Format. Dataset are extracted using Scope Data Logging. The Current and Voltage waveform are recorded at swing bus (Bus 1). The Dataset consists of time series data of phase wise voltage and phase wise current (i.e. three voltage and three current data). The Sampling time for data was taken as 5e-4 seconds and since it was simulated for 1.2 seconds the dataset size is 2400*6. Here current and voltage data are features and fault location data are labels/Target for the ML Model. Pre-processing or feature selection of data is not performed in this work only raw simulation output data are used for ML Model Development.

3.4 ML Model Development

The Generated Data sets are required for training and testing ML algorithms. Generated data are divided into train and test data for ML model development and testing.

Three different ML Algorithms are used to train the model. Mostly used algorithms are SVM, Decision Tree, RF etc. [11], due to their better performance in fault diagnosis. In this work also Algorithms are chosen according to their better performance and also algorithm of different nature shall be chosen for variation in training. The algorithms chosen in these work are **RF, SVM** and **GB**.

Firstly, **hyper parameters** of the algorithms are tuned using **PSO**. Most Effective Hyper parameters of RF, SVM and GB are **no. of estimators, C parameter** and **Learning Rate** Respectively. Then, the weights of ML algorithms for the **ensemble** with **bagging process** and based on the **soft-voting technique**, are calculated by **PSO** module. The ensemble process uses three ML algorithms with best hyper parameters and weights according to PSO results. In this process, **cross validation** is applied to the dataset to evaluate the predictive performance of the model results in

the training dataset step. In applying this approach, one of the well-known methods, **Stratified K-Fold Cross Validation**, is applied as **CV=5**. That means, each fold consists of 20% data i.e. for each process 80% of data will be chosen for the training part and 20% for testing purposes as discussed in 2.2.6. This ratio was chosen because our dataset contained 2400 samples, providing enough test instances (about 480) for reliable error estimation while maximizing training data. Also, Cross-validation improves model generalization by training on multiple data splits, reducing overfitting, and providing a more reliable performance estimate. Finally, the Hybrid AI Model is compared with weak learners (SVM,GB and RF) with its default Hyper parameters using CV performance Metrics.

For ML Model Development, Python Programming Language is used.

3.5 Work Flow of PSO Based Ensemble

The Work Flow of The Whole Model is as below:

1. Simulation
 - (a) Load Flow is Performed for IEEE 14 bus System
 - (b) LG Fault is provided for 5 different location between bus 9 and 14.
2. Dataset Generation using Scope Data Logging. Dataset Consists of:
 - (a) Phase Wise Voltage
 - (b) Phase Wise Current
 - (c) Fault location
3. Data Set Processing:
 - (a) 80% of data is taken for training and remaining are taken for testing randomly
 - (b) Voltage and Current are taken as Features and Fault Location as Target/Label for ML Model
 - (c) Scaling of data is done for better performance
4. Hyper Parameters Tuning.
 - (a) Three ML Algorithms chosen are:
 - i. RF
 - ii. SVM
 - iii. GB
 - (b) Hyper Parameters chosen for tuning keeping other parameters constant:
 - i. no. of estimators for RF
 - ii. C parameter for SVM
 - iii. Learning Rate for GB
 - (c) Limit taken for Hyper parameters:
 - i. no. of estimators: (50,200)

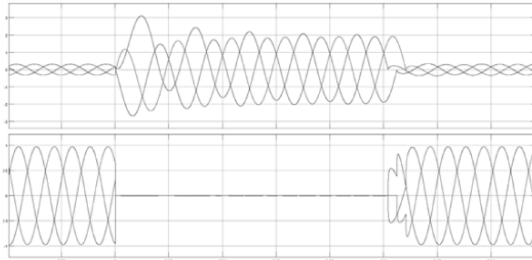


Figure 8: Waveform of LLLG Fault

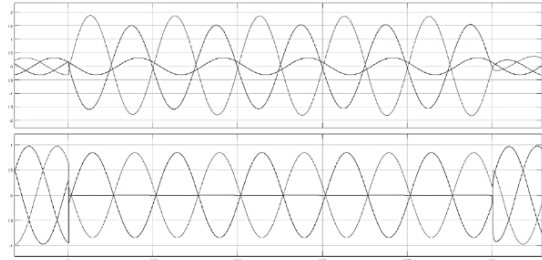


Figure 9: Waveform of LL Fault

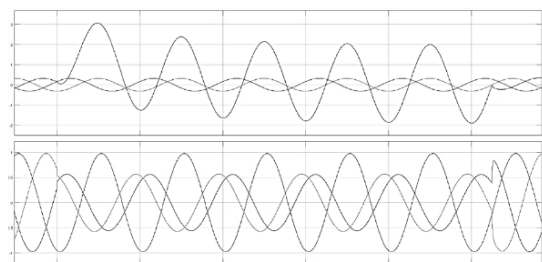


Figure 10: Waveform of LG Fault

- ii. C parameter: (0.1,10)
 - iii. Learning Rate: (0.5,2)
 - (d) PSO parameters (Randomly Chosen):
 - i. $c1=0.5$
 - ii. $c2=0.5$
 - iii. $w=0.9$
 - iv. no. of Particles=20
 - v. iteration no.=20
 - (e) Soft Voting Classifier is taken in ensemble
 - (f) Fitness function for PSO is taken as RMSE of ensemble algorithm
 - (g) Objective of PSO technique is to minimize RMSE
 - (h) From This we get the optimum Hyper Parameters
5. Calculating Optimized Weights for Soft Voting.
- (a) Limit of weights for all is taken between 1 to 5
 - (b) Fitness Function is again chosen as RMSE and PSO parameters are set as above steps.
 - (c) From this we get the optimum weights for all ML Algorithms.
6. Finally K Fold Cross Validation with CV=5 is performed with optimum weights and optimum Hyper Parameters.

4. Result and Discussion

4.1 Simulation Result

Simulation Result i.e. voltage and current waveform for different Types of faults, LLLG Fault, LL Fault and LG fault is as shown in Figure 8, Figure 9, Figure 10 respectively.

4.2 PSO Results

Results obtained by PSO Hyper Parameters Tuning are as follows:

- number of estimators for RF=150
- C parameter for SVM=7

- Learning Rate for GB=0.2

The weights for different ML Algorithms after optimizing from PSO are:

- for RF=4
- for SVM=3
- for GB=1.5

The weights for ML models have been given by PSO to minimize error in final prediction. Model with low error are weighted high and vice versa i.e. PSO have given weight-age to them according to their performance.

4.3 Model Testing:

Finally, by using above hyper parameters and weights ensemble machine learning Model with Soft Voting is deployed. RMSE and CV Mean score (i.e. Accuracy) are analyzed for checking model performance. The Final result is summarized in Table 1 which compares the base Models or weak Learners (SVM,GB and RF) with proposed Hybrid AI Model. The RMSE was found to be 0.47 and CV Mean score was found to be 94.07% for Hybrid AI Model. Since RF is also a ensemble Model, it has performed with good accuracy of 92.97% which is quit close with Hybrid AI Model. Here SVM and GB has been found very poor AI Models. Also, challenges aroused with distinguishing fault data in each transition or switching time, impacting the classification accuracy of ML Models.

ML Models	Accuracy (%)	RMSE
SVM	72.84	1.52
GB	84.40	1.184
RF	92.97	0.56
Hybrid AI Model	94.07	0.47

Table 1: Machine Learning Models Performance

5. Conclusion

This paper presents the study of Hybrid AI model consisting of ML algorithms and Optimization technique for classifying power system faults. Further, a brief theoretical background on each proposed machine learning algorithms and optimization technique has been introduced. Voltage and current signals were obtained from the IEEE 14-bus system and analyzed under both fault and no-fault conditions. Datasets were developed for the analysis of Fault identification. These datasets were used to train and evaluate the classifiers, with performance metrics such as CV Mean score and RMSE. CV Mean score was found to be greater than 94% which is rather good accuracy for detecting fault location. While IEEE 1547 standards permit a 2-second delay [17] to detect possible faults, the proposed method took as low as 0.1 seconds for detecting fault on this model. This means the computational time will be approximately 5% of the IEEE standards, which is significantly faster and more efficient for power system studies.

6. Future Works

Future studies can incorporate exploring deeper higher-level models, which, being more computationally expensive and requiring larger datasets, can possibly result in higher classification accuracy. Reinforcement learning, another method, is a class of machine learning aimed at explicitly solving multi-step problems through trial and error. Unlike supervised learning where the best outputs are given while training, reinforcement learning learns the best output itself in a trial and error process [14]. And the third one, Transfer learning is a machine learning where a pre-trained model for a task is used as a base to construct a model for another similar but different task. It is applied very widely in deep learning because training a large neural network from the beginning requires vast data and computational power. It can also enhance the generalizability and stability of the outcome using real-time checking of data. Its impact can be mitigated in terms of classification accuracy as well. Machine learning models designed in the literature are test data without training data base from actual measurement. It is a basic law of machine learning that model output quality will depend upon training data quality as well as how far they range the space of event possibilities to be found in the world within word use.

Acknowledgments

The authors are Thankful to the Department of Electrical Engineering, TU, IOE, Central Campus.

References


- [1] L. L. Grigsby. *Power System Stability and Control*. CRC Press, 2017.
- [2] A. Esmaeilian and M. Kezunovic. Prevention of power grid blackouts using intentional islanding scheme. *IEEE Transactions on Industry Applications*, 53(1):622–629, 2017.
- [3] S. Raza, H. Mokhlis, H. Arof, J. A. Laghari, and L. Wang. Application of signal processing techniques for islanding detection of distributed generation in distribution network: A review. *Energy Conversion and Management*, 96:613–624, 2015.
- [4] D. S. Pillai and N. Rajasekar. A comprehensive review on protection challenges and fault diagnosis in pv systems. *Renewable and Sustainable Energy Reviews*, 91:18–40, 2018.
- [5] D. S. Pillai and N. Rajasekar. Metaheuristic algorithms for pv parameter identification: A comprehensive review with an application to threshold setting for fault detection in pv systems. *Renewable and Sustainable Energy Reviews*, 82:3503–3525, 2018.
- [6] L. Yavuz, A. Soran, A. Onen, X. Li, and S. M. Muyeen. Adaptive fault detection scheme using an optimized self-healing ensemble machine learning algorithm. *CSEE Journal of Power and Energy Systems*, 8(4):1145–1156, 2022.
- [7] J. Keller and B. Kroposki. Understanding fault characteristics of inverter-based distributed energy resources. *NREL*, 2010.
- [8] H. G. Jiang, J. Zhang, W. Z. Gao, and Z. P. Wu. Fault detection identification, and location in smart grid based on data-driven computational methods. *IEEE Transactions on Smart Grid*, 5(6):2947–2956, 2014.
- [9] J. E. Tate and T. J. Overbye. Line outage detection using phasor angle measurements. *IEEE Transactions on Power Systems*, 23(4):1644–1652, 2008.
- [10] L. Che, M. E. Khodayar, and M. Shahidehpour. Adaptive protection system for microgrids. *IEEE Electrification Magazine*, 2(1):66–80, 2014.
- [11] M. M. Zaben, MUHAMMEDY.WORKU, and M. Hassan. Machine learning methods for fault diagnosis in ac microgrids: A systematic review. *IEEE Access*, 12:20260–20298, 2024.
- [12] H. M. Sharaf, H. H. Zeineldin, and E. El-Saadany. Protection coordination for microgrids with grid-connected and islanded capabilities using communication assisted dual setting directional overcurrent relays. *IEEE Transactions on Smart Grid*, 9(1):143–151, 2018.
- [13] S. Teimourzadeh, F. Aminifar, M. Davarpanah, and M. Shahidehpour. Adaptive protection for preserving microgrid security. *IEEE Transactions on Smart Grid*, 10(1):592–600, 2019.

- [14] M. Singh and P. Basak. Adaptive protection methodology in microgrid for fault location and nature detection using q_0 components of fault current. *IET Generation, Transmission & Distribution*, 13(6):760–769, 2019.
- [15] Y. G. Paithankar and S. Bhide. *Fundamentals of Power System Protection*. PHI Learning Pvt. Ltd., 2022.
- [16] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.
- [17] IEEE Standard. Ieee standard for interconnecting distributed resources with electric power systems. *IEEE Standard*, 2003:1547–2003, 2003.

Appendix L: Plagiarism Test Report

suresh prajapati

Adaptive Hybrid AI Model for the Detection of the Power System Faults

 Tribhuvan University

Document Details

Submission ID

trn:oid:::3117:451392379

Submission Date

Apr 22, 2025, 4:26 PM GMT+5:45

Download Date

Apr 22, 2025, 4:28 PM GMT+5:45

File Name

Thesis Report without reference.pdf

File Size

1.1 MB

48 Pages

10,718 Words

54,957 Characters





10% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- ▶ Small Matches (less than 8 words)

Match Groups


-  **96 Not Cited or Quoted 10%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **4 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 9%  Internet sources
- 6%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

-  **Hidden Text**
7 suspect characters on 1 page
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 96 Not Cited or Quoted 10%**
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**
Matches that are still very similar to source material
- 4 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 9% Internet sources
- 6% Publications
- 0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	eitca.org	<1%
2	Internet	www.geeksforgeeks.org	<1%
3	Internet	www.researchgate.net	<1%
4	Publication	Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023	<1%
5	Internet	researchbank.swinburne.edu.au	<1%
6	Publication	"Business Process Management", Springer Science and Business Media LLC, 2016	<1%
7	Internet	jai.front-sci.com	<1%
8	Internet	upcommons.upc.edu	<1%
9	Internet	www.coursehero.com	<1%
10	Internet	ebin.pub	<1%

11	Internet	www.electrical elites.com	<1%
12	Publication	Ton Duc Thang University	<1%
13	Internet	www.mdpi.com	<1%
14	Publication	Shazia Baloch, Sunil Srivatsav Samsani, Mannan Saeed Muhammad. "Fault Protec...	<1%
15	Internet	sciengtexpopen.org	<1%
16	Internet	etd.auburn.edu	<1%
17	Internet	iieta.org	<1%
18	Internet	www.springerprofessional.de	<1%
19	Internet	www.irejournals.com	<1%
20	Internet	www.ijert.org	<1%
21	Internet	www.lib.utexas.edu	<1%
22	Internet	www.slideshare.net	<1%
23	Publication	Scott, Graham. "Improving Clinical Text Classification Using Large Language Mod...	<1%
24	Internet	khazna.ku.ac.ae	<1%

25	Internet	pmc.ncbi.nlm.nih.gov	<1%
26	Internet	repository.its.ac.id	<1%
27	Internet	openscience.ub.uni-mainz.de	<1%
28	Internet	www.aanda.org	<1%
29	Internet	energetics.cosrec.org	<1%
30	Internet	techwatch.de	<1%
31	Internet	bes.ihep.ac.cn	<1%
32	Internet	link.springer.com	<1%
33	Publication	"Proceedings of Fifth International Conference on Computing, Communications, ...	<1%
34	Publication	Al-Abri, Tariq Ghusan. "Design and Development of the Pricing Framework for Pe...	<1%
35	Publication	Di Wu. "Data Mining with Python - Theory, Application, and Case Studies", CRC Pr...	<1%
36	Internet	fastercapital.com	<1%
37	Internet	loancalculatorcanada.ca	<1%
38	Internet	mSPACE.lib.umanitoba.ca	<1%

39	Internet	robots.net	<1%
40	Internet	vdoc.pub	<1%
41	Internet	www.powerworld.com	<1%
42	Publication	Batel Ziv, Yisrael Parmet. "Improving nonconformity responsibility decisions: a se...	<1%
43	Publication	Heda Song, Isaac Triguero, Ender Özcan. "A review on the self and dual interactio...	<1%
44	Publication	K. R. Padiyar, Anil M. Kulkarni. "Dynamics and Control of Electric Transmission an...	<1%
45	Publication	Mokhlis, Hazlie. "Fault Location on Distribution Networks Using Analytical Voltag...	<1%
46	Publication	S. Panda, D. P. Mishra, S. N. Dash. "Comparison of ANFIS and ANN Techniques in F...	<1%
47	Publication	Sallam, , and Malik. "Power flow analysis", Power System Stability Modelling Anal...	<1%
48	Internet	dokumen.pub	<1%
49	Internet	medium.com	<1%
50	Internet	theses.lib.polyu.edu.hk	<1%
51	Internet	vuir.vu.edu.au	<1%
52	Publication	Ayman El-Baz, Jasjit S. Suri. "Big Data in Multimodal Medical Imaging", CRC Press, ...	<1%

53	Publication	Dinesh Goyal, Bhanu Pratap, Sandeep Gupta, Saurabh Raj, Rekha Rani Agrawal, I...	<1%
54	Publication	H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Co...	<1%
55	Publication	Lee, Mang Hei Gordon. "The Cosmological Bootstrap and the Analytic Wavefuncti...	<1%
56	Publication	Rahul Nimbai, Mayur Patil, Veereshkumar Rathod, Santosh Pattar, Prema T. Akka...	<1%
57	Publication	Torabi, K.. "On-line adaptive Bayesian classification for in-line particle image m...	<1%
58	Internet	manualzz.com	<1%
59	Internet	pseccommunity.org	<1%
60	Internet	www.ajol.info	<1%
61	Internet	www.bartleby.com	<1%
62	Internet	www.jetir.org	<1%