Fault Detection in Power Electronics Converter with Machine Learning Approach

Project report submitted to
Visvesvaraya National Institute of Technology,
Nagpur in partial fulfillment of the requirements
for the award of
the degree

Bachelor of Technology
In
Electrical and Electronics Engineering

by

Prathmesh Pande (BT21EEE009)
Archis Prabhughate (BT21EEE013)
Gaurav Savale (BT21EEE019)
Kushal Tiwari (BT21EEE065)

under the guidance of
Dr. Krishnama Raju S

Department of Electrical Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

2024-25

Fault Detection in Power Electronics Converter with Machine Learning Approach

Project report submitted to
Visvesvaraya National Institute of Technology,
Nagpur in partial fulfillment of the requirements
for the award of
the degree

Bachelor of Technology
In
Electrical and Electronics Engineering

by

Prathmesh Pande (BT21EEE009)
Archis Prabhughate (BT21EEE013)
Gaurav Savale (BT21EEE019)
Kushal Tiwari (BT21EEE065)

under the guidance of
Dr. Krishnama Raju S

Department of Electrical Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

2024-25

Declaration

We, Prathmesh Pande, Archis Prabhughate, Gaurav Savale and Kushal Tiwari, hereby declare that this project work titled "Fault Detection In Power Electronics Converter With Machine Learning Approach" is carried out by us in the Department of Electrical Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution / University.

Date:

Sr.No.	Enrollment No	Names	Signature
1	BT21EEE009	Prathmesh Pande	
2	BT21EEE013	Archis Prabhughate	
3	BT21EEE019	Gaurav Savale	
4	BT21EEE065	Kushal Tiwari	

Certificate

This is to certify that the project titled "Fault Detection In Power Electronics Converter With Machine Learning Approach", submitted by **Prathmesh Pande, Archis Prabhughate, Gaurav Savale and Kushal Tiwari** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electrical and Electronics Engineering, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.

Dr. Krishnama Raju S
Assistant Professor,
Department of Electrical
Engineering, VNIT,
Nagpur

Prof. B.S.Umre
Head, Department of Electrical Engineering
VNIT, Nagpur
Date:

ACKNOWLEDGEMENT

First and foremost, we express our sincere gratitude to our guide, Dr. Krishnama Raju S, Assistant Professor, Department of Electrical Engineering, for his constant support and guidance throughout the course of this project. We are deeply grateful for his valuable suggestions during our regular meetings and seminars, from which we learned a great deal.

We would also like to thank Dr. Manojkumar R. Ramteke, Dr. Ritesh Kumar Keshri, and Dr. Bhooshan Rajpathak for their insightful recommendations and constructive criticism during various phases of project evaluation, which greatly contributed to the improvement of our work.

Our sincere thanks extend to all the esteemed faculty members of the Department of Electrical Engineering, VNIT Nagpur, for their excellent teaching and unwavering support throughout our undergraduate journey. We feel privileged to have had the opportunity to learn from and interact with such distinguished professors.

Lastly, we would like to express our heartfelt gratitude to our parents for their constant emotional and financial support, and for encouraging us to pursue engineering and follow our passions.

ABSTRACT

Power electronic converters are critical components in modern energy systems, enabling efficient conversion of DC to AC power. However, they are susceptible to faults, particularly open-circuit faults, which can lead to system failures and reduced reliability. Traditional fault diagnosis methods, such as time-domain or frequency-domain analysis, often require extensive manual effort and lack adaptability under varying load conditions.

This thesis presents a data-driven approach for fault diagnosis in 2-level and 3-level inverters using machine learning (ML). The proposed method leverages the Power-Invariant Clarke Transform to extract load-independent features from three-phase currents, ensuring robustness across different operating conditions. Key features, including the slopes ($\psi 1$,$\psi 2$) of transformed currents, are used to train and compare three ML models: Random Forest (RF), XGBoost, and a Hybrid (RF+XGBoost) classifier.

The dataset is generated via MATLAB Simulink simulations under varying resistive loads (10–40 Ω). The models are evaluated based on accuracy, computational efficiency, and adaptability to fault localization. Results demonstrate that the Hybrid model achieves the highest accuracy (98.5%), outperforming individual classifiers. Additionally, the sliding window technique and IQR-based outlier removal enhance real-time fault detection while reducing computational overhead.

This work contributes to reliable fault diagnosis in power electronics, with potential applications in renewable energy systems, electric vehicles, and industrial automation. Future research directions include extending the framework to higher-level inverters and integrating deep learning techniques for improved fault classification.

LIST OF FIGURES

21. DSO.

22. Inverter.

23. Pulse Generation.

24. Hardware Implementation.

LIST OF TABLES

NOMENCLATURE

1. Sa1 - A phase Switch 1
2. Sa2 - A phase Switch 2
3. Sa3 - A phase Switch 3
4. Sa4 - A phase Switch 4
5. Sb1 - B phase Switch 1
6. Sb2 - B phase Switch 2
7. Sb3 - B phase Switch 3
8. Sb4 - B phase Switch 4
9. Sc1 - C phase Switch 1
10. Sc2 - C phase Switch 2
11. Sc3 - C phase Switch 3
12. Sc4 - C phase Switch 4
13. Ia - current in A phase
14. Ib - current in B phase
15. Ic - current in C phase

INDEX

CHAPTER 1

Introduction

Inverters are essential components in power electronics, playing a pivotal role in a wide range of industries such as energy and power, manufacturing, industrial automation, electric vehicles, railways, construction, and consumer equipment. Their primary function is to convert direct current (DC) to alternating current (AC), which is critical for the operation of many electronic systems and machines. With the increasing integration of renewable energy sources like solar and wind, the demand for efficient and cost-effective inverters has risen significantly. This demand is driven by the need for enhanced energy efficiency and reduced costs in both residential and industrial applications. Inverters are not only vital for renewable energy systems but are also integral to the broader functioning of modern electrical and electronic infrastructure.

1.1 Classification of Inverters

Inverters can be broadly classified into:

- **2-Level Inverters:**

2-level inverters are suitable for low-to-medium power applications due to their simple design and cost-effectiveness. They switch between two voltage levels, making them ideal for systems with lower voltage and power requirements. While they are affordable and reliable, 2-level inverters tend to produce higher harmonic distortion and are less efficient at higher voltages or frequencies.

- **Multi-Level Inverters :**

Multi-level inverters are more suitable for high voltage and high power applications.They offer higher efficiency and better power handling capabilities.

Multi-level inverters were specifically designed to address the limitations of traditional 2-level inverters, offering several advantages that enhance their performance in high-power applications. They reduce voltage stress on switching devices by utilizing clamping diodes, which helps in preventing device failures and extending their lifespan. Additionally, multi-level inverters generate lower harmonic distortion, making them ideal for sensitive applications that require high-quality AC power. They also provide higher thermal stability and lower electromagnetic interference (EMI), which enhances overall system reliability and reduces operational noise. With the ability to handle higher power levels more efficiently, multi-level inverters contribute to reduced losses, improving system performance. The use of robust switching devices like MOSFETs and IGBTs further increases the reliability and longevity of the system, making them a preferred choice for demanding applications.

1.2 Faults in Inverters

Inverters are vulnerable to faults that can compromise system performance.

- Electrical faults like:
  - Undervoltage
  - Overvoltage
  - Overcurrent
- Faults can lead to disruption in system operations.

1.3 Types of Faults in Inverters

1. **Short-Circuit Faults**
   - Caused by unintended connections between two conductors.
   - Leads to excessive current flow.
   - Consequences:
     - Overheating
     - Component degradation
     - Fire hazards
2. **Open-Circuit Faults**
   - Occurs when a switch fails to conduct electricity due to a break in the circuit.
   - Leads to:
     - Distorted waveforms
     - Reduced power
     - Imbalance in inverter operation.

1.4 Traditional Fault Detection Methods:

- Methods include time-domain analysis, frequency-domain analysis, mathematical models, and expert systems.
- These methods require significant manual effort and expert knowledge, making them time-consuming.
- Mathematical models are complex and difficult to understand

1.5 Advantages of AI/ML for Fault Diagnosis:

- Artificial Intelligence (AI) and Machine Learning (ML) methods have emerged as powerful tools for fault detection.
- Advantages of AI/ML:
  - Fast diagnosis and localization of faults.
  - Ability to handle complex, non-linear datasets.
  - Improved accuracy and adaptability to varying load conditions.
- Machine Learning methods rely on historical fault data for classification.
- Feature engineering extracts relevant features from available data for fault diagnosis.
- AI/ML methods significantly reduce the time and manual efforts compared to traditional methods.

In recent years, Artificial Intelligence (AI) and Machine Learning (ML) have emerged as powerful tools for fault diagnosis in power electronics. The main advantages of AI/ML methods are their ability to offer fast fault detection and precise localization of faults. These methods can also handle large, complex, and highly non-linear datasets, making them particularly effective in dynamic environments with varying operational loads. AI/ML techniques are capable of delivering improved accuracy and better adaptability to fluctuating conditions compared to traditional methods. Moreover, these techniques rely on historical fault data, and feature engineering plays a crucial role in extracting relevant data features for classification, which simplifies and speeds up the process significantly. This ability to work with data directly, without requiring in-depth expert knowledge, drastically reduces the time and manual effort required for fault detection.

Machine learning methods, including Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Random Forests (RFs), Extreme Gradient Boosting (XGBoost), and Convolutional Neural Networks (CNNs), are widely used for fault detection in inverters. These algorithms are effective in processing large volumes of data and identifying complex patterns that might indicate faults in the system. For instance, the Power Invariant Clarke Transform is utilized in this project to convert three-phase AC currents into a two-dimensional orthogonal system, while maintaining power invariance. This transformation helps extract features that remain stable, even when the load changes, which is vital for accurate fault classification.

Once the features are extracted, they are analyzed using advanced ML algorithms like Random Forests and XGBoost. Random Forests are ensemble learning methods that combine multiple decision trees to improve classification accuracy. On the other hand, XGBoost is known for its superior performance in terms of accuracy and efficiency. The research employs a hybrid model combining both RF and XGBoost, which benefits from the strengths of each method, ultimately enhancing fault detection performance. These algorithms are tested on unseen data, which is crucial for improving the robustness and generalization of the classifier, ensuring its effectiveness under various real-world conditions.

In this research, the primary focus is on diagnosing faults in NPC (Neutral Point Clamped) inverters. NPC inverters, which are a type of multi-level inverter, are commonly used in high-power applications due to their ability to handle higher voltage levels efficiently. These inverters are susceptible to various faults, including open-circuit and short-circuit faults, which can significantly affect the performance and reliability of power systems.NPC inverters are particularly useful for medium- to high-power applications since they can produce a multilevel output that enhances power quality, decreases unwanted harmonic distortions, and reduces stress on switching devices. These features make NPC inverters a popular choice for advanced power systems.

A 3-level NPC inverter consists of four insulated-gate bipolar transistors (IGBTs) and two clamping diodes in each phase leg. By changing how these components switch on and off, the inverter can offer several voltage levels, improving the system's performance and efficiency while helping it handle faults better. This is essential for maintaining stability and reliability in the system.

However, NPC inverters are not without problems. One frequent issue is an open-circuit fault in the switching devices, which can severely disrupt the output electricity and, if unnoticed, cause further system damage. Traditional fault detection methods, like time-domain and frequency-domain analysis, often lack flexibility, are labor-intensive, may not be precise, and usually require expertise to interpret.

This project explores using machine learning to enhance fault detection in NPC inverters. The approach uses the Clarke transformation to extract valuable, load-independent features from three-phase current waveforms. These features are then processed by machine learning models such as Random Forest, XGBoost, and a hybrid of RF and XGBoost to accurately identify and classify faults. The dataset for training and testing these models comes from MATLAB Simulink simulations under loads ranging from 10 to 40 ohms.

The study shows that machine learning, combined with reliable feature extraction like the Clarke transformation, provides a strong and scalable solution for real-time fault detection in NPC inverters. This introduction lays the groundwork for a deeper discussion on inverter structure, fault mechanisms, and the application of intelligent fault diagnosis using data-driven methods.

1.6 Neutral Point Clamped (NPC) Inverters

NPC inverters are widely used in industries due to their unique advantages:

- Lower Harmonic Distortion: They produce a multilevel output that reduces Total Harmonic Distortion (THD).
- Reduced Voltage Stress:Clamping diodes limit voltage spikes across switching devices like IGBTs.
- High Efficiency:They have lower switching losses compared to 2-level inverters.
- Fault Tolerance:Their modular design allows for isolating faults without needing to shut down the entire system.

1.7 Structure of a 3-Level NPC Inverter

A 3-level NPC inverter comprises three phase legs, each with:

- 4 IGBTs (S_{x1}, S_{x2}, S_{x3}, S_{x4}) and 2 clamping diodes per phase (Figure 1.1).
- Three Output Voltage Levels:
  - +Vdc/2 (P-mode: Upper switches ON).
  - 0 (O-mode: Clamping diodes conduct).
  - Vdc/2 (N-mode: Lower switches ON).

1.8 Operating States

The inverter operates in three modes based on switching combinations:

- **P Mode:** Positive voltage (+Vdc/2) when the upper switches are ON.
- **O Mode:** Zero voltage (0) when the clamping diodes conduct.
- **N Mode:** Negative voltage (−Vdc/2) when the lower switches are ON.

Figure 1 shows the basic circuit of a 3 Level Neutral-Point-Clamped (NPC) inverter. This topology uses multiple levels to achieve higher output voltage quality, lower harmonic distortion, and reduced voltage stress on switches.

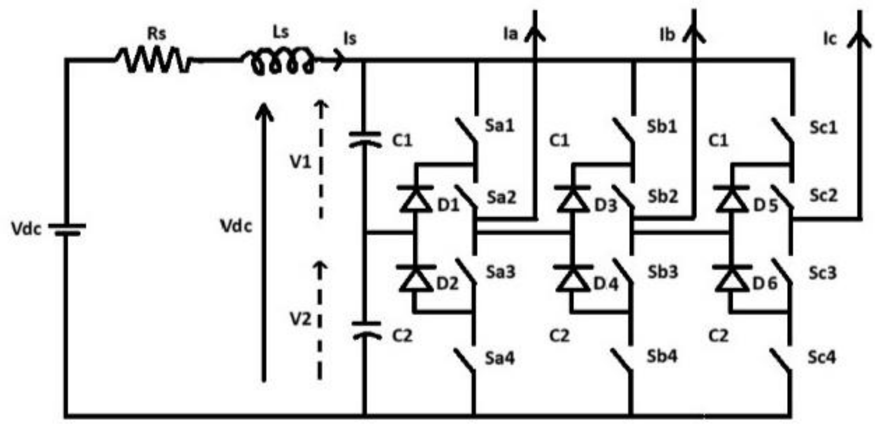


Figure 1: 3-Level Neutral-Point-Clamped Inverter Circuit Diagram

1.9 Working of NPC Inverter

The function of Neutral Point Clamped (NPC) inverter inverter works combines the switches, clamping diodes and DC-link capacitors to generate multiple voltage levels as output. So one full bridge leg of a typical 3-level NPC inverter contains four switches (S1 to S4) and two clamping diodes where capacitors split common DC voltage by half. The inverter can have a 3 level output: +Vdc/2, 0, −Vdc/2

When +Vdc/2 needs to be output it both the top switches (S1 & S2) are ON now current to go from positive terminal to output (positive pole). When 0V output required, S2 and S3 are enabled and a current path goes through either of the clamping diodes to the mid point of capacitors output. For getting −Vdc/2 at the output, switches bottom two (S3 and S4) are to be turned ON, so that the current will flow from output to negative part of DC link. NPC inverter use PWM techniques to generate a pulse waveform very close to sinusoidal output.

CHAPTER 2

Literature Review

In this research, we have studied several key papers related to fault detection in power inverters, particularly focusing on methods used for detecting faults in NPC (Neutral Point Clamped) inverters. Through our review, we have identified a number of crucial findings and common trends that shape the current state of fault detection techniques.

Firstly, it was evident that traditional methods for fault detection, such as time-domain analysis and frequency-domain analysis, are still widely used. However, these techniques often require extensive manual effort and expert knowledge, which can be time-consuming. Moreover, they are primarily based on mathematical models that can be quite complex, limiting their practical application in real-time systems.

2.1 Method for Detecting an Open-Switch Fault in a Grid-Connected NPC Inverter System

By Ui-Min Choi, Hae-Gwang Jeong, Kyo-Beum Lee, and Frede Blaabjerg

This paper introduces a novel technique for detecting open-switch faults in Neutral-Point-Clamped (NPC) inverters connected to the grid. The method utilizes Concordia current patterns to not only detect faults but also pinpoint the location of the faulty switch. Unlike traditional methods, this approach does not require additional sensors or complex hardware. It effectively detects faults within two fundamental periods by analyzing the variation in phase current patterns. The paper also covers different fault scenarios including faults in switches Sx1 through Sx4 and explains how changes in voltage and current patterns reveal the faulty component. Simulation and experimental validation confirm the robustness and accuracy of the proposed system, making it suitable for real-world high-power applications.

2.2 Diagnostic System for a Multilevel Inverter Using a Neural Network

By Surin Khomfoi and Leon M. Tolbert

This paper presents a diagnostic system based on neural networks for detecting faults in multilevel inverter drives (MLIDs). The challenge of diagnosing faults in complex MLID systems, which contain multiple switching devices, is addressed using multilayer perceptron (MLP) neural networks. The system is trained using inverter output voltage data transformed through fast Fourier transform (FFT), allowing it to classify both the type and the location of faults. The classification accuracy between normal and abnormal states is around 90%, while among various fault types it's about 85%. This approach is non-invasive and does not depend on mathematical modeling, making it highly adaptable and efficient for use in high-power industrial applications where reliability and minimal downtime are critical.

2.3 Ultrafast Transmission Line Fault Detection Using a DWT-Based ANN

By Ahmad Abdullah

In this paper, a new ultrafast method for detecting transmission line faults using Discrete Wavelet Transform (DWT) and Artificial Neural Networks (ANN) is proposed. Unlike conventional protection systems that rely on voltage and current phasors, this method only uses the high-frequency content of local current signals, specifically the two aerial modes. These are processed through wavelet transformation to extract detailed coefficients, which are then used to train an ANN. The technique enables the detection of transients and fault classification with high speed and accuracy—using just one-eighth of a cycle of post-fault data. Additionally, the system can distinguish between faults on the protected line and those on adjacent lines, as well as identify non-fault transients such as lightning strikes or switching events, all without requiring voltage data or communication links.

2.4 Enhanced-Online-Random-Forest Model for Static Voltage Stability Assessment Using Wide Area Measurements

By Heng-Yi Su and Tzu-Yi Liu

This paper introduces an advanced real-time framework for assessing static voltage stability using PMU (Phasor Measurement Unit) data and an Enhanced Online Random Forest (EORF) model. Unlike traditional offline models that require full retraining with new data, this method supports online learning through drift detection and online bagging, enabling continuous model adaptation. The model combines multiple decision trees via weighted majority voting to ensure robust and accurate predictions even under varying operating conditions or changes in system topology. The authors test the model on both the IEEE 57-bus system and Taiwan's Taipower system, demonstrating its high speed, accuracy, and adaptability. The system also identifies the most important input variables through variable importance ranking, which helps improve computational efficiency and reliability in voltage collapse prediction.

Table 1: Summary of related research papers

Sr. No.	Paper Title	Authors	Conference	Outcome
1	Method for Detecting an Open-Switch Fault in a Grid-Connected NPC Inverter System	Ui-Min Choi, Hae-Gwang Jeong, Kyo-Beum Lee	IEEE	Studied switch operating conditions under Positive (P), Open (O), and Negative (N) scenarios, analyzed current patterns in healthy and open-circuit faults, evaluated power transfer cases, and developed a fault diagnosis algorithm
2	Fault detection of voltage-source inverter using pattern recognition of the 3D current trajectory	V-Fernao Pires, Tito G.Amaral, Duarte Sousa, G.D. Marques	IEEE	The analysis of 3D space vectors for various faults was conducted. Utilizing a pattern recognition approach, fault locations can be accurately identified.

3	Fault diagnostic system for A multilevel inverter using Neural network.	Surin Khomfoi, Leon M. Tolbert	IEEE	The study analyses a single phase H-bridge inverter Voltage output using FFT to extract features.This feature is used by Neural Network to classify fault with over 85% accuracy.
4	Method for Detecting an Open-Switch Fault in a Grid-Connected NPC Inverter System	Hao Yan, Yumeng Peng, Wenjun Shang, Dongdong Kong	IEEE	The Deep Neural Network utilizes ten signal features extracted from three-phase currents for fault diagnosis. It includes multiple hidden layers and is trained using stochastic gradient descent
5	Method for Detecting an Open-Switch Fault in a Grid-Connected NPC Inverter System	Ui-Min Choi, Hae-Gwang Jeong, Kyo-Beum Lee	IEEE	Studied switch operating conditions under Positive (P), Open (O), and Negative (N) scenarios, analyzed current patterns in healthy and open-circuit faults, evaluated power transfer cases, and developed a fault diagnosis algorithm

6	Fault detection of voltage-source inverter using pattern recognition of the 3D current trajectory	V-Fernao Pires, Tito G.Amaral, Duarte Sousa, G.D. Marques	IEEE	The analysis of 3D space vectors for various faults was conducted. Utilizing a pattern recognition approach, fault locations can be accurately identified.
7	Fault diagnostic system for A multilevel inverter using Neural network	Surin Khomfoi, Leon M. Tolbert	IEEE	The study analyses a single phase H-bridge inverter Voltage output using FFT to extract features.This feature is used by Neural Network to classify fault with over 85% accuracy

2.5 Observation from Literature Review:

A significant shift in fault detection approaches has been the integration of Artificial Intelligence (AI) and Machine Learning (ML) algorithms. These methods have been found to offer substantial advantages over traditional techniques. Notably, AI and ML-based methods, including Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and Random Forests (RF), enable faster diagnosis and can handle large datasets with ease. These methods can identify and classify faults with greater accuracy, especially when the system is exposed to varying operational conditions.

In particular, studies involving wavelet-based ANNs have demonstrated promising results in detecting high-frequency current transients, a key indicator of faults in inverters. These methods offer ultrafast fault detection and have the ability to adapt to dynamic system conditions. Additionally, FFT (Fast Fourier Transform) techniques, when applied to single-phase inverters, have enhanced the classification capabilities of neural networks, although their applicability is generally limited to certain inverter configurations.

One of the key observations from the literature is that while several studies have integrated advanced techniques such as the Concordia Transformation to improve feature extraction in NPC inverters, these methods still face challenges under variable load conditions. In particular, many of the existing algorithms struggle to generalize across different types of inverters, such as two-level and three-level inverters, due to their sensitivity to load changes.

This gap in the research highlights the need for the development of new algorithms that are not only accurate and robust but also versatile enough to operate across a variety of inverter types and conditions. As such, the study of hybrid models that combine the strengths of different algorithms, such as Random Forests and XGBoost, appears to be a promising direction for future work. These models have the potential to provide better performance and improved fault detection capabilities, particularly when tested with unseen data.

In conclusion, our review of the literature underscores the importance of integrating advanced machine learning techniques for fault detection in NPC inverters and similar power electronic systems. The research points toward a future where AI and ML can provide more efficient, reliable, and real-time fault detection, ultimately contributing to the overall performance and longevity of inverter-based systems.

CHAPTER 3

Current Waveform Analysis

Current waveform analysis is a fundamental step in diagnosing faults in power electronic inverters. By examining the phase current behaviour under different switching conditions and fault scenarios, critical patterns and signatures emerge that help in identifying the location and type of faults. This section elaborates on the effect of open-circuit faults on the current waveform in both 2-level and 3-level NPC inverters.

3.1. Operating States in Inverters:

Power electronic inverters operate by switching semiconductor devices (like IGBTs or MOSFETs) to alternate current flow paths. The number of available output voltage levels defines whether an inverter is 2-level or 3-level. These switching states directly influence the output waveform shape and are crucial for understanding how faults distort them.

a. 3-Level NPC Inverter:

A 3-level NPC inverter has four switches per leg (Sx1 to Sx4), allowing three operational modes: Positive (P), Zero (O), and Negative (N). Each switching combination produces a different pole voltage VxO, as shown in Table I.

Table 2: Operating States of the Switch and Pole Voltage for a 3-Level Inverter

Work Modes	**Switching States (x = a, b, c)**				**Pole Voltage**
	Sx1	Sx2	Sx3	Sx4	V x O
P	on	on	off	off	+Vdc/2
O	off	on	on	off	0
N	off	off	on	on	-Vdc 2

b. 2-Level Inverter:

In contrast, the 2-level inverter uses only two switches per phase leg (Sx1 and Sx2). Its operation is limited to two states: Positive and Negative.

Table 3: Operating States of the Switch and Pole Voltage for a 2-Level Inverter

Work Modes	Switching States (x=1,2,3) Sx1	Sx2	Output Voltage VxO
P	on	off	+Vdc/2
N	off	on	-Vdc 2

3.2. Impact of Open-Circuit Faults on Switching States:

When an open-circuit fault occurs in any of the switches, the expected conduction path is disrupted, resulting in missing current segments or distortion in output waveform. These faults cause:

- Reduction in output phase current
- Distorted waveform shape
- Abnormal zero crossings
- Phase imbalance

In 3-level inverters:

- **Sx1 fault:** Affects P-mode, reduces positive current.
- **Sx2 fault:** Affects P and O modes.
- **Sx3 fault:** Affects O and N modes.
- **Sx4 fault:** Affects N-mode, reduces negative current.

3.3. Observed Distortions in Current Waveforms:

Open-circuit faults in inverters lead to distinct and measurable distortions in the phase current waveforms. These distortions arise due to the interruption of current paths when one or more switches fail to conduct. The nature of the distortion depends on the inverter topology (2-level or 3-level), the location of the fault, and the corresponding operating mode of the inverter.

A) Faults in 2-Level Inverter:

A 2-level inverter consists of two switches per leg (Sx1 and Sx2) that alternate to generate positive and negative voltage levels across the load. During healthy operation, the output current waveform for each phase appears as a near-perfect sine wave.

- **Sx1 (upper switch) open-circuit fault:** The positive half-cycle of the current is interrupted. The switch fails to conduct when it should, resulting in a clipped or flattened positive region in the waveform.
- **Sx2 (lower switch) open-circuit fault:** The negative half-cycle of the current is suppressed or missing entirely due to the inability of the lower switch to complete the conduction path.

These effects are illustrated in Figure 2, where the A-phase current under S11 and S12 faults demonstrates asymmetric and truncated waveform characteristics. These signatures are clear and consistent, making 2-level inverter faults easier to detect based on current waveform shape alone.

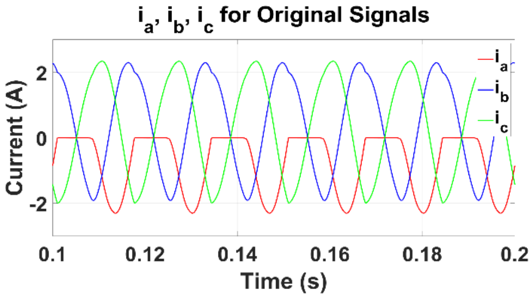
B) Faults in 3-Level NPC Inverter:

A 3-level NPC inverter features four switches per leg (Sx1 to Sx4), which generate three distinct voltage levels: +Vdc/2 (P-mode), 0 (O-mode), and –Vdc/2 (N-mode). The complexity of the switching arrangement introduces more nuanced current distortions under fault conditions.

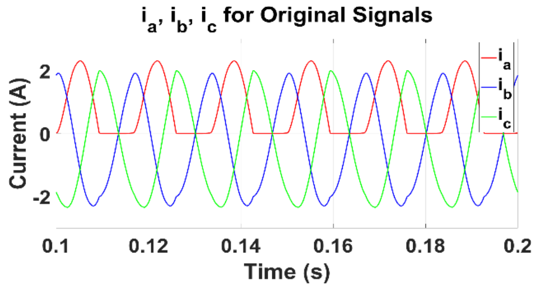
- **Sx1 fault:** Disrupts the Positive (P) mode. The inability of the top switch to conduct reduces the positive peak of the output current.
- **Sx2 fault:** Affects both P-mode and Zero (O) mode. The resulting waveform shows partial positive current and irregular zero-level transitions.
- **Sx3 fault:** Affects Zero (O) mode and Negative (N) mode. The waveform exhibits asymmetry and decreased negative current.
- **Sx4 fault:** Directly impacts the Negative (N) mode, suppressing the negative half-cycle of the waveform.

These waveform changes are depicted in Figure 3, which shows the A-phase current under all four switch faults. The faults introduce sharp discontinuities, slope changes, and cycle asymmetries. These patterns are fault-specific and phase-specific, allowing for accurate identification and classification.

The analysis of these waveform distortions provides the foundational data used in the later stages of fault diagnosis. However, due to the influence of load variations on raw current values, these waveforms are not ideal for load-independent fault detection. Therefore, they are further transformed using the Power-Invariant Clarke Transform (discussed in Section IV), enabling consistent feature extraction across varying operating conditions.

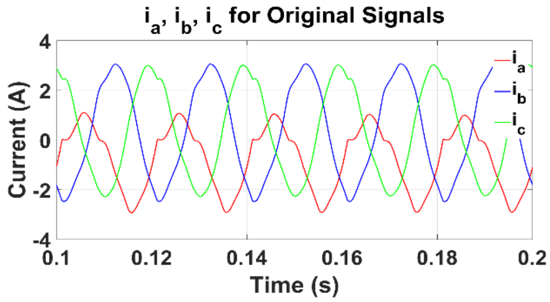


(a) S11 fault

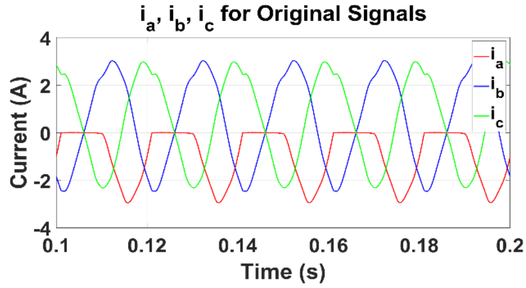


(b) S12 fault

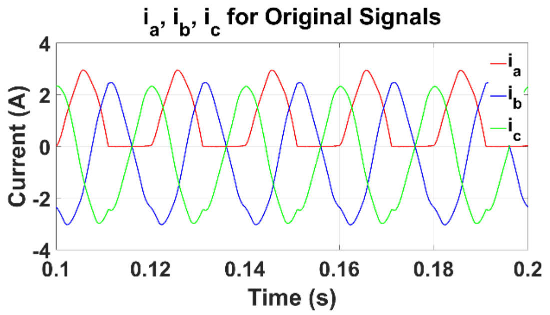
Fig.2. Plot for A-phase fault (2-lvl Inverter).



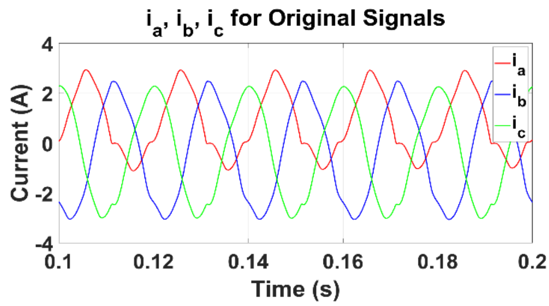
(a) S11 fault



(b) S12 fault



(c) S13 fault



(d) S14 fault

Fig.3. Plot for A-phase fault (3-lvl Inverter).

3.4. Switching Function and Output Voltage Expression:

The inverter's output phase voltage VK is directly linked to the switching state via:

$$V_K = \frac{Vdc}{2} S_K$$

Where Sk is the switching function for each phase k = a,b,c defined as:

$$S_K = \begin{cases} 1 & if\ in\ P - mode \\ 0 & if\ in\ O - mode \\ -1 & if\ in\ N - mode \end{cases}$$

This expression relates the fault-induced disruptions in switching state to their effect on current flow paths and observed voltage levels.

3.5. Significance in Fault Diagnosis:

The distortions described above are key diagnostic features. They:

- Help identify the faulty switch based on missing current segments
- Reveal the phase where the fault has occurred
- Provide input patterns for machine learning-based classifiers.

CHAPTER 4

Feature Extraction Using Concordia Transform

The Concordia transform is a mathematical technique used in the analysis of three-phase electrical systems, particularly for converting three-phase signals into two-phase signals that are easier to analyze and process. It was introduced by CIGRÉ (International Council on Large Electric Systems) as an alternative to other transformations like the Clarke and Park transforms. The primary goal of the Concordia transform is to reduce the complexity of analyzing three-phase systems by representing them in a two-phase coordinate system, making fault detection and analysis more efficient. Unlike the Clarke and Park transforms, which require knowledge of both the phase voltages and the reference frame, the Concordia transform focuses on simplifying the calculation of unbalanced systems, making it particularly useful in systems with asymmetric load or fault conditions. Its uniqueness lies in its ability to directly measure the negative and zero-sequence components, which are essential for fault diagnosis in power systems. This transform finds applications in various fields, including electric power systems, motor drives, and fault detection in power electronics.

Fault diagnosis of multi-level inverters requires robust feature extraction techniques that can adapt to varying load conditions. To achieve this, the Concordia transformation is applied to process three-phase AC currents, converting them into a two-phase system. This transformation ensures that the extracted features remain consistent across different loads, making them ideal for fault diagnosis.

4.1 Mathematical Representation

The Concordia transform converts three-phase currents into a two-dimensional system, reducing load dependence and enabling effective feature extraction for fault diagnosis. The transform is expressed mathematically as:

$$\begin{bmatrix} i_{\propto} \\ i_{\beta} \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

This scaling factor $\sqrt{2/3}$ ensures that the total power is preserved during the transformation.

4.2 Expression of Phase Currents

For a balanced three-phase system, the phase currents are as follows:

ia $= I_m \sin(\omega t)$, ib $= I_m \sin(\omega t - 2\pi/3)$, ic $= I_m \sin(\omega t + 2\pi/3)$

where: Im : Amplitude of the current, ω: Angular frequency of the AC waveform, t: Time variable. The condition 'ia + ib + ic = 0' holds for balanced systems, simplifying calculations for transformation consistency.

4.3 Slope of Current Trajectory

In the two-phase frame obtained from the power-invariant Clarke transform, the current trajectory forms patterns that are useful for fault diagnosis. The slopes obtained are essential for fault detection. The slopes are calculated as:

$$\psi_1 = \frac{i_\propto(t)}{i_\beta(t)}$$

$$\psi_2 = \frac{i_\propto(t) - i_\propto(t - T)}{i_\beta(t) - i_\beta(t - T)}$$

Here T is the sampling interval. To achieve load independent fault diagnosis these slopes are calculated, as these slopes depend solely on time and sampling interval. This trains the model to be load independent.

4.4 Extended Derivations for Individual Phases

The Clarke transform for individual phase is shown below:

For phase A:

$$i_{A\propto} = \sqrt{\frac{3}{2}}\, i_a$$

$$i_{A\beta} = \sqrt{2}\, i_b + \frac{i_a}{\sqrt{2}}$$

$$\psi_{A1} = \frac{i_{A\propto}(t)}{i_{A\beta}(t)}$$

$$\psi_{A2} = \frac{i_{A\propto}(t) - i_{A\propto}(t - T)}{i_{A\beta}(t) - i_{A\beta}(t - T)}$$

For phase B:

$$i_{B\propto} = \sqrt{\frac{3}{2}}\, i_b$$

$$i_{B\beta} = \sqrt{2}\, i_c + \frac{i_b}{\sqrt{2}}$$

$$\psi_{B1} = \frac{i_{B\propto}(t)}{i_{B\beta}(t)}$$

$$\psi_{B2} = \frac{i_{B\propto}(t) - i_{B\propto}(t - T)}{i_{B\beta}(t) - i_{B\beta}(t - T)}$$

For phase C:

$$i_{C\propto} = \sqrt{\frac{3}{2}}\, i_c$$

$$i_{C\beta} = \sqrt{2}\, i_a + \frac{i_c}{\sqrt{2}}$$

$$\psi_{C1} = \frac{i_{C\propto}(t)}{i_{C\beta}(t)}$$

$$\psi_{C2} = \frac{i_{C\propto}(t) - i_{C\propto}(t - T)}{i_{C\beta}(t) - i_{C\beta}(t - T)}$$

The slope derivatives ψA2, ψB2, ψC2 are essential features for the fault diagnosis.

CHAPTER 5

Machine Learning Applications in Fault Diagnosis

In recent years, Machine Learning (ML) has emerged as a transformative tool in the field of fault diagnosis, offering the ability to learn complex patterns from large datasets without requiring explicit programming rules. In the context of power electronic inverters, especially multi-level topologies, fault diagnosis becomes challenging due to nonlinear behaviours, load variations, and subtle waveform distortions. Traditional rule-based or analytical methods often fall short in such scenarios. By contrast, ML models can automatically extract meaningful features from current waveforms, learn the relationship between these features and fault conditions, and accurately classify fault types such as open-circuit faults. In this work, a supervised ML approach is adopted, where historical labelled data generated through simulations is used to train classification models like Random Forest, XGBoost, and Artificial Neural Networks. These models are capable of identifying fault locations with high accuracy, even under varying load conditions, thereby enabling a robust and load-independent diagnostic system.

5.1 Introduction to Machine Learning Lifecycle

The development of an ML algorithm follows a structured sequence of steps to ensure effective learning and generalization. The first step is problem definition, where the objective of the model is clearly stated. This is followed by data collection, where relevant data is gathered from sensors, experiments, or simulations. The data preprocessing stage involves cleaning the data, handling missing values, scaling, and normalizing features to ensure consistency. Next, feature engineering is carried out to select, extract, or transform input variables that are most relevant to the learning task. After this, an appropriate ML model is chosen based on the nature of the problem—classification, regression, or clustering.

The selected model is then subjected to training, where it learns patterns from a training dataset, and its performance is evaluated using a validation set. Model evaluation is done using metrics such as accuracy, precision, recall, or mean squared error, depending on the task. To further enhance performance, hyperparameter tuning is performed using techniques like grid search or random search. Once an optimal model is obtained, it is deployed into a real-time environment where it can make predictions or classifications. Finally, continuous monitoring and maintenance are essential to track the model's performance over time and retrain it if necessary due to changes in input data patterns or operating conditions.

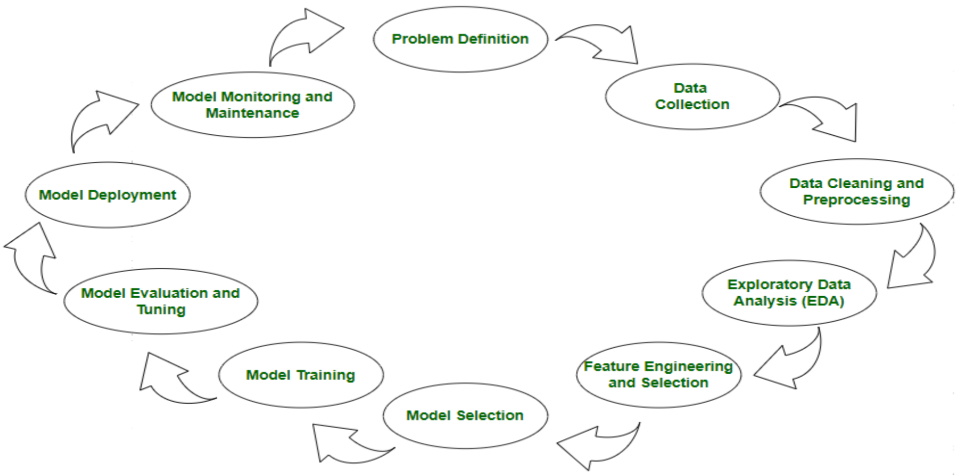


Figure 4: Machine Learning life cycle.

5.2 Overview of Machine Learning Techniques

Machine Learning (ML) techniques are generally categorized into three types: supervised, unsupervised, and reinforcement learning. Supervised learning involves training models on labeled data, making it ideal for classification tasks such as identifying inverter faults. Unsupervised learning finds patterns in unlabeled data and is mainly used for clustering or anomaly detection. Reinforcement learning is based on feedback through rewards and penalties and is typically applied in decision-making environments. In this project, supervised learning is employed, as the inverter fault classification requires a model trained on data with known fault labels.

a. Supervised Learning

Supervised learning is a type of machine learning where the algorithm learns from a labeled dataset. Each training example consists of input features and a corresponding label, allowing the algorithm to learn the relationship between them. Over time, the model adjusts its internal parameters to minimize the error in predicting outputs from new, unseen inputs. This approach is particularly well-suited for classification problems, such as identifying fault types in power electronic inverters. Since the dataset in this project contains clearly labelled fault conditions, supervised learning enables the development of a reliable model capable of recognizing patterns associated with different fault scenarios, even under varying operational conditions.

- **Examples of Algorithms:**

i. **Linear Regression :** Used for predicting continuous values (e.g., temperature forecasting, house price prediction).

ii. **Logistic Regression :** Used for binary classification (e.g., spam vs. non-spam email).

iii. **Decision Trees :** Simple tree-based model for both classification and regression tasks.

iv. **Random Forest :** An ensemble of decision trees used for robust classification (e.g., fault detection in electrical systems).

v. **Support Vector Machine (SVM) :** Effective for high-dimensional classification tasks (e.g., face recognition).

vi. **K-Nearest Neighbors (KNN) :** Classifies data based on similarity to nearby data points (e.g., handwriting recognition).

vii. **Naive Bayes :** Based on probability theory; useful for text classification and sentiment analysis.

viii. **Gradient Boosting Machines (e.g., XGBoost) :** Advanced boosting algorithm for high accuracy in classification (e.g., medical diagnosis, fault classification).

ix. **Artificial Neural Networks (ANNs) :** Mimics the human brain and excels in complex pattern recognition (e.g., speech and image recognition).

b. Unsupervised Learning

Unsupervised learning is a machine learning approach where the model is trained on data without any explicit labels or predefined outputs. Instead of learning from known answers, the algorithm tries to identify hidden patterns, structures, or groupings within the data. This technique is commonly used for clustering, dimensionality reduction, and anomaly detection. In the context of engineering systems, it can be useful for uncovering unusual behavior or detecting unknown fault conditions when labeled data is not available. Although unsupervised learning is not directly applied in this project, it provides valuable insights in exploratory data analysis and can serve as a foundation for semi-supervised or hybrid diagnostic systems in the future.

- **Examples of Algorithms:**

i. **K-Means Clustering :** Partitions data into a predefined number of clusters by minimizing the distance between points and their cluster centroids (e.g., grouping similar sensor measurements for pattern discovery).

ii. **Hierarchical Clustering :** Builds a tree-like structure of nested clusters without needing to predefine the number of groups (e.g., revealing hierarchical relationships in machinery vibration data).

iii. **Principal Component Analysis (PCA) :** Reduces high-dimensional datasets to a smaller set of uncorrelated variables that capture most of the variance (e.g., compressing waveform features for visualization and noise reduction).

iv. **Autoencoders :** Neural networks trained to reconstruct their input, learning a compact representation in the hidden layers; often used for anomaly detection (e.g., spotting unusual inverter behaviour by reconstruction error).

c. Reinforcement Learning

Reinforcement learning involves an agent that interacts with its environment, making sequential decisions to maximize a notion of cumulative reward. Rather than learning from labeled examples, the agent explores different actions, receives feedback in the form of rewards or penalties, and gradually refines its strategy—or policy—to achieve the best long-term outcome. Common algorithms include Q-learning, SARSA, and more advanced deep reinforcement learning methods like Deep Q-Networks (DQN). Although reinforcement learning is not directly applied to static fault classification in this study, it holds promise for adaptive control and self-healing strategies in power electronic systems, where an inverter could learn optimal fault-tolerant actions through continual interaction and feedback.

- **Examples of Algorithms:**

i. **Q-Learning :** An off-policy algorithm that learns the value of state-action pairs

(Q-values) through trial-and-error, enabling an agent to derive an optimal policy; for example, it could be used to select inverter switching strategies that minimize power losses under varying load conditions.

ii. **Deep Q-Networks (DQN) :** Extends Q-Learning by using a deep neural network to approximate Q-values in high-dimensional state spaces (e.g., feeding Clarke-transform features into a DQN to learn fault-tolerant control policies for multi-level inverters).

iii. **Policy Gradient Methods :** Directly optimize the policy by estimating gradients of the expected cumulative reward with respect to policy parameters; these methods can handle continuous actions, such as tuning PWM duty cycles for adaptive control and self-healing in power electronic systems.

5.3 Description of Machine Learning models used in project

a. Random Forest

Random Forest is an ensemble learning method that builds a large number of decision trees and combines their predictions to improve accuracy and control overfitting. Each tree is trained on a random subset of the data (bagging) and considers only a random subset of features at each split, which introduces diversity and reduces correlation among trees. During inference, the forest aggregates individual tree votes (for classification) to produce a final decision, making it robust to noise and outliers. Random Forests also provide measures of feature importance, helping to identify which input signals or derived features—such as Clarke-transform slopes—contribute most to fault discrimination. In fault diagnosis of power electronic inverters, this robustness and interpretability make Random Forests particularly effective at handling complex, nonlinear patterns in current waveforms across varying load conditions.

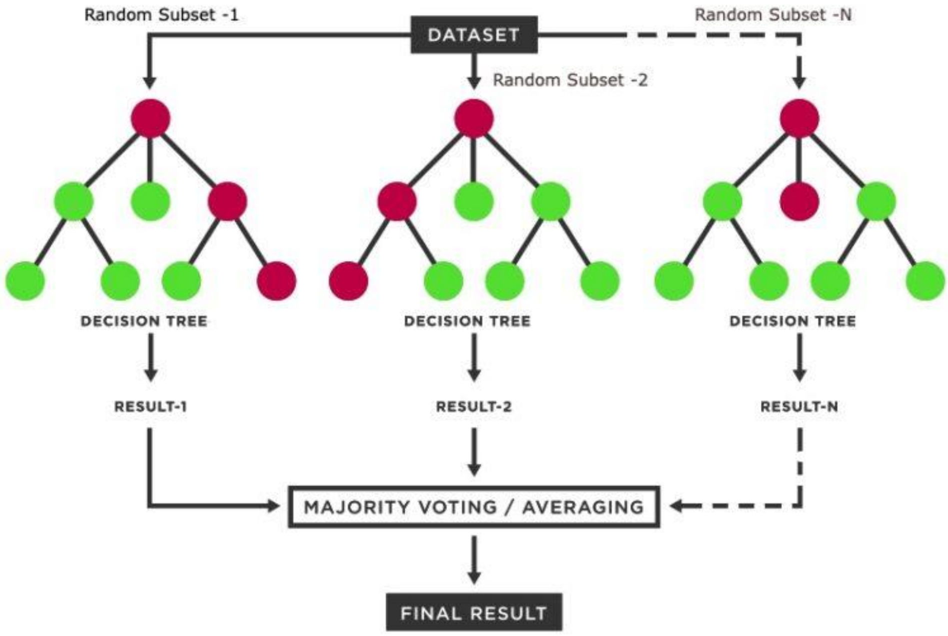


Figure 5: Random Forest Structure

- **Working :**

Random Forest works by constructing a collection of decision trees, each trained on a random subset of the data, and then combining their outputs to make predictions. Here's how it operates step-by-step:

i. **Data Sampling (Bootstrapping) :** Random Forest starts by creating multiple datasets through a process called bootstrapping, where random samples are drawn from the original dataset (with replacement). This means some data points may appear multiple times in one subset, while others may be left out.

ii. **Building Multiple Decision Trees :** For each bootstrapped dataset, a decision tree is built. Each tree is trained independently using a recursive process where the data is split based on the best feature at each node, aiming to separate the data into increasingly pure subsets. The decision on which feature to split on is made by selecting the one that best improves the classification or regression criterion (such as Gini impurity for classification or variance reduction for regression).

iii. **Random Feature Selection :** To further increase diversity among trees, Random Forest introduces a second level of randomness by selecting a random subset of features at each node split. This ensures that each tree explores different aspects of the data and reduces the correlation between trees.

iv. **Voting for Classification or Averaging for Regression:** Once all the decision trees are trained, Random Forest makes predictions by aggregating the individual tree predictions. For classification tasks, each tree votes for a class, and the majority vote determines the final prediction. For regression tasks, the average of all tree predictions is used as the final output.

v. **Final Output and Accuracy :** By averaging the predictions of many trees, Random Forest reduces overfitting and improves the model's generalization to unseen data. This makes it more accurate and robust than individual decision trees, especially in noisy or complex datasets.

In fault diagnosis, such as for power electronic inverters, the ensemble of trees ensures that the model is resilient to noise in the data and can handle variations in operational conditions, making it highly effective for classifying fault types even under diverse scenarios.

- **Advantages:**

i. **High accuracy and robustness :** By aggregating the predictions of many uncorrelated trees, Random Forest often achieves better performance and generalization than individual decision trees.

ii. **Resilience to overfitting :** The combination of bootstrapping and random feature selection at each split reduces the chance that the model will overfit noisy or idiosyncratic patterns in the training data.

iii. **Built-in feature importance :** Random Forest computes measures of variable importance, helping you identify which features (e.g., Clarke-transform slopes) are most informative for fault classification.

iv. **Handles high-dimensional and mixed data :** It works well when you have many features, including numerical and categorical variables, without requiring extensive preprocessing.

v. **Robust to outliers and missing values :** Individual trees can effectively ignore outliers, and the ensemble nature of the forest mitigates their influence. It also provides strategies for dealing with missing data.

vi. **Minimal hyperparameter tuning :** With sensible defaults (number of trees, max depth, etc.), Random Forests often perform well without extensive grid searches.

- **Disadvantages:**

i. **Reduced interpretability:** While individual trees are easy to interpret, the aggregated forest behaves like a "black box," making it harder to explain specific predictions.

ii. **Increased computational cost:** Training and inference require building and querying dozens or hundreds of trees, which can be resource-intensive in terms of CPU time and memory.

iii. **Slower prediction time:** Especially with large forests, real-time applications may suffer from latency as each tree must cast its vote.

iv. **Bias with imbalanced classes:** If fault classes are highly imbalanced, Random Forest may favour the majority class unless corrective measures (e.g., class weights, resampling) are applied.

v. **Limited extrapolation:** Like other tree-based methods, Random Forest cannot reliably predict outcomes outside the range observed in the training data.

b. XG Boost (Gradient Boosting)

XGBoost (Extreme Gradient Boosting) is a powerful ensemble learning technique that builds models in a sequential, additive fashion, where each new decision tree is trained to correct the errors of the combined ensemble so far. It optimizes a regularized objective function that balances the model's fit to the data against its complexity, reducing the risk of overfitting. By computing both first- and second-order gradients of the loss function, XGBoost makes more informed split decisions and converges faster than traditional gradient-boosting methods. Additional enhancements—such as column subsampling, sparse data handling, and parallelized tree construction—make XGBoost exceptionally fast and capable of managing large, high-dimensional datasets. In fault diagnosis for power electronic inverters, XGBoost's ability to capture subtle nonlinear relationships in waveform features, while remaining robust against noise and missing values, translates to highly accurate classification of fault types under varying operational conditions.

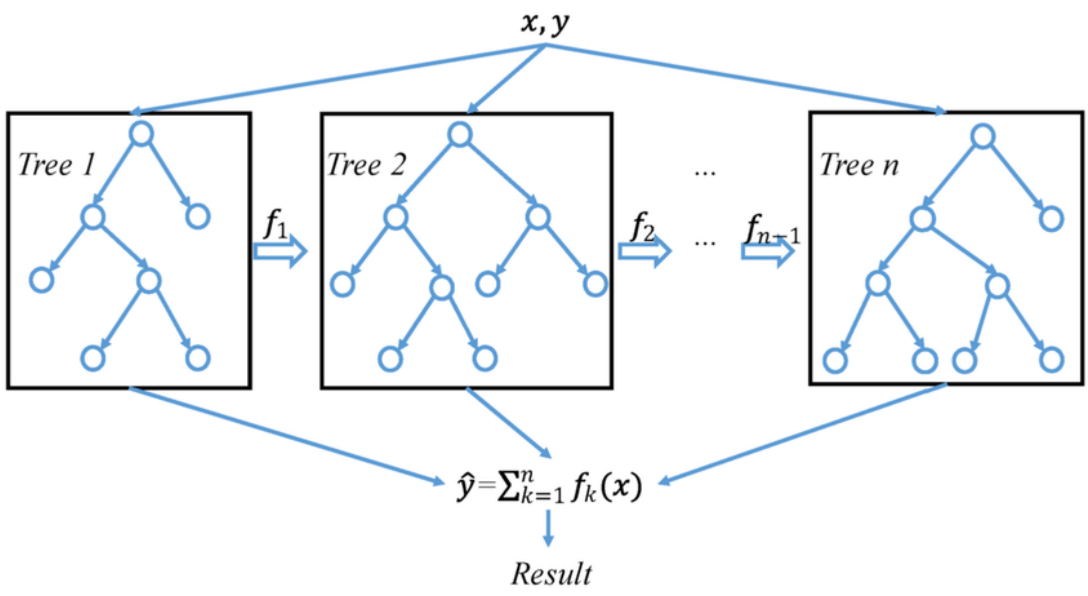


Figure 6: XG Boost Structure

- **Working :**

XGBoost (Extreme Gradient Boosting) is based on the principle of gradient boosting, where new models (trees) are added to an ensemble to correct the residual errors made by the previous models. Here's how XGBoost works step by step:

i. **Initialization :** The process begins by initializing a model with a simple prediction, such as the mean for regression or the log-odds for classification.

ii. **Compute Residuals (Gradients) :** For each sample in the training data, XGBoost calculates the gradient (first derivative) of the loss function with respect to the current model prediction. This represents how much the model's current predictions are off from the true values.

iii. **Construct Trees to Fit Residuals :** A decision tree is built to predict the residuals, i.e., the errors of the model. This tree aims to fit the negative gradient (or pseudo-residuals) and reduce the prediction error. The tree-building process is like a standard decision tree, but with the goal of predicting errors rather than classifying or regressing directly.

iv. **Model Update :** The newly built tree is then added to the existing ensemble, and the predictions are updated by adding the new tree's predictions, scaled by a learning rate (shrinkage factor). This learning rate helps control how much each tree contributes to the final prediction, preventing overfitting.

v. **Regularization :** XGBoost applies L1 (Lasso) and L2 (Ridge) regularization to the tree's leaf weights to avoid overfitting. Regularization penalizes overly complex trees, ensuring a balance between model accuracy and complexity.

vi. **Pruning :** Trees are pruned to avoid overfitting by removing branches that do not provide significant improvements in the objective function. Pruning is done using a technique called maximum depth or by using a threshold that evaluates the gain of a split.

vii. **Iteration :** Steps 2 to 6 are repeated iteratively, with each new tree refining the predictions made by the ensemble. The process continues until a predefined number of trees is reached, or the improvements between iterations become minimal.

viii. **Final Prediction :** Once the model has built all the trees, XGBoost makes predictions by summing the contributions of all the trees. For classification tasks, the output is passed through a sigmoid function to produce a probability, while for regression tasks, the final prediction is the average of all trees' outputs.

This combination of boosting, regularization, and efficient computation makes XGBoost one of the most powerful algorithms for classification and regression tasks, particularly when dealing with large, complex datasets.

- **Advantages:**

i. **High Predictive Accuracy :** XGBoost consistently delivers top-notch performance, often achieving superior accuracy compared to other machine learning models due to its ability to fine-tune errors at each step and minimize bias through gradient boosting.

ii. **Speed and Efficiency :** XGBoost is designed to be computationally efficient. It uses techniques like parallelization and hardware optimization (e.g., using multiple cores), making it faster than many other boosting algorithms, especially with large datasets.

iii. **Regularization :** XGBoost includes L1 (Lasso) and L2 (Ridge) regularization to control model complexity, preventing overfitting. This is a significant advantage over other gradient boosting methods, which typically lack this feature.

iv. **Handling Missing Data :** XGBoost can handle missing values in the dataset directly, without requiring imputation or removal of missing data. It effectively learns how to treat missing values during the training process, improving model robustness.

v. **Robust to Overfitting :** The built-in regularization, along with careful tree pruning and feature selection, helps XGBoost avoid overfitting, especially when the dataset has high dimensionality or noise.

vi. **Feature Importance :** XGBoost provides an easy-to-interpret ranking of feature importance, helping users understand which features (e.g., Clarke-transform slopes or current waveforms) contribute most to the model's decisions, aiding in feature engineering and model transparency.

- **Disadvantages:**

i. **Complex Hyperparameter Tuning :** XGBoost has a large number of hyperparameters that require careful tuning, such as learning rate, tree depth, and regularization parameters. Finding the optimal combination can be time-consuming and may require a grid search or random search.

ii. **Memory Intensive :** While XGBoost is efficient, it can be memory-intensive, especially when working with large datasets or deep trees. The model may consume significant computational resources, particularly during training.

iii. **Limited Interpretability :** Like other tree-based models, XGBoost's ensemble of decision trees makes it difficult to interpret specific decisions. While feature importance can be derived, understanding the reasoning behind individual predictions is challenging, making it less transparent compared to simpler models.

iv. **Overfitting with Insufficient Regularization :** Although XGBoost includes regularization, the model is still prone to overfitting if the hyperparameters are not well-tuned. In particular, if too many trees are added or if regularization terms are not carefully adjusted, the model can become overly complex and lose generalization.

v. **Slow Prediction Time with Large Ensembles :** Since XGBoost relies on the predictions of many trees, the inference time can be slower when dealing with large numbers of trees or when making predictions in real-time, especially in resource-constrained environments.

c. Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) are a class of machine learning models inspired by the structure and function of the human brain. They consist of layers of interconnected nodes, or neurons, each performing a simple computation. The network typically has an input layer, one or more hidden layers, and an output layer. The neurons in each layer are connected by weighted edges, and each connection has a weight that adjusts during the training process to minimize prediction errors. ANNs are capable of learning complex, non-linear relationships between inputs and outputs, making them particularly useful for tasks such as image recognition, natural language processing, and fault diagnosis in systems with intricate patterns. During training, ANNs use backpropagation, an algorithm that computes gradients and adjusts weights using optimization techniques like gradient descent. Although ANNs can model highly complex relationships, they require large amounts of data and computational power to train effectively, and their "black-box" nature makes them less interpretable than simpler models like decision trees. Despite this, their ability to adapt and learn from vast amounts of data makes them highly effective for fault detection in power electronic inverters, where the patterns in current waveforms are intricate and non-linear.

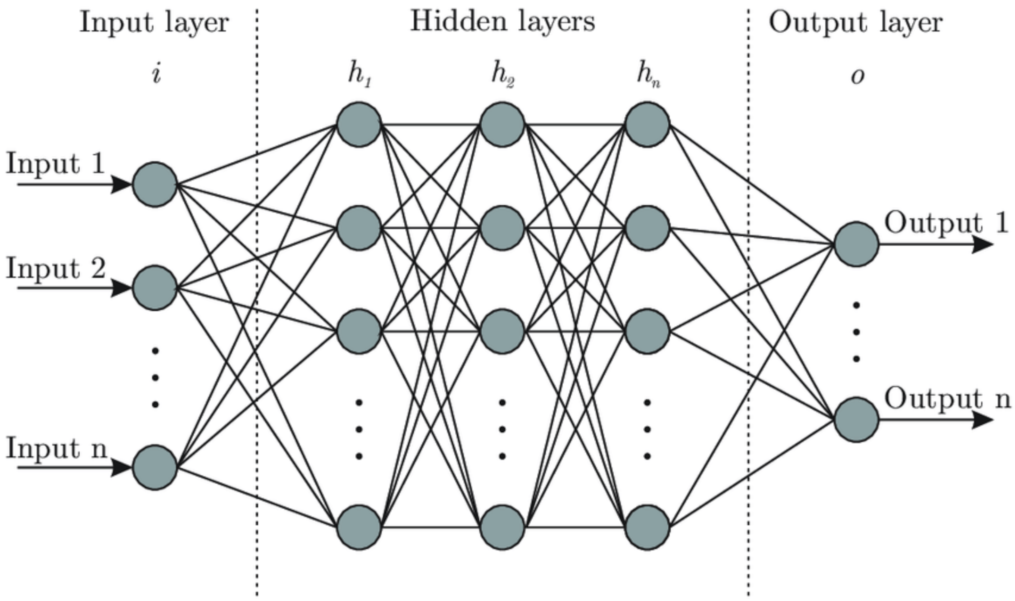


Figure 7: ANN Structure

- **Working:**

Artificial Neural Networks learn to map inputs to outputs through a process of forward and backward passes across interconnected layers of neurons. Here's how they operate step by step:

i. **Network Initialization:** Define the architecture by specifying the number of layers (input, hidden, output) and the number of neurons in each layer. Initialize all weights and biases to small random values (often using methods like Xavier or He initialization) to break symmetry.

ii. **Forward Propagation**
   - **Input Layer:** Feed the raw input features (e.g., Clarke-transform slopes) into the network.
   - **Hidden Layers:** At each neuron, compute a weighted sum of its inputs plus a bias term, then apply an activation function (e.g., ReLU, sigmoid, or tanh) to introduce nonlinearity.
   - **Output Layer:** Produce final scores or probabilities for each class by applying a suitable activation (e.g., softmax for multi-class classification).

iii. **Loss Computation:** Compare the network's output to the true labels using a loss function—commonly cross-entropy loss for classification. The loss value quantifies how far the model's predictions deviate from the actual labels.

iv. **Backward Propagation:** Use the chain rule to compute gradients of the loss with respect to every weight and bias in the network. Starting from the output layer, propagate these gradient values backward through each hidden layer, determining how each parameter contributed to the overall error.

v. **Weight and Bias Update:** Adjust weights and biases in the direction that minimizes loss, typically via gradient descent or one of its variants (e.g., Adam, RMSProp).

vi. **Iteration Over Epochs:** Repeat the forward and backward passes for multiple epochs (full passes through the training data), optionally using mini-batches to balance convergence speed and stability. Monitor training and validation loss to ensure the network is learning effectively and to detect overfitting.

vii. **Inference (Prediction) :** Once training is complete, use the learned weights and biases to perform a forward pass on new, unseen data. The output layer's activations (e.g., highest softmax probability) determine the predicted class, enabling the network to classify fault types in power electronic inverters.

- **Advantages:**

i. **Model Complex Nonlinear Relationships :** ANNs excel at capturing intricate, nonlinear patterns in data, making them well-suited for tasks—like fault diagnosis—where waveform features interact in complicated ways.

ii. **Automatic Feature Learning :** Unlike traditional models that rely on handcrafted features, ANNs can automatically discover and extract relevant representations from raw inputs, reducing the need for manual feature engineering.

iii. **Scalability with Data :** As the size of the training dataset grows, ANNs often continue to improve in performance, leveraging large volumes of data to refine their internal representations.

iv. **Flexibility in Architecture :** A wide variety of network structures (e.g., feedforward, convolutional, recurrent) and activation functions allow ANNs to be tailored to specific problem domains and data modalities.

v. **Robustness to Noise :** When properly regularized (using techniques like dropout or weight decay), ANNs can generalize well and remain resilient to noisy or partially corrupted inputs.

vi. **Versatility Across Tasks :** Beyond classification, ANNs can be applied to regression, sequence modelling, anomaly detection, and more—offering a unified framework for many types of predictive analytics.

- **Disadvantages:**

i. **Require Large Datasets :** ANNs typically require a large amount of data to train effectively. Without enough data, the network may not learn meaningful patterns and could easily overfit to the training set.

ii. **High Computational Cost :** Training deep neural networks can be resource-intensive, requiring significant computational power and memory, especially for large datasets or complex models.

iii. **Lack of Interpretability :** ANNs are often referred to as "black-box" models because it can be difficult to interpret how the network arrives at its decisions. This lack of transparency makes it challenging to explain specific predictions, which is a drawback in safety-critical applications.

iv. **Overfitting Risk :** Without proper regularization techniques, ANNs can easily overfit the training data, especially when the network is too complex relative to the amount of data available.

v. **Sensitive to Hyperparameter Choices :** ANN performance is highly dependent on the selection of hyperparameters (e.g., learning rate, number of layers, and activation functions). Tuning these parameters can be time-consuming and may require trial-and-error or sophisticated optimization techniques.

vi. **Long Training Times :** Training a neural network can take a long time, especially with deep architectures or large datasets, making it less suitable for real-time or resource-constrained applications.

vii. **Risk of Local Minima :** The optimization process can sometimes converge to a local minimum rather than the global minimum, especially when the loss function is complex and non-convex. While techniques like stochastic gradient descent help mitigate this, the problem still exists.

CHAPTER 6

Simulation

6.1 Setup and Methodology:

To validate the proposed fault diagnosis framework, MATLAB Simulink simulations were conducted on a 3-level Neutral-Point-Clamped (NPC) inverter. The simulation model incorporates 12 IGBT switches, 6 freewheeling diodes, a DC power supply of 200 V, and a purely resistive load. The inverter was configured to operate at a switching frequency of 12.8 kHz with an output frequency of 50 Hz. Faults were artificially introduced by simulating open-circuit conditions in specific switches of the inverter.

The simulations were performed under varying load resistances 10 Ω, 30 Ω, and 40 Ω to test the robustness of the proposed feature extraction and classification methods under different operating conditions. The three-phase current signals obtained from these simulations were processed using the Power-Invariant Clarke Transform to extract the essential fault features particularly the slope-based features ($\psi1$ and $\psi2$). These features were then used to train and evaluate multiple machine learning models, as discussed in subsequent sections.

6.2 MATLAB Simulink Model

The MATLAB Simulink model of the Neutral-Point-Clamped (NPC) inverter was simulated using 12 IGBT switches,6 freewheeling diodes, a DC power supply, and a resistive load. The model represents a three-level inverter, designed to convert DC to three-phase AC, allowing for performance analysis under various operating conditions. Each phase (A, B, and C) is controlled via gate pulses and includes fault injection mechanisms for open-circuit faults at different switches. The circuit structure allows clear isolation of each switching device and mimics the behavior of real-world NPC inverters under faulted and healthy scenarios. The simulation runs in discrete-time mode with a fixed step size of **1µs (1e-6 s)**, which is appropriate for high-frequency switching events (such as the inverter's 12.8 kHz switching frequency). This time step ensures accurate capture of switching transients and current distortions due to faults.

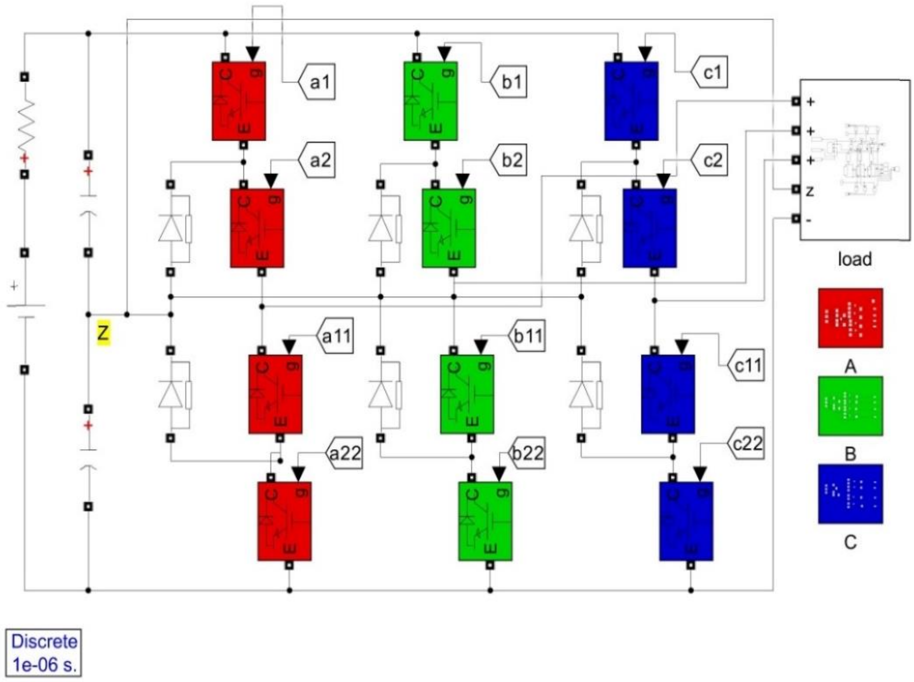


Figure 8: Model of 3 level Neutral-Point-Clamped Inverter

Table 4: System Parameters

Parameters	**Value**
Output Frequency (Hz)	50
DC Voltage (V)	200
Switching Frequency (kHz)	12.8
Load Resistance (Ω)	10

6.3 Slope Features Under Varying Loads:

The following figures (Figures 9 to 16) present a visual analysis of the current waveforms and derived slope features corresponding to various open-circuit faults (S11 to S14) occurring in a three-phase inverter system. These plots are generated for two different load conditions—40 Ω and 30 Ω to demonstrate the effectiveness of the power-invariant Clarke transformation in achieving load-independent fault diagnosis. Each figure consists of three subplots: the first displays the original three-phase currents (ia, ib, and ic), while the second and third depict the transformed slope features (Ψ1 and Ψ2) for each phase. Notably, despite the changes in load, the slope trajectories maintain similar patterns across the corresponding faults, underscoring the robustness of the proposed feature extraction technique. These consistent characteristics form the foundation for accurate fault classification using machine learning algorithms, and visually validate the reliability and adaptability of the diagnostic framework across varying operating conditions.

For 30 Ω Load:

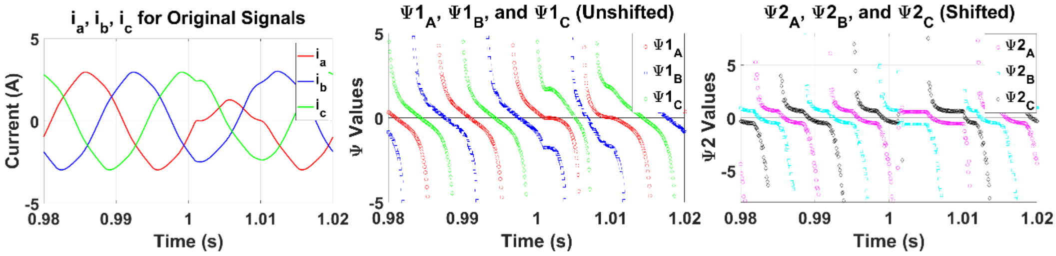


Figure 9: Plots for Sa1 fault with 40 Ω load.

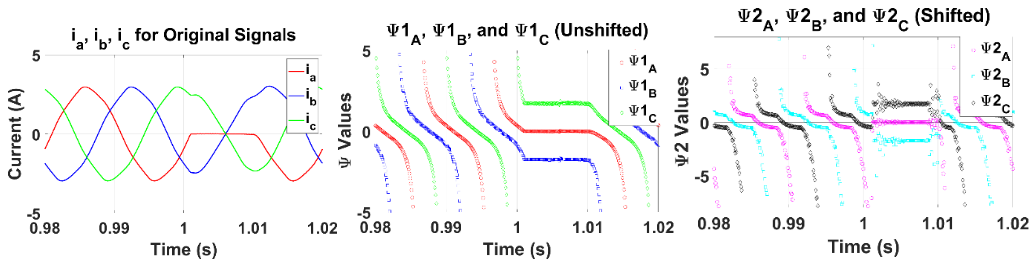


Figure 10: Plots for Sa2 fault with 40 Ω load.

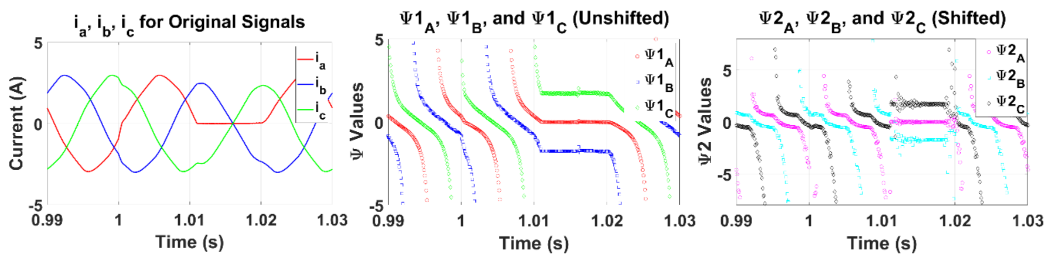


Figure 11: Plots for Sa3 fault with 40 Ω load

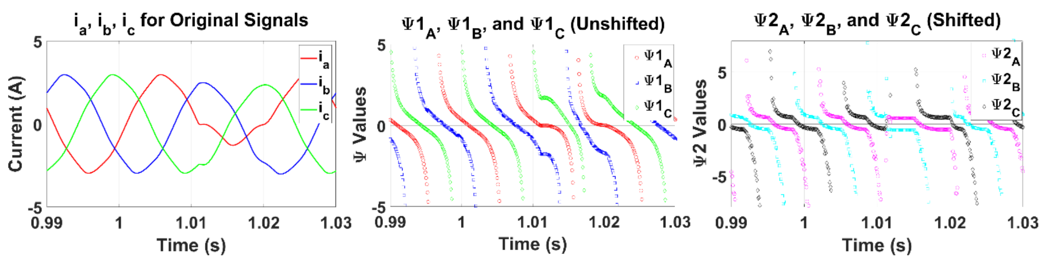


Figure 12: Plots for Sa4 fault with 40 Ω load.

For 40 Ω Load:

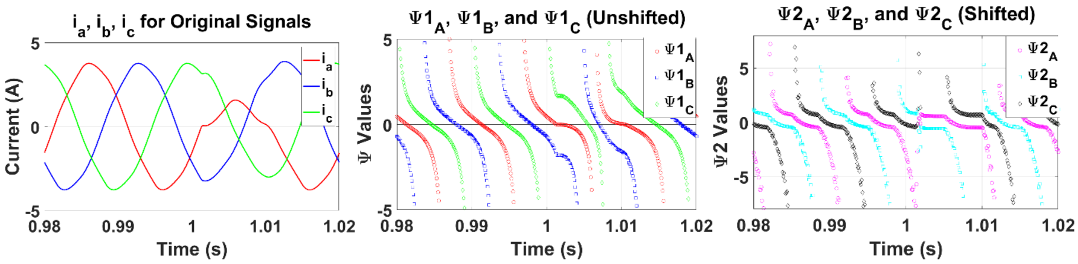


Figure 13: Plots for Sa1 fault with 30 Ω load.

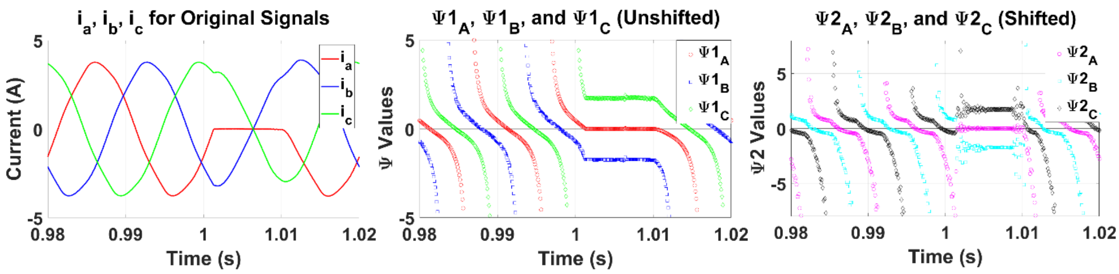


Figure 14: Plots for Sa2 fault with 30 Ω load.

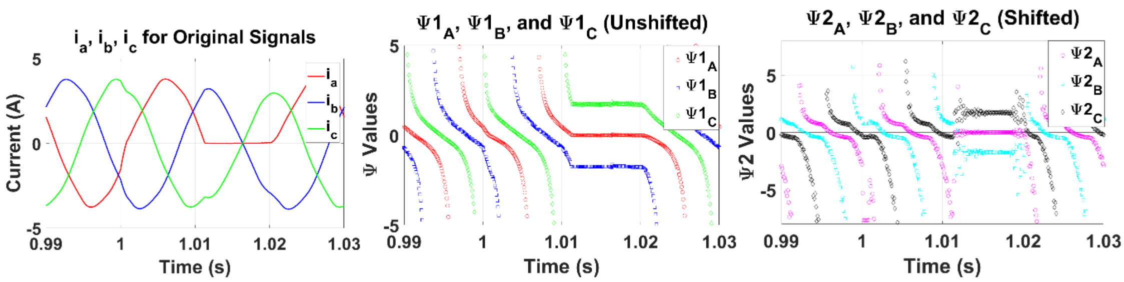


Figure 15: Plots for Sa3 fault with 30 Ω load.

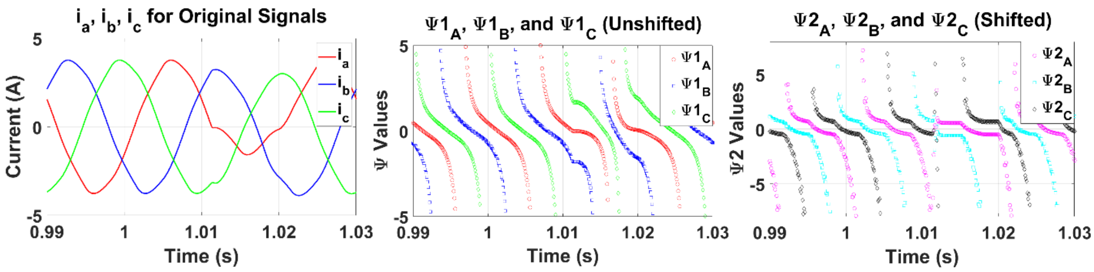


Figure 16: Plots for Sa4 fault with 30 Ω load.

CHAPTER 7

Evaluation of ML Model

In this chapter, we focus on evaluating various machine learning (ML) models used for detecting faults in NPC (Neutral Point Clamped) inverters. The models were trained and tested on datasets extracted from MATLAB Simulink simulations under various load conditions and fault scenarios. The objective was to identify which models offer the highest accuracy and generalization capabilities for fault diagnosis. The evaluation includes: Random Forest (RF), Artificial Neural Networks (ANN), Extreme Gradient Boosting (XGBoost), and a Hybrid RF + XGBoost model. Each model was tested using training and unseen testing datasets to assess its load independence, robustness, and effectiveness in identifying fault types.

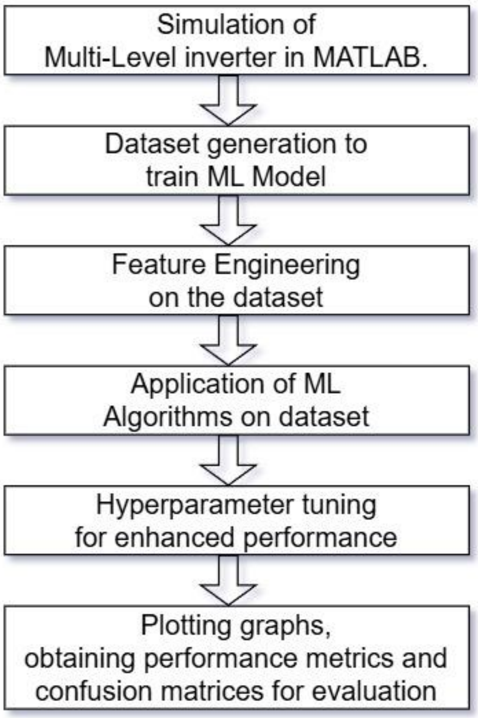


Figure 17: Flow diagram of project implementation.

7.1 Dataset Preparation

The dataset was generated from an independently developed MATLAB Simulink model. The system parameters were as follows, output frequency 50 Hz, the switching frequency is 12.8 kHz, the DC Voltage is 200 V and the load resistances are 10 Ohms, 20 Ohms, 30 Ohms and 40 Ohms. The datasets for each load were obtained separately. Each dataset contained 50,005 rows and 16 features. The first 15 features were used to detect the fault location. The dataset contained one dependent variable 'Y' which was used to label the dataset. The first 15 features of the dataset are as follows:

- Three-phase currents ia, ib, and ic.
- The transformed currents iα and iβ with respect to each phase A, B, and C.
- Slopes of transformed currents Ψ1 and Ψ2 with respect to each phase A, B, and C.

7.2 Steps for Evaluation

The methodology for the research is as follows:

- **Step 1:** Simulation of multi-level inverter in MATLAB.
- **Step 2:** Dataset generation to train ML Model.
- **Step 3:** Feature Engineering on the dataset.
- **Step 4:** Application of ML Algorithms on dataset.
- **Step 5:** Hyperparameter tuning for enhanced performance.
- **Step 6:** Plotting graphs, obtaining performance metrics and confusion matrices for evaluation.

The fault diagnosis classifier integrates knowledge-driven feature transformation with a data-driven approach to achieve robust classification. The features are extracted using the power-invariant Clarke transformation, which converts three phase AC currents into a two-phase system, ensuring consistency across varying loads. Three algorithms are employed for classification: Random Forest (RF), XGBoost, and a Hybrid (RF+XGBoost) model. These models leverage advanced ensemble learning techniques to improve accuracy and generalization across diverse operating conditions.

7.3 Data Preprocessing

The Power-Invariant Clarke Transform processes the three phase currents (ia, ib, ic) into two-phase components ($i\alpha$, $i\beta$). This transformation ensures that the slope of the current trajectory ($\psi 1$, $\psi 2$) remains stable under varying loads. The Interquartile Range (IQR) filter is used for outlier removal from the dataset. This method filters the extremities which cause misclassifications.

7.4 Performance Analysis

The following results were obtained for three different models namely, Random Forests(RFs), Extreme Gradient Boosting(XGBoost) and Hybrid (RF+XGBoost) model. It is observed that, these models perform well even on unseen datasets giving accuracy above 82% for different loads. The models were also tested on 2 Level inverter datasets, it was observed that more than 99% accuracy can be achieved. This shows the adaptability of algorithm to various inverters. A comparative analysis of these models show that satisfactory fault detection can be achieved for different loads as well as for different types of inverters.

Table 5: Model Accuracy on Training Data

Model	**Accuracy on 30 Ohm Training data%**
Random Forest	98.87
XG Boost	98.91
Hybrid (RF+XG Boost)	99.08

Table 6: Model Accuracy on Test Data

Model	**Accuracy on 10 Ohm Data%**	**Accuracy on 20 Ohm Data%**	**Accuracy on 40 Ohm Data%**
RF	83.01	91.96	84.30
XG	83.41	86.01	85.01
RF+XG	82.47	82.12	85.73

Table 7: Performance Metrics for 40 Ohm Dataset (3 Level Inverter)

Model	Accuracy	Precision	Recall	F1 Score
RF	0.8430	0.85	0.82	0.83
XG	0.8501	0.85	0.83	0.84
RF+XG	0.8573	0.86	0.84	0.85

7.5 Model Accuracy for Number of Trees Plot

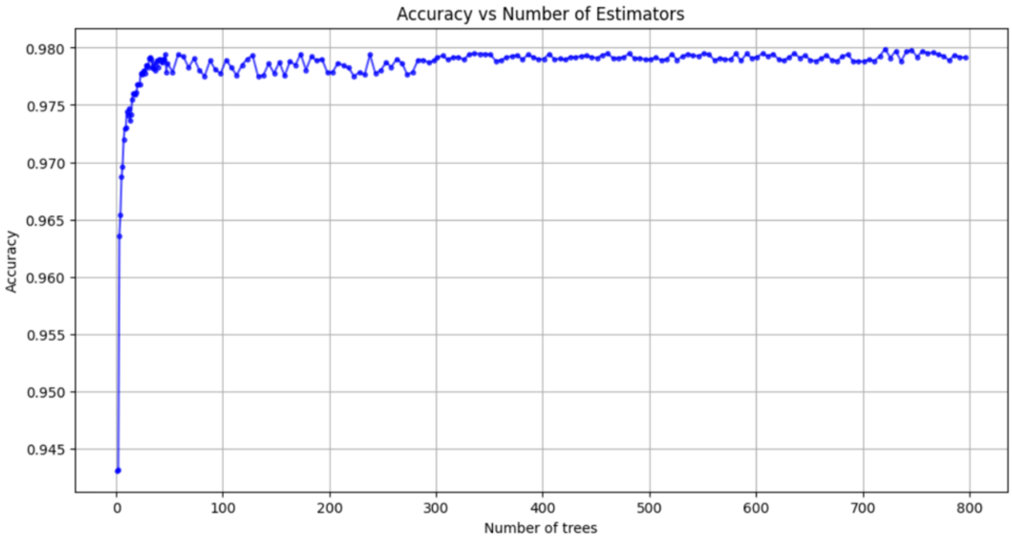


Figure 18: Accuracy vs Number of Trees

7.6 Optimization Techniques Used

To further enhance the performance of the classifiers, the following novel optimization techniques were applied:

- **Instantaneous Fault Samples:** In this method, fault samples at a sampling interval of 20 milliseconds were obtained. Predictions are made based on these 50,000 samples, ensuring quick and efficient diagnosis during runtime.
- **Sliding Window Method:** A sliding window containing 100 samples was created, where predictions are based on this collection of samples. This method helps in reducing the computational load while maintaining high accuracy for time-sensitive fault detection.

- **Ratio of Slopes:** The ratio of ψ1 and ψ2, obtained from the Clarke Transformation, are used as features for classification. This technique yields higher accuracy by capturing subtle variations in the signal patterns.

This chapter highlighted the strengths and limitations of various ML models for NPC inverter fault detection. Concordia transformation, feature engineering, and model optimization played key roles in achieving high accuracy. Each model offers unique benefits, and the final choice depends on the specific application and hardware requirements.

CHAPTER 8

Hardware Implementation

8.1 Introduction:

The hardware implementation forms a crucial part of the fault detection system for NPC inverters. After validating machine learning algorithms in simulation, the next step involves realizing the system on physical hardware to perform real-time fault classification. This chapter presents the selection of hardware components and the current progress in implementing PWM signal generation using the TMS320F28379D Digital Signal Processor (DSP).

8.2 Selection of Components:

Careful selection of hardware components was essential to ensure compatibility, real-time responsiveness, and scalability of the fault detection system. The major components chosen are detailed below:

1) Digital Signal Processor – TMS320F28379D:

Texas Instruments produces the C2000 Delfino series of Digital Signal Controllers: TMS320F28379D is the 16-bit DSP by Texas Instruments in the high performance category. Targeted at real-time control applications such as motor drives, power electronics & renewable energy systems, automotive control and industrial automation.

The F28379D is based on a powerful dual core 32 bit floating point C28x CPU architecture that integrates with real-time control co-processors called Control Law Accelerators (CLA) and provides parallel processing for low-latency control tasks. A high-speed 200 MHz CPU, multi-ePWM (up to 24 ePWM channels for precise control of power converters and inverters), dual(16-bit) ADCs with simultaneous sampling for high-resolution measurements with an FPGA implementation on F28379D.

It also includes a multitude of communication peripherals (SPI, I2C, CAN, UART and USB) which makes for easy integration into complex hardware. The TMS320F28379D has 256 KB RAM up to 1 MB flash and versatile motor control algorithms, which is ideal for computation-intensive applications requiring real-time control and signal processing.

Figure 19: TMS320F28379D

- **Reason for Selection:**

  The TMS320F28379D is a high-performance DSP from Texas Instruments, designed for real-time motor control and power electronics applications. It offers advanced PWM modules, ADC channels, and sufficient processing power for running machine learning models and control algorithms.

- **Features:**

  - Dual-core 200 MHz C28x CPU
  - Enhanced PWM modules (ePWM1 to ePWM12)
  - Fast ADC sampling
  - GPIO support for triggering and diagnostics

2) Current Sensors:

A current probe is a distinct measuring instrument that can observe and count the current that runs through a conductor, usually used with oscilloscope or data acquisition system.

Current probes: Current probes (not to be confused with general-purpose current sensors) are almost invariably used only in laboratory testing, circuit diagnostics or power electronics development — typically for non-invasive, high-frequency or high- accuracy measurements.

Current probes have two major classifications: passive and active. The most common type of CTs (current transformers), used for AC current measurement are passive and based on the electromagnetic induction principle. Easy to use, but only on a single voltage. Active current probes (Hall effect probes or Rogowski coils) work for both AC and DC currents. Hall probes: With magnetic field sensing produce a voltage output proportional to the magnetic flux, Rogowski coils: for measuring derivative or rate-of-change of current hence essentially transients or high-speed signal.

Current probes are used for their isolation, high bandwidth and wide dynamic range. In applications like inverters, converter and motor drives, which are high frequency switching circuits and waveform precision is required then these features are very essential. Their use with tools like digital storage oscilloscopes (DSO) allows engineers to conduct detailed and correct power electronic system analysis.

Figure 20: Current Measurement Probe

- Used to measure phase currents (Ia, Ib, Ic) from the inverter.
- The sensor outputs are digitized and sent to the DSP for real-time processing.

3) DSO:

A Digital Storage Oscilloscope (DSO) is an essential tool in embedded and power electronics systems for visualizing electrical signals over time. It enables engineers and technicians to capture and analyze waveforms, making it easier to troubleshoot, test, and verify circuit behavior. A DSO converts analog signals into digital data for display and analysis, providing features such as signal capture, measurement, and analysis. Modern DSOs offer high bandwidth, multiple channels, and advanced features like FFT analysis, triggering, and data storage. They can be connected to external devices for communication, often supporting protocols such as USB, LAN, or Wi-Fi for remote control and data transfer. The display typically shows waveforms, voltages, and time divisions, offering real-time feedback for system performance and diagnostics. DSOs are crucial for ensuring the stability and functionality of embedded systems, especially in complex power electronics applications.

Figure 21: Dso

- Communication lines connect the DSP with an LCD or serial interface for displaying fault information.
- Display of fault states helps with monitoring and debugging.

4) Inverter Setup:

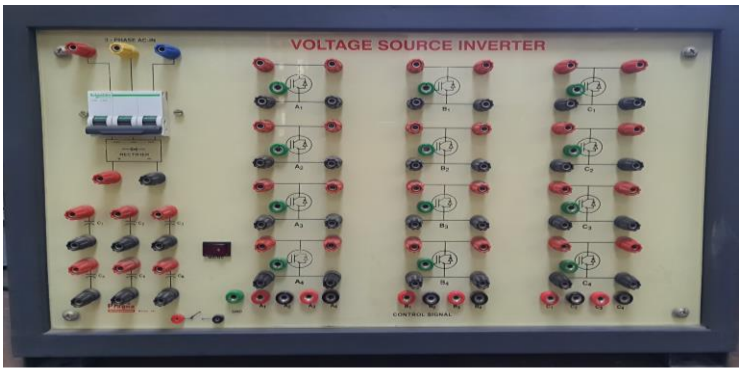


Figure 22: Inverter

- A 2-level NPC inverter was constructed to introduce and study switch faults.

- Gate driver circuitry is connected to the DSP, enabling control over the IGBT/MOSFET switches via PWM signals.

8.3 Pulse Generation using TMS320F28379D:

Generating precise PWM signals is critical for operating the inverter and simulating real-world switching faults. Currently, successful generation of ePWM pulses using the TMS320F28379D DSP has been achieved.

- **PWM Configuration:**
  - PWM Frequency: 50 kHz
  - Duty Cycle: 50%
  - Phase Shifted Outputs:
  - ePWM1: Reference pulse (0°)
  - ePWM2: 120° phase shift
  - ePWM3: 240° phase shift
- **Key Implementation Steps:**

1. **Clock and Time Base Configuration:**
   The time base period and prescaler values were set to achieve the desired PWM frequency.
2. **Phase Shifting:**
   Each ePWM module was configured with appropriate phase shift registers to achieve 120° spacing.
3. **Complementary Output Setup:**
   High and low side gate signals were generated using complementary output settings with dead-band insertion.
4. **GPIO Pin Mapping:**
   Specific GPIO pins on the DSP were configured for PWM output mode and mapped to the gate driver circuit.

5. **Code Composer Studio (CCS):**
   TI's CCS IDE was used to program and debug the DSP. Code was written in C, making use of the F2837xD DriverLib.

8.4 Result

Figure 23: Pulse Generation

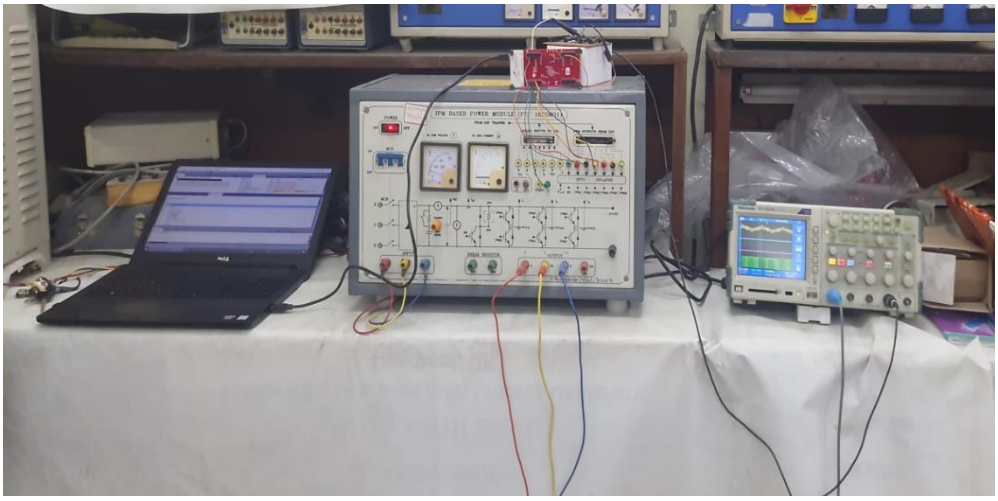


Figure 24: Hardware Implementation

8.5 Current Status and Next Steps:

At this stage, the PWM pulses for controlling the 2-level inverter have been successfully generated. These pulses are being used to simulate switching operations under various conditions. The next steps involve:

- Interfacing the DSP with current sensors to acquire phase current data.
- Performing Concordia transformation on real-time data.
- Running the trained machine learning model on the DSP or a connected embedded platform (e.g., Raspberry Pi).

Conclusion

Inverter fault detection, particularly for NPC inverters, initially relied on traditional methods like time-domain and frequency-domain analysis, which were time-consuming and lacked real-time accuracy. Recognizing these limitations, we shifted to Machine Learning (ML) techniques, such as ANNs, Random Forests, and XGBoost, which successfully improved fault detection by providing faster, more accurate results and better handling of complex datasets.

Our ML-based approach has proven successful, outperforming traditional methods and adapting well to varying load conditions. By integrating feature extraction techniques, the accuracy of fault classification was further enhanced. Moving forward, hybrid models and more generalized algorithms for different inverter topologies, such as two-level and multi-level systems, could offer even greater performance.

In conclusion, the shift to ML-based methods for inverter fault detection represents a significant improvement in speed, accuracy, and automation, ultimately boosting the reliability and efficiency of power systems.An effective machine learning-based framework for diagnosing open-circuit faults in Multi-Level inverters by leveraging the Power-Invariant Clarke Transform for robust and load independent feature extraction is developed. Novel feature engineering techniques like ratio of slopes and sliding window method further enhance the diagnosis model. By integrating advanced machine learning classifiers and their hybrid combination—the proposed methods demonstrate superior performance in fault detection and classification. The proposed algorithms work seamlessly on 2 Level and 3 Level Inverters, showcasing it’s ability to generalize across various inverters.

This project a comparative study of three models used for inverter fault diagnosis. The comparison of results highlights that the Hybrid (RF + XGBoost) model performs well on the 40 Ohm unseen dataset. The training accuracy for RF, XGBoost, and the hybrid model was 98.87%, 98.91%, and 99.08% on 30 Ohm training dataset, respectively, while the test accuracy was 84.30%, 85.01%, and 85.73% on 40 Ohm unseen data. This proves the robustness of the model on variable loads. The hybrid model demonstrated higher precision (0.86), recall (0.84), and F1-

score (0.85) compared to RF and XGBoost. Additionally, confusion matrix analysis confirmed that the hybrid model exhibited fewer misclassifications than the individual classifiers. The three models give more than 99% accuracy on 2 Level Inverter testing dataset.

The project concludes that while RF achieves high accuracy under nominal conditions, it struggles with load variability. XGBoost offers better adaptability, excelling in handling imbalanced datasets. The hybrid model combines the strengths of both RF and XGBoost, resulting in a more robust and accurate fault detection. This makes it highly suitable for real-world applications in industrial drives and renewable energy systems. From the results it can be concluded that the algorithms work well on unseen data sets, implying load independence.

Future Work

Future advancements, include, testing the algorithms on higher level inverters, the incorporation of deep learning techniques, can further enhance diagnostic precision and adaptability, making this approach a highly promising solution for fault diagnosis in power electronics.

To further improve the model’s performance and robustness:

- **Data Augmentation**: To enhance the model's generalization capabilities, introducing a broader range of load conditions and noise levels will help the model adapt to various operational scenarios, improving its robustness and reliability across different environments.
- **Model Optimization**: Exploring other machine learning models, such as Support Vector Machines (SVM) or deep learning architectures, will provide a basis for comparing performance and identifying the most efficient and accurate approach for fault detection.
- **Feature Engineering**: Further investigation into additional signal features or domain-specific transformations can enrich the dataset, providing more relevant information for the model to learn from and potentially improving its fault classification accuracy.
- **Real-Time Testing**: Implementing the fault detection model in a real-time environment with dynamic load variations will allow for an assessment of its practical effectiveness and performance under real-world conditions, where system behavior may fluctuate.
- **Transfer Learning**: Adapting the current model to different systems or component configurations using transfer learning techniques can help extend its applicability to a wider range of inverters, making it a more versatile solution across various power systems.

References

1. Y. Yu and S. Pei, “Open-circuit fault diagnosis of neutral point clamped three-level inverter based on sparse representation,” IEEE Access, vol.6, pp. 73499–73508, 2018.
2. S. Ceballos, J. Pou, E. Robles, et al., “Performance evaluation of faulttolerant neutral-point-clamped converters,” IEEE Trans. Ind. Electron., vol. 57, no. 8, pp. 2709–2718, 2010.
3. J. Lamb and B. Mirafzal, “Open-circuit IGBT fault detection and location isolation for cascaded multilevel converters,” IEEE Trans. Ind. Electron., vol. 64, no. 6, pp. 4846–4856, 2017.
4. X. Li, D. Xu, H. Zhu, et al., “Indirect IGBT over-current detection technique via gate voltage monitoring and analysis,” IEEE Trans. Power Electron., vol. 34, no. 4, pp. 3615–3622, 2019.
5. X. R. Fan, M. Z. Kang, E. Heuvelink, et al., “A knowledge-and-datadriven modeling approach for simulating plant growth: a case study on tomato growth,” Ecol. Model., vol. 312, pp. 363–373, 2015.
6. I. M. Karmacharya, R. Gokaraju, “Fault location in ungrounded photovoltaic system using wavelets and ANN,” IEEE Trans. Power Deliv., vol. 33, no. 2, pp. 549–559, 2018.
7. A. Abdullah, “Ultrafast transmission line fault detection using a DWTbased ANN,” IEEE Trans. Ind. Appl., vol. 54, no. 2, pp. 1182–1193, 2018.
8. V. F. Pires, T. G. Amaral, D. Sousa, et al., “Fault detection of voltagesource inverter using pattern recognition of the 3D current trajectory,” 2010 IEEE Region 8 Int. Conf. Computational Technologies in Electrical and Electronics Engineering (SIBIRCON), Listvyanka, 2010, pp. 617–621.
9. J. Sun, G. Zhong, J. Dong, et al., “Cooperative profit random forests with application in ocean front recognition,” IEEE Access, vol. 5, pp. 1398–1408, 2017.
10. A. M. Shah, B. R. Bhalja, “Fault discrimination scheme for power transformer using random forest technique,” IET Gener. Transm. Distrib., vol. 10, no. 6, pp. 1431–1439, 2016.
11. R. Pivovarov, E. No´emie, “A hybrid knowledge-based and data-driven approach to identifying semantically similar concepts,” J. Biomed. Inf., vol. 45, no. 3, pp. 471–481, 2012.
12. U. Choi, H. Jeong, K. Lee, et al., “Method for detecting an open-switc fault in a grid-connected NPC inverter system,” IEEE Trans. Power Electron., vol. 27, no. 6, pp. 2726–2739, 2012.
13. S. Khomfoi, L. M. Tolbert, “Fault diagnostic system for a multilevel inverter using a neural network,” IEEE Trans. Power Electron., vol. 22, no. 3, pp. 1062–1069, 2007.
14. L. Kou, C. Liu, G.-w. Cai, J.-n. Zhou, Q.-d. Yuan, and S.-m. Pang, ”Fault diagnosis for open-circuit faults in NPC inverter based on knowledgedriven and data-driven approaches,” IET Power Electronics, vol. 13, no. 6, pp. 1236–1245, 2020.
15. C. N. Ibem, M. E. Farrag, A. A. Aboushady, and S. M. Dabour, “Multiple Open Switch Fault Diagnosis of Three Phase Voltage Source Inverter Using Ensemble Bagged Tree Machine Learning Technique,” IEEE Access, vol. 11, pp. 85865–85877, 2023.
16. C. J. O’Rourke, M. M. Qasim, M. R. Overlin, and J. L. Kirtley, “A Geometric Interpretation of Reference Frames and Transformations: dq0, Clarke, and Park,” IEEE Transactions on Energy Conversion, vol. 34, no. 4, pp. 2070–2083, Dec. 2019.