

Machine Learning–Based Protection and Fault Identification of 100% Inverter-Based Microgrids

Milad Beikbabaie,¹ *Graduate Student Member, IEEE*, Michael Lindemann,² *Graduate Student Member, IEEE*,
Mohammad Heidari Kapourchali,² *Member, IEEE*, and Ali Mehrizi-Sani,¹ *Senior Member, IEEE*

¹ The Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061

² Department of Electrical Engineering, University of Alaska, Anchorage, AK 99508

e-mails: miladb@vt.edu, mlindemann@alaska.edu, mhkapourchali@alaska.edu, mehrizi@vt.edu

Abstract—100% inverter-based renewable units are becoming more prevalent, introducing new challenges in the protection of microgrids that incorporate these resources. This is particularly due to low fault currents and bidirectional flows. Previous work has studied the protection of microgrids with high penetration of inverter-interfaced distributed generators; however, very few have studied the protection of a 100% inverter-based microgrid. This work proposes machine learning (ML)–based protection solutions using local electrical measurements that consider implementation challenges and effectively combine short-circuit fault detection and type identification. A decision tree method is used to analyze a wide range of fault scenarios. PSCAD/EMTDC simulation environment is used to create a dataset for training and testing the proposed method. The effectiveness of the proposed methods is examined under seven distinct fault types, each featuring varying fault resistance, in a 100% inverter-based microgrid consisting of four inverters.

Index Terms—Fault identification, inverter-based resources (IBR), microgrid, protection.

I. INTRODUCTION

With the increasing prevalence of renewable energy resources in the form of inverter-based distributed generation units, traditional protection strategies for microgrids connecting multiple such resources may become insufficient. The protection of 100% inverter-based microgrids presents significant challenges [1], primarily due to the reduced inertia of inverter-based units [2]. A microgrid with low inertia sources could experience stability problems if line faults are not quickly cleared. In inverter-interfaced distributed generators, the current contribution is limited during short circuits, resulting in much lower fault currents. Limited fault current, combined with the bidirectional power flow and intermittent generation, presents a challenge for fault protection when employing traditional high fault current methods.

Reference [3] studies the challenges faced when converting

an existing distribution feeder to an inverter-based microgrid and suggests the use of adaptive settings for relays to tackle the protection challenges. Protection challenges are more prominent in isolated microgrids, which have lower fault levels compared to grid-connected microgrids with infeed power from the main grid [4]. A protection scheme for isolated microgrids with high penetration of inverter-interfaced distributed generators should be fast, adaptable, and accurate in order to uphold microgrid stability and safeguard critical loads. However, the research on microgrid protection has not yet led to a commercially available microgrid relay [5]. This is because many solutions rely on communication methods or complex learning-based relay systems [6]. Using communication makes the power system susceptible to delays and cyberattacks, reducing the grid's resiliency [7]. Communication delays can affect the dynamic behaviors of communication-based IBR control during faults, which makes protection more challenging [8]. Many existing microgrid protection schemes lack adaptability to diverse topologies and source types, are not cost-effective, or rely on communication, and have not been tested under scenarios with extremely high IBR penetration, which is likely in the near future [6].

Machine learning (ML)–based protection methods have shown great potential in accurately detecting and identifying faults. However, they have been mostly employed in transmission and distribution systems protection, and very few studies have explored the applications of ML in detecting and identifying faults in microgrids with high renewable resource penetration [5]. Authors in [5] propose an intelligent fault diagnosis method based on deep learning, utilizing wavelet transformation and sequence components. Deep learning models developed in Pytorch are employed to train and validate fault detection, classification, and location identification. Authors in [9] present a general-purpose support vector machine (SVM)-based adaptive scheme to identify normal and fault conditions in AC microgrids and detect fault types. Reference [10] develops an ML-based protection method for AC microgrids that detects and classifies faults. The proposed ML methods use complex deep learning algorithms or require high sampling rates.

This paper presents decision tree-based protection solutions that combine fault detection and fault type classification in a fully inverter-based microgrid, using local measurements with-

This work of Virginia Tech is supported in part by the National Science Foundation (NSF) under award ECCS-1953213, in part by the State of Virginia's Commonwealth Cyber Initiative (www.cyberinitiative.org), in part by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number 38637 (UNIFI Consortium led by NREL), and in part by Manitoba Hydro International. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government. The work of UAA is supported in part by the U.S. National Science Foundation (NSF) under awards OIA-2229772, RISE-2220624, RISE-2022705, and RISE-2318385.

out any communication. The effectiveness of the protection solution is studied on a microgrid equipped with four inverters, where half operate in grid-following mode, and the remaining two function in grid-forming mode. The proposed method only uses the root mean square (RMS) value of the three-phase current, three-phase voltage, active and reactive power with 1 ms intervals, which reduces the computational process and memory use. The low computational burden facilitates practical implementation on a microcontroller. The salient features of the proposed method are:

- The proposed method, designed for simplicity, integrates fault detection and identification for efficient implementation in digital relays.
- The decision tree algorithm trains in less than 9 s makes it applicable for real-world grid applications.
- The proposed approach can detect faults in less than 5 ms for both low- and high-impedance faults.

II. TEST SYSTEM

A. Basics of Inverter Control

This subsection discusses the basics of inverter control, where it can be controlled either in the grid-following or the grid-forming mode [2].

1) Grid-Following Mode

The inverter receives real power set points P_{ref} and reactive power set points Q_{ref} , and the inverter adjusts its output power to be as close as possible to the received power set points. A phase-locked loop (PLL) is used for estimating the voltage phase angle, helping with converting the current and voltage form abc -frame to dq -frame and back. Fig.1 shows the grid-following conventional decoupled current control loop for an inverter connected to the grid through an RL filter, where the output real power is P_t , the output reactive power is Q_t , inverter current is i_t , and V_t and V_s are the terminal and the grid voltages, respectively [11].

2) Grid-Forming Mode

The voltage and frequency of the grid are subject to disturbances and changes; however, the voltage and frequency magnitude need to be maintained within nominal ranges. As a result, $Q-V$ and $P-f$ droop controls are used in grid-forming inverters. The $Q-V$ droop control updates the voltage set point of the inverter controller, resulting in adjusting the reactive power to maintain the voltage as shown in Fig. 2(a). The $P-f$ droop control updates the voltage angle of the inverter, resulting in modifying the real power output to maintain the frequency as shown in Fig. 2(b). The PI controllers are used in the grid-forming inverter droop control, as shown in Fig. 2.

B. 100% Inverter-Based Microgrid

A 4-bus microgrid is developed in PSCAD/EMTDC as shown in Fig. 3, where the first and third inverters are grid-forming, and the rest are grid-following. The microgrid base power is 1.5 MVA, the low-voltage base voltage is 480 V, and the medium-voltage base voltage is 12.47 kV. Each inverter has a DC voltage source where its primary side voltage is fixed to 1.2 kV. The maximum output power of each inverter

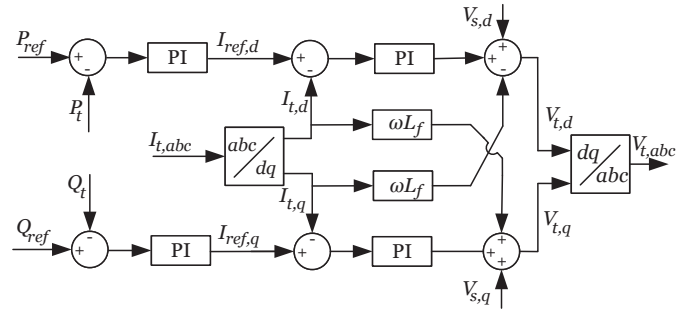


Fig. 1. Decoupled control scheme of a grid-following inverter for d - and q -axes.

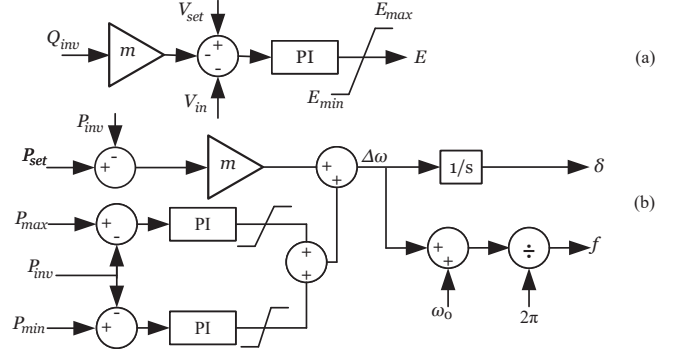


Fig. 2. Droop control scheme of a grid-forming inverter for: (a) $P - f$, (b) $Q - V$.

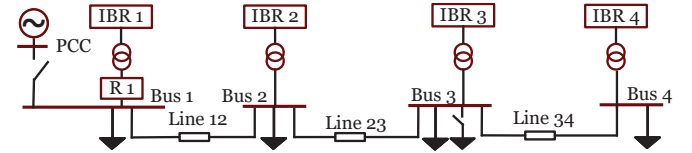


Fig. 3. Single-line diagram of the microgrid.

is limited to 1.5 MVA. Each grid-following unit is connected to a transformer through an RL filter, where its resistance is 1.5 m Ω , and its inductance is 20 μ H. $k_P = 2$ and $k_I = 0.0025$ are the gains for the grid following PI blocks. Gains for the PI block in $P-f$ droop control of grid-forming inverters are $k_P = 0.6$ and $k_I = 0.003$, and they are $k_P = 0.6$ and $k_I = 0.002$ for the $Q-V$ droop control. Line 12 resistance is 1.4 Ω , and line 12 inductance equals to 2 Ω . Line 23 resistance is 2.2 Ω , and line 23 inductance equals to 3.16 Ω . Line 34 resistance is 0.6 Ω , and line 34 inductance is 3.16 Ω . A load is connected to bus 3 through a breaker, where its real power can be between 0.1 pu to 0.6 pu, and its reactive power can vary from 0.01 pu to 0.06 pu. The protection relay R1 is at bus 1.

III. METHODOLOGY

This work utilizes a decision tree-based method to detect and identify faults in the microgrid, as elaborated in the following subsections.

A. Algorithm Selection

Since the protection algorithm needs to detect a fault within a few milliseconds it needs to be run every few milliseconds. Due to this requirement, making algorithm less computational expensive is preferable. Previous work has utilized deep learning methods for protection; however, real-time implementation using off-the-shelf microcontrollers is still a challenge. Furthermore, microgrids are small grids with a few nodes, and classification algorithms can show effective results for protection against short circuit faults. The most basic type of algorithm is the support vector machine (SVM) model. However, SVM models scale between either quadratically or cubically with training data complexity. Due to this scaling penalty and this projects datasets complexity, SVM was not a viable option. Therefore, this work uses a decision tree model since decision tree models scale linearly with the data complexity.

B. Decision Tree Basics

Decision trees work via a tree like structure in which each node represents a feature or attribute that has been determined as important. Branches correspond to a decision based on the previous feature and lead to the next node. The decision on when to split in the tree is decided using the Gini impurity method. During training, the goal is to constantly minimize the Gini impurity [12]. The Gini impurity of the dataset is calculated using the equation:

$$\text{Gini}(D) = 1 - \sum_{i=1}^J p_i^2, \quad (1)$$

where D represents the dataset for the node, J is the number of classes, and p_i is the proportion of samples that belong to class i in the dataset D .

C. Dataset Preparation and Feature Selection

A dataset is developed by simulating short-circuit faults using PSCAD/EMTDC. The PSCAD automation Python API is utilized to run multiple cases and create the dataset [7]. For each simulation run, a distinct fault type, fault location, fault resistance, and fault duration are considered. The seven fault types selected are *AG*, *BG*, *CG*, *ABG*, *ACG*, *BCG*, and *ABCG*. The chosen fault resistance values are 100 Ω , 10 Ω , 1 Ω , 0.1 Ω , and 0.001 Ω . The faults are located at buses 1–4. For the fault duration, 0.05 s, 0.1 s, and 0.2 s are selected. All faults are introduced at $t = 0.05$ s, and each simulation is conducted for a duration of 1 second. As a result, a total of 420 cases are simulated and used for both training and testing the proposed protection algorithm.

In every simulation, the three-phase current I , three-phase voltage V , real power P , and reactive power Q data are recorded with a 1 ms time intervals. Furthermore, a fault detection signal is added to the dataset, where it becomes 1 during the short-circuited faults and is zero when there is no fault. Additionally, a column is added for the fault type. A number is assigned to each fault type as shown in Table I, where it can be between 0 and 7. Fault type 0 represents no faults.

TABLE I
FAULT TYPE NUMBER TABLE

Fault Number	Description	Fault Number	Description
0	No faults	4	<i>ABG</i> faults
1	<i>AG</i> faults	5	<i>ACG</i> faults
2	<i>BG</i> faults	6	<i>BCG</i> faults
3	<i>CG</i> faults	7	<i>ABCG</i> faults

TABLE II
TABLE OF THE ACCURACY OF DIFFERENT INPUT FEATURES

Inputs	Fault Prediction Accuracy	Fault Type Prediction Accuracy
I	95.58%	95.50%
I, V	95.92%	95.82%
I, P, Q	96.26%	96.05%
I, V, P, Q	96.33%	96.04%

D. Training and Hyperparameters Selection

Table II shows how different selections of the input features affect the accuracy of the decision tree. Four combinations of inputs are selected for the R1 relay, where the first one only uses I . The second one uses both V and I . The third one uses I , P , and Q . The fourth one uses I , V , P , and Q . In order to achieve the highest accuracy the fourth feature set are used in this work, where it shows the best performance for fault detection compared to other combinations.

Scikit-learn library is used in this work, where the decision tree model maximum depth is set to 43 and it has a total of 4044 leaves. Training and validation data consisted of 715,959 1 ms time samples. Input variables consisted of the three-phase current, the three-phase voltage, the real power output, and the reactive power output of the bus where the relay is located. The data is split utilizing `train-test-split` function in scikit-learn. 80 percent of the data is used for training, and the remaining 20 percent is used for validation. A shuffle is applied with a random state setting of 20 being utilized for repeatability. No further preprocessing is required before training the decision tree model. True output variables are split into their respective categories and fed into the model.

IV. SIMULATION RESULTS AND EVALUATION

This section presents the simulation results for different types of faults at different buses.

1) Case 1: *AG* Fault on Bus 1.

Figs. 4 shows fault detection and fault type for a phase A to ground fault with a 0.01 Ω resistance located at bus 1, starting at $t = 50$ ms and clearing at $t = 150$ ms. The decision tree detects the fault occurrence at $t = 50$ ms and detects fault clearance at $t = 155$ ms, with 5 ms delays. The decision tree detects the fault type number 1, *AG* fault. The decision tree detects the fault type and fault occurrence correctly during the fault period except for one sample at $t = 0.104$ s; however, it quickly corrects its prediction in the next time step.

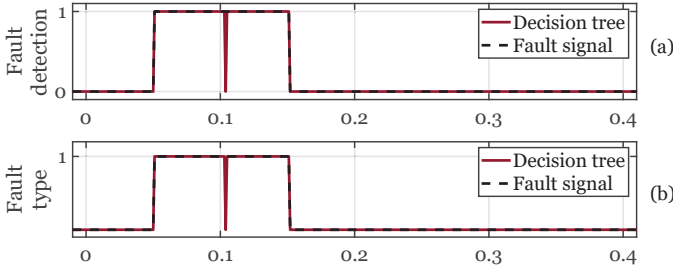


Fig. 4. *AG* fault located at bus 1: (a) fault detection and (b) fault type.

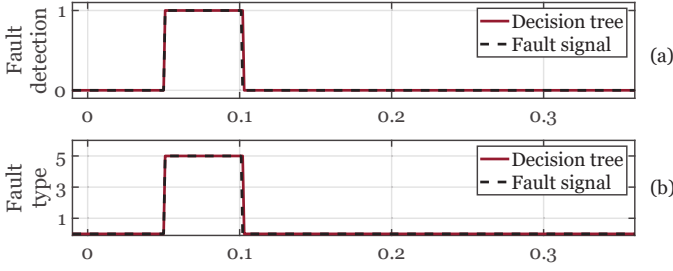


Fig. 5. *ACG* fault located at bus 2: (a) fault detection and (b) fault type.

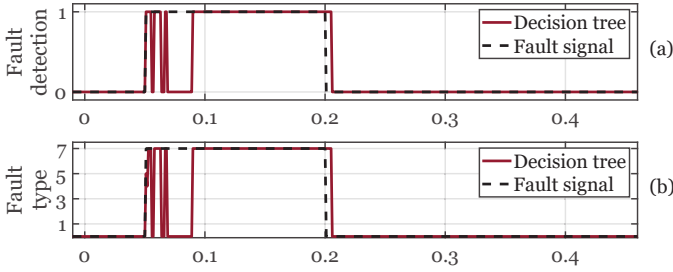


Fig. 6. *ABCG* fault located at bus 3: (a) fault detection and (b) fault type.

2) Case 2: *ACG* Fault on Bus 2.

Figs. 5 shows fault detection and fault type for phases *A* and *C* to ground fault with a $1\ \Omega$ resistance located at bus 2, starting at $t = 50$ ms and clearing at $t = 100$ ms. The decision tree detects the fault occurrence at $t = 50$ ms and fault clearance at $t = 100$ ms. The decision tree detects the fault type correctly. The decision tree indicates that the fault type number is 5, which is equivalent to an *ACG* fault shown in Fig. 5(b).

3) Case 3: *ABCG* Fault on Bus 3.

Figs. 6 shows fault detection and fault type for a three-phase to ground fault with a $40\ \Omega$ impedance located at bus 3, starting at $t = 50$ ms and clearing at $t = 200$ ms. Unlike Case 1 and Case 2, the decision tree has not been trained for a $40\ \Omega$ fault value, decreasing its accuracy. The decision tree detects the fault occurrence at $t = 50$ ms and fault clearance at $t = 210$ ms; however, it cannot detect correctly after the fault occurrence in a few time steps shown in Fig. 6(a). Furthermore, the decision tree has difficulty identifying the fault type at the first 40 ms of the simulation; however, it can detect the fault with certainty after that. The decision tree detects the fault clearance with a 10 ms delay. The decision

tree indicates that the fault type number is 7 after 40 ms, which is equivalent to an *ABCG* fault using shown in Fig. 6(b). Fig. 6 indicates that the proposed method still works well for an untrained high-impedance fault.

V. CONCLUSION

Low fault current and bidirectional current are major challenges in the protection of fully inverter-based microgrids. This work proposes Decision tree-based protection solutions that incorporate fault detection and identification. These solutions are designed for efficient implementation in digital relays due to their simplicity. The effectiveness of the proposed method is examined for both low and high-impedance faults across seven distinct fault types characterized by varying fault resistance. The analyses are conducted in the context of a 100% inverter-based microgrid equipped with four inverters. This work indeed facilitates the utilization of a ML-based method for the protection of fully inverter-based microgrids. Future work includes investigating the efficacy of the ML-based methods for a larger microgrid and assessing their accuracy under various network configurations.

REFERENCES

- [1] J. Sharma and T. S. Sidhu, "A new protection scheme for feeders of microgrids with inverter-based resources," *Electric Power System Research*, vol. 224, p. 109632, Nov. 2023.
- [2] M. Beikbabaie, M. Montano, A. Mehrizi-Sani, and C.-C. Liu, "Mitigating false data injection attacks on inverter set points in a 100% inverter-based microgrid," in *IEEE Innovative Smart Grid Technologies Conference (ISGT)*, Washington DC, Feb. 2024.
- [3] T. Patel, S. Brahma, J. Hernandez-Alvidrez, and M. J. Reno, "Adaptive protection scheme for a real-world microgrid with 100% inverter-based resources," in *IEEE Kansas Power and Energy Conference (KPEC)*, Aug. 2020.
- [4] D. Liu, A. Dyśko, Q. Hong, D. Tzelepis, and C. D. Booth, "Transient wavelet energy-based protection scheme for inverter-dominated microgrid," *IEEE Trans. on Smart Grid*, vol. 13, no. 4, pp. 2533–2546, Mar. 2022.
- [5] L. M. Kandasamy and K. Jaganathan, "Intelligent fault diagnosis using Deep Learning for a microgrid with high penetration of renewable energy sources," *Electric Power System Research*, vol. 51, no. 4, pp. 332–350, Feb. 2023.
- [6] P. Gadde and S. M. Brahma, "Topology-agnostic, scalable, self-healing, and cost-aware protection of microgrids," *IEEE Transactions on Power Delivery*, vol. 37, no. 4, pp. 3391–3400, Nov. 2021.
- [7] M. Beikbabaie, A. Venkataramanan, and A. Mehrizi-Sani, "EMT-based co-simulation of power system and communication networks," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Nov. 2023.
- [8] M. Beikbabaie, A. Mohammadhassani, V. Krishnan, A. Gorski, A. Mehrizi-Sani, V. K. Shah, A. P. Da Silva, and J. H. Reed, "Experience in real-time simulation of the power system with 5g communication," in *IEEE Innovative Smart Grid Technologies Conference (ISGT)*, Feb. 2024, pp. 1–5.
- [9] R. Aiswarya, D. S. Nair, T. Rajeev, and V. Vinod, "A novel SVM based adaptive scheme for accurate fault identification in microgrid," *Electric Power System Research*, vol. 221, p. 109439, Aug. 2023.
- [10] M. Uzair, M. Eskandari, L. Li, and J. Zhu, "Machine learning based protection scheme for low voltage AC microgrids," *Energies*, vol. 15, no. 24, p. 9397, Dec. 2022.
- [11] N. Souri, A. Mehrizi-Sani, and K. Tehrani, "Stability enhancement of LCL-type grid-following inverters using capacitor voltage active damping," in *IEEE PES General Meeting*, Seattle, WA, Jul. 2024.
- [12] S. Lee, C. Lee, K. G. Mun, and D. Kim, "Decision tree algorithm considering distances between classes," *IEEE Access*, vol. 10, pp. 69 750–69 756, Jun. 2022.