

Enhanced Fault Detection in Power Electronics-Dominated Microgrids: A Machine Learning Framework with Residual-Based Features and Class Imbalance Mitigation

Shrunkhala Satish Wankhede^{1*}, Khushboo Jain², Nileshchandra Pikle³

¹Phd Scholar, Department of Computer Science and Engineering, Indian Institute of Information Technology, Nagpur (IIITN)
Email: shrunkhalawankhede30@gmail.com, ORCID ID: 0000-0002-7515-0930

²Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Information Technology, Nagpur (IIITN)
Email: khushboo.jain@iiitn.ac.in, ORCID ID: 0000-0001-5643-3647

³Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Information Technology, Nagpur (IIITN)
Email: npikle@iiitn.ac.in, ORCID ID: 0000-0002-0106-4618

ABSTRACT

This study presents a machine learning and deep learning framework for fault detection in power electronics-dominated microgrids using residual-based features and class-imbalance mitigation. The framework compares the performance of Support Vector Machine, K-Nearest Neighbors, Logistic Regression, Gradient Boosting, Convolutional Neural Network, and Long Short-Term Memory models, with data created from a simulated environment for a microgrid. To enhance the representation of fault related deviations, residual based and temporal features are created, and class imbalance is compensated for by applying Synthetic Minority Over-sampling Technique (SMOTE) to the training data. The metrics employed to evaluate the models include accuracy, precision, recall, F1 score and other metrics. The experimental results demonstrate that the Gradient Boosting model with 99.80% accuracy is best in the evaluated dataset and the models based on CNN are providing a competitive performance in the fault detection task. The results show the feasibility of a machine learning method based on residuals to enable automatic microgrid fault detection. But it is still required to validate with measured hardware or data from the field before use in practice.

Keywords: Microgrid Fault Detection, Machine Learning, Class Imbalance, SMOTE, Gradient Boosting, Predictive Maintenance, LSTM, CNN

1. INTRODUCTION

The integration of distributed energy resources (DERs) and power electronic interfaces in modern microgrids is transforming traditional power systems into more flexible, resilient, and intelligent networks. However, this transition brings significant challenges in system monitoring and fault detection, especially in power electronics-dominated grids (PEDGs), where high-frequency switching and non-linear dynamics complicate conventional diagnostic techniques [1], [2]. Detecting faults in such systems is critical for ensuring grid reliability and safety. Yet, the timely and accurate identification of internal inverter faults—particularly in the presence of load and renewable generation variability—remains a complex and unsolved problem [3]. Traditional fault detection methods, such as threshold-based or rule-based algorithms, often fail to generalize under changing operating conditions [4]. These methods lack the adaptability required to handle the dynamic behavior of DERs and frequently misclassify transient anomalies or miss rare but critical fault events [5].

In response, machine learning (ML) methods have been increasingly applied for fault classification, anomaly detection, and predictive maintenance in microgrids. Techniques such as support vector machines (SVM) [6], artificial neural networks (ANN) [7], and deep learning models including CNNs and LSTMs [8]–[10] have shown improved accuracy in modeling complex fault dynamics. However, a persistent challenge across studies is the class imbalance between normal operations (majority class) and rare fault conditions (minority class). This imbalance leads to models that are biased toward the majority class, resulting in poor detection of faults despite high overall accuracy [11], [12]. While several studies propose class balancing techniques like SMOTE, ensemble learning, or GAN-based augmentation [13], [14], comparative evaluation of these strategies under real-world microgrid conditions is rarely conducted. Furthermore, the integration

of forecast-based residual features—which quantify deviations between expected and actual values—has not been widely explored in fault detection systems.

To overcome these drawbacks, a unified experimental platform is used in this study to demonstrate the performance of the conventional machine learning method and deep learning approach on fault detection in a typical power-electronics dominated microgrid system.

The framework is designed to combine feature engineering via residual with class-imbalance mitigation and test several model configurations in an identical experimental environment.

1. Residual-based feature engineering using PV and load forecasts;
2. Advanced class imbalance mitigation techniques including SMOTE, cost-sensitive learning, and ensemble modeling;
3. Evaluation of traditional ML classifiers (SVM, KNN, Gradient Boosting) and deep models (CNN, LSTM) under balanced and imbalanced conditions;
4. Assessment of the impact of loss functions (MAE vs. MSE) on model sensitivity to faults.

The main contributions of this work are:

1. A novel combination of forecast residuals and ML models for improved microgrid fault detection;
2. A comparative analysis of imbalance handling techniques, revealing their effects on detecting minority class faults;
3. Demonstration that Gradient Boosting and CNNs with MAE loss and softmax activation significantly outperform baseline methods;
4. A reproducible, deployment-oriented framework with fault alerting and visualization tools.

Novelty and Contributions:

This research addresses a critical and underexplored challenge in microgrid fault detection—the reliable identification of rare fault events in the presence of severe class imbalance, which is often overlooked in traditional ML studies.

The key novel aspects and contributions of this work include:

1. Residual-based Feature Engineering: We introduce the use of residuals between actual and forecasted load and PV generation as primary features, which effectively capture system anomalies and improve model sensitivity to faults.
2. Comparative Evaluation of Class Imbalance Strategies: We benchmark SMOTE, cost-sensitive learning, and ensemble techniques across diverse ML models (SVM, KNN, Gradient Boosting, CNN, LSTM), offering the study provide a comparison in this context.
3. Impact of Loss Function on Minority Class Detection: We demonstrate that using Mean Absolute Error (MAE) over Mean Squared Error (MSE) significantly improves fault detection for deep learning models trained on imbalanced data.
4. Deployment-Oriented Framework: We implement real-time visualization and alert generation components, making the solution suitable for practical deployment in microgrid operational environments.
5. Superior Fault Detection Performance: Using the Gradient Boosting model, the framework can detect faults with a precision of 99.80%, while the CNN model can detect faults with a precision of about 90% on the simulated test set that is evaluated.

These contributions collectively advance the current state of ML-based microgrid fault detection by offering a practical, accurate, and generalizable approach to handling imbalanced datasets.

The remainder of the paper is organized as follows: Section 2 reviews related literature; Section 3 describes the proposed methodology; Section 4 presents results and analysis; Section 5 addresses overfitting and deployment; and Section 6 concludes with future directions.

Research Gap

Despite the increasing use of machine learning for microgrid Binary fault detection, four methodological gaps remain.

1. First, many existing fault-detection approaches primarily use electrical measurements or handcrafted features. The potential of forecast deviations between expected and measured load/PV behavior as complementary fault-sensitive features remains comparatively underexplored in power-electronics-dominated microgrids.
2. Second, severe class imbalance remains an important methodological problem. A model may obtain high accuracy by correctly classifying the dominant normal class while failing to identify rare fault

events. Therefore, Existing studies do not adequately report PR-AUC, balanced accuracy and MCC.

3. Third, studies often evaluate individual algorithms without systematically separating the effects of feature engineering, class balancing, and classifier selection. Consequently, it can be difficult to determine whether performance improvements originate from the proposed features, the balancing method, or the classifier itself.
4. Fourth, reproducibility and temporal generalization require greater attention. For time-dependent microgrid data, random splitting can allow highly correlated observations from neighboring time periods to appear in both training and test sets. A chronological evaluation protocol is therefore more appropriate for estimating performance on future unseen operating conditions.

To address these gaps, this study investigates a residual-guided and imbalance-aware fault-detection framework using a chronological experimental protocol. The framework combines operational measurements, forecast residuals, temporal statistics, and environmental variables and compares conventional machine learning and deep learning classifiers.

2. LITERATURE SURVEY

The increasing penetration of renewable energy sources, inverter-based distributed energy resources (DERs), energy storage systems, and power electronic converters has significantly transformed the operating characteristics of modern microgrids. Unlike conventional power systems, microgrids can experience rapid changes in voltage, current, frequency, power flow, and operating topology because of variable renewable generation, converter control actions, load switching, islanding, and grid-connected or islanded operation. These characteristics make reliable fault and anomaly detection increasingly challenging.

Traditional protection and fault-detection techniques are generally designed around predefined operating limits and electrical signatures. However, the dynamic behavior of power-electronics-dominated grids (PEDGs) can make fixed thresholds less effective. Consequently, data-driven approaches based on machine learning (ML) and deep learning (DL) have attracted considerable research attention. This section reviews the development of fault detection techniques, ML-based methods, class-imbalance mitigation strategies, feature-engineering approaches, and deployment-oriented solutions for microgrid applications.

2.1 Traditional Fault Detection Approaches

Early fault detection methods in microgrids were primarily based on deterministic rules, threshold comparisons, relay characteristics, and signal-processing techniques. These methods commonly monitor electrical quantities such as phase voltage, phase current, frequency, rate of change of frequency, power, and impedance. When a measured parameter exceeds a predefined threshold, the corresponding protection mechanism identifies a possible fault condition [1].

Overcurrent protection is one of the most widely used approaches because fault conditions generally result in significant increases in current. Similarly, under-voltage and over-voltage protection methods identify abnormal operating conditions based on voltage limits. Frequency-based techniques are particularly useful for detecting disturbances associated with generation-load imbalance and islanding conditions.

Although these methods are computationally efficient and comparatively easy to implement, their effectiveness depends strongly on appropriately selected thresholds. A threshold suitable for one operating condition may not be appropriate under another condition. Changes in load demand, renewable generation, converter operating modes, and system topology can significantly alter the normal operating range of electrical variables. With increasing penetration of inverter-based DERs, the electrical signature of faults can also differ from that observed in conventional synchronous-generator-dominated systems [2]. Power electronic converters can limit fault current, modify transient responses, and introduce fast switching components. As a result, conventional protection techniques may experience difficulties in distinguishing between genuine faults and normal transient events.

Signal-processing-based approaches were subsequently introduced to extract additional information from measured electrical signals. Techniques such as Fourier analysis, wavelet transforms, empirical mode decomposition, and time-frequency analysis have been investigated for identifying transient fault signatures. These methods can provide useful information about the frequency components and temporal characteristics associated with abnormal events.

However, signal-processing techniques generally require carefully designed features and may be sensitive to measurement noise and changes in operating conditions. These limitations have encouraged researchers to investigate intelligent data-driven techniques that can automatically learn relevant patterns from historical measurements.

2.2 Machine Learning for Microgrid Fault Detection

Machine learning has emerged as an alternative to conventional rule-based detection because it can learn relationships between measured system variables and operating conditions from historical data. Instead of relying entirely on manually defined thresholds, ML models can identify patterns associated with normal and abnormal operating states.

Supervised learning approaches have been widely used for fault classification. In such approaches, historical measurements are labelled according to their operating condition, such as normal operation, line-to-line fault, line-to-ground fault, three-phase fault, inverter fault, or other disturbance categories. The trained classifier can subsequently assign new observations to the corresponding operating class [3].

Artificial Neural Networks (ANNs) have been investigated extensively because of their capability to approximate nonlinear relationships between input measurements and fault classes. ANN-based models can process multiple electrical variables simultaneously and can therefore capture interactions that may not be apparent when individual threshold values are considered.

Support Vector Machines (SVMs) have also been widely applied to power-system fault classification because of their effectiveness in high-dimensional feature spaces. SVMs can provide strong classification performance when appropriate kernels and hyper parameters are selected [5]. However, their computational complexity can become significant for large datasets. In addition, conventional SVM training may be influenced strongly by class imbalance when fault observations are substantially fewer than normal observations.

Decision Trees provide an interpretable classification framework in which decisions are obtained through a sequence of feature-based conditions. Random Forests extend this concept by combining multiple decision trees, thereby reducing the dependence on a single tree and improving generalization [6]. Gradient Boosting and other ensemble-learning methods have similarly demonstrated the ability to model nonlinear relationships between system measurements and fault conditions [7].

Roy et al. [8] investigated a deep-learning-based relay system for fault detection, classification, and localization in microgrids. Such studies demonstrate the potential of data-driven models to perform multiple fault-diagnosis tasks using electrical measurements.

Despite these developments, the effectiveness of ML models depends strongly on the quality and distribution of the training dataset. A classifier trained predominantly on normal operating samples may achieve high overall accuracy while providing inadequate detection of rare fault events. Therefore, class imbalance has become an important research issue in microgrid fault detection.

2.3 Deep Learning-Based Fault Detection

Deep learning techniques have attracted increasing attention because they can automatically learn hierarchical representations from raw or minimally processed data. Unlike conventional ML models, which often require explicit feature engineering, deep neural networks can learn complex features directly from large datasets.

Convolutional Neural Networks (CNNs) have been applied to electrical fault detection by converting time-series signals into suitable representations or by processing multi-channel measurement data. CNNs can learn local patterns and transient signatures that may correspond to different fault conditions [3].

Long Short-Term Memory (LSTM) networks are particularly relevant to microgrid applications because electrical measurements are inherently time-dependent. LSTM networks use memory mechanisms to learn temporal dependencies and can therefore capture changes occurring across successive measurement intervals [4]. This capability makes LSTM-based models suitable for early warning and sequential anomaly-detection applications.

Hybrid architectures have also been investigated. For example, CNN-LSTM structures can combine CNN-based feature extraction with LSTM-based temporal modelling. Such architectures can capture both local patterns and temporal dependencies in electrical signals.

Auto-encoders and other unsupervised or semi-supervised deep-learning approaches provide another direction for anomaly detection. These models can be trained primarily using normal operating data and subsequently identify abnormal observations based on reconstruction error. Such methods are particularly useful when labelled fault samples are limited.

However, deep-learning approaches generally require sufficient training data, computational resources, and appropriate model regularization. Their performance can also be affected by imbalanced datasets, noisy measurements, changing operating conditions, and differences between simulated and real-world data.

2.4 Feature Engineering for Fault Detection

The selection and representation of input features significantly influence the performance of ML-based fault detection systems. Electrical measurements such as voltage, current, active power, reactive power, frequency, phase angle, and rate-of-change quantities are commonly used as input features.

Statistical features such as mean, standard deviation, variance, skewness, kurtosis, minimum, and maximum values can be extracted from measurement windows. Frequency-domain features can also be derived using spectral analysis, while time-frequency techniques can provide information about transient behavior.

In renewable-energy-based microgrids, additional features may be obtained from photovoltaic generation, wind generation, load demand, battery operation, and environmental measurements. These variables can provide useful contextual information for distinguishing genuine faults from normal fluctuations.

Feature engineering can therefore be viewed as a mechanism for incorporating domain knowledge into an ML model. Nevertheless, conventional features may not always capture deviations from expected system behavior. This motivates the use of forecasting-based residual features.

2.5 Forecasting-Based Residual Features

Forecast residual was calculated as the difference between the observed value of load and PV and their respective forecast value. Gradient Boosting Regression (GBR) is applied for load forecasting and an RNN-based forecasting model is used for PV generation. The residual for each time step was obtained by subtracting the observed value from the forecast value:

$$r(t) = y(t) - \hat{y}(t) \quad (1)$$

where $y(t)$ is the observed value at time (t) and $\hat{y}(t)$ is the corresponding forecast value, r_t is the forecast residual. These residual sub-streams for load and PV signals were created and fed into the next feature set for fault detection.

Data used for training the forecasting models were limited to time prior to the forecast time to maintain the chronological structure of the data set and avoid using the future information for the generation of the residuals. Residual-based features therefore provide information that is different from the absolute measurement itself. For example, a PV output of a particular magnitude may be normal under one irradiance condition but abnormal under another. A forecasting residual can capture this deviation relative to the expected operating condition.

Time-series forecasting models such as Artificial Neural Networks, LSTM networks, and other recurrent architectures can be employed to generate expected system behavior. The resulting residuals can subsequently be supplied to an ML classifier or anomaly-detection model.

Despite their potential, forecasting residuals have received comparatively less attention as complementary features for microgrid fault classification. Integrating residual-based information with electrical measurements and imbalance-handling methods therefore represents an important research direction.

2.6 Oversampling and Synthetic Data Generation

Several techniques have been proposed to address class imbalance. Random oversampling increases the representation of minority samples by replicating existing observations. Although simple, excessive duplication can increase the risk of overfitting.

The Synthetic Minority Over-sampling Technique (SMOTE) addresses this limitation by generating synthetic minority samples using interpolation between existing minority observations [9]. SMOTE has been widely adopted in Binary fault detection and other classification applications.

Variants of SMOTE have also been developed to focus synthetic samples on difficult or borderline observations. Tomek Links can additionally be used to identify ambiguous samples near class boundaries. Chandra et al. [10] combined SMOTE and Tomek Links to improve data distribution and classification generalization in an air-quality prediction application.

GAN-based approaches provide another mechanism for minority-class data generation. Generative Adversarial Networks can learn the underlying distribution of minority-class samples and generate synthetic observations [11]. Such techniques may be useful when actual fault data are scarce.

However, synthetic data generation must be carefully controlled. Poorly generated synthetic fault samples may not represent physically meaningful operating conditions and can potentially introduce unrealistic patterns into the training dataset.

2.7 Class Imbalance in Fault Detection

Class imbalance is one of the most important challenges in supervised learning for critical infrastructure applications. In a typical microgrid dataset, normal operating conditions may account for the majority of observations, whereas actual fault events occur relatively infrequently.

When a classifier is trained on such a dataset, the learning process may become dominated by the majority class. A model can consequently achieve high overall accuracy simply by correctly classifying normal observations while incorrectly classifying a significant proportion of fault samples.

For example, if 95% of the observations correspond to normal operation and only 5% correspond to faults, a classifier that predicts every observation as normal could achieve 95% accuracy while providing no practical fault-detection capability. Therefore, accuracy alone is not sufficient for evaluating highly imbalanced fault-detection problems.

Metrics such as precision, recall, F1-score, balanced accuracy, specificity, sensitivity, and the confusion matrix provide a more informative assessment of classifier performance.

Table 1: Comparison of Research Directions

Reference	Dataset / Domain	Electrical Features	Forecast Residuals	SMOTE / Resampling	Deep / Machine Learning	Imbalance Metrics Reported	Real / Simulated Data
Kasimalla et al. [1]	Inverter Faults / Microgrid	Yes	No	No	DL (LSTM/ANN)	Precision, Recall	Simulated
Pan & Mandal [2]	DC Microgrid Clusters	Yes	No	No	SVM, Decision Trees	Accuracy	Simulated
Beikbabaie et al. [3]	100% Inverter-Based Grids	Yes	No	No	CNN, LSTM	Accuracy, F1	Simulated
Deng et al. [4]	Industrial Anomaly / Signals	Yes	No	GAN (Generative)	CNN	PR-AUC	Real
Liu et al. [5]	Photovoltaic System Faults	Yes	No	Semi-Supervised	SSL Models	F1, Precision	Real
Swana et al. [7]	Rotating Machinery	Signal	No	SMOTE + Tomek	SVM, Random Forest	F1, Recall	Real
Roy et al. [8, 11]	Grid-Connected Microgrid	Yes	No	No	Deep Relay	Accuracy, Precision	Simulated
Fahim et al. [10]	Microgrid Protection	Yes	No	No	KNN, SVM, Decision Tree	Accuracy	Simulated
Proposed Framework	Power Electronics-Dominated Microgrid (PEDG)	Yes	Yes	Yes (SMOTE on Train Set)	GB, CNN, LSTM	Precision, Recall, F1, Accuracy	Simulated (MIRIS)

Table 1 summarizes the existing research on microgrid fault detection, which focuses mainly on the use of raw electrical measurements (such as phase voltages, currents, frequency) and standard machine learning or deep learning models. In addition to the studies that tackle class imbalance with oversampling or GAN-based methods, some studies consider time-series prediction, while none of the studies considered both class imbalance and deep learning with power electronics dominated microgrid along with predictive forecast residuals (load and PV). The proposed framework addresses this gap by showing that the residual deviations can be very useful complementary features to detect rare fault events in severe class imbalance scenario.

2.8 Identified Gaps

The literature review reveals several research gaps that motivate the proposed work:

1. Limited integration of domain-specific residual features: Existing ML-based fault-detection studies predominantly use direct electrical measurements, while forecasting residuals representing deviations from expected load or renewable generation are less extensively explored.
2. Limited comparative analysis of imbalance-handling methods: Different strategies such as SMOTE, Tomek Links, cost-sensitive learning, and ensemble methods are often evaluated independently. A systematic comparison under a common experimental framework remains valuable.
3. Insufficient emphasis on minority-class performance: Many studies primarily report overall accuracy. More comprehensive evaluation using precision, recall, F1-score, balanced accuracy, and confusion matrices is required for rare-event fault detection.
4. Limited integration of forecasting and classification: Forecasting models and fault classifiers are frequently treated as independent components. Integrating forecast residuals into the classification

process may provide additional information for identifying abnormal behavior.

5. Limited deployment-oriented implementations: Several studies focus on offline classification performance without addressing real-time visualization, anomaly indication, alert generation, and practical inference requirements.
6. Limited analysis of model-training objectives: The influence of forecasting loss functions, including Mean Absolute Error (MAE) and Mean Squared Error (MSE), on residual quality and downstream anomaly sensitivity requires further investigation.
7. Generalization under changing operating conditions: Models trained under specific load, renewable-generation, and topology conditions may not generalize adequately to unseen operating scenarios. Evaluation under multiple operating conditions is therefore necessary.
8. Need for integrated fault-detection frameworks: There remains a need for a unified framework that combines forecasting, residual-based feature extraction, class-imbalance mitigation, ML-based classification, performance evaluation, and real-time visualization.

3. METHODOLOGY

This section presents the comprehensive methodological framework employed for fault detection in power electronics-dominated microgrids (PEDGs), emphasizing data-driven modeling, class imbalance mitigation, and robust model evaluation. The workflow comprises five core stages: data preparation, feature engineering, data balancing, model training and evaluation, and real-time visualization and alert generation.

3.1 Dataset Description and Preprocessing

The dataset employed in this study is derived from a simulated microgrid environment constructed using the MIRIS platform, which provides high-resolution real-time data for load demand, photovoltaic (PV) generation, and weather conditions. The data spans a continuous time window of several weeks, with a sampling frequency of 15 minutes, enabling fine-grained temporal analysis for anomaly and fault detection.

Table 2a: Summary of Datasets, Sampling Rates, and Key Variables

Dataset	Sampling	Reported size	Reported variables
MIRIS load	15 min	10,000 time steps	Timestamp, active power (kW), node ID
MIRIS PV	15 min	10,000 time steps	Timestamp, DC output (kW), inverter ID
Weather	15 min	10,000 time steps	Temperature, irradiance, wind speed, humidity
Forecast datasets	15 min	10,000 time steps	GBR/RNN forecasts for load/PV

3.1.1 Load Data (MIRIS Load Dataset)

The load dataset records active power consumption across various nodes in the microgrid. It reflects daily and seasonal variability, consumer behavior, and potential load anomalies. Key characteristics include:

1. Resolution: 15-minute intervals
2. Features: Timestamp, Active Power (kW), Node ID Size: 10,000 time steps
3. Format: CSV
4. Notable Attributes: Captures sudden drops or spikes in load that may indicate transient faults or system instabilities.

3.1.2 PV Generation Data (MIRIS PV Dataset)

The PV dataset measures solar power output from multiple photovoltaic arrays integrated within the microgrid. PV generation is highly sensitive to weather fluctuations and system-level faults such as inverter failures or panel mismatches.

1. Resolution: 15-minute intervals
2. Features: Timestamp, DC Output (kW), Inverter ID Size: 10,000 time steps
3. Forecast Models Used: GBR and RNN-based forecasting models for residual generation

3.1.3 Weather Data

Weather conditions significantly affect PV generation and can indirectly impact fault occurrence due to thermal or irradiance stress on electronic components.

The weather dataset includes:

- Parameters: Ambient Temperature ($^{\circ}\text{C}$), Solar Irradiance (W/m^2), Wind Speed (m/s), Humidity (%)
Resolution: 15-minute intervals
- Integration: Aligned by timestamp with the load and PV datasets for multi-variable feature construction

3.1.4 Forecast Datasets and Residual Generation

To enable residual-based feature engineering, two separate forecasting models were applied:

- Gradient Boosted Regressor (GBR): Used to generate point forecasts for load and PV
- Recurrent Neural Network (RNN): Captures sequential dependencies and non-linearities for improved forecast accuracy

The residuals—i.e., the difference between actual and forecasted values—serve as indicators of abnormal system behavior, and are key features for fault detection modeling.

3.1.5 Feature Set

Table 3: Classification and Description of Engineering Feature Groups

Feature group	Examples	Purpose
Raw operational	Load, PV output	Represent absolute operating state
Forecast residual	Actual – forecast	Represent deviation from expected behavior
Temporal	Lag 1, lag 2	Capture short-term dynamics
Rolling statistics	Mean, SD, min, max	Represent local variation
Residual statistics	Residual volatility	Represent instability/anomalous deviation
Weather	Temperature, irradiance, wind, humidity	Contextualize PV/load variability

3.1.6 Fault Labeling and Labeling Methodology

Specific physical and operational disturbance scenarios were injected in the MIRIS simulation environment under different operating conditions (such as high solar irradiance, high load, dynamic weather, etc.) to create the binary fault labels (0 – normal, 1 – fault). The faults were added as distinct fault events in the 10,000 time-step data set without overlapping each other. To simulate realistic microgrid dynamic behavior, fault durations ranged from single 15-minute intervals (transient disconnects) up to 2-hour sustained periods (communication blackouts). In this study, fault detection is modeled as a binary classification problem in which all individual fault scenarios are grouped together into a single fault class (Class 1), which corresponds to about 3.5% (~350 samples) of the total data set to simulate the worst-case scenario in a real world application of class imbalance. The injection parameters, thresholds, event counts, and sample distributions are summarized in Table 2b.

Table 2b: Summary of Fault Injection Mechanisms and Sample Distributions

Fault Type	Injection Mechanism / Modified Parameter	Threshold / Severity Definition	Duration Per Event	Number of Events	Total Samples (15-min rate)
Inverter Failure	Direct drop/zeroing of inverter AC power output	$P_{\text{out}} \text{ drop} \geq 80\%$ relative to PV forecast	30–60 min (2–4 steps)	25	~80
Over-Voltage	Terminal grid voltage step boost	$V > 1.10 \text{ p.u.}$ ($>110\%$ nominal)	15–45 min (1–3 steps)	30	~60
Under-Voltage	Bus voltage drop / simulated grid sag	$V < 0.90 \text{ p.u.}$ ($<90\%$ nominal)	15–45 min (1–3 steps)	30	~60
Load Disconnect	Instantaneous branch load shedding	Active drop $\Delta P_{\text{load}} \geq 50\%$	15–30 min (1–2 steps)	40	~50
Communication	Packet loss / frozen	Static readings	60–120	20	~100

Failure	telemetry feed	held for >2 intervals	min (4–8 steps)		
Total Class 1 Faults	Aggregated binary fault class	Any active fault state	Varies (15–120 min)	145 events	~350 (~3.5%)

3.1.7 Preprocessing Steps

To prepare the dataset for modeling:

1. All datasets were joined using a common timestamp key
2. Missing values were imputed using forward/backward filling and interpolation
3. Data was normalized using Min-Max scaling for neural network models
4. Rolling statistical features (mean, std, min, max) were computed over short (1–2 hour) windows to capture local patterns
5. Exploratory Data Analysis (EDA) was conducted to verify fault scenarios, detect noise, and visualize trends

3.2 Feature Engineering

Effective feature engineering was critical to enhancing the sensitivity of the models to fault events in a class-imbalanced environment. The process involved transforming raw time-series data into informative predictors that capture both system dynamics and anomalies.

3.2.1 Residual-Based Features

A novel combination in this study is the use of residuals—the deviation between actual and forecasted values—for both load and PV generation. Residuals were calculated using predictions from two forecasting models:

1. Gradient Boosted Regression (GBR) for load forecasts
2. Recurrent Neural Networks (RNNs) for PV forecasts

These residuals act as indicators of abnormal behavior, flagging potential fault events where system behavior diverges from learned patterns.

3.2.2 Temporal and Statistical Features

To capture short-term and long-term patterns, the following features were generated using sliding windows:

1. Moving average, standard deviation, min, and max over 1-hour and 2-hour intervals
2. Lagged values (e.g., $t - 1$, $t - 2$) to introduce time dependency for sequential models like LSTM
3. Rolling residual volatility to indicate system instability

3.2.3 External Environmental Features

Weather-based features were included to account for external influence on PV performance and microgrid balance. These included:- Ambient temperature, Solar irradiance, Wind speed, Humidity
These features helped to distinguish between natural fluctuations (e.g., cloudy skies) and genuine system anomalies.

3.2.4 Feature Normalization and Transformation

All continuous features were scaled using Min-Max normalization to ensure uniformity across models. Correlation analysis and mutual information scores were used to validate feature relevance.

3.3 Data Splitting and Class Balancing

The microgrid data was split chronologically into three parts: training, validation, and final testing, to avoid temporal leakage and maintain the temporal structure of the microgrid data. The first part of the data was used to train the model, the second part of the data was used to validate the model and select the best model and the last unseen chronological part of the data was used to evaluate the performance of the best model. Only the training data was used for SMOTE, the validation and test sets kept their original class distribution.

Table 4: Chronological Data Splitting and Evaluation Protocol

Subset	Recommended protocol	Purpose
Training	Earlier chronological period	Model fitting and SMOTE
Validation	Later part of training period	Hyperparameter tuning / early stopping
Test	Final unseen chronological period	Independent evaluation

3.4 Model Development and Training

Multiple machine learning and deep learning models were trained on the balanced dataset. These include:

1. Traditional ML Models: Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Logistic Regression, Gradient Boosting Classifier
2. Deep Learning Models: Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM)

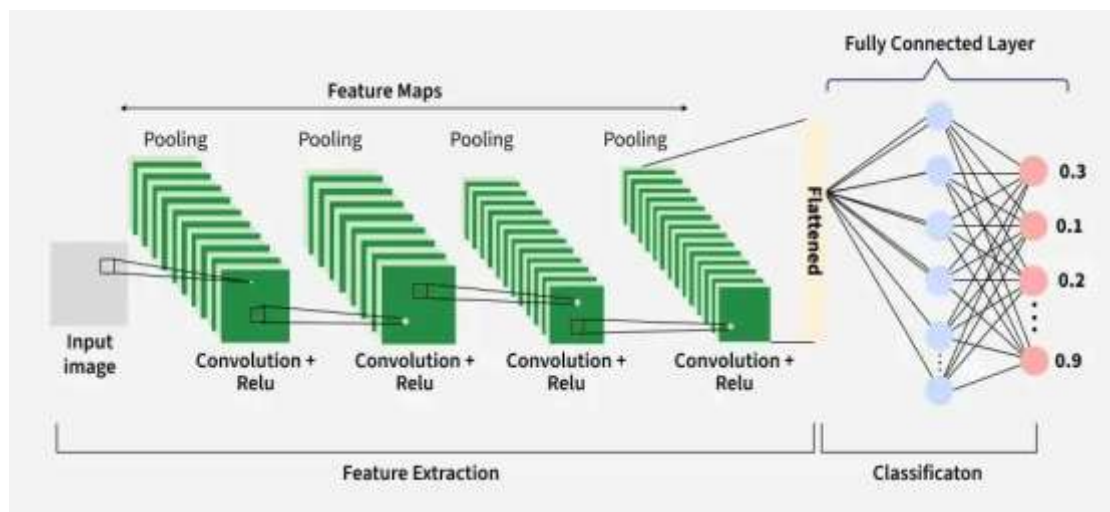
3.4.1 CNN model

A sequential CNN model was developed, consisting of:

1. Two 1D convolutional layers
2. A flattening layer
3. A dense layer with dropout regularization
4. A two-neuron output layer with softmax activation was used for binary classification.

The CNN was trained using both Mean Absolute Error (MAE) and Mean Squared Error (MSE) loss functions. Early stopping and softmax activation were used to prevent overfitting and improve generalization.

Table 5 presents the architectural summary of the Convolutional Neural Network (CNN) model referred to as Sequential 2. The network is composed of six layers, designed to process one-dimensional time-series data. The model starts with two consecutive Conv1D layers, which learn local temporal patterns from the input residual features. These are followed by a Flatten layer that converts the 3D output into a 1D vector, preparing it for dense classification layers. The penultimate dense layer includes a Dropout operation to prevent overfitting.

**Figure 1:** CNN Model

(Credits:https://media.geeksforgeeks.org/wp-content/uploads/20260217113409515144/working_of_cnn_.webp)

The final dense layer uses a softmax activation function to output binary class probabilities (fault or no fault). This model has a total of 440,802 trainable parameters, indicating a moderately complex architecture suitable for structured time-series classification tasks.

Table 5: Model Summary for sequential 2

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 214, 64)	256
conv1d 1 (Conv1D)	(None, 212, 32)	6,176
flatten (Flatten)	(None, 6784)	0

dense (Dense)	(None, 64)	434,240
dropout (Dropout)	(None, 64)	0
dense 1 (Dense)	(None, 2)	130

3.4.2 LSTM model

The LSTM model comprised:

1. Two LSTM layers for sequential pattern learning
2. Dropout regularization
3. Fully connected dense layers for classification

The impact of MAE vs. MSE loss functions was similarly assessed in this model.

Table 6 outlines the architecture of the Long Short-Term Memory (LSTM) network designated as Sequential 3. The model is constructed using five layers. It begins with two stacked LSTM layers that capture temporal dependencies and sequential trends in the data. These are followed by a Dropout layer to reduce the risk of overfitting during training. The dense layer (dense 2) then transforms the LSTM output into a compact feature representation, which is passed to a final classification layer (dense 3) to generate binary predictions. This model has 31,554 trainable parameters, making it computationally lighter than the CNN architecture and better suited for learning long-term dependencies in time-series fault patterns.

Table 6: Model Summary for sequential 3

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 216, 64)	16,896
lstm 1 (LSTM)	(None, 32)	12,416
dropout 1 (Dropout)	(None, 32)	0
dense 2 (Dense)	(None, 64)	2,112
dense 3 (Dense)	(None, 2)	130

3.5 Model Performance

The performance of each model was evaluated using standard metrics, including accuracy, precision, recall, MSE, MAE and F1-score, calculated separately for the majority class (Class 0: Non-fault) and the minority class (Class 1: Fault). The results are summarized in Table 7.

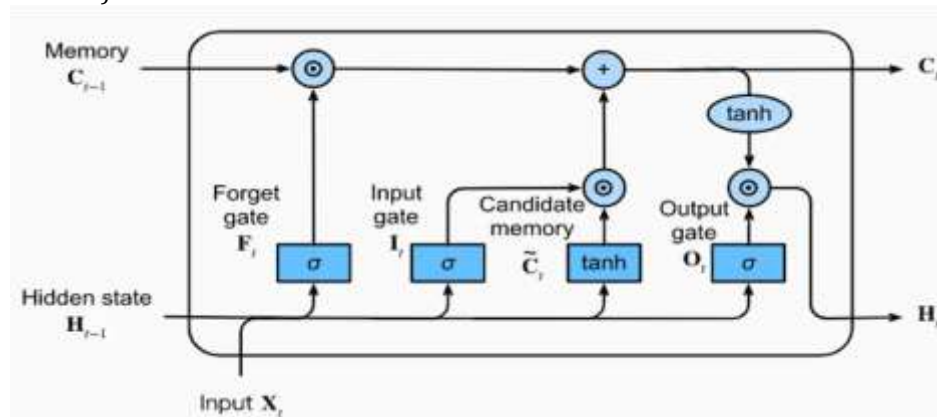


Figure 2: Architecture of an LSTM Unit (Credits: https://d2l.ai/chapter_recurrent-modern/lstm.html)

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

(Proportion of correct predictions over total cases)

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

(Proportion of true positives among predicted positives)

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

(Proportion of actual positive fault events correctly identified)

$$F1-Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

(Harmonic mean of precision and recall, crucial for imbalanced evaluation)

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - Y'_i)^2 \quad (5)$$

(Mean Squared Error: penalizes larger errors more heavily)

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - Y'_i| \quad (6)$$

(Mean Absolute Error: average of absolute prediction errors)

3.6 Visualization

To ensure the interpretability of model performance and to support the analysis of fault detection outcomes, a set of visualization tools was developed. These tools assist in understanding how the machine learning models respond to variations in system behavior, especially under imbalanced data conditions.

Time-series plots were used to compare:

1. Actual vs. Forecasted Load and PV Generation: These plots reveal deviations captured by residual features and help highlight system anomalies visually.
2. Residual Trends Over Time: Residuals, as engineered features, are plotted to detect abnormal spikes or patterns that correlate with known fault events.

These visualizations aid in identifying temporal alignment between model-predicted faults and real deviations in microgrid performance.

4. RESULTS AND DISCUSSION

This section presents a comprehensive evaluation of the fault detection models developed using traditional machine learning and deep learning techniques. Particular emphasis is placed on class-wise performance, the impact of class imbalance mitigation, and the choice of loss functions on model behavior. The results validate the proposed methodology and its effectiveness in detecting rare fault events in power electronics-dominated microgrids (PEDGs).

Table 7: Comparison of Machine Learning Models for Fault Detection Performance

Model	Class 0 Precision	Class 1 Precision	Class 0 Recall	Class 1 Recall	Class 0 F1-score	Class 1 F1-score	Accuracy
SVM	0.99	0.02	0.81	0.01	0.89	0.01	80.24%
Gradient Boosting Classifier	0.99	1.00	1.00	0.96	0.99	0.98	99.80%
K-Nearest Neighbors (KNN)	0.99	0.04	0.89	0.02	0.94	0.03	88.23%
Logistic Regression	1.00	0.21	0.97	0.15	0.98	0.17	97.18%
CNN Model	1.00	0.4	0.99	0.32	0.99	0.36	99.00%
CNN (MAE Loss)	1.00	0.38	0.99	0.3	0.99	0.33	99.00%
CNN (MSE Loss)	1.00	0	1.00	0	1.00	0	99.00%
LSTM (MAE Loss)	0.99	0	1.00	0	0.99	0	99.00%
LSTM (MSE Loss)	1.00	0.07	0.89	0.04	0.94	0.05	88.00%
CNN (Softmax & Early Stopping)	0.99	0.9	0.99	0.84	0.99	0.87	99.00%

4.1 Majority Class (Class 0) Analysis

Most models demonstrated near-perfect precision and accuracy for Class 0, which is expected due to its dominance in the dataset. Logistic Regression, CNN, and Gradient Boosting achieved precision scores above 0.99 for normal condition detection, suggesting their reliability in identifying standard operating scenarios.

4.2 Minority Class (Class 1) Analysis

Detection of rare fault events proved more challenging due to severe class imbalance. Several key observations were made:

1. Gradient Boosting was the highest overall accuracy (99.80%) and with precision (1.00), recall (0.96) and F1-score (0.98) for the fault class. The result shows that the model made no mistake in the test set it was evaluated with in predicting faults, but correctly predicted 96% of the actual faults.
2. CNN with Softmax and Early Stopping performed strongly (0.90 precision for Class 1), showing that deep models can generalize well when regularized appropriately.
3. The CNN network that was trained with the use of MSE had a perfect accuracy of 99% but has a fault-class recall and F1-score of 0.00. This means that the high accuracy was mainly attributed to correctly classify the majority of normal class and not to detect the faults well. The findings underscore the need to assess minority-class precision, recall, F1 score and imbalance-aware metrics besides accuracy.
4. The results of comparing MAE- and MSE-based training were model dependent. For CNN, MAE outperformed MSE, achieving 0.33 fault-class F1 score while MSE achieved 0.00. The LSTM results were not consistent, though: Both loss functions had very low performance on the minority classes. The results indicate that the loss function effect is not universally best and do not imply a better loss function, but rather depend on the model architecture and the data.
5. Other models like SVM, KNN, and LSTM (with MSE or MAE) struggled to detect fault events effectively, producing low or zero precision.
6. The present study formulates fault detection as a binary classification problem, where all fault scenarios are grouped into a single fault class.

This highlights the critical role of model selection in imbalanced classification contexts.

4.3 Overall Accuracy

Gradient Boosting worked best in the test set, with an overall accuracy of 99.80% and a fault-class precision of 1.00. But accuracy and precision are not enough to describe the performance for the class imbalance observed. Hence, the effectiveness of the model should also be judged using fault-class recall, F1-score, PR-AUC, balanced accuracy and MCC. This indicates that while most models performed well in detecting the majority class, the Gradient Boosting Classifier and the enhanced CNN models were particularly effective in handling the class imbalance and achieving high overall accuracy.

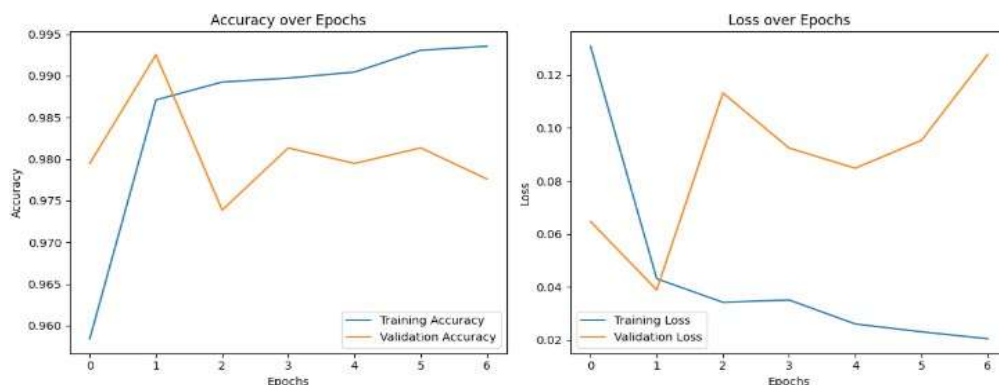


Figure 3: Training and Validation Accuracy/Loss over Epochs for CNN (MAE Loss)

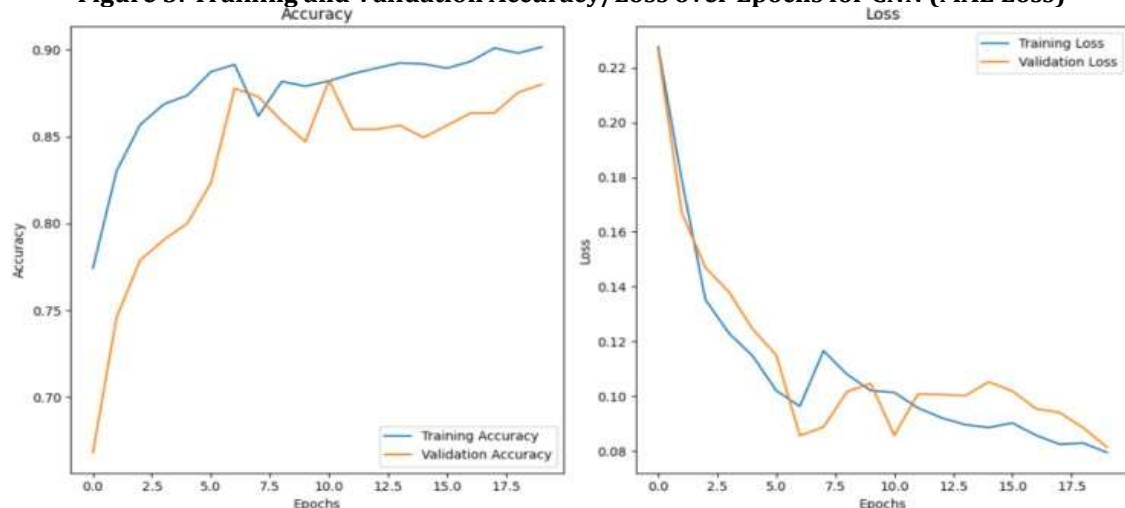


Figure 4: Training and Validation Accuracy/Loss over Epochs for Sequential LSTM).

4.4 Impact of Loss Functions

The choice of loss function was found to significantly affect performance in deep learning models:

1. The effect of MAE and MSE loss functions was model dependent.
2. For the CNN model, MAE resulted in improved minority-class performance compared with MSE.
3. However, the LSTM results did not demonstrate a similar improvement, indicating that the influence of the loss function depends on the model architecture and data characteristics.

4.5 Effectiveness of Class Imbalance Mitigation

The application of SMOTE played a pivotal role in balancing the training set. Without SMOTE, fault detection performance dropped sharply, especially in non-ensemble models. Only the training set was used for SMOTE to prevent data leakage, after the split of chronological data. The validation and test sets were kept as is with regard to their class distribution with no oversampling. To assess the model performance, data was used that was not shown to the model in the oversampling and training process.

1. Gradient Boosting naturally handles imbalance well due to its sequential training mechanism.
2. SMOTE combined with CNN and MAE loss led to high performance for Class 1, validating the effectiveness of the hybrid approach.

Figure 3 displays two graphs illustrating the performance of a machine learning model during the training and validation phases over several epochs: "Accuracy over Epochs" and "Loss over Epochs."

Figure 4 displays the training performance of a machine learning model, showing both accuracy and loss over 18 epochs.

4.6 Key Findings and Practical Implications

1. Gradient Boosting Classifier is highly effective and interpretable, suitable for real-world deployment in microgrids.
2. CNN with Softmax and MAE offer a strong balance of accuracy and generalization, especially in high-dimensional or temporal settings.
3. Loss function and class balancing strategy significantly influence fault detection performance and should be carefully selected in future deployments.

5. HANDLING OVERFITTING

The overfitting observed in the CNN model's performance graphs, where training accuracy continues to improve while validation accuracy plateaus or fluctuates and validation loss increases, can be addressed by employing several techniques:

5.1 Early Stopping:

Monitor the model's performance on the validation set during training. When the validation performance (e.g., validation loss) stops improving for a certain number of epochs (patience), halt the training process and revert to the model weights from the epoch with the best validation performance. This prevents the model from continuing to learn noise in the training data and memorizing it, ensuring better generalization to unseen data.

5.2 Regularization Techniques (L1/L2 Regularization):

Add penalty terms to the model's loss function based on the magnitude of the model's weights (L1 penalizes the absolute value, L2 penalizes the squared value). Discourages large weights, which can lead to complex models that overfit. L1 can also lead to sparse models, effectively performing feature selection.

5.3 Dropout:

Randomly "drop out" (temporarily deactivate) a fraction of neurons during each training iteration. Prevents neurons from becoming overly reliant on specific connections and encourages the network to learn more robust features, as different subsets of neurons are active in each training step.

5.4 Data Augmentation (especially for Image Data like CNNs):

Artificially increase the size and diversity of the training dataset by applying various transformations (e.g., rotations, flips, shifts, zooms, brightness changes) to existing data. Exposes the model to a wider range of variations, making it more robust and less likely to overfit to the specific examples in the original training set.

5.5 Model Simplification:

If the model is too complex for the given data (e.g., too many layers, too many neurons per layer), reduce its complexity. A simpler model is less likely to memorize the training data and more likely to capture the underlying patterns, leading to better generalization.

5.6 Batch Normalization:

Normalize the inputs to each layer within a neural network, usually before the activation function. Stabilizes and speeds up training by reducing internal covariate shift, and can also act as a mild regularizer, potentially reducing the need for other strong regularization techniques.

These findings have significant implications for the design and implementation of fault detection systems in microgrids. The choice of model and the strategies employed to address class imbalance are critical for ensuring the accuracy and reliability of these systems. The superior performance of the Gradient Boosting Classifier and the enhanced CNN models suggests that ensemble methods and deep learning architectures, when appropriately configured, can effectively handle the challenges posed by class imbalance in microgrid fault detection. The results also highlight the importance of considering the impact of loss function selection on model performance.

The detailed analysis presented in this section provides valuable insights into the effectiveness of various machine learning models for microgrid fault detection and the importance of addressing class imbalance. These findings contribute to the development of more robust and reliable fault detection systems for improved microgrid operation and safety. Further research could explore hybrid models combining the strengths of different techniques and investigate more advanced class imbalance mitigation strategies.

6. CONCLUSION

This study investigated a forecast-residual-based machine learning framework for binary fault detection in power electronics-dominated microgrids under severe class imbalance. The framework combines load and photovoltaic measurements with forecast residuals, temporal statistics, lagged variables, and environmental information. Multiple conventional machine learning and deep learning models were evaluated, while SMOTE was applied exclusively to the training data to reduce the influence of class imbalance.

The currently reported experiments show that Gradient Boosting achieved an accuracy of 99.80%, with 1.00 precision, 0.96 recall, and 0.98 F1-score for the fault class, demonstrating strong fault-detection performance on the evaluated test set. These results indicate the potential of ensemble and deep-learning methods for identifying rare fault events in the evaluated simulated microgrid environment.

Nevertheless, the results also demonstrate the importance of minority-class evaluation. High overall accuracy can coexist with poor fault detection when the normal class dominates the dataset. Consequently, future versions of the experimental evaluation should emphasize fault recall, F1-score, PR-AUC, balanced accuracy, MCC, and confusion-matrix analysis.

The principal methodological contribution of the study is the integration of forecast residuals with operational and temporal features within an imbalance-aware fault-detection framework. To establish the independent contribution of the proposed representation, feature ablation, imbalance ablation, repeated experiments, and chronological generalization tests should be included.

The present study is based on a simulated microgrid environment and therefore does not establish real-world deployment performance. Future work should validate the framework using measured microgrid data, multiple operating conditions, unseen fault types, hardware-in-the-loop experiments, and real-time implementation. Extension from binary fault detection to multiclass fault diagnosis and subsequent integration with component-health estimation and maintenance scheduling are also important directions. Overall, the study provides a structured basis for investigating forecast deviations as complementary information for machine-learning-based fault detection in power-electronics-dominated microgrids.

7. FUTURE WORK

Building on the findings of this study, future research directions include:

1. Development of hybrid models combining ensemble learning and deep architectures for improved robustness
2. Application of generative models (e.g., GANs) for more realistic data augmentation in imbalanced scenarios
3. Extension to multi-class fault classification, enabling finer-grained fault diagnostics
4. Real-world validation using live microgrid operational data, ensuring generalization and practical deployment feasibility

These enhancements aim to further strengthen the reliability, adaptability, and deployment readiness of fault detection systems in modern smart grid environments.

Declaration of AI Assistance

AI-based tools were used during manuscript preparation for language editing, grammar correction, and improvement of readability. The authors reviewed and validated all content and take full responsibility for the final manuscript.

References

- [1] S. R. Kasimalla, K. Park, J. Hong, Y.-J. Kim, and H. Lee, "AI-enhanced inverter fault and anomaly detection system for distributed energy resources in microgrids," arXiv preprint, arXiv:2411.08761, 2024. [Online]. Available: <https://doi.org/10.48550/ARXIV.2411.08761>
- [2] P. Pan and R. K. Mandal, "Fault detection and classification in DC microgrid clusters," *Engineering Research Express*, vol. 5, no. 2, p. 025010, 2023. [Online]. Available: <https://doi.org/10.1088/2631-8695/accad2>
- [3] M. Beikbabaie, M. Lindemann, M. H. Kapourchali, and A. Mehrizi-Sani, "Machine learning-based protection and fault identification of 100% inverter-based microgrids," arXiv preprint, arXiv:2405.07310, 2024. [Online]. Available: <https://doi.org/10.48550/ARXIV.2405.07310>
- [4] C. Deng, Z. Deng, S. Lu, M. He, J. Miao, and Y. Peng, "Fault diagnosis method for imbalanced data based on multi-signal fusion and improved deep convolution generative adversarial network," *Sensors*, vol. 23, no. 5, p. 2542, 2023. [Online]. Available: <https://doi.org/10.3390/s23052542>
- [5] Q. Liu, X. Wang, B. Yang, Z. Wang, Y. Liu, and X. Guan, FSR-SSL: A fault sample rebalancing framework based on semi-supervised learning for PV fault diagnosis, *IET Renewable Power Generation*, vol. 16, no. 15, pp. 2667–2681, 2022. [Online]. Available: <https://doi.org/10.1049/rpg2.12458>
- [6] M. Qian and Y.-F. Li, Weakly supervised-based oversampling for high imbalance and high dimensionality data classification, arXiv preprint, arXiv:2009.14096, 2020. [Online]. Available: <https://doi.org/10.48550/ARXIV.2009.14096>
- [7] E. F. Swana, W. Doorsamy, and P. Bokoro, "Tomek Link and SMOTE approaches for machine fault classification with an imbalanced dataset," *Sensors*, vol. 22, no. 9, p. 3246, 2022. [Online]. Available: <https://doi.org/10.3390/s22093246>
- [8] M. Isangediok and K. Gajamannage, "Fraud detection using optimized machine learning tools under imbalance classes," in *Proc. 2022 IEEE Int. Conf. on Big Data (Big Data)*, Osaka, Japan, 2022, pp. 4275–4284. [Online]. Available: <https://doi.org/10.1109/bigdata55660.2022.10020723>
- [9] B. Roy, S. Adhikari, S. Datta, K. J. Devi, A. D. Devi, F. Alsaif, S. Alsulamy, and T. S. Ustun, "Deep learning based relay for online fault detection, classification, and fault location in a grid-connected microgrid," *IEEE Access*, vol. 11, 2023. [Online].
- [10] S. Rahman Fahim, S. K. Sarker, S. M. Mueen, Md. R. I. Sheikh, and S. K. Das, "Microgrid fault detection and classification: Machine learning based approach, comparison, and reviews," *Energies*, vol. 13, no. 13, p. 3460, 2020. [Online]. Available: <https://doi.org/10.3390/en13133460>
- [11] Bappa Roy, Shuma Adhikari, Subir Datta, and Kharibam J. Devi, "Deep Learning Based Relay for Online Fault Detection, Classification, and Fault Location in a Grid-Connected Microgrid," *IEEE Access*, vol. PP, no. 99, pp. 1–1, Jan. 2023, doi: 10.1109/ACCESS.2023.3285768. Licensed under CC BY 4.0.
- [12] Abdelsattar Mohamed Saeed M, Rasslan A, Emad-Eldeen A, "Comparative Analysis of Machine Learning Techniques for Fault Detection in Solar Panel Systems," *SVU-International Journal of Engineering Sciences and Applications*, vol. 5, pp. 140–152, 2024. <https://doi.org/10.21608/svusrc.2024.279389.1198>
- [13] Z. Zhao, T. Cui, S. Ding, J. Li, and A. G. Bellotti, Resampling Techniques Study on Class Imbalance Problem in Credit Risk Prediction, vol. 12, no. 5, Feb. 2024. <https://doi.org/10.3390/math12050701>.
- [14] W. Chandra, B. Suprihatin, and Y. Resti, Median-KNN Regressor-SMOTE-Tomek Links for Handling Missing and Imbalanced Data in Air Quality Prediction, vol. 15, no. 4, Apr. 2023. <https://doi.org/10.3390/sym15040887>.

- [15] J. X. C. Ke, A. DhakshinaMurthy, R. B. George et al., The effect of resampling techniques on the performances of machine learning clinical risk prediction models in the setting of severe class imbalance: development and internal validation in a retrospective cohort, vol. 4, Nov. 2024. <https://doi.org/10.1007/s44163-024-00199-0>.
- [16] A. Newaz, A. U. R. Adib, and T. Jabid, iCost: A Novel Instance Complexity Based Cost-Sensitive Learning Framework, arXiv preprint, arXiv:2401.11526, 2024. <https://arxiv.org/abs/2401.11526>
- [17] L. Wu, H. He, J. Zhao, and S. Yu, "Handling Imbalanced Medical Datasets: A Decade Review," *Artificial Intelligence Review*, 2024. <https://doi.org/10.1007/s10462-024-10546-1>
- [18] Z. Fang, Y. Zhang, and J. Tang, "A Cost-Sensitive Fault Diagnosis Method Based on Ensemble Learning for Imbalanced Industrial Data," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1434–1442, Feb. 2023. <https://doi.org/10.1109/TII.2022.3186931>
- [19] A. A. Khan, O. Chaudhari, and R. Chandra, "A Review of Ensemble Learning and Data Augmentation Models for Class Imbalanced Problems: Combination, Implementation and Evaluation," arXiv preprint, arXiv:2304.02858, 2023.
- [20] Y. Li and C. Liu, "Attention-stacked Generative Adversarial Network (AS-GAN)-Empowered Sensor Data Augmentation for Online Monitoring of Manufacturing System," arXiv preprint, arXiv:2306.06268, 2023.
- [21] J. Huang, F. Zhao, and L. Chen, "Defect Detection Network in PCB Circuit Devices Based on GAN-Enhanced YOLOv11," arXiv preprint, arXiv:2501.06879, 2025.
- [22] X. Gui and J. Zhang, "Deep Metric Learning Model for Imbalanced Fault Diagnosis," 2021. [Online]. Available: <http://arxiv.org/abs/2107.03786>
- [23] O. Wu, "Rethinking Class Imbalance in Machine Learning," vol. 14, no. 8, pp. 1–14, 2023. [Online]. Available: <http://arxiv.org/abs/2305.03900>
- [24] S. Lu, Z. Gao, Q. Xu, C. Jiang, A. Zhang, and X. Wang, "Class-Imbalance Privacy-Preserving Federated Learning for Decentralized Fault Diagnosis With Biometric Authentication," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9101–9111, 2022, doi: 10.1109/TII.2022.3190034.
- [25] M. Uzair, M. Eskandari, L. Li, and J. Zhu, "Machine Learning Based Protection Scheme for Low Voltage AC Microgrids," *Energies*, vol. 15, no. 24, pp. 1–19, 2022, doi: 10.3390/en15249397.
- [26] N. C. Yang and H. Ismail, "Robust Intelligent Learning Algorithm Using Random Forest and Modified-Independent Component Analysis for PV Fault Detection: In Case of Imbalanced Data," *IEEE Access*, vol. 10, pp. 41119–41130, 2022, doi: 10.1109/ACCESS.2022.3166477.
- [27] A. Rohan, "Holistic Fault Detection and Diagnosis System in Imbalanced, Scarce, Multi-Domain (ISMD) Data Setting for Component-Level Prognostics and Health Management (PHM)," *Mathematics*, vol. 10, no. 12, 2022, doi: 10.3390/math10122031.
- [28] B. Hu, J. Liu, R. Zhao, Y. Xu, and T. Huo, "A New Fault Diagnosis Method for Unbalanced Data Based on 1DCNN and L2-SVM," *Applied Sciences*, vol. 12, no. 19, 2022, doi: 10.3390/app12199880.
- [29] Y. M. Nsaif, M. S. Hossain Lipu, A. Hussain, A. Ayob, Y. Yusof, and M. A. A. M. Zainuri, "A Novel Fault Detection and Classification Strategy for Photovoltaic Distribution Network Using Improved Hilbert–Huang Transform and Ensemble Learning Technique," *Sustainability*, vol. 14, no. 18, 2022, doi: 10.3390/su141811749.
- [30] B. L. H. Nguyen, T. Vu, T. T. Nguyen, M. Panwar, and R. Hovsapien, "1-D Convolutional Graph Convolutional Networks for Fault Detection in Distributed Energy Systems," in *1st IEEE Industrial Electronics Society Annual On-Line Conference (ONCON)*, 2022, doi: 10.1109/ONCON56984.2022.10126859.