eng

MDPI

Article

Tree-Based Algorithms and Incremental Feature Optimization for Fault Detection and Diagnosis in Photovoltaic Systems

Khaled Chahine

College of Engineering and Technology, American University of the Middle East, Kuwait; khaled.chahine@aum.edu.kw

Abstract: Despite their significant environmental benefits, solar photovoltaic (PV) systems are susceptible to malfunctions and performance degradation. This paper addresses detecting and diagnosing faults from a dataset representing a 250 kW PV power plant with three types of faults. A comprehensive dataset analysis is conducted to improve the dataset quality and uncover intricate relationships between features and the target variable. By introducing novel feature importance averaging techniques, a two-phase fault detection and diagnosis framework employing tree-based models is proposed to identify faults from normal cases and diagnose the fault type. An ensemble of six tree-based classifiers, including decision trees, random forest, Stochastic Gradient Boosting, LightGBM, CatBoost, and Extra Trees, is trained in both phases. The results show 100% accuracy in the first phase, particularly with the Extra Trees classifier. In the second phase, Extra Trees, XGBoost, LightGBM, and CatBoost achieve similar accuracy, with Extra Trees demonstrating superior training and convergence speed. This study then incorporates Explainable Artificial Intelligence (XAI), utilizing LIME and SHAP analyzers to validate the research findings. The results highlight the superiority of the proposed approach over others, solidifying its position as an innovative and effective solution for fault detection and diagnosis in PV systems.

Keywords: fault detection; fault diagnosis; binary classification; multi-class classification; tree-based classifiers; feature importance; explainable AI

Academic Editors: Juvenal Rodriguez-Resendiz, Akos Odry, José Manuel Álvarez-Alvarado and Marco Antonio Aceves-Fernandez

Received: 13 November 2024
Revised: 30 December 2024
Accepted: 16 January 2025
Published: 20 January 2025

Citation: Chahine, K. Tree-Based Algorithms and Incremental Feature Optimization for Fault Detection and Diagnosis in Photovoltaic Systems. *Eng* **2025**, *6*, 20. https://doi.org/10.3390/eng6010020

1. Introduction

Solar PV technology plays a crucial role in achieving a global low-carbon energy system and progressing toward carbon neutrality. Over the past decade, advancements in the PV industry have reduced the Levelized Cost of Electricity (LCoE) for PV energy by 85%, making it one of the most cost-effective sources of electricity worldwide. For instance, in Saudi Arabia, the cost of PV-generated electricity reached as low as USD 0.0104 per kWh in April 2021 [1]. The growing interest in renewable energy, particularly solar PV systems, stems from the need to reduce the environmental impact and market dependence on fossil fuels, which are both non-renewable and polluting [2]. In contrast, PV systems rely on solar irradiance, providing a sustainable and long-lasting energy solution.

Like all industrial assets and power systems, solar PV systems are prone to faults at various stages of operation. These faults, whether related to maximum power point tracking, environmental conditions, or component malfunctions, can lead to power generation losses, supply disruptions, or even complete system shutdowns. A 2010 study estimated that faults contribute to at least an 18.9% reduction in annual energy output from PV systems [3]. Therefore, effective fault detection and diagnosis are essential to

ensure reliable operation, especially in industries that rely heavily on PV systems for power stability and safety.

Traditional fault detection methods in PV arrays, such as GFDI fuses, residual current measurement, isolation resistance measurement, and reflectometry, require specialized domain knowledge. However, the rise of artificial intelligence, particularly machine learning, has transformed fault detection and diagnosis in the energy sector. By utilizing data generated from system operations, machine learning models can provide deeper insights into the behavior of energy systems, improving parameter estimation, anomaly detection, and fault diagnosis in complex environments [4–8]. This emerging approach offers a powerful, data-driven solution for identifying and addressing faults in PV systems.

This paper contributes to both sustainability goals and advancements in renewable energy by improving fault detection and diagnosis in solar PV systems. By reducing the environmental footprint of energy production and enhancing the efficiency of these systems, this work supports the transition to cleaner energy sources. The main contributions of this paper are as follows:

- Innovative Framework: Introducing an innovative tree-based two-phase framework for fault detection and diagnosis, achieving an unprecedented 100% accuracy.
- Novel Feature Importance Technique: Developing a novel technique of feature importance averaging employing tree-based algorithms and demonstrating high reliability and alignment with the final results.
- Optimization of Data Dimensions: Proposing a distinctive technique for optimizing data dimensions and feature contributions through an incremental approach, contributing to enhanced efficiency.
- Explainable Artificial Intelligence: Using two XAI techniques, ensuring high transparency and establishing the validity of findings in fault detection and diagnosis.

The remainder of this paper is structured as follows. Section 2 discusses various PV fault types and their impacts on the system's performance. Section 3 presents a comprehensive literature review, examining the diverse landscape of machine learning models within PV systems. In Section 4, the proposed framework is described, detailing its key components and methodology. The results and discussions are introduced in Section 5, which also presents a detailed analysis of the framework's performance, providing valuable insights. Finally, Section 6 concludes with a summary of key findings, contributions, and directions for future research in the dynamic intersection of PV systems and machine learning applications.

2. Photovoltaic Fault Types

A comprehensive understanding of PV systems requires a solid grasp of the various fault types affecting their performance. By examining these fault types in detail, one can better understand their potential impacts on PV module efficiency and reliability. This knowledge underscores the importance of effective fault detection and diagnosis, essential for maintaining optimal system operation. This section provides an overview of the different fault types, laying the groundwork for later discussions on advanced detection methodologies and innovative approaches within PV systems. PV system faults can occur for a variety of reasons and can have a substantial impact on the system's performance and dependability. Below is an overview of the faults typically seen in PV systems:

- Partial Shading: Partial shading occurs when specific portions of a PV module or array receive less sunlight due to objects blocking sunlight, such as trees and buildings. As a result, the power generation may become imbalanced, resulting in a lower overall system performance [9].

- Hotspots: Hotspots are isolated areas of high temperature in PV modules. Cell abnormalities, non-uniform soiling, or shadowing can cause them. If not treated, hotspots can cause cell degradation, lower module efficiency, and can lead to permanent damage [10].
- Cell Cracks: Mechanical stress, temperature variations, and external factors can all cause cracks in PV cells. Cell cracks lower the electrical output of the affected cells and can spread over time, affecting the module or array's overall performance [10].
- Degradation and Aging: PV modules are subject to gradual degradation and aging over time. Factors such as exposure to sunlight, temperature variations, and environmental conditions can decrease module efficiency and power output. Common degradation mechanisms include light-induced degradation (LID), potential-induced degradation (PID), and moisture ingress [11,12].
- Open and Short Circuits: Open circuits arise when an electrical route is broken, blocking current flow. Short circuits, on the other hand, occur when two points are unintentionally connected, resulting in excessive current flow. Open and short circuits can cause power outages in the affected areas of the PV system [13].

Among the different faults, the following three are considered catastrophic for PV systems according to [14]:

- Ground Faults: Ground faults occur when an unintentional electrical connection is made between the PV system and the ground. This can lead to safety risks, inefficiencies in the system, and potential damage to the PV equipment.
- Line-to-Line Faults: Line-to-line faults in PV systems are electrical faults that occur when two conductors in the system are directly shorted. These faults often include a direct connection between two of the three phases of the PV system's alternating current output, known as a three-phase fault.
- Arc Faults: PV arc faults refer to electrical arcs in PV systems. An electrical arc is a high-energy discharge of electricity that flows through the air or across a gap between conductive materials. In PV systems, arc faults can arise from various causes, including insulation breakdown, loose connections, damaged wiring, or component failures.

Table 1 lists different types of PV faults and gives their descriptions.

Table 1. Types of PV system faults.

Fault Type	Description
Variation in Irradiance	Fluctuations in sunlight intensity throughout the day
Soiling	Accumulation of bird droppings and dirt on the surface of PV modules
Environmental Effects (Snow Covering and Hotspots)	Extreme temperature variations based on geographical location and weather conditions
Earth Fault (Upper Ground Fault)	Unintended grounding with zero fault impedance between the last two modules in a PV string
Earth Fault (Lower Ground Fault)	Unintended grounding with zero fault impedance between the second and third modules in a PV string, often associated with high back-feed current
Arc Faults (Series and Parallel)	Arcing caused by disruptions in current-carrying conductors due to solder disjunction, cell damage, connector corrosion, rodent interference, or abrasion
Bypass Diode Faults	Short circuits resulting from incorrect diode connections
Bridging Faults	Low-resistance connections between points of different potential within a string of modules or cabling
Maximum Power Point Tracking (MPPT) Faults	Malfunctions in MPPT charge controllers
Cabling Faults	Issues related to cable connections

The dataset used in this work includes the following three faults: string faults, string-to-ground faults, and string-to-string faults. More details are given in Section 4.2.

3. Machine Learning Models in Photovoltaic Systems

Machine learning has become a crucial tool in solar PV systems, with applications spanning fault detection and diagnosis, performance optimization, and predictive maintenance. A summary of the most widely used machine learning algorithms in the literature for fault detection and diagnosis is provided in Figure 1. By analyzing historical data, weather patterns, and system parameters, machine learning algorithms can also predict optimal operating conditions, thereby maximizing energy output. Another essential application of machine learning in PV systems is predictive maintenance. By evaluating sensor data and historical performance records, machine learning algorithms can forecast potential equipment failures, enabling predictive maintenance actions. This predictive capability reduces system downtime and enhances the overall reliability and lifespan of PV components. Indeed, integrating machine learning into PV systems enhances the sustainability and efficiency of solar energy by enabling intelligent, data-driven solutions. A summary of ML applications in PV systems is presented in Table 2.

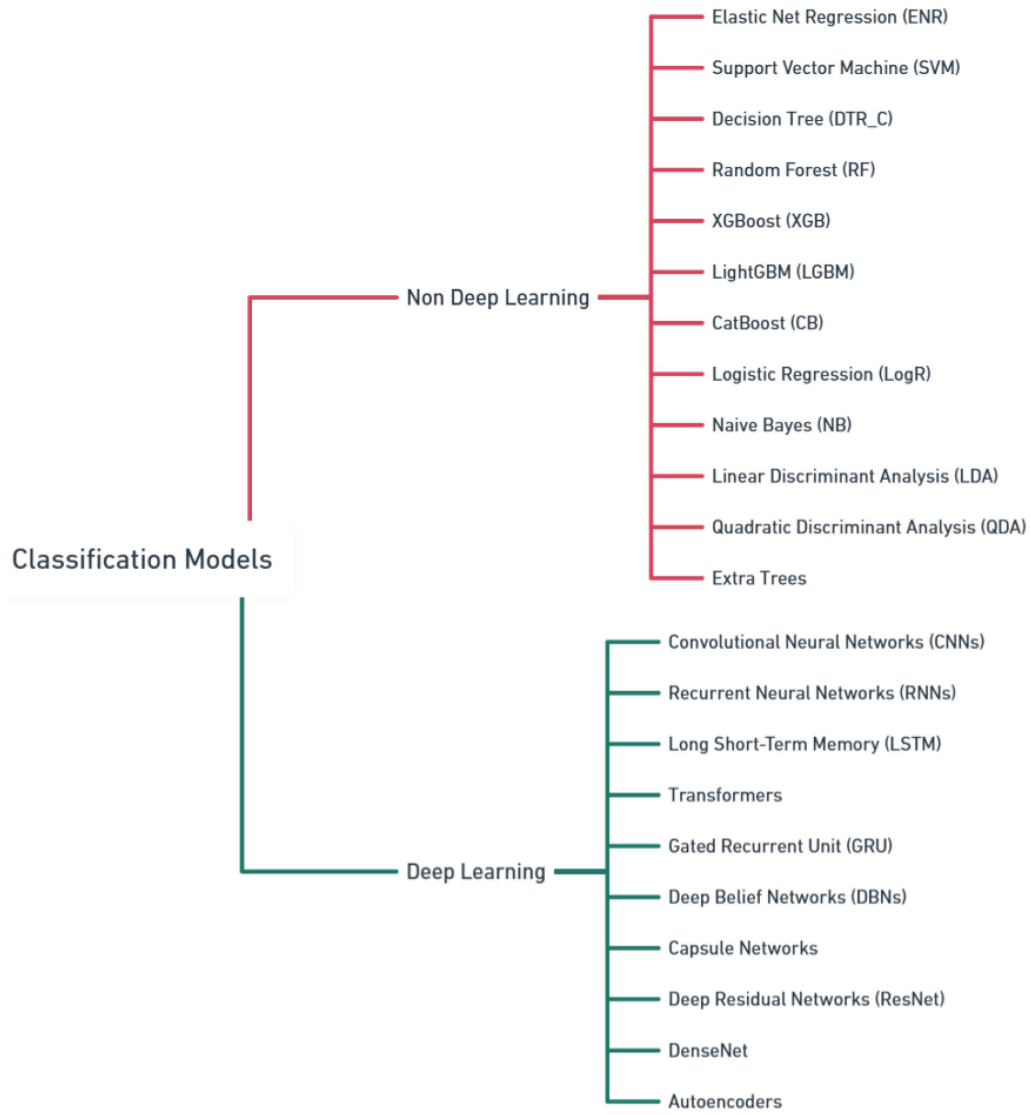


Figure 1. Machine learning models.

Table 2. Applications of machine learning in PV systems.

Domain of Use	Reference	Description
Performance Prediction	[15]	ML models examine historical data, weather conditions, and other pertinent aspects to predict the electricity production of PV systems. They aid in energy forecasting, system planning, and grid integration.
Fault Detection and Diagnosis	[16]	By evaluating real-time data from sensors, monitoring devices, and historical records, ML algorithms detect and diagnose defects in PV systems. They detect irregularities, allowing for timely repair and reducing downtime.
Energy Optimization	[17]	ML techniques optimize energy production by considering weather conditions, load demand, and grid requirements. They dynamically adjust the system parameters to maximize energy capture and improve the system's overall efficiency.
Soiling Detection and Cleaning Strategies	[18]	ML algorithms examine sensor data and weather patterns to identify the accumulation of dust, dirt, or other debris in PV modules. They recommend appropriate cleaning schedules and tactics to keep system performance high.
Lifetime Assessment and Degradation Modeling	[19]	ML forecasts the degradation of PV modules over time. ML models estimate the remaining usable life of modules by assessing previous performance data, environmental variables, and material qualities, assisting in asset management and maintenance planning.
Load Forecasting and Demand Response	[20]	ML algorithms forecast energy demand by analyzing historical load data and external factors (e.g., weather, time of day). This information helps grid operators and utilities manage energy supply and demand, allowing the optimal integration of PV systems into the grid.
Data Analytics and Decision Support	[21]	In PV systems, ML approaches enable data-driven decision-making. ML models may detect trends, correlations, and improvement possibilities by processing and analyzing massive amounts of data, making system monitoring, performance evaluation, and maintenance scheduling easier.

3.1. Tree-Based Models

The tree-based algorithms in this work, namely, decision trees, random forest, Stochastic Gradient Boosting, LightGBM, CatBoost, and Extra Trees, have shown remarkable effectiveness in detecting and diagnosing faults in various applications. These algorithms are based on decision trees (illustrated in Algorithm 1) as their core component and offer high accuracy in identifying and classifying faults, as demonstrated in this study.

Algorithm 1 Decision tree algorithm

```
procedure DecisionTress(Instances, Target_feature, Features)
  if all instances at the current node belong to the same category then
    Create a leaf node of the corresponding class
  else
    Find the feature A that maximizes the goodness measure
    Make A the decision feature for the current node
    for each possible value v of A do
      Add a new branch below node testing for A = v
      Instances_v ← subset of Instances with A = v
      if Instances_v is empty then
        Add a leaf with the most common value of Target_feature in Instances
      else
        Below the new branch, add a subtree
        DecisionTress( Instances_v, Target_feature, Features − {A})
      end if
    end for
  end if
end procedure
```

A detailed overview of these models is presented below.

- Decision Trees: Decision trees provide a tree-like model of decisions and their potential outcomes. They divide the data into several classes based on different attributes and features. Decision trees can handle both category and numerical data and are interpretable.
- Random Forest: Random forest is an ensemble learning technique that makes predictions by combining numerous decision trees. It generates a set of decision trees and aggregates their outputs to obtain a final forecast. Random forests are well known for their robustness and capacity to handle large amounts of data.
- Extra Trees: Like random forest, the Extra Trees classifier builds multiple decision trees but introduces additional randomness during node splitting, leading to a more diverse ensemble.
- XGBoost (Extreme Gradient Boosting): XGBoost is a gradient boosting framework optimized to solve machine learning issues quickly and accurately. It employs a scalable and adaptable tree-boosting approach. XGBoost is a popular choice for data scientists and practitioners due to its sophisticated features, such as regularization, parallelization, and handling of missing inputs. It can handle classification and regression tasks and is widely used in competitions and real-world applications.
- LightGBM (Light Gradient Boosting Machine): LightGBM is an efficient and scalable gradient boosting system. It employs a unique technique known as Gradient-based One-Side Sampling (GOSS) to accelerate training while consuming less memory and retaining good prediction accuracy. LightGBM is well known for its quick training speed and ability to handle big datasets. It can handle various tasks, such as binary classification, multi-class classification, and regression. LightGBM also includes sophisticated features, including categorical feature support, parallel learning, and customized loss algorithms.
- CatBoost: It is a gradient boosting system that focuses on adequately managing categorical features. It employs cutting-edge approaches such as Ordered Boosting, which optimizes the learning process by considering the natural ordering of categories. CatBoost automatically handles category features by translating them into numerical representations, removing the need for manual feature engineering. It contains built-in capabilities, such as effective handling of missing values and the ability to train on GPU, and gives excellent accuracy on a wide range of tasks. CatBoost is user-friendly and integrates well with common programming languages.

3.2. Evaluation Metrics

Classification is a fundamental activity in machine learning and data analysis, which categorizes or classifies incoming data examples such as this case. It involves creating models to learn from labeled training data and predict new unlabeled data. Classification metrics are used to assess the effectiveness and performance of classification models. These metrics provide quantifiable measures of the model's predictions, such as accuracy, precision, recall, etc. Insights into the model's strengths and limitations can be obtained by studying these indicators and making informed decisions about its performance. The classification problem's specific goals and constraints determine the metrics used. Different metrics highlight certain aspects of the model's performance and may be more meaningful in particular scenarios. Table 3 shows the metrics used in this paper.

Table 3. Metrics and formulas.

Metric	Description	Formula
False Positive	False positive rate measures the proportion of actual negatives incorrectly identified as positive	$\frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$
False Negative	False negative rate measures the proportion of actual positives incorrectly identified as negative	$\frac{\text{False Negatives}}{\text{False Negatives} + \text{True Positives}}$
Accuracy	Measures the overall correctness of the model's predictions	$\frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}}$
Precision	Measures the proportion of true positives among all positive predictions	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$
Recall	Measures the proportion of true positives identified correctly among all actual positives	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$
F1 Score	Combines precision and recall into a single metric, thus balancing both measures	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

3.3. Related Work on Fault Detection and Diagnosis in PV Systems

Researchers have investigated machine learning and deep learning approaches for fault detection in PV systems over the years. These methods use the power of data-driven algorithms to evaluate the complex patterns and correlations found in the data from the PV system, allowing for the diagnosis of faults and irregularities that may affect the performance and dependability of the system. In [22], an SVM-based model was developed to detect three types of faults: short circuit, open circuit, and lack of irradiation. An optimization procedure was performed to achieve a test accuracy of 97% and a generalization ability superior to prior methods. In [23], and using a MATLAB Simulink dataset, the random forest, Ada-Boost, CN2, logistic regression, and naive Bayes algorithms were built for fault identification in PV farms. In total, 30 features were employed, including plant current, distributed current measurements, temperature, and radiation. The Ada-Boost classifier performed the best with a precision of 0.95. In [24], a newly developed fault detection framework—based on deep residual models and a deep learning algorithm that adapts moment estimates—was used to extract features from essential parameters such as temperature and current voltage and improve performance with a deeper network. The model was tested using Simulink MATLAB simulation data from a PV farm, and it gave improved results in terms of precision, dependability, generalization, and training efficiency. In [25], autonomous feature extraction was developed for LSTM and BiLSTM DNN and used for fault classification in PV systems. The suggested framework demonstrated excellent performance in applying to a grid-connected PV system, achieving 100% in fault classification for the BiLSTM. In [26], deep belief networks and genetic algorithms were coupled to address the fault diagnosis problem, and the new method was compared to the classical DBN, SVM, and back-propagating network with GA, demonstrating a high degree of generalization and recognition of PV array faults. In [27], an ML structure for fault detection and diagnosis was proposed to address multiple fault types such as arc faults, line-to-line faults, maximum power point tracking unit failure, and open-circuit faults. Cubic SVM and GRB kernels performed the best in terms of accuracy. In [28], a probabilistic neural network was used to detect faults in the DC portion of the PV array. Four operational scenarios were examined in a 9.54 kWp grid-connected PV system: a healthy system, three modules short-circuited in one string, ten modules short-circuited in one string, and a string unplugged from the array in a healthy system. The proposed method was highly effective in detecting DC-side anomalies.

4. Proposed Framework

The proposed framework for this application starts with the refinement of the dataset, which involves eliminating redundant samples and transforming categorical features, encompassing fault types and normal cases. The framework proceeds with various exploratory data analysis and preprocessing stages to attain the optimal dataset. This refined dataset is then fed into the first-phase tree-based classifier, segregating faulty samples from normal ones. The identified faulty samples proceed to the second phase for fault diagnosis, classifying them into one of the three existing fault types (F1, F2, or F3). The models are then examined for performance and convergence time. A post-processing step follows, focusing on feature number reduction based on optimal results and explainability. Figure 2 illustrates the proposed framework.

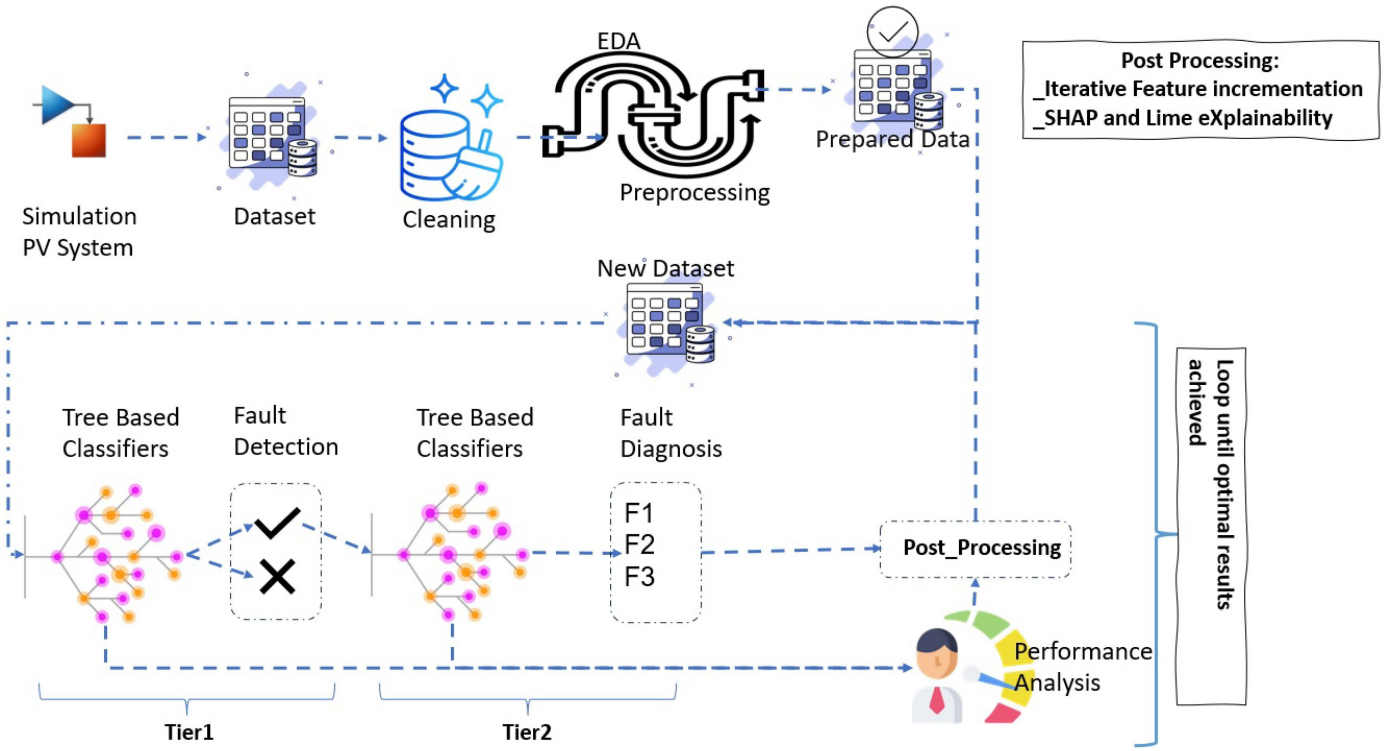


Figure 2. The proposed two-phase framework.

4.1. Simulink Model

Figure 3 represents a 250 kW grid-connected PV power system developed in MATLAB/Simulink (Version R2018a). The system integrates a PV array comprising 88 parallel strings, each containing seven series-connected modules (SunPower SPR-415E-WHT-D). Each module consists of 128 cells with a maximum power of 414.801 W, an open-circuit voltage of 85.3 V, a short-circuit current of 6.09 A, and an operating voltage of 72.9 V at 5.69 A. The PV system is connected to the grid through a three-level IGBT inverter operating under PWM control and implementing maximum power point tracking using the perturb and observe method. The power from the PV array is stepped up by a three-phase 0.25/250 kV transformer to interconnect with the grid. The model features two transmission lines: a 14 km feeder leading to a 120 kV power equivalent grid and an 8 km feeder supplying a static load. The model includes detailed measurements, filtering, and control to simulate dynamic grid-tied operations under varying environmental conditions, ensuring efficient energy transfer and grid stability.

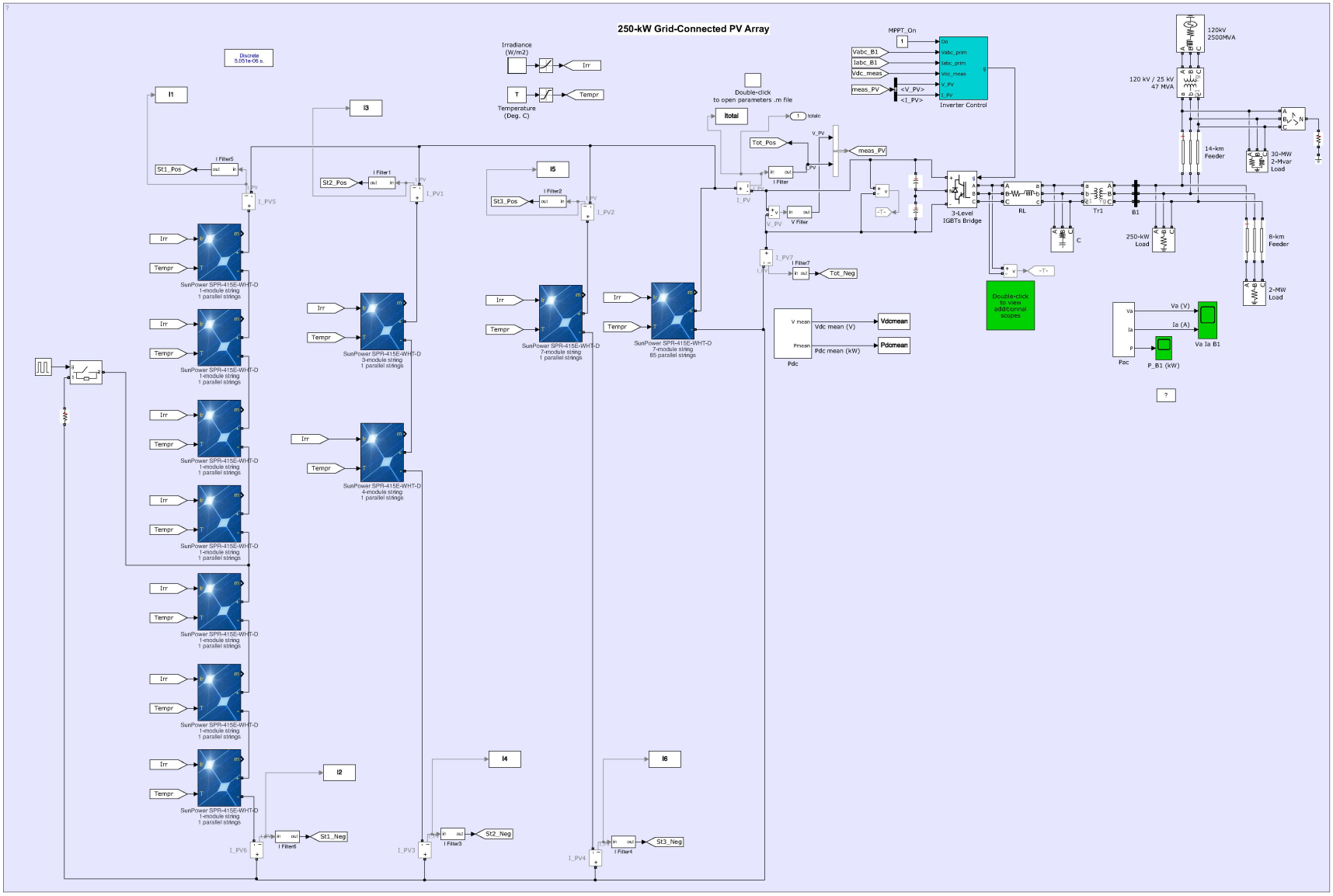


Figure 3. The Simulink model of the 250 kW PV power plant.

4.2. Dataset Description

The simulated faults represent string faults in string 1 (F1), string-to-ground faults in string 1 (F2), and string-to-string faults between strings 1 and 2 (F3), as illustrated in Figure 4. Table 4 provides the normal and faulty cases in both training and testing datasets, whereas Table 5 provides the list of features and their descriptions in the dataset. As can be seen from Table 5, these features represent the electric current average, maximum, minimum, and variance from the affected strings, namely, strings 1, 2, and 3, where each string is equipped with two ammeters to measure the current at its top and bottom during the simulation, as can be seen from Figure 3. In addition, the total average DC power, total current, total average DC voltage, solar irradiance, and temperature are also included as features. Currents, voltages, and power are used for fault detection and diagnosis in PV systems since deviations from standard I-V and P-V curves indicate anomalies such as shading, open circuits, or short circuits. As for environmental features, solar irradiance normalizes electrical performance and allows fault detection by comparing expected against observed output, while ambient temperature impacts performance and helps differentiate between environmental effects and actual faults.

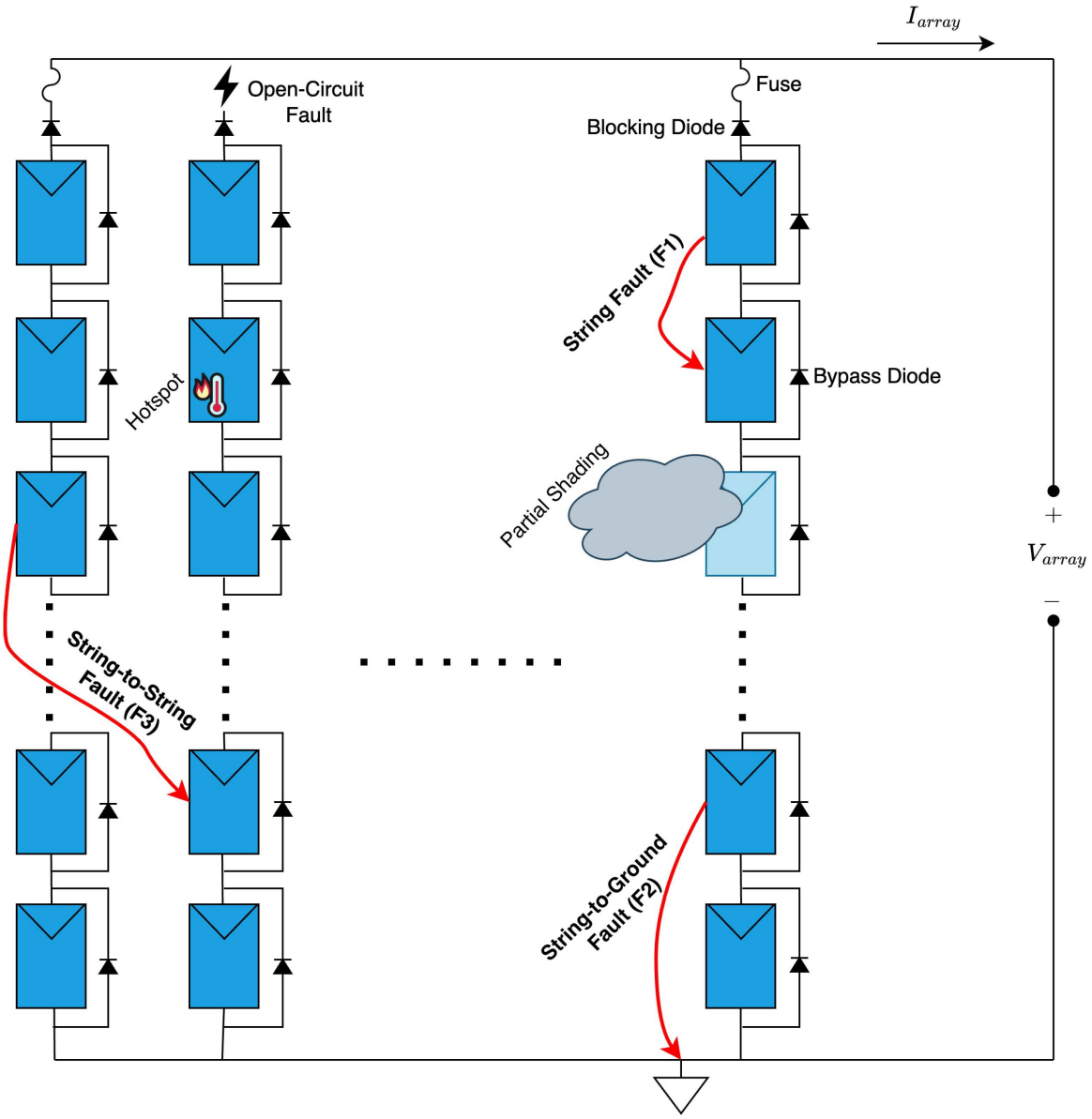


Figure 4. String (F1), string-to-ground (F2), and string-to-string (F3) faults.

Table 4. Dataset of normal and faulty cases.

Dataset	Case	Number of Samples
Training Data	Normal	100
	F1	153
	F2	149
	F3	198
Testing Data	Normal	25
	F1	25
	F2	25
	F3	25
Total	-	700

Table 5. The dataset features and their description.

Feature	Description
I1	Average of the current at the top of string 1
I2	Average of the current at the bottom of string 1
$I1_{max}$	Maximum of the current at the top of string 1
$I1_{min}$	Minimum of the current at the top of string 1
$I1_{VAR}$	Variance of the current at the top of string 1
$I2_{max}$	Maximum of the current at the bottom of string 1
$I2_{min}$	Minimum of the current at the bottom of string 1
$I2_{VAR}$	Variance of the current at the bottom of string 1
I3	Average of the current at the top of string 2
I4	Average of the current at the bottom of string 2
$I3_{max}$	Maximum of the current at the top of string 2
$I3_{min}$	Minimum of the current at the top of string 2
$I3_{VAR}$	Variance of the current at the top of string 2
$I4_{max}$	Maximum of the current at the bottom of string 2
$I4_{min}$	Minimum of the current at the bottom of string 2
I5	Average of the current at the top of string 3
I6	Average of the current at the bottom of string 3
I_{total1}	Average of the total current of the PV power plant
$I_{totalmax1}$	Maximum of the total current
$I_{totalmin1}$	Minimum of the total current
$V_{dcmean1}$	Average of the total DC voltage
V_{dcmax1}	Maximum of the total DC voltage
V_{dcmin1}	Minimum of the total DC voltage
$P_{dcmean1}$	Average of the total DC Power
IR	Solar irradiance ranging from 100 to 1000 W/m^2
T	Temperature ranging from 10 to 35 °C
Range1	$I1_{max}$−$I1_{min}$
Range2	$I2_{max}$−$I2_{min}$
Range3	I1−I2
Range4	I3−I4
Class	Normal: 0, F1: 1, F2: 2, F3: 3

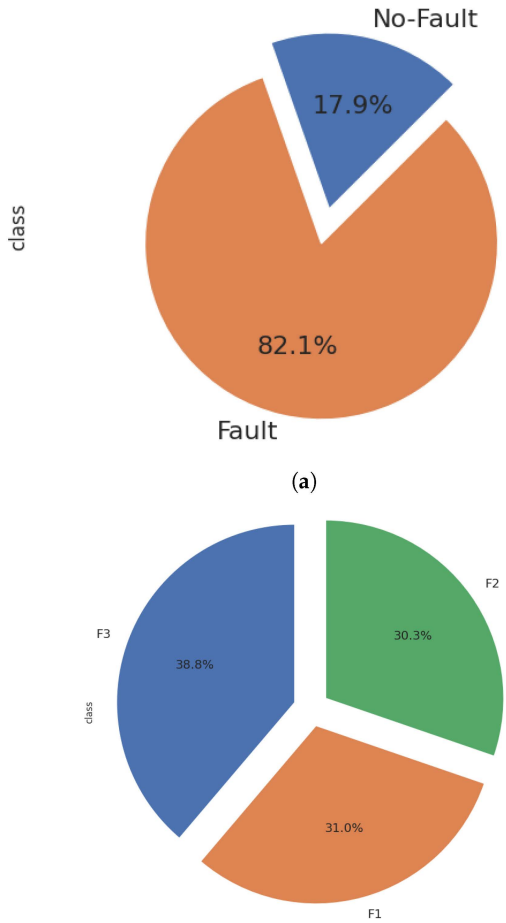
4.3. Data Preprocessing

Data preprocessing refers to the actions and techniques performed on raw data before they are fed into a machine learning model. It entails converting, cleansing, and organizing data so that it may be analyzed and modeled. Preprocessing is critical to the success of a machine learning project. In the following lines, the preprocessing steps are formulated in the form of inquiries to be answered:

- How well are the dataset classes balanced? The initial analysis focused on assessing the balance within the dataset, primarily addressing the two key phases of the system. The first phase involves the fault detection system with two classes (No-Fault and Fault), while the second phase, dedicated to fault diagnosis, encompasses three classes (F1, F2, and F3). Maintaining balance among these classes is crucial for specific machine learning models to overcome biases towards the more prevalent class. In this study, the primary classifiers employed were tree-based. Initially, the dataset was not balanced, but a decision was made to consider balancing as a post-processing option if the results proved insufficiently accurate. This approach explored the possibility

of improving the outcomes through balancing measures. It can be seen in Figure 5 that the classes within the first detection system exhibit an imbalance, with a higher prevalence of faulty samples. Conversely, the second phase is characterized by a balanced distribution among the three fault types, with a slight prevalence of 7% for F3. This balanced distribution in the second phase contributes to a more uniform consideration of the different types of faults during diagnosis.

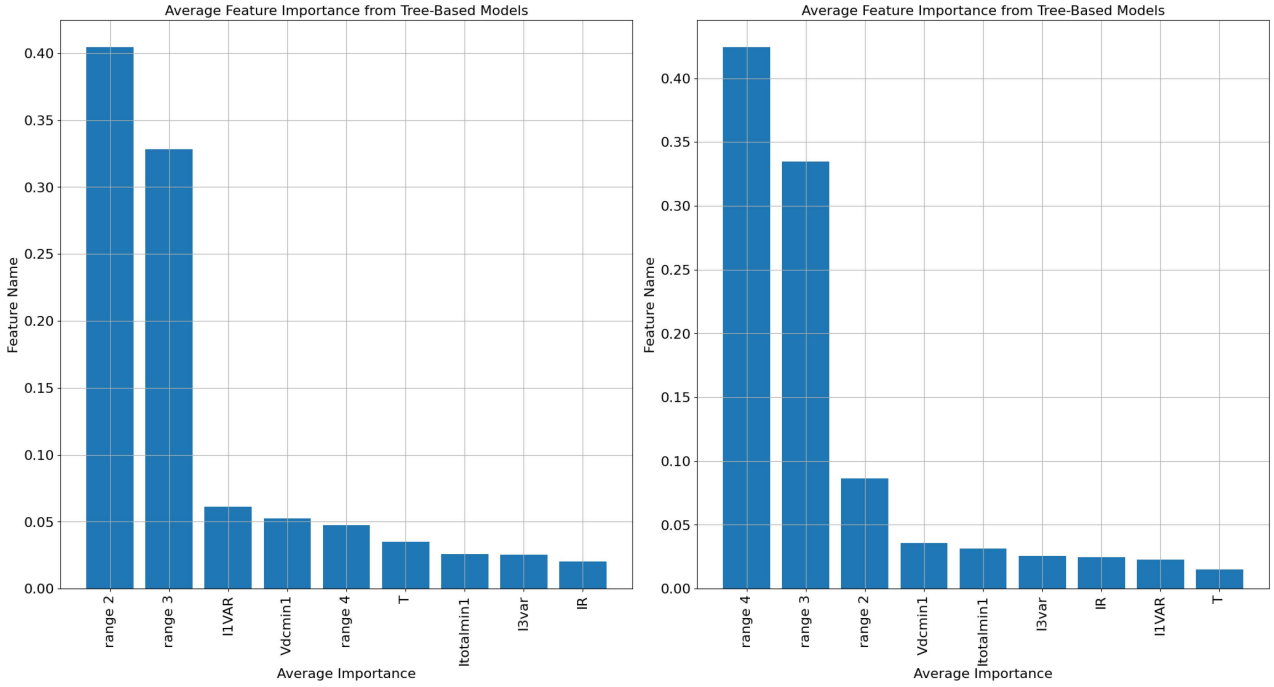
- What are the key features influencing the models? An averaging technique based on the average importance of the tree-based classifier was used to identify the most impactful features in the model's decisions. Figure 6 presents the findings of this study using the average feature importance technique for both the first-phase fault detection system and the second-phase fault diagnosis system, as well as a combined analysis of both systems in a naive solution. Regardless of the system under consideration, the features range 4, range 3, range 2, V_{dcmin1}, $I3_{VAR}$, IR, $I1_{VAR}$, and T consistently emerge as the most influential. Notably, one of the range *i* features is slightly superior in each phase. These consistent observations across all phases suggest a potential hidden pattern within these series, potentially linked to faults. Consequently, based on the results obtained through this technique, the dataset was refined to include the top eight most influential features. These selected features are the starting point for feeding the dataset into the classifiers, aiming to optimize the model's performance by focusing on the most significant contributors to decision-making.
- Are there any outliers? Analyzing and handling outliers is a pivotal step in the complete preprocessing pipeline. Outliers, defined as values outside the range of a given series, can considerably influence a model. These values can potentially degrade the model's performance if not handled correctly. Interestingly, in certain cases, outliers can provide valuable insights, aiding in identifying hidden anomalies associated with existing fault types. The outlier analysis, illustrated in Figure 7, indicates the potential presence of points aligning with this perspective. Nevertheless, on the whole, the series appears to be predominantly clean, suggesting that outliers, while possible, may not be pervasive throughout the dataset.
- What are the possible existing correlations? Both the full and the fault datasets underwent a correlation analysis of the selected eight features. Mutual correlations exceeding 90% were systematically eliminated, indicating analogous behavior between the series. The results of this analysis are presented in Figure 8, where the series class is the target. Notably, no substantial mutual correlations were detected, with the upper threshold set at 72% in the full dataset and 71% in the fault dataset. Furthermore, the feature exhibiting the highest correlation with the target in both datasets is range 4, with percentages of 36% and 38% for the full dataset and the fault dataset, respectively.



(**a**)

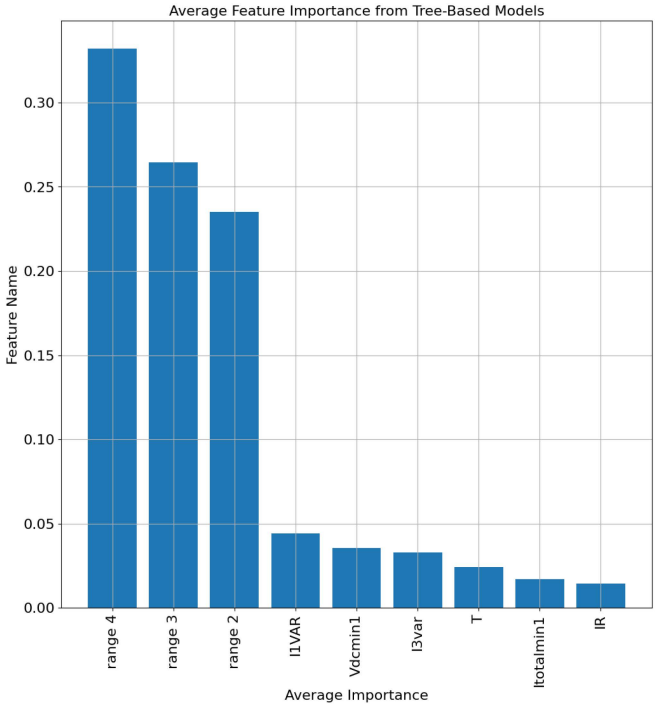
(**b**)

Figure 5. Pie charts of the (**a**) faulty and normal cases and (**b**) fault types.



(**a**) (**b**)

Figure 6. *Cont.*



(**c**)

Figure 6. Averaging feature importance results: (**a**) fault detection phase, (**b**) fault diagnosis phase, and (**c**) naive solution.

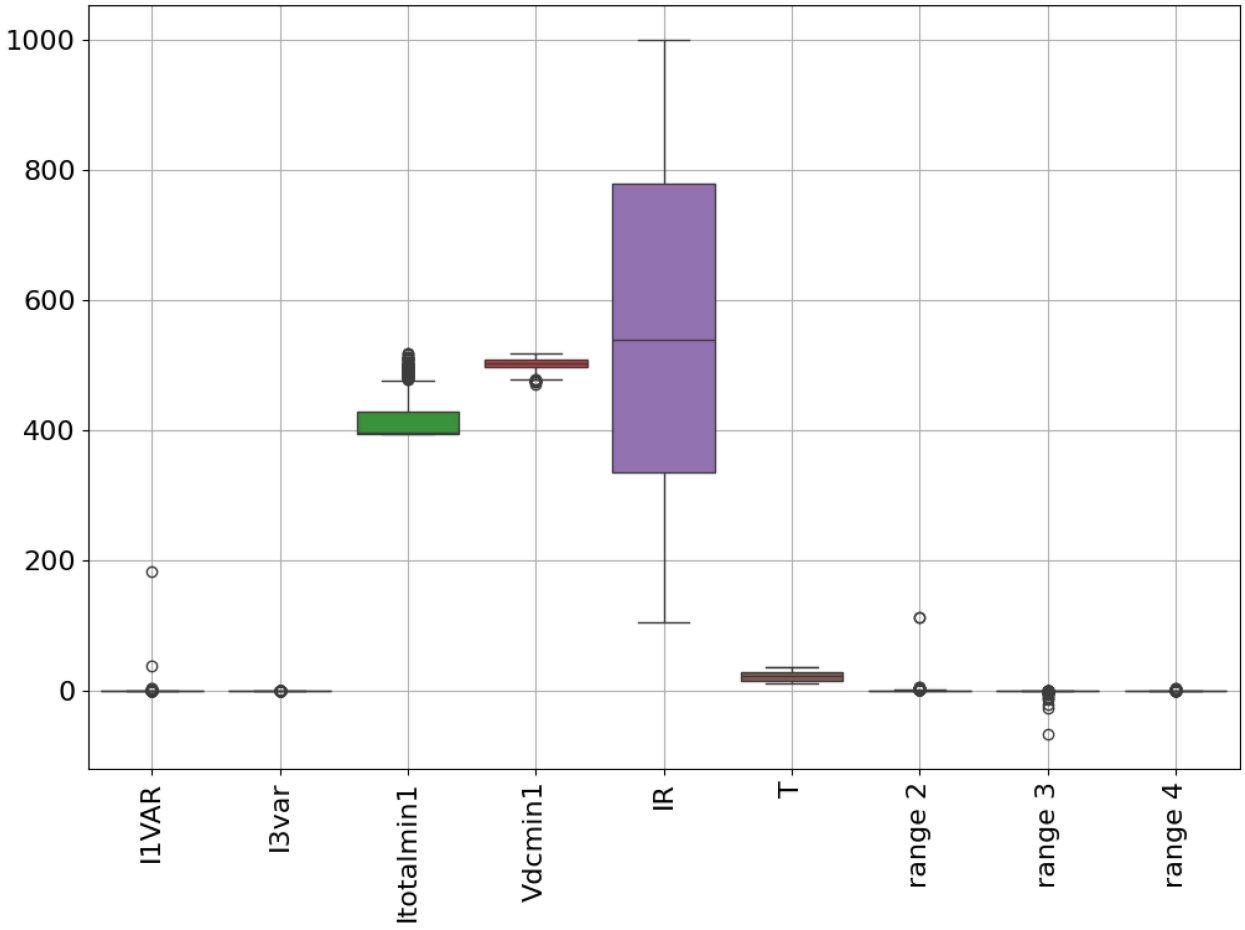
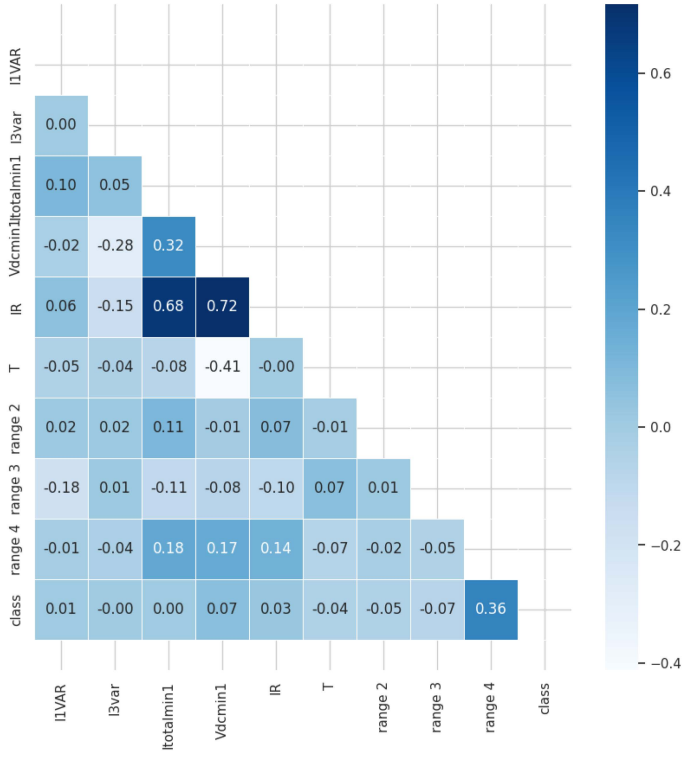
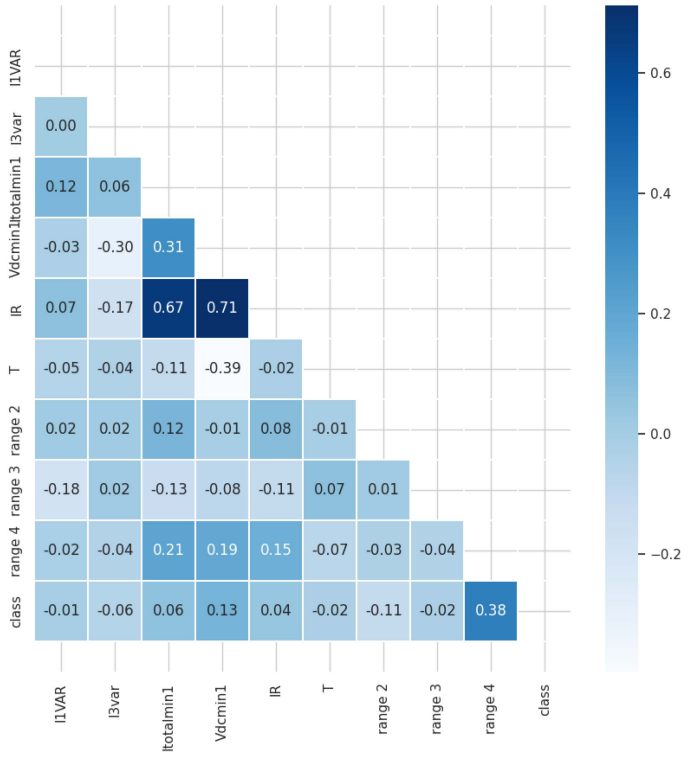


Figure 7. Outlier detection using boxplot.



(**a**)



(**b**)

Figure 8. Correlation analysis: (**a**) full dataset and (**b**) fault dataset.

5. Results and Discussion

After completing the preprocessing steps, the dataset consisted of 700 rows and 8 columns. It was then split into training and testing sets, with 80% allocated for training and 20% for testing. The six machine learning tree-based classifiers (decision trees, random

forest, Extra Trees, XGBoost, CatBoost, and LightGBM) were trained using Google Colab, a free platform for Python 3.10code development, machine learning, and data analysis provided by Google. The results of the experiments are categorized as follows:

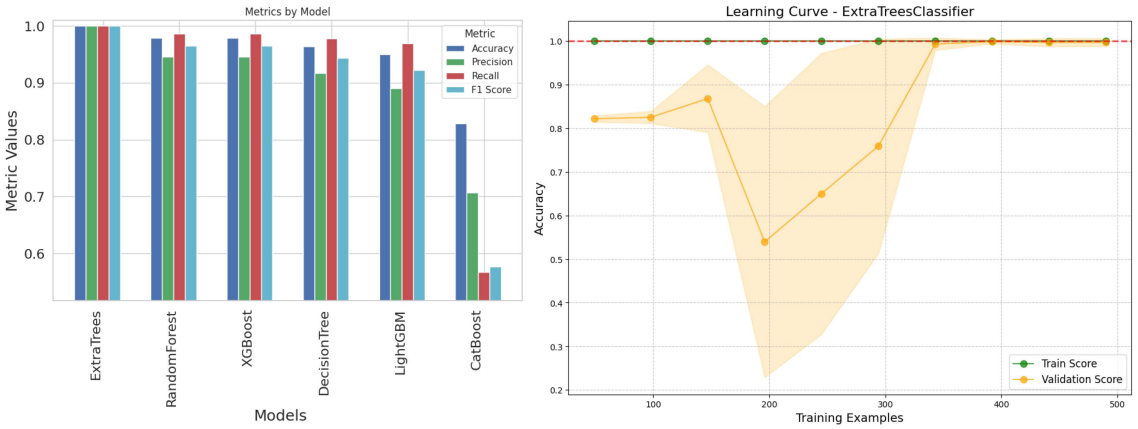
- Fault detection results;
- Fault diagnosis results;
- Comparison with previous works;
- eXplainability through XAI.

5.1. Binary Classification Results

The clean data were fed into the first classification stage (binary classification), which included six classifiers. This first stage enabled determining a superior classifier, which can be further improved using hyperparameter tuning or optimizing the number of features. The results of the binary classification, as showcased in Table 6, demonstrate the remarkable effectiveness of the Extra Trees classifier in distinguishing between faulty and normal samples, surpassing both the random forest and XGBoost classifiers. The outstanding accuracy of 100% attests to its exceptional performance. Additionally, the model's effectiveness is verified by examining the normalized confusion matrix. For the remaining classifiers, their overall accuracy exceeded 95%, except for the CatBoost classifier. A 5-fold cross-validation method was employed to provide an unbiased evaluation of the proposed models. The dataset was divided into five equal parts, with four parts (80% of the data) used for training and the remaining part (20% of the data) used for testing in each iteration. To ensure fairness, each fold preserved the class distribution of the dataset. The final cross-validation accuracy, designated by Accuracy-CV in Table 6, was determined by averaging the results across all five folds. Except for CatBoost, the difference between the cross-validation accuracy and test accuracy is small for all models, indicating that the models generalize well. A comparative analysis of the classifiers' performance is visually presented as a barplot in Figure 9a, offering a comprehensive performance overview. The performance of the ExtraTree classifier was monitored through its learning curve, as depicted in Figure 9b. The initial stages of the model's training exhibit some fluctuations, suggesting a learning curve characterized by a degree of struggle. However, the model ultimately converges to its optimal state after analyzing approximately 350 samples. Notably, this indicates rapid convergence, even though it occurs later. It is crucial to consider the relatively small size of the datasets, especially after partitioning into training and testing sets. Furthermore, the training time for the ExtraTree classifier was found to be 0.5218 s, highlighting its remarkably fast training capability.

Table 6. Evaluation metrics for the binary classification models trained by using only 6 features (range 2, range 3, I1VAR, Vdcmin1, range 4, T).

Model	Accuracy-CV	Accuracy	Precision	Recall	F1 Score	Normalized Confusion Matrix
ExtraTrees	0.998214	1.000000	1.000000	1.000000	1.000000	$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$
RandomForest	1.000000	0.978571	0.946429	0.986957	0.965090	$\begin{bmatrix} 1.0 & 0.0 \\ 0.02608696 & 0.97391304 \end{bmatrix}$
XGBoost	1.000000	0.964286	0.916667	0.978261	0.943434	$\begin{bmatrix} 1.0 & 0.0 \\ 0.04347826 & 0.95652174 \end{bmatrix}$
DecisionTree	1.000000	0.964286	0.916667	0.978261	0.943434	$\begin{bmatrix} 1.0 & 0.0 \\ 0.04347826 & 0.95652174 \end{bmatrix}$
LightGBM	0.998214	0.950000	0.890625	0.969565	0.922901	$\begin{bmatrix} 1.0 & 0.0 \\ 0.06086957 & 0.93913043 \end{bmatrix}$
CatBoost	1.000000	0.828571	0.706767	0.566957	0.576613	$\begin{bmatrix} 0.16 & 0.84 \\ 0.02608696 & 0.97391304 \end{bmatrix}$



(**a**) (**b**)

Figure 9. First-phase results: (**a**) barplots (barplot for the first-phase results) and (**b**) learning curve of the Extra Trees classifier.

Fault Detection Optimization

Optimization typically involves fine-tuning hyperparameters to achieve optimal and improved metric values for a model. However, in the context of the fault detection problem (first phase), the focus of the optimization process was directed at determining the optimal number of features that yield the highest observed accuracy. The chosen model for this phase is the ExtraTree classifier, which is identified as the most effective in fault detection, as highlighted in the previous section. The objective was to streamline the model by simultaneously reducing complexity and training time, providing a more reliable and lightweight solution. The approach involved initiating the process with the most important features identified through the averaging importance technique. Subsequently, the ExtraTree classifier was trained in a loop with increasing features until the optimal values were reached. Figure 10 illustrates the outcomes of this optimization process.

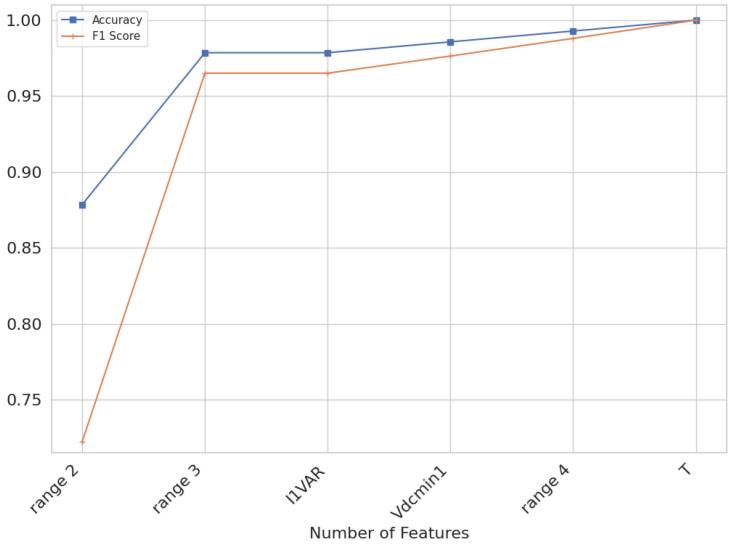
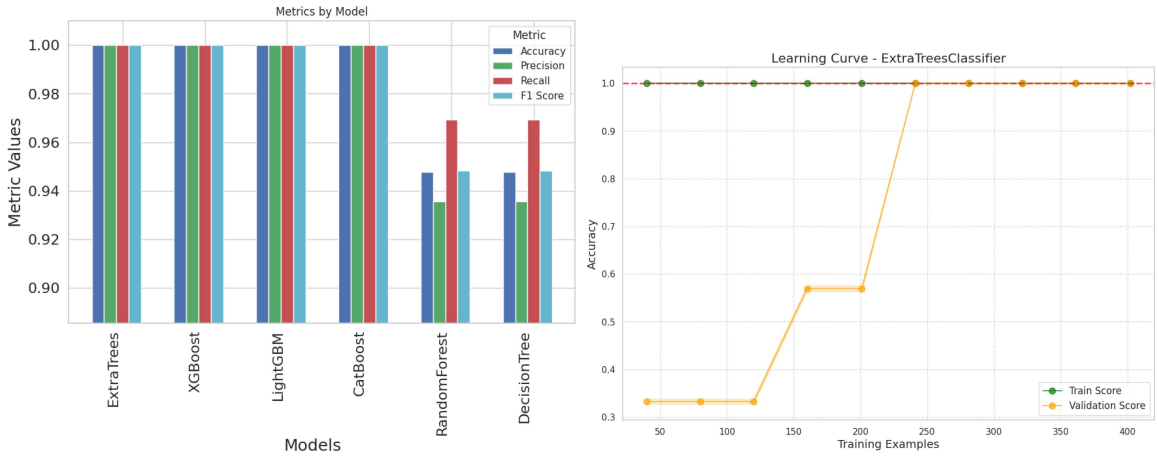


Figure 10. Binary fault detection: performance versus cumulative important features.

The observation from Figure 11 reveals that the Extra Trees classifier initiates its convergence toward the optimal result at an early stage. Notably, the accuracy experiences a significant increase immediately after adding the second feature, range 3. However, the model does not attain its full performance until the inclusion of the sixth feature, which is the temperature T. In summary, the model achieves its highest performance in the fault

detection phase with six essential features: range 2, range 3, I1VAR, Vdcmin1, range 4, and T. These features play a crucial role in enhancing the model's accuracy and effectiveness in detecting faults.



(**a**) (**b**)

Figure 11. Fault diagnosis phase: result analysis. (**a**) Barplot for the six classifiers. (**b**) Learning curve for the Extra Trees classifier.

5.2. Multi-Class Classification Results

The second-phase fault diagnosis system classifies fault types into three main categories: F1, F2, and F3. The system's performance, employing the six tree-based classifiers, is outlined in Table 7 and broken to barplots in Figure 11a. Several classifiers demonstrate exceptional results, achieving 100% accuracy, precision, recall, and F1 scores for each fault type. Specifically, classifiers such as Extra Trees, XGBoost, LightGBM, and CatBoost perform equally. Moreover, the 5-fold cross-validation reveals little to no difference between validation and testing accuracies, suggesting that the models generalize well to unseen data. However, there are variations in convergence speed among these classifiers. A comparative analysis of the training and testing times for Extra Trees and XGBoost reveals that XGBoost takes 0.9509 s for the entire process, while Extra Trees completes it in 0.4204 s. Figure 11b depicts the learning curve of the Extra Trees classifier, showing that it attains its peak value after analyzing approximately 240 samples. This represents less than 50% of the complete dataset, indicating superior generalization and rapid convergence toward optimal results.

Table 7. Performance of the multi-class classification trained by using only 2 features: range 3 and range 4.

Model	Accuracy-CV	Accuracy	Precision	Recall	F1 Score	Normalized Confusion Matrix
ExtraTrees	1.000000	1.000000	1.000000	1.000000	1.000000	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
XGBoost	1.000000	1.000000	1.000000	1.000000	1.000000	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
LightGBM	1.000000	1.000000	1.000000	1.000000	1.000000	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
CatBoost	1.000000	1.000000	1.000000	1.000000	1.000000	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
RandomForest	1.000000	0.947826	0.935484	0.969231	0.948157	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.09230769 & 0.90769231 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
DecisionTree	1.000000	0.947826	0.935484	0.969231	0.948157	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.09230769 & 0.90769231 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$

Fault Diagnosis Optimization

A similar strategy was employed for the fault diagnosis system in a manner akin to the approach taken for the fault detection system. Given achieving optimal results with 100% accuracy among four classifiers, the focus shifted to determining the optimal number of features required for the model to converge efficiently and accurately. To conduct this analysis, the process was initiated with the most influential feature identified through the feature importance averaging technique, as illustrated in Figure 6b. Subsequently, the number of features was iteratively increased until reaching 100% accuracy. The results of this investigation are presented in Figure 12a. Remarkably, the results reveal that the second-phase system can attain optimal performance with just two features, specifically the range 4 and range 3 features. Beyond this point, the model stabilizes at 100% accuracy, suggesting that a reduction in complexity and dimensionality is achievable. Furthermore, upon visualizing the scatter plots of these two features in the presence of the three fault types, as depicted in Figure 12b, it becomes apparent that these features exhibit higher separation between the faults. This implies that they can serve as effective indicators for each fault type and may be utilized independently for fault identification.

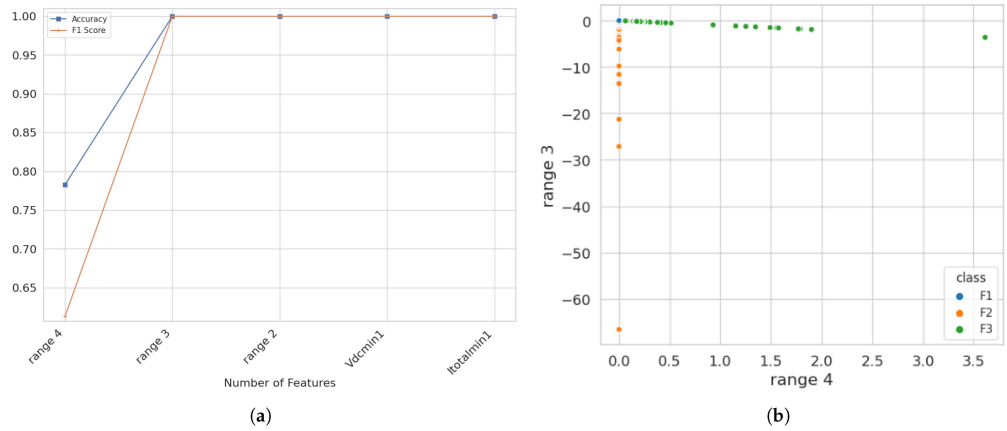


Figure 12. (**a**) Performance variation versus cumulative important feature number and (**b**) scatter plot between the range 4 and range 3 features colored by fault type.

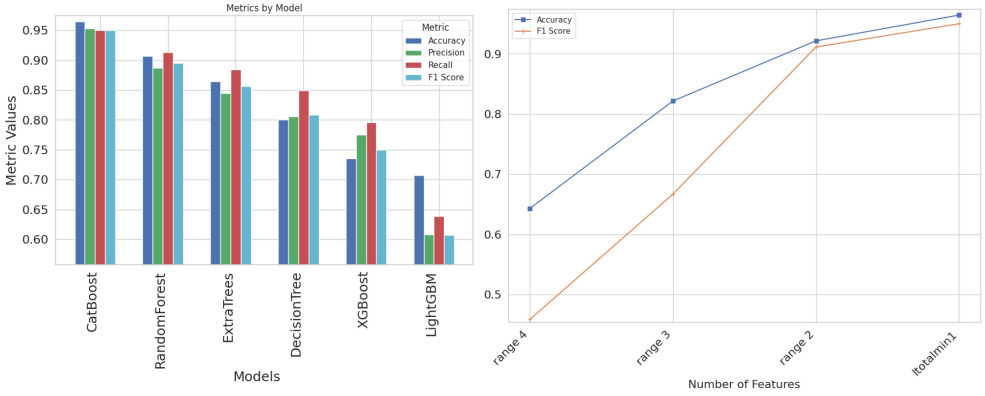
5.3. Comparison with Previous Works

To validate the superiority of the proposed methodology, a comparative analysis was conducted with previous works. This comparison focused on three aspects. First, a contrast was drawn with a naive solution encompassing fault detection and diagnosis in a single layer. In this approach, normal and faulty samples were collectively considered in a multi-class classification (four classes). The same classifiers were employed to ensure fairness, and the results are presented in Table 8. This table shows that none of the classifiers achieved 100% accuracy in any considered metric. The best accuracy of 96.4% was obtained by CatBoost, which falls short of the proposed framework. Further insight from the confusion matrix of CatBoost reveals that only two fault types, F2 and F3, were classified with 100% accuracy, while significant classification errors occurred for the F1 fault and non-faulty samples. The results, illustrated in barplots in Figure 13a, expose the diminished performance of classifiers when employing the naive solution. These findings underscore that the two-phase proposed solution is more adept at addressing the problem. Additionally, the performance was tracked by increasing the number of features, and Figure 13b indicates that it takes four features to achieve the highest accuracy of 96.4%. The proposed solution surpasses the naive one in terms of accuracy and the number of features required for convergence. Notably, the complexity of the naive solution is higher, as it involves classification among four classes. In contrast, despite requiring

two phases, the proposed solution reduces the complexity of each phase to two classes in the first and three classes in the second. The models that were used are the subject of the second comparison. All the trained classifiers are tree-based, meaning they do not require complicated building patterns. Moreover, the only methodology that achieved 100% high-performance results in the literature was described in [25], which employed an autonomous feature extraction for LSTM and BiLSTM. Deep learning was mainly used to achieve these outcomes. However, because tree-based algorithms are solely used in this work, the proposed methodology does not require as much computing complexity, which shows that it is superior in model selection. The two new approaches of averaging importance and increasing feature iteratively, leading to optimal dimensionality, were used to make the final comparison. The current work, which has only six features in the initial phases of fault detection and two in fault diagnosis, outperforms all previous works in terms of accuracy and speed of convergence. Training and testing of the algorithms can be completed in less time, as evidenced by the proposed framework's total time of 0.48 s. The proposed solution stands out among all the earlier studies on the topic due to these comparisons and the innovative nature of the suggested methodology.

Table 8. Performance of the one-phase naive classification model (all classes).

Model	Accuracy-CV	Accuracy	Precision	Recall	F1 Score	Normalized Confusion Matrix
CatBoost	1.000	0.964	0.953	0.950	0.950	[0.960 0.040 0.000 0.000 0.160 0.840 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 1.000]
RandomForest	1.000	0.907	0.887	0.913	0.895	[1.000 0.000 0.000 0.000 0.240 0.760 0.000 0.000 0.031 0.077 0.892 0.000 0.000 0.000 0.000 1.000]
ExtraTrees	1.000	0.864	0.845	0.884	0.857	[1.000 0.000 0.000 0.000 0.280 0.720 0.000 0.000 0.015 0.169 0.815 0.000 0.000 0.000 0.000 1.000]
DecisionTree	1.000	0.800	0.805	0.849	0.809	[1.000 0.000 0.000 0.000 0.280 0.720 0.000 0.000 0.031 0.292 0.677 0.000 0.000 0.000 0.000 1.000]
XGBoost	0.993	0.736	0.775	0.796	0.750	[0.960 0.040 0.000 0.000 0.360 0.640 0.000 0.000 0.292 0.123 0.585 0.000 0.000 0.000 0.000 1.000]
LightGBM	1.000	0.707	0.608	0.639	0.607	[0.040 0.960 0.000 0.000 0.360 0.640 0.000 0.000 0.000 0.123 0.877 0.000 0.000 0.000 0.000 1.000]



(**a**) (**b**)

Figure 13. Performance analysis of the naive solution: (**a**) barplot of the obtained results (barplot of naive solution) and (**b**) the variation of accuracy and F1 score versus cumulative important feature numbers (performance versus the cumulative important features).

5.4. ML Explainability

Explainable machine learning solutions offer insights into how the models make decisions, the most influential attributes, and why certain predictions are made to provide human–machine decision comprehension. These solutions are critical in fault detection because they increase the confidence and transparency of the models, allowing users to validate the judgments taken. In this study, XAI is employed by two distinct tools. The first tool is the SHAP explainer, also known as Shapley Additive exPlanations, which is utilized to discern the influence of features on the model's outputs. This is achieved by calculating the Shapley values for each feature, representing the average contribution of a particular feature across all conceivable feature permutations. The second tool is the LIME analyzer, which is applied to assess two specified samples and the corresponding decisions made by the models. LIME works by approximating the decision boundary of a complex model in the vicinity of a specific instance or data point. It generates a simpler, interpretable model that approximates the behavior of the complex model for that particular instance. This simpler model is often more understandable, allowing users to gain insights into how individual predictions are influenced by different features.

Figure 14a illustrates that the binary Extra Trees classifier necessitates the utilization of six features for optimal decision-making. The magnitudes of importance and their impacts on performance vary. Specifically, the range *i* features emerge as primary contributors to the Extra Trees decisions, exhibiting a balanced influence on the detection of both faulty and non-faulty samples. In the fault diagnosis phase, as presented in Figure 14b, it is observed that the range 4 and range 3 features alone are adequate for the model to formulate decisions. Notably, range 4 appears more directed towards detecting F3 faults, while range 3 significantly impacts identifying F1 and F2 faults. Nevertheless, both features contribute to the diagnosis of each fault type to some extent. This analysis highlights individual features' differential contributions and influences on the binary Extra Trees classifier's decision-making process.

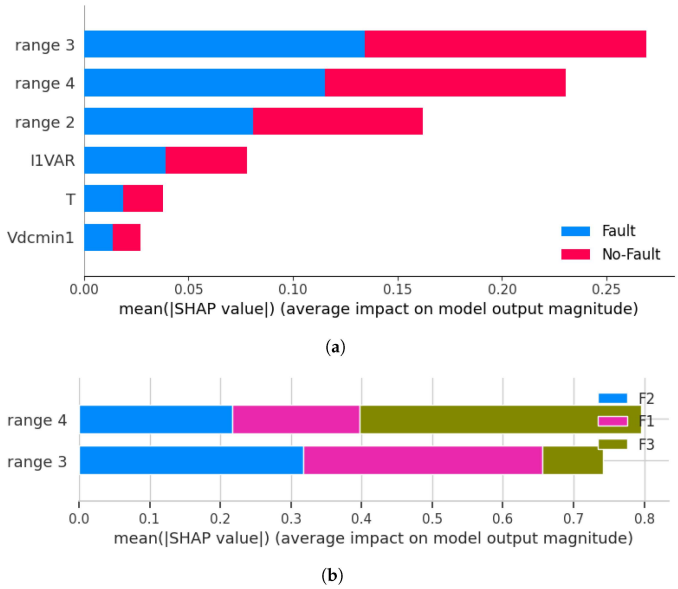
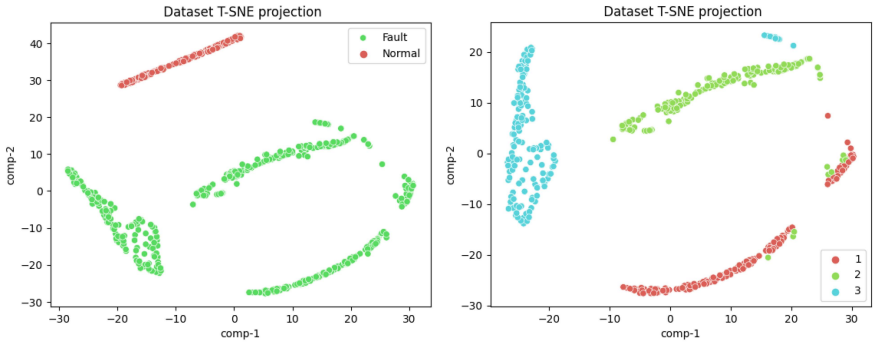


Figure 14. SHAP explainability of the model's output: (**a**) fault detection (first phase—first-phase fault detection: feature impact on the model's output) and (**b**) fault diagnosis (second phase—second-phase fault diagnosis: feature impact on the model's output).

Two samples were examined for the first phase and two for the second phase. The decision-making process of the model was observed at the sample level, and the features it relies on were identified using the LIME analyzer. Analyzing the LIME results in Table 9, it

becomes apparent that, for these specific samples in the binary classification first-phase problem, the key contributors to the model's decision in the first sample are range 1, 3 with significant impact, while I5, Itotalmin1, and Vdcmax1 contribute to a lesser extent. The same pattern is observed in the second sample, with varying degrees of contribution for each feature. Transitioning to the second phase of multi-class classification, the analysis indicates that the range 3, 4 features are predominant in detecting the fault class among the three provided. These findings align with those of the SHAP analyzer, which provides an aggregation of contributors across the entire dataset. The SHAP analyzer suggests a total of six contributors in the first fault detection phase and two in the second phase. The obtained results and explanations by SHAP were also reviewed using the t-distributed Stochastic Neighbor Embedding (t-SNE) perplexity projection in a 2D space in Figure 15. The use of the range features as components of the t-SNE projection aids in the separation of normal and faulty cases, which aids the model in predicting each one. This projection also helps in recognizing and categorizing fault cases based on their types (normal, F1, F2, or F3).



(**a**) (**b**)

Figure 15. t-SNE projection of (**a**) faulty and normal cases in binary classification (binary classification) and (**b**) faulty cases in multi-class classification (faulty multi-class classification). (1, 2, and 3 represent the first, second, and third faults, respectively).

Table 9. Explainability analysis results.

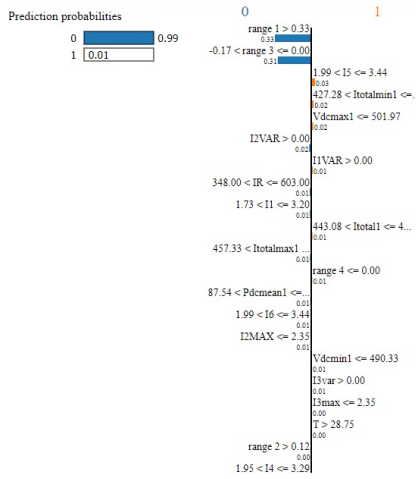
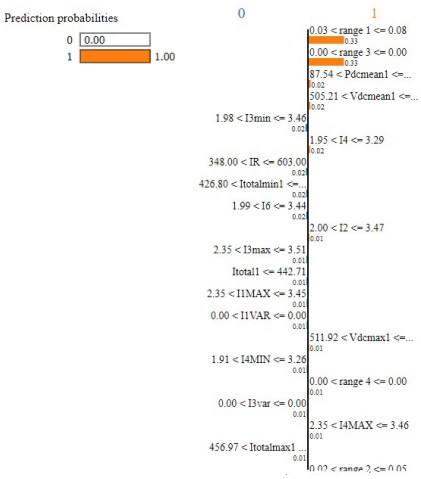
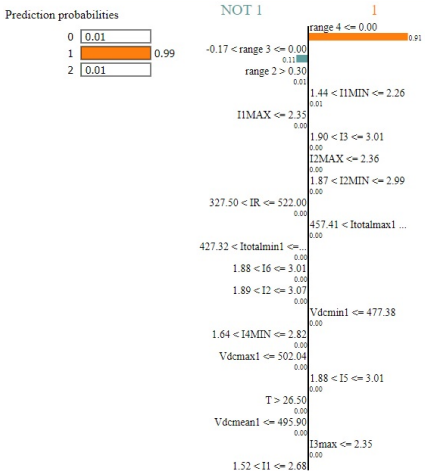
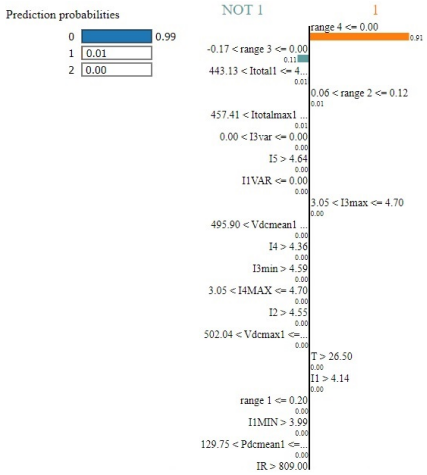
Classification	**Samples**	
	Sample 1	**Sample 2**
Binary		
Explanations	In this particular instance, focusing on binary classification, the model exhibits a high level of confidence (99%) in categorizing the case as a normal (0) operation of a PV panel. This determination is strongly supported by two features, namely range 1 and range 3, while three other features exhibit a negative correlation with this classification. At least five features play a significant role in fine-tuning the model's decisions toward achieving an optimal result.	This decision to consider the sample faulty was supported by two features, "range 1" and "range 3", each by 33%. In total, a 66% contribution is made. While no significant contribution to the opposing decision is observed (0.02c), the model is 100% certain that it is a Faulty sample.

Table 9. *Cont.*

Classification	Samples	
	Sample 1	**Sample 2**
Multi-Class		
Explanations	In this situation, just two features have a choice about the type of fault; range 4 believes (91% of the time) that it is the second fault type, whereas range 2 believes (11% of the time) that it is not; the model favors the certainty of the range 4 feature in this decision.	In this situation, just two features have a choice regarding the type of fault; range 4 believes it is the First Fault type by 91%, while range 2 believes it is not by 11%; the model chooses the certainty of the range 4 feature toward this decision.

6. Conclusions

This research focused on detecting and diagnosing faults in PV array systems. This study utilized a dataset generated from the MATLAB Simulink tool, simulating a 250 kW PV power plant. The dataset comprised 700 samples, encompassing three fault types and normal cases. Initially, a comprehensive dataset analysis was conducted to improve its quality and understand the relationships between its features and the target variable. A novel feature importance averaging technique was then employed to identify the most influential features on the decisions of tree-based models. This technique effectively reduced the number of features to nine. The research employed a two-phase system for fault detection and diagnosis. The first phase focused on detecting faults from normal cases, while the second phase involved diagnosing the exact fault type (F1, F2, or F3) in faulty samples. An ensemble of six tree-based classifiers, including decision trees, random forest, Stochastic Gradient Boosting, LightGBM, CatBoost, and Extra Trees, was used in both phases. The results indicated that the first phase achieved the highest accuracy, reaching 100% with the Extra Trees classifier. In the second phase of fault diagnosis, four classifiers—Extra Trees, XGBoost, LightGBM, and CatBoost—achieved 100% accuracy, with Extra Trees demonstrating the fastest training and convergence time. A novel post-processing technique was integrated to determine the optimal number of features for achieving 100% accuracy. This study revealed that six features were necessary for the first phase and two for the second, with the significant contribution coming from the range *i* features. XAI was finally employed to validate the research findings through two analyzers: LIME and SHAP. Future work will focus on validating the proposed framework with publicly available real-world PV datasets to assess its practical applicability.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in GitHub at https://github.com/amrrashed/Fault-Detection-Dataset-in-Photovoltaic-Farms/tree/main (accessed on 10 October 2024).

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Li, Z. Prospects of Photovoltaic Technology. *Engineering* **2023**, *21*, 28–31. [CrossRef]
2. Sabbaghpur Arani, M.; Hejazi, M.A. The comprehensive study of electrical faults in PV arrays. *J. Electr. Comput. Eng.* **2016**, *2016*, 8712960. [CrossRef]
3. AbdulMawjood, K.; Refaat, S.S.; Morsi, W.G. Detection and prediction of faults in photovoltaic arrays: A review. In Proceedings of the 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018), Doha, Qatar, 10–12 April 2018; pp. 1–8.
4. Allal, Z.; Noura, H.N.; Salman, O.; Chahine, K. Leveraging the power of machine learning and data balancing techniques to evaluate stability in smart grids. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108304. [CrossRef]
5. Yao, Z.; Lum, Y.; Johnston, A.; Mejia-Mendoza, L.M.; Zhou, X.; Wen, Y.; Aspuru-Guzik, A.; Sargent, E.H.; Seh, Z.W. Machine learning for a sustainable energy future. *Nat. Rev. Mater.* **2023**, *8*, 202–215. [CrossRef]
6. Allal, Z.; Noura, H.N.; Vernier, F.; Salman, O.; Chahine, K. Wind turbine fault detection and identification using a two-tier machine learning framework. *Intell. Syst. Appl.* **2024**, *22*, 200372. [CrossRef]
7. Chahine, K.; Baltazart, V.; Wang, Y. Parameter estimation of dispersive media using the matrix pencil method with interpolated mode vectors. *IET Signal Process.* **2011**, *5*, 397–406. [CrossRef]
8. Chahine, K. Rotor fault diagnosis in induction motors by the matrix pencil method and support vector machine. *Int. Trans. Electr. Energy Syst.* **2018**, *28*, e2612. [CrossRef]
9. Lu, F.; Guo, S.; Walsh, T.M.; Aberle, A.G. Improved PV module performance under partial shading conditions. *Energy Procedia* **2013**, *33*, 248–255. [CrossRef]
10. Goudelis, G.; Lazaridis, P.I.; Dhimish, M. A review of models for photovoltaic crack and hotspot prediction. *Energies* **2022**, *15*, 4303. [CrossRef]
11. Knausz, M.; Oreski, G.; Eder, G.C.; Voronko, Y.; Duscher, B.; Koch, T.; Pinter, G.; Berger, K.A. Degradation of photovoltaic backsheets: Comparison of the aging induced changes on module and component level. *J. Appl. Polym. Sci.* **2015**, *132*, 42093. [CrossRef]
12. Luo, W.; Khoo, Y.S.; Hacke, P.; Naumann, V.; Lausch, D.; Harvey, S.P.; Singh, J.P.; Chai, J.; Wang, Y.; Aberle, A.G.; et al. Potential-induced degradation in photovoltaic modules: A critical review. *Energy Environ. Sci.* **2017**, *10*, 43–68. [CrossRef]
13. Nadeem, A.; Sher, H.A.; Murtaza, A.F.; Ahmed, N. Online current-sensorless estimator for PV open circuit voltage and short circuit current. *Sol. Energy* **2021**, *213*, 198–210. [CrossRef]
14. Alam, M.K.; Khan, F.; Johnson, J.; Flicker, J. A comprehensive review of catastrophic faults in PV arrays: Types, detection, and mitigation techniques. *IEEE J. Photovolt.* **2015**, *5*, 982–997. [CrossRef]
15. Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Mekhilef, S.; Idris, M.Y.I.; Van Deventer, W.; Horan, B.; Stojcevski, A. Forecasting of photovoltaic power generation and model optimization: A review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 912–928. [CrossRef]
16. Basnet, B.; Chun, H.; Bang, J. An intelligent fault detection model for fault detection in photovoltaic systems. *J. Sens.* **2020**, *2020*, 6960328. [CrossRef]
17. Kolodziejczyk, W.; Zoltowska, I.; Cichosz, P. Real-time energy purchase optimization for a storage-integrated photovoltaic system by deep reinforcement learning. *Control Eng. Pract.* **2021**, *106*, 104598. [CrossRef]
18. Martin, J.; Jaskie, K.; Tofis, Y.; Spanias, A. PV Array Soiling Detection using Machine Learning. In Proceedings of the 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, 12–14 July 2021; pp. 1–4.
19. Aslam, M.; Lee, J.M.; Altaha, M.R.; Lee, S.J.; Hong, S. AE-LSTM based deep learning model for degradation rate influenced energy estimation of a PV system. *Energies* **2020**, *13*, 4373. [CrossRef]
20. Rafati, A.; Joorabian, M.; Mashhour, E.; Shaker, H.R. Machine learning-based very short-term load forecasting in microgrid environment: Evaluating the impact of high penetration of PV systems. *Electr. Eng.* **2022**, *104*, 2667–2677. [CrossRef]
21. Chen, X.; Qu, G.; Tang, Y.; Low, S.; Li, N. Reinforcement learning for decision-making and control in power systems: Tutorial, review, and vision. *arXiv* **2021**, arXiv:2102.01168.
22. Wang, J.; Gao, D.; Zhu, S.; Wang, S.; Liu, H. Fault diagnosis method of photovoltaic array based on support vector machine. *Energy Sources Part A Recover. Util. Environ. Eff.* **2019**, *45*, 5380–5395. [CrossRef]
23. Ghoneim, S.S.; Rashed, A.E.; Elkalashy, N.I. Fault Detection Algorithms for Achieving Service Continuity in Photovoltaic Farms. *Intell. Autom. Soft Comput.* **2021**, *30*, 467–479. [CrossRef]

24. Chen, Z.; Chen, Y.; Wu, L.; Cheng, S.; Lin, P. Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions. *Energy Convers. Manag.* **2019**, *198*, 111793. [CrossRef]
25. Hichri, A.; Hajji, M.; Mansouri, M.; Bouzrara, K.; Nounou, H.; Nounou, M. Deep Learning based Fault Diagnosis in a Grid-Connected Photovoltaic Systems. In Proceedings of the 2022 19th International Multi-Conference on Systems, Signals & Devices (SSD), Setif, Algeria, 6–10 May 2022; pp. 1150–1155.
26. Tao, C.; Wang, X.; Gao, F.; Wang, M. Fault diagnosis of photovoltaic array based on deep belief network optimized by genetic algorithm. *Chin. J. Electr. Eng.* **2020**, *6*, 106–114. [CrossRef]
27. Badr, M.M.; Hamad, M.S.; Abdel-Khalik, A.S.; Hamdy, R.A.; Ahmed, S.; Hamdan, E. Fault identification of photovoltaic array based on machine learning classifiers. *IEEE Access* **2021**, *9*, 159113–159132. [CrossRef]
28. Garoudja, E.; Chouder, A.; Kara, K.; Silvestre, S. An enhanced machine learning based approach for failures detection and diagnosis of PV systems. *Energy Convers. Manag.* **2017**, *151*, 496–513. [CrossRef]