Improved Reliability of Photovoltaic Systems through Iterative Dataset and Feature Simplification

ShivaPrakash S[1], *Selvaraju. M*[2*], *Suresh.G*[3], *M. Muthuraj*[4], *Gokilhashree. M*[5], *Pon Mahesh* Kumar[6]

[1]Department of Mechanical Engineering, New Horizon College of Engineering, Outer Ring Road, Bellandur, Bengaluru, Karanataka – 560103
[2]Department of Mechanical Engineering, Rathinam Technical Campus, Coimbatore, Tamilnadu, India.
[3]Department of Mathematics, Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai 600062
[4]Department of Mechanical Engineering, NPR College of Engineering and technology, Natham, Dindigul, Tamilnadu
[5]Department of Medical Electronics, Sengunthar Engineering College, Thirucegode, Namakkal, Tamilnadu - 637205
[6]Department of Mechanical Engineering, Nandha college of Technology, Perundurai, Erode, Tamilnadu, India - 638052

[*]Corresponding author email: *kannanarchieves@gmail.com*

ABSTRACT

Photovoltaic (PV) systems have a high number of potential problems. The conventional types of security are often broken down. This resulted in the development of state-of-the-art and fully automated, AI-based, methods, specifically, ML, which have already proven their utility in PV prevention. The cost of AI algorithms is a significant factor because it is highly complex, although they are rapidly developed and evolved. To ensure that the traditional ML algorithms can be applied to protect PV arrays under the new AI algorithm development, this study proposes how to simplify the models of the ML. In this paper, a framework of constructing an aggregative model with numerous ML methods is presented. It is an iterative method, whose main aim is to simplify model training. To simplify the training the model process, it uses two methods. The first step that we undertake is to reduce the dataset of classes through a horizontal simplification strategy. The main assumption made in the first approach is that in case an algorithm is performing poorly in an attempt to correctly label smaller data sets, then it would most definitely fail even larger data sets. The second step is to apply a vertical simplification strategy that is implemented in a random forest algorithm to select the most effective characteristics and further reduce the dataset on each iteration. The proposed strategy proves to be effective and sound by applying it to one of the laboratory PV systems to two experimental situations with different datasets. We have tested the proposed method on a large-scale PV system based on MATLAB/Simulink. The figures indicate that the test accuracy is 100 % in the case, 99.59 % in the second and 99.17 % in the third. The approach is also more effective in a variety of aspects compared to other similar studies that have been published in the past.

Keywords: Photovoltaics, Machine learning, Fault detection, Dataset, Random Forest.

1. Introduction

Photovoltaic fault detection is critical for ensuring the safety and reliability of PV arrays, as conventional protection mechanisms often fail to detect complex electrical faults. Although AI-based machine learning techniques have demonstrated strong fault detection

capability, their high computational complexity limits practical deployment. Therefore, this study introduces a simplified machine learning framework using dataset shrinkage techniques and random forest–based feature selection to enable efficient and accurate PV fault protection. An AI-based stacking regression model is proposed for photovoltaic fault detection and power prediction using machine learning. By integrating random forest with ensemble learners, the model efficiently captures fault-induced variations in current-carrying conductors under real operational conditions [1]. A comparative AI-based photovoltaic fault detection study employing machine learning classifiers highlights the effectiveness of random forest and boosting models. Electrical faults affecting current-carrying conductors are accurately identified using simulated datasets without complex feature reduction [2]. AI-based machine learning framework for photovoltaic fault detection using optimized ensemble models. Random forest combined with dataset balancing techniques improves fault detection accuracy for current-carrying conductor anomalies under partial shading conditions [3]. An AI-based machine learning approach for photovoltaic fault detection is validated using real-time experimental data. Random forest and ensemble methods demonstrate superior fault detection performance for current-carrying conductor faults under practical operating conditions [4]. Deep neural network feature extraction with machine learning ensembles for photovoltaic fault detection under limited data. Dataset shrinkage techniques combined with random forest significantly enhance AI-based fault detection accuracy in current-carrying conductors [5].

A simplified AI-based photovoltaic fault detection framework is proposed using iterative dataset shrinkage techniques. Random forest–driven feature reduction enables effective machine learning–based fault detection in current-carrying conductors with reduced computational cost [6]. AI-based photovoltaic inverter fault detection using machine learning with dimensionality reduction. Dataset shrinkage techniques combined with random forest enable accurate fault detection in current-carrying conductor pathways with improved computational efficiency [7]. An AI-based machine learning investigation compares advanced ensemble models for photovoltaic fault detection and power prediction. Random forest and boosting algorithms achieve near-perfect fault detection accuracy for current-carrying conductor failures in large-scale datasets [8]. AI-based photovoltaic fault detection using machine learning classifiers and deep learning image analysis. Random forest–driven fault detection effectively identifies current-carrying conductor degradation and thermal anomalies under realistic fault scenarios [9]. An AI-based machine learning framework is proposed for photovoltaic fault detection across multiple operating modes. Ensemble models, particularly random forest and XGBoost, enable high-accuracy fault detection in current-carrying conductor systems with minimal false alarms [10].

There are still some considerable barriers to overcome, even though there is sample information in the literature on the use of ML models to detect PV failures: (i) Most models in the literature are not capable of detecting complex and serious faults: they can only detect simple and easily-identifiable faults. (ii) Many of the models need massive training data, difficult to obtain, particularly in some settings. (iii) There are models that lack the required accuracy to be useful in the real world, particularly in terms of detecting critical faults that are not unlike those of normal operational conditions. Studies should therefore focus at developing practical, efficient and accurate models of fault detection to ensure overcome barriers and can extensively applied in PV applications in the real world.

2. Problem formulation

As illustrated in Figure 1, environmental stressors that face PV components result in electrical malfunctions and failures that occur out of the constant operation of these components in open environments.

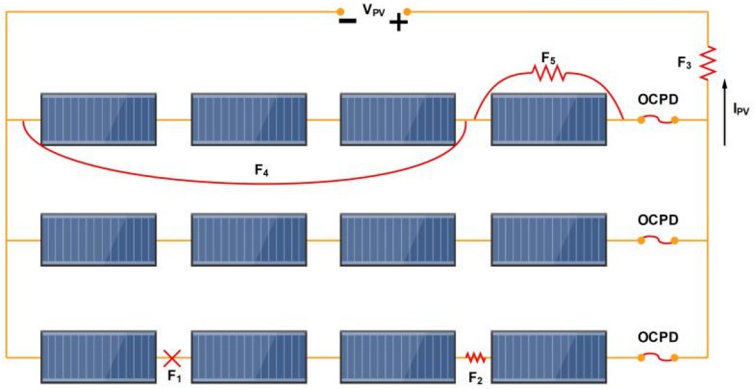


Fig. 1. Schematic view of PV Electrical Faults

Figure 1 defines a current-carrying conductor (CCC) as an unexpected fault in the current carrying conductor (OCF), which is referred to as open-circuit (OC) fault. The typical offenders of these types of issues are the microcracks within photovoltaic cells or the malfunctioning PV modules, or improper connections, in particular, the junction and wiring boxes.

Moreover, long term outlay on the sun radiation may bring deterioration problems in PV systems [11]. Destruction of the entire PV array or of individual strings can lead to a reduction of power generation. As illustrated in figure 1, A2 is degradation of the strings (String DEG) and A3 degradation of the arrays (Array DEG). Fig. 1 shows that the impedance is a character that represents these flaws of degradation that lower the power generation efficiency of the array.

Finally, A4 and A5 shows line-to-line fault events that arise when two sites of a PV array with varying voltage potentials by chance come into contact with each other. The main causes of LL faults are (i) unintentional short-circuing of two CCCs, (ii) gross insulation failure in the cables, or (iii) mechanical stress, water intrusion or corrosion which leads to internal short-circuiting in the DC junction boxes.

Common protective equipment including the overcurrent protection device (OCPD) in Figure 1 are always present to protect the PV arrays against fault and failures. Regrettably, there are various aspects that may bring about the protective devices to act inadequately and swiftly in the event of an issue. The primary impediments are low levels of irradiation, blocking diodes, and MPPTs. In general, they hinder the capabilities of protection equipment to detect faults. This is because in case of any fault, the current into the protection devices would be constricted. Consequently, the failure takes place because of insufficient current on the protection mechanisms. This implies that the issues are not being noticed within the system, and they may end up destabilizing the processes up to the point of causing fire.

In Fig. 2(a) and Fig. 2(b), the I-V curve of PV under various circumstances is observed. As shown in the figure, the mentioned faults may cause output voltages and currents to be less

than expected, thereby resulting in the deviation and disturbance of PV operating currentvoltage (I -V) point and the highest power production both prior to the failures and after. This has led to the fact that fault detection and removal must be timely and precise to ensure that the PV components would be safeguarded against possible effects.

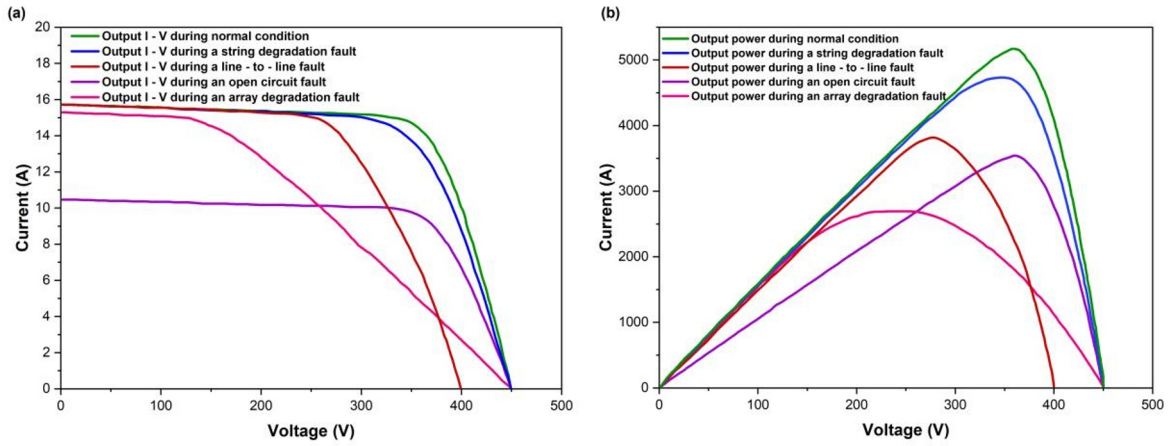


Fig. 2. PV I–V and P–V Under Faults

Protective devices have an even harder task detecting fault in PV arrays as soon as critical conditions occur. In such instances, fault impedance and/or a few PV modules are critical to the fault. The critical conditions complicate the task of protection devices to detect issues because they are so close to the usual operating conditions in a PV array. Figure 3 shows that there is a serious case of LL fault where the fault impedance value (20 in this case) has exceeded the ideal value and only one out of ten modules has been affected. Following the malfunction, the operating point of the PV array which is the output voltage and current, exhibits only a minor change. Due to the existence of high level of resemblance and the bounds of OCPDs, it is almost difficult to detect this blemish by them.

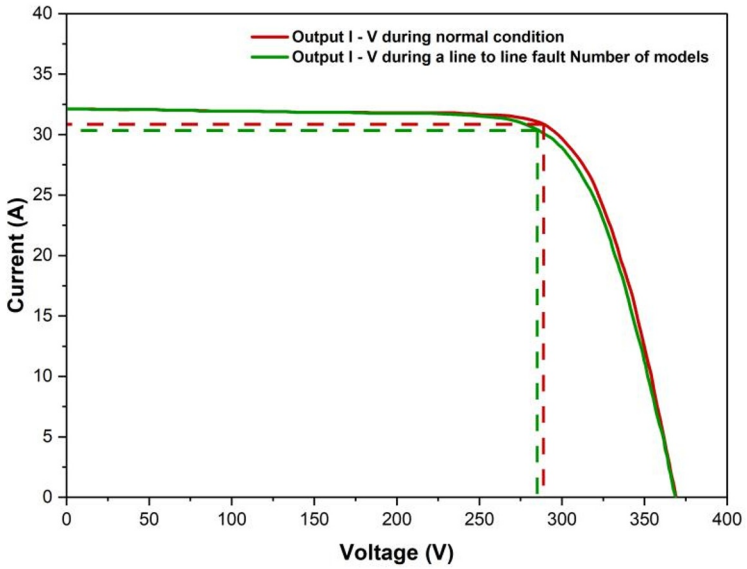


Fig. 3. Fault–Normal Overlap at STC

1.1. Paper contributions

The study presents some of the novelties and contributions that assist in addressing the gaps mentioned above and overcoming the challenges identified:

Finally, the aggregative model considers all the best classifiers and eliminates those that are unable to deal with more difficult datasets. That is because the elimination process (SPE) of all the algorithms was done with care and nomination. The paper goes further to explore the issue of how many classifiers are best to build a SPE-based aggregate model in light of the trade-offs between simplicity and accuracy.

The former models have utilized some of the properties of the I-V curve of PV arrays. Conversely, our study is the first to apply Euclidean distance approach in order to extract attributes of the I-V curve of a PV array, on the basis of a specified number of predefined points. A RF algorithm is used to remove redundant attributes and minimize the dimensionality of a dataset, which further simplifies the final model computationally.

Experiments on the performance of final SPE aggregative model tested in critical faults situations, illustrate the exceptional faultfinding capability of the model; it is the conditions that are infamously challenging and difficult to detect faults at a PV array. By adding an experimental case of PV array faults, we can also test the strength of the proposed model.

2. Proposed Aggregative ML Model

The initial steps of the process and its development can be observed by the further steps in the figure in Figure 4. Early detection of line-to-line and open-circuit faults reduces the probability of arc formation and thermal runaway, thereby mitigating fire risks in large-scale PV installations. In the first place, to collect data samples, the proposed model defines five different points on the I-V curve of the PV array output. After this, a set of attributes is extracted, based on Euclidean distances between every two of these five points. The data is then entering the elimination process in a systematic manner. Finally, the process ought to give a model that is aggregative and capable of ranking the test dataset. In this section, the complexity of Figure 4 is discussed.

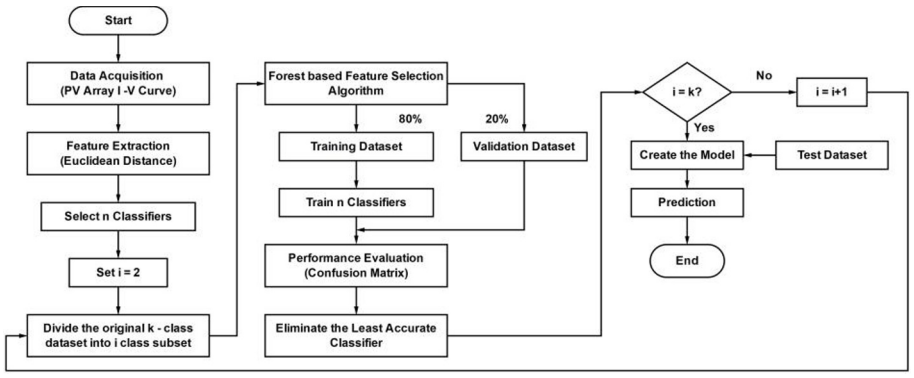


Fig. 4. Proposed Method Flowchart

Figure 4 illustrates the iterative progression from dataset formation to horizontal simplification, classifier elimination, vertical feature reduction, and final aggregative reconstruction.

2.1. Formation of Initial Dataset

The first part of the data collection is to split the training dataset into an 80/20 split to train the model and validate it and collect the test data at five set points at I-V curve.

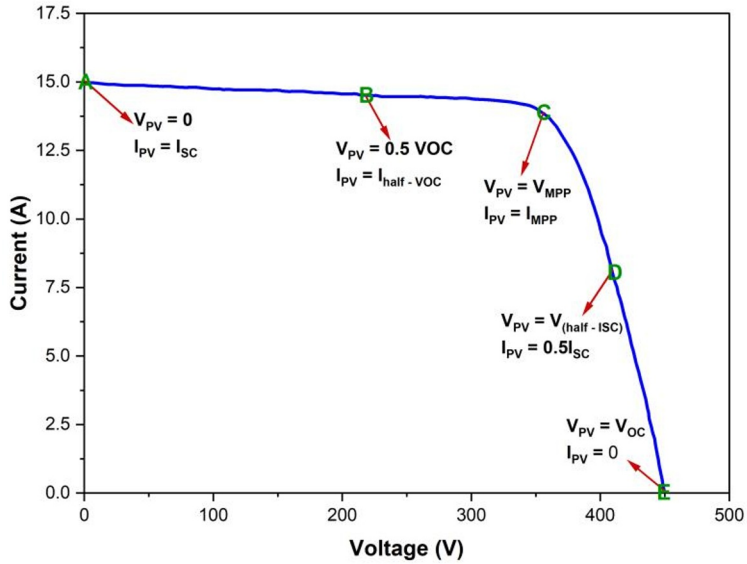


Fig. 5. Five I–V Curve Points

Points were extracted on I-V curves with different fault as well as normal conditions (i.e., no fault). Other electrical parameters including the amount of modules being affected by the faults (differing values of mismatch) and the values of the fault impedance, and other environmental variables, including different temperatures and irradiance values are also taken into consideration during the formation of the curves.

2.2. Euclidean Distance–Based Attributes

When constructing ML models feature extraction reduces amount of time it takes to train the model and makes it easier to interpret the model. In the present research, a comprehensive analysis of the I-V curve PV array were carried as a part of the feature extraction process. Here, there were five points at I-V curve that have already been formed; with the distances that are measured between them, ten attributes (A1-A10) are formed.

One mathematical definition of Euclidean distance is the length of the straight line segment between two points in space. In the n-dimensional Euclidean space and using Cartesian coordinates of the points, the distance can be written as (1):

$$d(p,q) = \sqrt{\sum_{j=1}^{n} \left(q_j - p_j\right)^2} \tag{1}$$

The p_j and q_j of the n-dimensional space defined by the Euclidean geometry, and the vectors p and q which are the extensions of the base of the space. As a result, ten extracted attributes (A1-A10) are displayed in Fig. 6 as per the distance measured in centimeters as outlined in the I-V curve of the PV array.

The Euclidean distances between selected I–V characteristic points capture geometric deformation patterns of the curve under different fault conditions. Since PV faults primarily

alter slope, knee position, and curvature of the I–V profile, distance-based metrics provide a compact representation of these nonlinear deviations.

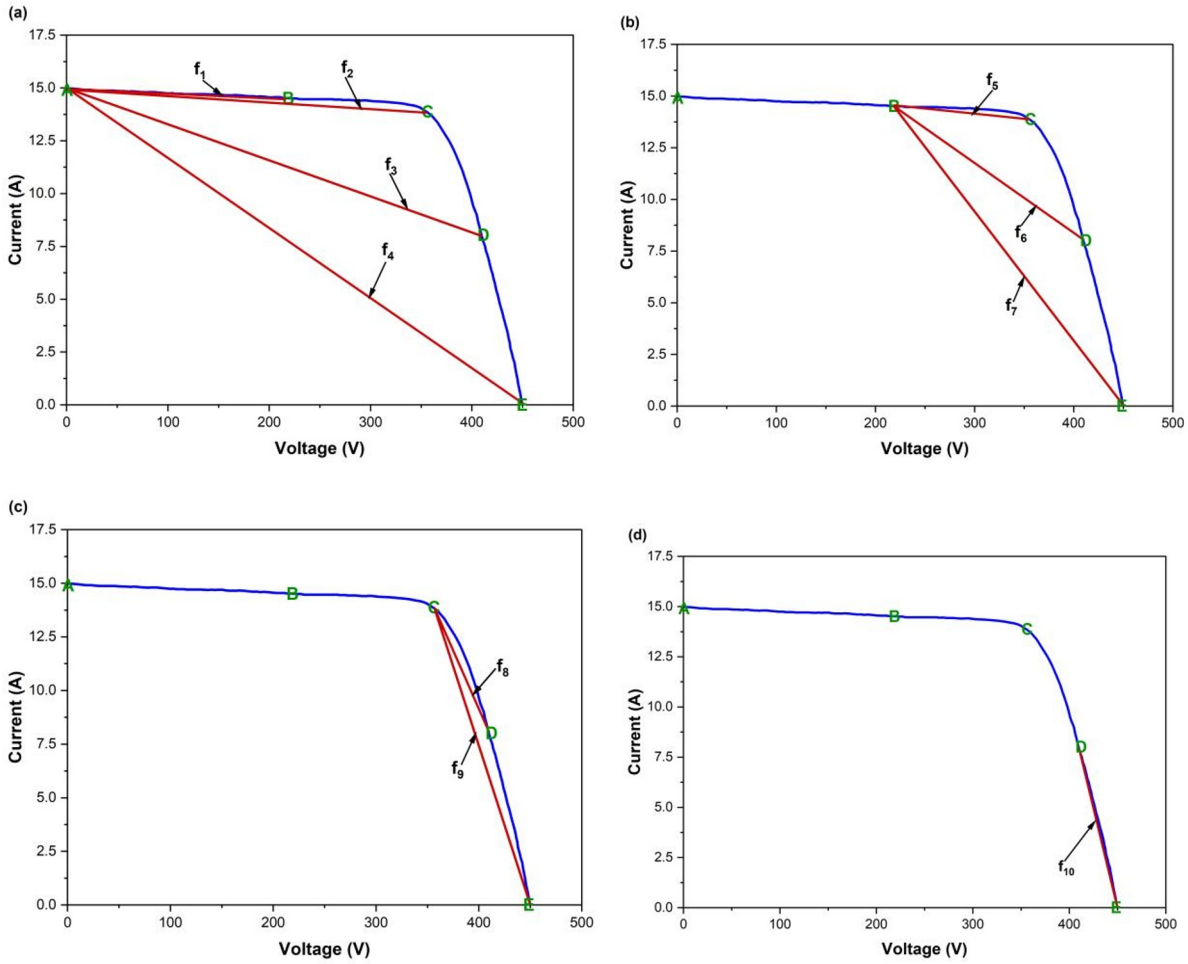


Fig. 6. Euclidean Distance Feature Set

2.3. Sequential Elimination Process

It is necessary to initially designate the most accurate classifiers in order to construct an accurate aggregative model. In this paper, we present iterative multi-sequence method based on sequential elimination procedure. At the end of each cycle, or iteration, the algorithm discards the classifier. The quantity of sequences are easily figure out by plugging the number of classes into Eq. (2), which is directly proportional to the dataset size:

$$NS = NC\text{-}1 \tag{3}$$

The first step is to pick n classifiers at random, ideally from a big pool. With all n classifiers operational in the first iteration, we partition the dataset into several two-class subsets using every conceivable mathematical permutation from Eq. (3).

$$C(k,i) = \frac{k!}{i!(k-i)!} \tag{4}$$

Fig. 7 shows that, there is a horizontal simplification process taking place, which involves reducing number of classes. The main reason for this is that if optimized classifier can't fall a dataset with small classes were obviously easier to classify—definitely struggle to provide accurate results at datasets with a large class. This work makes use of a horizontal dataset simplification, as shown in Fig. 7, which involves dividing a huge dataset into smaller portions.

Sample Number	Feature 1	Feature 2	Feature 3	Feature f	Classes
1	V_{1f1e1}	V_{1f2c1}	V_{1f3d1}	V_{1fnf1}	Class 1
2	V_{2f1e1}	V_{2f2c1}	V_{2f3d1}	V_{2fnf1}	
3	V_{3f1e1}	V_{3f2c1}	V_{3f3d1}	V_{3fnf1}	
n	V_{nf1e1}	V_{nf2c1}	V_{nf3d1}	V_{nfnf1}	
1	V_{1f1e2}	V_{1f2c2}	V_{1f3d2}	V_{1fnf2}	Class2
2	V_{2f1e2}	V_{2f2c2}	V_{2f3d2}	V_{2fnf2}	
3	V_{3f1e2}	V_{3f2c2}	V_{3f3d2}	V_{3fnf2}	
m	V_{mf1e2}	V_{mf2c2}	V_{mf3d2}	V_{mfnf2}	
1	V_{1f1ecc}	V_{1f2ccc}	V_{1f3dcc}	V_{1fnfcc}	Class n
2	V_{2f1ecc}	V_{2f2ccc}	V_{2f3dcc}	V_{2fnfcc}	

Fig. 7. Horizontal Simplification via Class Reduction

The next step is to divide the subsets into two datasets: one for validation (20%) and one for training (80%). We train each classifier with the data from the two-class combinations' training sets, and then we use the validation sets to assess how well the classifiers did using the A1-score assessment measure. That is why there are as many iterations of training and validation for each classifier as there are subgroups. Consider a dataset with four classes; according to Eq. (3), the first sequence would have six subsets. Hence, we will train each classifier six times. Every sequence concludes with the computation of the mean validation A1-score for every classifier. The first step was to remove the classifier with the lowest mean A1-score. Next, we eliminate the least accurate classifier and start over with three subsets of n-1 remaining classifiers and original dataset. When all classes have been added to the subset, the process ends and the initial dataset is rebuilt. The last step involves using the entire training dataset to build aggregative model with classifiers. The proposed horizontal and vertical simplification strategies reduce computational burden by limiting class combinations and feature dimensions during early iterations. The effective training complexity is reduced approximately proportional to $O(n \times m \times k)$, where n represents classifiers, m attributes, and k class subsets.

2.4. RF-Based Feature Selection

Dimensionality reduction is crucial for shrinking the dataset and reducing model complexity for better analysis. Dataset shrinkage techniques by feature selection significantly improve random forest and SVM performance in distinguishing current-carrying conductor faults with similar I–V profiles [12]. To determine which traits are most relevant in each scenario across all sequences, this research employs forest-based feature approach.

Using the "entropy" or "gini" index, the method finds most important attributes to reduce dataset impurity for each decision tree, and then iteratively creates the best nodes. With the help of Eq. (4) and Eq. (5), one may determine the indices. As an appropriate indicator for

determining the feature's efficacy in enhancing the model's correctness, it subsequently returns a mean value pertaining to number of trees.

$$\text{Entropy} = \sum_{i=1}^{c} - p_i \log_2 p_i \quad (4)$$

$$\text{Gini} = 1 - \sum_{i=1}^{c} (p_i)^2 \quad (5)$$

Sample Number	Feature 1	Feature 2	Feature 3	Feature f	Classes
1	V_{1f1e1}	V_{1f2c1}	V_{1f3d1}	V_{1fnf1}	Class 1
2	V_{2f1e1}	V_{2f2c1}	V_{2f3d1}	V_{2fnf1}	
3	V_{3f1e1}	V_{3f2c1}	V_{3f3d1}	V_{3fnf1}	
n	V_{nf1e1}	V_{nf2c1}	V_{nf3d1}	V_{nfnf1}	
1	V_{1f1e2}	V_{1f2c2}	V_{1f3d2}	V_{1fnf2}	Class2
2	V_{2f1e2}	V_{2f2c2}	V_{2f3d2}	V_{2fnf2}	
3	V_{3f1e2}	V_{3f2c2}	V_{3f3d2}	V_{3fnf2}	
m	V_{mf1e2}	V_{mf2c2}	V_{mf3d2}	V_{mfnf2}	
1	V_{1f1ecc}	V_{1f2ccc}	V_{1f3dcc}	V_{1fnfcc}	Class n
2	V_{2f1ecc}	V_{2f2ccc}	V_{2f3dcc}	V_{2fnfcc}	

Fig. 8. Feature-Reduction Simplification Technique

In this research, we made use of this quality to rank the attributes according to how important they are to the model's correctness. Hence, at vertical simplification process attributes that are not going to improve the model's accuracy are removed in every iteration. This allows for a smaller set of initial attributes to be used to train the model with fewer data samples. Figure 8 shows how the dataset presented in Figure 7 is further compressed and streamlined by removing unnecessary attributes.

2.5. Evaluation metrics

Partitioning the entire dataset as training (80%) and validation (20%) allows us to evaluate efficiency of individual classifiers with all combination and final ensemble model. To top it all off, we will use a test dataset that was not a part of the training or validation set to see how well the final aggregative model performed. The study begins by calculating the confusion matrix and then uses it to compute the class-wise A1-score. In machine learning, the A1-score is a measure of a model's predictive ability that focuses on its performance inside each class rather than its overall performance, as is the case with accuracy. As reported in previous studies, AI-based photovoltaic fault detection framework using machine learning with a two-layer stacking ensemble. Random forest–based meta-learning enables accurate classification of current-carrying conductor faults and shading conditions despite overlapping I–V characteristics, achieving high fault detection robustness [13]. Its typical application is in cases of data imbalance, such as when one class's sample size is dramatically higher than another's. Partitioning the entire dataset as training (80%) and validation (20%) allows us to evaluate efficiency of individual classifiers.

Hyperparameters were optimized using five-fold cross-validation with grid-search tuning. Random Forest was configured with 100 trees using the Gini impurity criterion. SVM kernels were tuned across regularization parameter C ∈ {0.1, 1, 10} and γ ∈ {0.01, 0.1, 1}. Logistic regression used L2 regularization.

Table 1 Typical confusion matrix

		Predicted Label	
		Positive	Negative
Labels	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

• **TP (True Positive):** Cases where good results were actually positive

• **TN (True Negative):** Examples of accurately identified negative cases

• **FP (False Positive):** Examples of the Type I error include the incorrect classification of negative cases as positive.

• **FN (False Negative):** The occurrence of false negatives due to positive case classification (Type II error)

$$\mathrm{A1-\ Score\ } = \frac{\mathrm{TP}}{\mathrm{TP}+0.5(\mathrm{FP}+\mathrm{FN})} \tag{6}$$

3. Results and Discussion

3.1. Experimental testbed

We developed and erected a 3 × 6 PV array to test and put suggested model into action. The array consists of three strings, with six PV modules in each string. To round things off, Table 2 details the specs of the boost converter and PV module.

Table 2 Specifications of PV modules and the boost converter.

System/Module	Parameter	Value
Boost Converter	Inductor	2 mH
	Capacitor	390 μF, 400 V
	Switching frequency	40 kHz
	Load resistance	200 Ω
Yingli YL010D-18b PV Module	Isc	0.61 A
	Voc	22.5 V
	Impp	0.56 A
	Vmpp	18 V

Evaluating a PV array's I-V curve requires measuring the current from zero to ISC. This needs the use of a DC-DC boost converter, which is a programmable DC load. A DC-DC boost converter, an ARM controller, a sensing board, an InstruStarISDS205A oscilloscope card, and a 200 Ω resistive load are all parts of the I-V tracer system. In this setup, the duty cycle of the PWM signal controls the boost converter's resistance, making it act as a variable resistor. Sensing board records current and voltage at PV array's output. The microcontroller processes the values and sends them to a computer through the oscilloscope card at a sampling rate of 1 kHz. To create the I-V curves, the duty cycle is varied. Gaussian noise (σ = 2%) was

synthetically injected into voltage and current measurements during validation experiments. The final model maintained A1-score above 97%, confirming resilience to measurement noise.

Important considerations for the suggested method's realistic deployment in utility-scale PV facilities include the following. Field measurements have typically necessitated the use of specialist commercial tracers, which can be rather costly, or the tedious and time-consuming process of manually disconnecting strings for testing. This approach presents its fair share of logistical issues and the risk of revenue loss due to downtime. For most large-scale installations, the cost of continuously monitoring all strings' I-V levels would be too high due to this practical constraint.

The strategic benefits offered by the suggested method, however, more than make up for these difficulties. When put into practice, the system would continuously monitor using regular SCADA data and only initiate targeted I-V measurements in response to potential anomalies detected by the initial problem detection. While reducing the operational burden of I-V analysis, this hybrid monitoring technique maintains its diagnostic precision.

It is worth devoting special attention to the topic of how computational efficiency relates to actual implementation. Our emphasis on simplification resolves numerous undervalued real-world limitations, even though contemporary computing resources might potentially support more complicated algorithms. To start, lightweight models that cut down on hardware costs and energy usage are great for edge deployment at combiner boxes or inverters. Secondly, field technicians can make quicker maintenance decisions with simpler models because of their improved interpretability.

Finding a happy medium between diagnostic resolution and practicality of execution is crucial. By improving fault identification accuracy without necessitating continuous acquisition of I-V curves, the suggested technique strikes a reasonable compromise.

In conclusion, temporal restrictions may make it impractical to scale this system to PV monitoring throughout the entire facility. Targeted analysis, such as fixing underperforming strings identified by production data, is the primary emphasis of this research. In contrast to real-time production measurements, I-V curves reveal finer details, such the extent to which shading or degradation has occurred, or the degree to which there has been a mismatch. This compromise validates their application in particular contexts.

In order to ensure that the suggested strategy is robust, we have set up three separate datasets in a large-scale MATLAB/Simulink scenario and two separate experimental situations. Here we break out the outcomes for the first, second, and third scenario. In order to provide a comprehensive explanation, we will go into detail about the first scenario's results while providing brief illustrations of the second and third scenarios.

3.2. Results for the first scenario

Normal situations, three types of LL faults, and array degradation faults make up the sole five classes in the first scenario's baseline datasets. The first scenario's training and test datasets, retrieved under various electrical and environmental conditions, were detailed at Table 3. To cover broad spectrum of fault and normal conditions, cases have temperature values between 0 and 40◦C, irradiation between 100 and 1100 W/m^2, number of faulty modules in a string ranging from 1 to 6, and fault impedance between 0 and 25 Ω with 5 Ω increments. In addition, the model is ready to start the first sequence after constructing ten attributes based on

Euclidean distance. To prevent overfitting, stratified splitting ensured no overlap between training, validation, and test samples. Environmental parameters and fault impedance levels in the test dataset were distinct from training conditions, confirming independent evaluation.

Table 3 Data at First Scenario

Event	Label	Test samples	Training samples
line mismatch faults 20%	LL3	73	343
line mismatch faults 20%	LL2	25	115
line mismatch faults 10%	LL1	25	115
Array degradation faults	ARRDEG	50	580

String degradation (STRDEG) refers to localized mismatch within a single string, whereas array degradation (ARRDEG) represents uniform impedance-related power loss across multiple strings. The first step in preparing for the first scenario is to identify several traditional ML classifiers. On the other hand, to improve the model accuracy, it is best to start with a large number of classifiers. Since the dataset contains five classes (NC = 5), four sequences will be involved in building the model (NS = 5), as per Eq. (2).

3.2.1. Results for the first sequence

Every one of the ten classifiers is there in the first sequence. In this case, we split the original dataset into two classes. C (5,2) = 10 (refer to Eq. (3)) is the mathematical combination that determines the number of subsets, as previously mentioned, where k= 5 and i= 2. Next, we train ten classifiers using training datasets for all combination. Then, we use validation dataset to evaluate each classifier's A1-score. Figure 9(a) validation A1-scores for first sequence over each two-class subsets. Additionally, for each pair of classes, it displays which classifier has generated the lowest A1-score. The C (5,2) = 10 classification process, the first purpose of first series were identify classifier with lowest mean validation A1-score and exclude it from further consideration. Figure 9(b) shows that out of all ten combinations, MLP has the most eliminations and lowest mean validation A1-score (90.08%). Hence, MLP is removed from the first sequence and does not move on to the second.

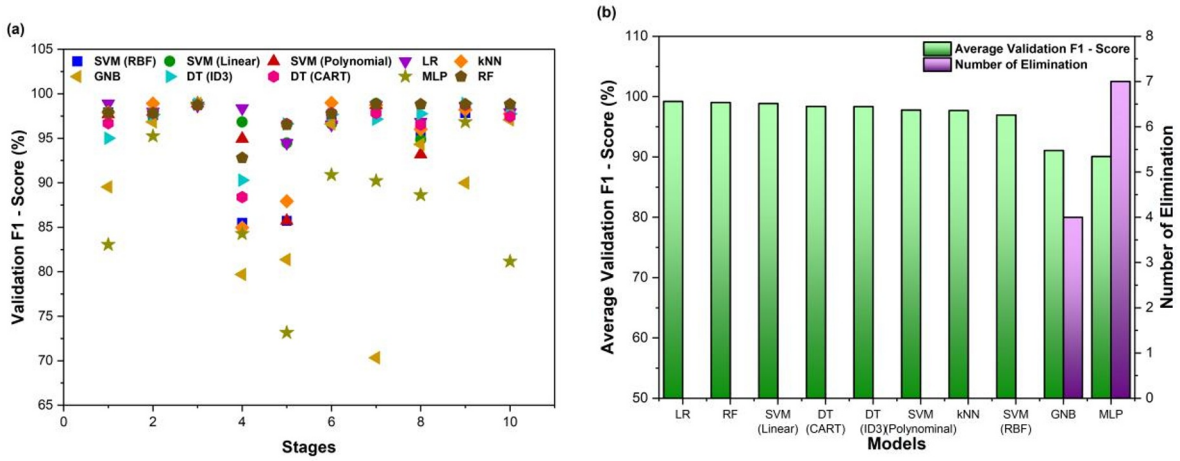


Fig. 9. (a) Validation A1-Score Evolution (b) Mean A1-Score and Eliminations

Table 4 RF Feature Selection Results

Grouping	**Attributes**
LL1	A_8, A_{10}
LL2	A_2, A_5
LL3	A_5
Degree	A_5, A_9, A_{10} A_8
LL1–LL2	A_5, A_2, A_8, A_9
LL1–LL3	A_2, A_5, A_3, A_6, A_8, A_9, A_4, A_{10}
LL1–Deg	A_{10}, A_8,
LL2–LL3	A_6, A_2, A_{10}, A_9, A_5,
LL2–Deg	A_7, A_4, A_8, A_{10}, A_1,
LL3–Deg	A_1, A_7, A_3, A_8, A_4,

4.3. Second Sequence Results

With i= 3 and nine methods left over from the first series (MLP excluded), C(5,3)= 10 3 subsets of original dataset are available in second sequence. Figure 10(a) shows A1-scores for the second sequence over all three subsets. Additionally, it displays classifier that has achieved the lowest A1-score for each set of three classes. In addition, as demonstrated in Figure 10(b), GNB were removed at conclusion of second sequence following C (5,3) = 10 classification operations utilizing 9 classifiers. With a mean A1-score of 80.01%, GNB had the fewest eliminations at nine.

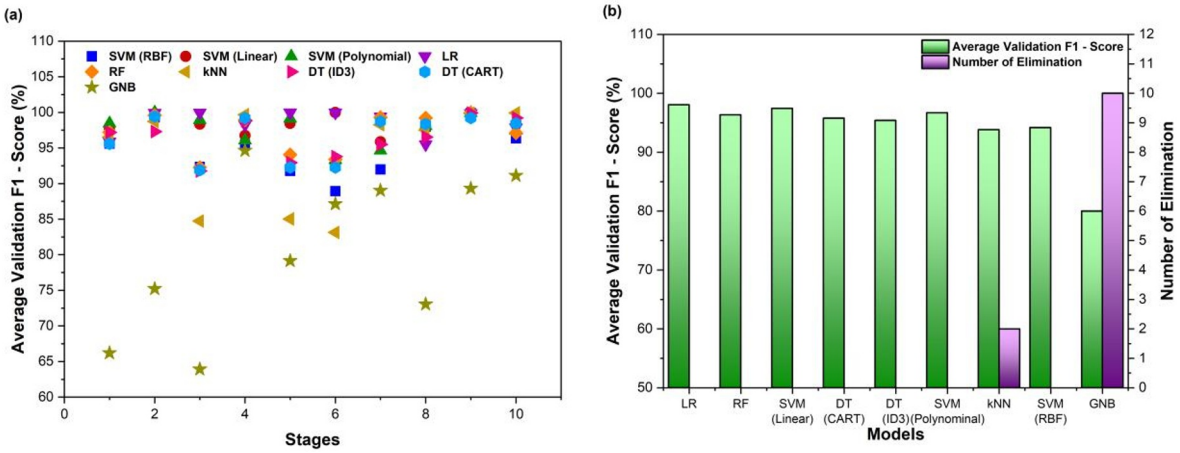


Fig. 10. (a) Validation A1-Score Evolution (b) Mean A1-Score and Eliminations

Table 5 RF Feature Selection Results

Grouping	**Attributes**
LL1–LL2	A_5, A_2, A_{10}, A_8,
LL1–LL3	A_5, A_2, A_6, A_3, A_{10}, A_5,
LL1–Degree	A_6, A_{10}, A_5, A_8,
LL2–LL3	A_3, A_2, A_9, A_6, A_8, A_5,
LL2–Degree	A_5, A_{10}, A_8, A_9,
LL3–Degree	A_1, A_7, A_9, A_5, A_{10}, A_2, A_8, A_4,
LL1–LL2–LL3	A_6, A_2, A_8, A_{10}, A_5,
LL1–LL2–Deg	A_7, A_{10}, A_1, A_8,
LL1–LL3–Deg	A_1, A_7, A_8, A_5, A_{10}, A_4
LL2–LL3–Deg	A_1, A_7, A_2, A_{10}, A_5, A_8, A_6, A_9 A_4

4.3.1. Results for third sequence

Up till now, the weakest classifiers on the list have been MLP at first sequence and GNB at second. Hence, eight of the ten algorithms are still present in this sequence. We also increase the number of classes in each subgroup by one, bringing the total to four. In this case, we generate five C(5),(4) = four subsets. Figure 11(a) displays validation A1-scores for the third sequence over all four classes (horizontal axis). Figure 11(b) shows that even though the least accurate classifier changes for all combination, model eventually excludes kNN by two red bars representing eliminations and a mean A1-score of 93.09%.

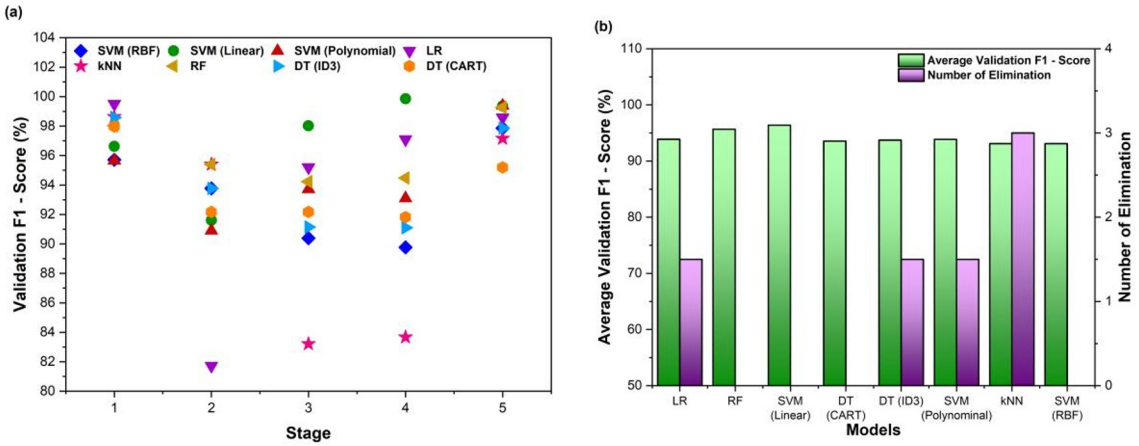


Fig. 11 (a) Validation A1-Score Evolution (b) Mean A1-Score and Eliminations

Table 6 RF Feature Selection Results

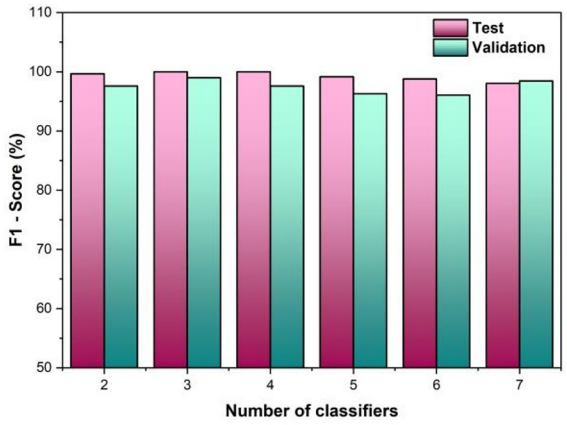
Grouping	**Attributes**
LL1–LL2–LL3	A_8, A_2, A_6, A_{10}, A_5
LL1–LL2–Degree	A_{10}, A_8
LL1–LL3–Degree	A_1, A_4, A_8, A_{10} A_7
LL2–LL3–Degree	A_1, A_4, A_2, A_5, A_8, A_{10}, A_9, A_6 A_7
LL1–LL2–LL3–Degree	A_1, A_7, A_{10}, A_8, A_2, A_6, A_4

4.4. Results for last sequence

As previously mentioned, last sequence starts when the final aggregative model is being created, i = k, i were number of classes added all subset and k were total number of classes at all sequence. After removing MLP, GNB, and kNN from the dataset in earlier sequences, seven of the ten remaining classifiers—among most accurate—make up final aggregative model at last sequence, which regenerates the entire five-class dataset. Furthermore, as mentioned earlier, this work determines the ideal classifiers—that is, m-classifier for m = 2, …, 7—as simplicity and achieving best test A1-score. Following the confirmation of the remaining classifiers, this procedure finds the best classifiers to build aggregate model using validation dataset. With an A1-score of 98.16%, three-classifier SPE- aggregative model comprising SVM (Linear), outperforms all other combinations of classifiers in the validation process (Fig. 12). Additionally, the winning aggregative model achieves maximum A1-score of 100% when tested with unknown data samples. Fig. 12 also includes testing accuracies of m-classifier models, which were not required but make for a better comparison. Thus, even though four-classifier aggregative model had also achieved a perfect score on the test A1-score, we have decided against nominating it as our final model for two reasons: first, it is more complicated than the other model due to the inclusion of an additional classifier, and second, we use

validation A1-score rather than test evaluation metric to nominate models. To add, keep in mind that the final sequence also employs RF feature selection method to identify optimal attributes for every model.

(a)



(b)

Combination	Selected Features (Prioritized)
Two - classifier model	$f_{10}, f_1, f_8, f_4, f_2, f_5$
Three - classifier model	$f_{10}, f_1, f_8, f_4, f_2$
Four - classifier model	$f_{10}, f_1, f_8, f_4, f_2, f_5$
Five - classifier model	$f_{10}, f_1, f_8, f_4, f_2, f_5, f_9, f_7, f_6$
Six - classifier model	$f_{10}, f_1, f_8, f_4, f_2, f_5, f_9, f_7, f_6$
Seven - classifier model	$f_{10}, f_1, f_8, f_4, f_2, f_5, f_9, f_7, f_6, f_3$

Fig. 12 (a) Validation–Test A1 Performance (b) Selected Feature Prioritization

4.5. Model further verification (Second scenario)

Adding string open circuit faults and degradation faults to existing dataset and creating larger dataset for second scenario allows us to further test the method's practicality and robustness. Table 7 shows all the information about second dataset, including test and training samples.

Table 7 Second Scenario Data samples

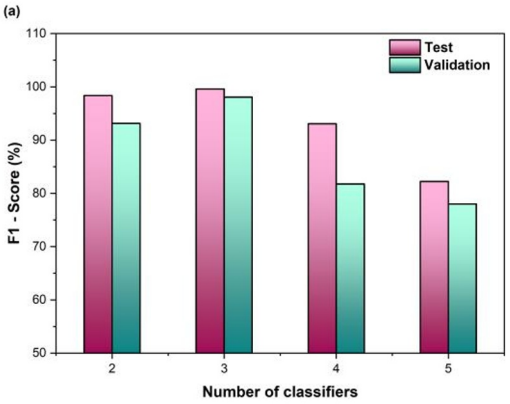
Method	**Label**	**Training**	**Test**
String faults	STRDEG	574	50
Array faults	ARRDEG	574	50
Normal condition	N	414	62
Open faults	OC	422	62
Line mismatch 20%	LL2	126	26
Line mismatch 10%	LL1	126	26
Line mismatch 20%	LL3	314	74

Using Eq. (2), we can deduce that NS = 6 since NC = 7 in the updated dataset. Figure 13 shows the five sequences' average A1-scores. At conclusion of each sequence, the least accurate classifiers are also displayed using a cross. At conclusion of first sequence, MLP were excluded with an validation A1-score of 85.34%, as shown in Figure 13. Next, we go on to the removal of GNB (73.11%), kNN (74.29%), SVM (RBF) (75.23%), and LR, in that order. Finding the best combination to build aggregative model is the last (sixth) step in the process.

First Sequence	Second Sequence	Third Sequence	Fourth Sequence	Fifth Sequence
SVM (RBF) 93.12%	SVM (RBF) 86.07%	SVM (RBF) 81.36%	SVM (Linear) 87.49%	SVM (Linear) 79.67%
SVM (Linear) 96.27%	SVM (Linear) 91.88%	SVM (Linear) 90.47%	SVM (Polynomial) 79.28%	SVM (Polynomial) 79.58%
SVM (Polynomial) 95.07%	SVM (Polynomial) 89.49%	SVM (Polynomial) 84.13%	LR 78.14%	DT (ID3) 72.05%
LR 96.63%	LR 91.36%	LR 84.05%	DT (ID3) 76.93%	DT (CART) 72.26%
GNB 86.37%	kNN 91.17%	DT (ID3) 78.62%	DT (CART) 75.75%	RF 72.13%
kNN 91.17%	DT (ID3) 94.09%	DT (CART) 78.39%	RF 75.11%	LR 62.24%
DT (ID3) 94.09%	DT (CART) 93.32%	RF 79.04%	SVM (RBF) 75.23%	
DT (CART) 93.32%	RF 93.97%	kNN 74.29%		
RF 93.97%	GNB 73.11%			
MLP 85.34%				

Fig. 13. Second scenario Model performance

Figure 14 shows that, similar to the previous case, all potential combinations were explored and models with the highest validation A1-scores were presented. The model's excellent A1-score of 99.59% on the unknown dataset is further evidence of its impressive performance. For a better comparison, Fig. 14 also provides test accuracies of various m-classifier aggregative models at case, similar to the first scenario.



(b)

Combination	Selected Features (Prioritized)
Two - classifier model	$f_{10}, f_1, f_8, f_4, f_2, f_5, f_9$
Three - classifier model	f_{10}, f_1, f_8, f_4
Four - classifier model	$f_{10}, f_1, f_8, f_4, f_2, f_5$
Five - classifier model	$f_{10}, f_1, f_8, f_4, f_2, f_5, f_9, f_7$

Fig. 14 (a) Validation–Test A1 Performance (b) Selected Feature Prioritization

4.6. Large-scale PV system

Earlier tests on a miniature experimental PV array demonstrated the suggested model's exceptional ability to identify and categorize a wide range of PV failures. To test the model's functionality on large-scale PV system, this section involves meticulous development of a 122.8-kW 10 × 30 PV utilizing Blue-chip ASP-410 M PV.

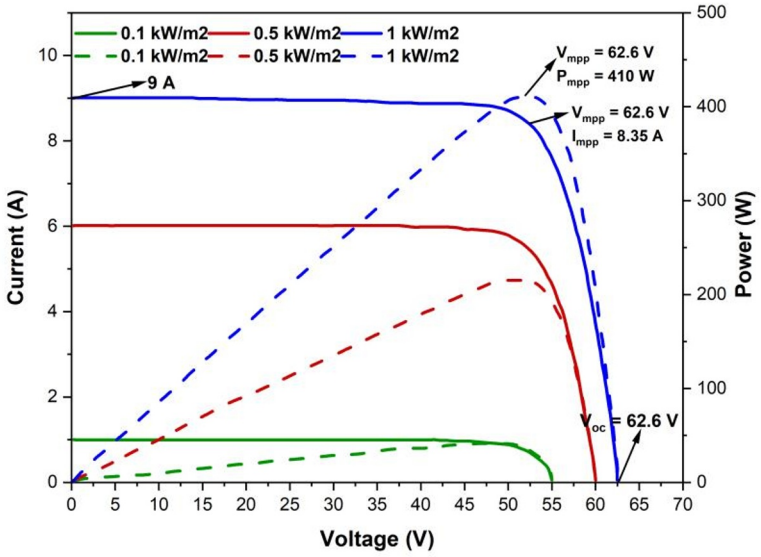


Fig. 15. ASP-405 M I–V/P–V specification

Like the second scenario, the extracted dataset includes normal conditions, array deterioration, open circuit, string degradation, and different types of line failures, among others.

Table 8. Data at third scenario

Method	Label	Training	Test
Faults	ARRDEG	1185	235
Condition	N	950	165
Open circuit faults	OC	945	165
String faults	STRDEG	1185	235
Line mismatch 20%	LL2	535	105
Line mismatch 10%	LL1	535	105
Line mismatch 20%	LL3	705	155

Consistent with the second case, NS = 6 (Refer to Eq. (2)) when NC = 7 at third dataset. Figure 16 displays the five sequences' average validation A1-scores. After each sequence, a cross is also marked to indicate the least accurate classifiers that should be deleted. Figure 16 demonstrates that GNB is removed after the first sequence with an average validation A1-score of 84.63%. Following the subsequent sequences, average validation A1-scores for RF, DT (C), MLP and kNN, are 63.67, 70.08%, 77.71%, and 83.57%, respectively. The end sequence also determines the ideal classifiers create final model. In prior demonstrated that, AI-based ensemble machine learning model is developed for photovoltaic fault detection by combining random forest, KNN, and boosting techniques. The framework enhances fault detection reliability for current-carrying conductor abnormalities in complex PV operating scenarios [14]. Since feature extraction is geometry-based rather than module-specific, the framework is transferable to different PV module ratings with minimal retraining. To evaluate scalability, the proposed framework was further validated on a 122.8 kW large-scale PV system with a 10 × 30 module configuration. Despite differences in array size, module ratings, and fault distribution, the model maintained high generalization performance with a test A1-score of

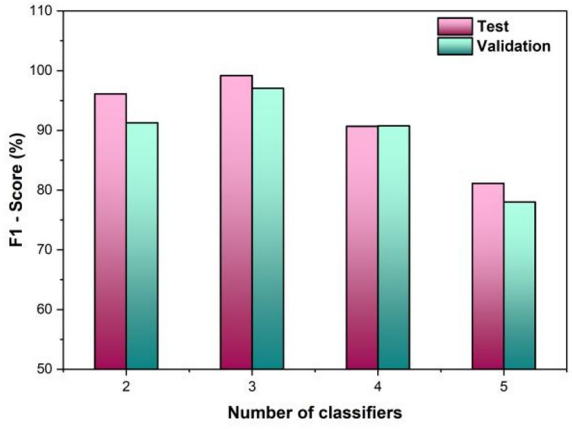
99.17%. Since feature extraction relies on geometric deformation characteristics of the I–V curve rather than system-specific electrical ratings, the framework demonstrates configuration-independent adaptability with minimal recalibration requirements.

First Sequence	Second Sequence	Third Sequence	Fourth Sequence	Fifth Sequence
SVM (RBF) 92.77%	SVM (RBF) 89.34%	SVM (RBF) 83.24%	SVM (RBF) 79.35%	SVM (RBF) 71.15%
SVM (Linear) 97.12%	SVM (Linear) 94.62%	SVM (Linear) 89.97%	SVM (Linear) 82.26%	SVM (Linear) 78.83%
SVM (Polynomial) 95.26%	SVM (Polynomial) 90.75%	SVM (Polynomial) 82.05%	SVM (Polynomial) 75.84%	SVM (Polynomial) 73.26%
LR 95.03%	LR 90.03%	LR 82.56%	LR 75.23%	LR 70.02%
kNN 90.26%	DT (ID3) 88.17%	DT (ID3) 81.09%	DT (ID3) 74.41%	DT (ID3) 68.84%
DT (ID3) 95.18%	DT (CART) 87.92%	DT (CART) 79.93%	RF 72.37%	RF 63.67%
DT (CART) 92.24%	RF 90.45%	RF 80.94%	DT (CART) 70.08%	
RF 95.06%	MLP 84.89%	MLP 77.71%		
MLP 88.18%	kNN 83.57%			
GNB 84.63%				

Fig. 16. Model efficiency at third scenario

Figure 17 shows that A1-score is 97.04%, and final were three model using SVM (Poly), SVM (RBF)and SVM (Linear). When tested on the unseen dataset, the final model achieved an A1-score of 99.17%, demonstrating its exceptional performance. Fig. 17 also includes the results of m-classifier aggregative models that used test dataset, which helps with comparison.

(a)



(b)

Combination	Selected Features (Prioritized)
Two - classifier model	$f_8, f_{10}, f_1, f_2, f_5, f_9, f_4, f_3$
Three - classifier model	$f_8, f_{10}, f_1, f_2, f_5, f_9$
Four - classifier model	$f_8, f_{10}, f_1, f_2, f_5, f_9, f_4, f_3$
Five - classifier model	$f_8, f_{10}, f_1, f_2, f_5, f_9, f_4, f_3, f_7, f_6$

Fig. 17. (a) Validation–Test A1 Performance (b) Selected Feature Prioritization

It is possible to deduce the following information and draw the following conclusions by investigating the three cases presented above: We explored all feasible combinations of classifiers in both circumstances and ultimately settled on three aggregative models with three classifiers. It follows that this is the sweet spot for algorithm density while building an aggregative model. SVM is one of capable and powerful algorithms at PV fault detection and array classification. Also, as DT classifiers are present in most aggregative models, either as an RF with a plethora of DTs or with the ID3 or CART algorithms, it is reasonable to assume that they are an effective algorithm in this respect [15]. Figure 18 shows average validation A1-scores for all classifier at second scenario of each sequence, and figure 19 shows the same thing for the third scenario. When given a dataset with just two classes to sort, all of the classifiers perform admirably in the first sequence. The classification process is increasingly challenging for the algorithms, nevertheless, because the process exhibits a declining tendency in subsequent sequences as number of classes raises.

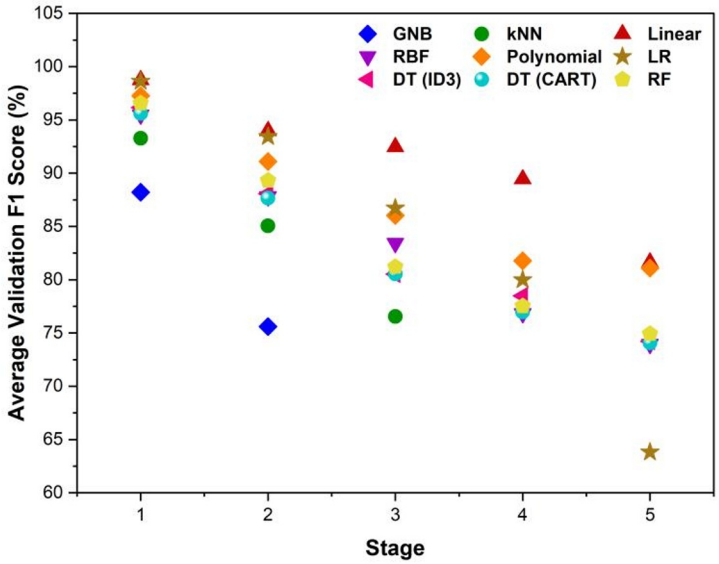


Fig. 18. Scenario 2 Classifier Evaluation

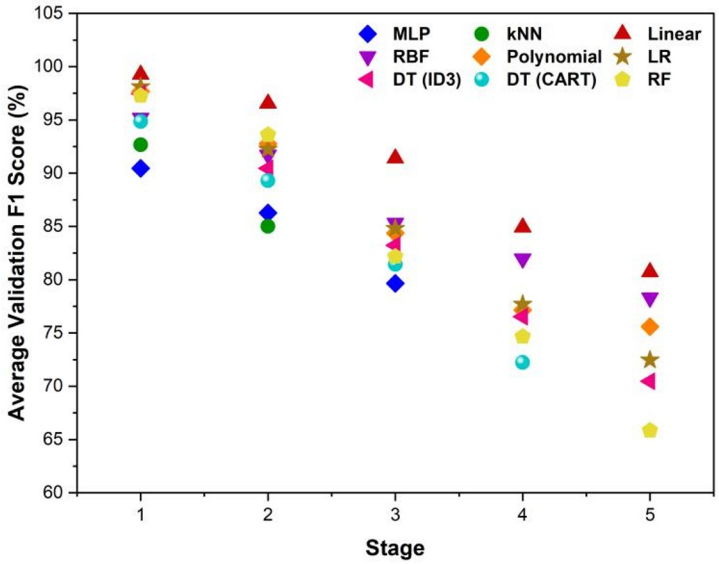


Fig. 19. Scenario 3 Classifier Evaluation

4.7. Comparison

There have been two comparisons to show the benefits of the suggested model. We began by comparing the model's output (A1-scores) in two separate scenarios to those of the individual classifiers tested here. Results from the validation and test procedures for first scenario and second scenario show that suggested model in this study performed better, as shown in Figures 20(a) and 20(b).

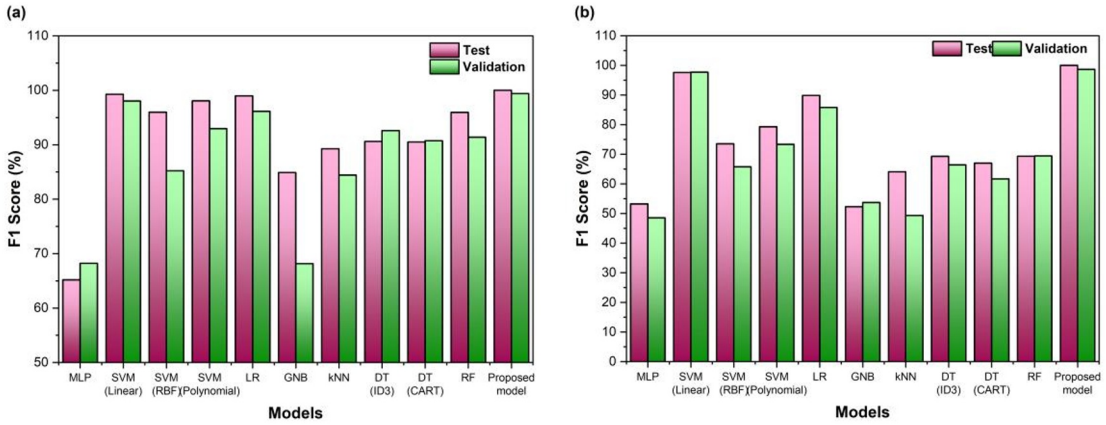


Fig. 20. (a) A1-Score Comparison: Scenario 1 (b) Scenario 2

Deep learning methods were not included in benchmarking due to their dependency on large-scale labeled datasets and high computational requirements. The proposed framework prioritizes lightweight implementation suitable for edge-level PV monitoring systems. It was found that, crucial defect detection circumstances have only been thoroughly considered by the suggested model. But the model's lack of precision is obvious. Similar to the models given, this paper's proposed model uses dataset with number of classes. The suggested model in this study, however, outperforms every one of those models in terms of accuracy. As a conclusion, the model appears to have a high level of accuracy; however, it is important to note that the model takes into account just a limited number of classes and fails to account for all of the crucial circumstances involved in defect identification. The present framework is limited to predefined steady-state fault categories and static I–V measurements. Future work will incorporate dynamic fault progression modeling, real-time streaming analytics, and hybrid physics-informed learning models to enhance adaptability.

4. Conclusion

This study rethought traditional machine learning classifiers by proposing a sequential elimination-based aggregative model for defect classification and detection at photovoltaic arrays with limited data. In order to accomplish this, we began by collecting our initial dataset from five discrete locations along the I-V curve. The next step was to extract ten characteristics by calculating the squared distances between every pair of points. By removing the least accurate classifier from the original set of machine learning classifiers, the elimination process was carried out sequentially. At last, all of the remaining classifiers came together to create an aggregative model that could identify and categorize several problems at once. To get the best possible accuracy on the new test datasets, we also looked at optimal number classifiers to use at final model. Simplifying training process was the goal of the iterative strategy that included two simplification techniques. To start, we reduced number of classes at dataset using a horizontal simplification strategy. The second step was to use vertical RF simplification method to pick best attributes and further reduce the dataset. Using three separate datasets, the approach was applied to a PV system in one large-scale MATLAB/Simulink scenario and two experimental situations. The findings demonstrated the suggested model's robustness and applicability with test accuracy rates of 100% in the first case, 99.59% in the second, and 99.17% in the third. The proposed dual-stage simplification framework provides a scalable pathway for integrating lightweight AI diagnostics into smart grid PV infrastructures, bridging the gap between computational efficiency and high-fidelity fault detection.

References:

[1] F. Alpsalaz, Y. Özüpak, E. Aslan, and H. Uzel, "Hybrid Machine Learning Approach for Enhanced Fault Detection and Power Estimation in Photovoltaic Systems," *IET Renewable Power Generation*, vol. 20, no. 1, 2026, doi: 10.1049/rpg2.70153.

[2] V. Khandeparkar and S. K. Senthil Kumar, "Effectiveness of supervised machine learning models for electrical fault detection in solar PV systems," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-18802-4.

[3] Y. Özüpak, "Real-time detection of photovoltaic module faults using a hybrid machine learning model," *Solar Energy*, vol. 302, 2025, doi: 10.1016/j.solener.2025.114014.

[4] M. V Prashanth *et al.*, "Machine Learning Approaches for Solar PV Fault Identification," *SN Comput. Sci.*, vol. 6, no. 7, 2025, doi: 10.1007/s42979-025-04364-9.

[5] M. Parvin, H. Yousefi, and B. Mohammadi-ivatloo, "Novel stacking algorithm with feature extraction for photovoltaic fault classification," *Solar Energy*, vol. 305, 2026, doi: 10.1016/j.solener.2025.114270.

[6] A. Nedaei, A. Eskandari, and M. Aghaei, "Photovoltaic fault detection and classification: Reconsideration of classic machine learning and dataset shrinkage techniques for simplification," *Results in Engineering*, vol. 27, 2025, doi: 10.1016/j.rineng.2025.106356.

[7] S. Tufail and A. I. I Sarwat, "A Comparative Study of Dimensionality Reduction Methods for Accurate and Efficient Inverter Fault Detection in Grid-Connected Solar Photovoltaic Systems," *Electronics (Switzerland)*, vol. 14, no. 14, 2025, doi: 10.3390/electronics14142916.

[8] M. Abdelsattar, A. AbdelMoety, and A. Emad-Eldeen, "Advanced machine learning techniques for predicting power generation and fault detection in solar photovoltaic systems," *Neural Comput. Appl.*, vol. 37, no. 15, pp. 8825–8844, 2025, doi: 10.1007/s00521-025-11035-6.

[9] A. Namoune, A. Chaker, and I. Saouane, "A dual-approach machine learning and deep learning framework for enhanced fault detection in photovoltaic systems: Incorporating SDM parameter analysis and thermal imaging," *Journal of Renewable and Sustainable Energy*, vol. 17, no. 3, 2025, doi: 10.1063/5.0264116.

[10] G. Nassreddine, A. Arid, M. Nassereddine, and O. Al-Khatib, "Fault Detection and Classification for Photovoltaic Panel System Using Machine Learning Techniques," *Applied AI Letters*, vol. 6, no. 2, 2025, doi: 10.1002/ail2.115.

[11] K. Ratsheola, D. Setlhaolo, A. Rasool, A. Ali, and N. E. Mabunda, "A Hybrid Artificial Intelligence for Fault Detection and Diagnosis of Photovoltaic Systems Using Autoencoders and Random Forests Classifiers," *Eng*, vol. 6, no. 10, 2025, doi: 10.3390/eng6100254.

[12] M. S. Hassan, V. J. Chin, and L. Gopal, "Accurate diagnosis of concurrent faults in photovoltaic systems using CONMI-based feature selection and Support vector

machines," *Energy Convers. Manag.*, vol. 344, 2025, doi: 10.1016/j.enconman.2025.120293.

[13] M. Parvin, H. Yousefi, and B. Mohammadi-ivatloo, "Photovoltaic fault detection algorithm using ensemble learning enhanced with deep neural network feature engineering," *Results in Engineering*, vol. 27, 2025, doi: 10.1016/j.rineng.2025.106491.

[14] M. S. Ibrahim, H. K. Almulla, A. D. Sallibi, A. A. Nafea, A. K. Kareem, and K. M. Ali Alheeti, "Enhanced fault detection in photovoltaic systems using an ensemble machine learning approach," *International Journal of Reconfigurable and Embedded Systems*, vol. 14, no. 2, pp. 507–517, 2025, doi: 10.11591/ijres.v14.i2.pp507-517.

[15] A. Freej, A. S. Sabik, and I. A. Nassar, "Performance Improvement of Photovoltaic Panels Through Advanced Fault Detection Techniques," *Processes*, vol. 13, no. 12, 2025, doi: 10.3390/pr13123831.