Interpretable Rule-based Classification for Photovoltaic Fault Detection Using Discrete Tuna Swarm Optimization

Souria Sif, Ouahiba Chouhal, Yassine Beddiaf, Abdeldjalil Ledmi, Rafik Mahdaoui, Hichem Haouassi, and L. Chrifi-Alaoui

***Abstract*—The reliability of photovoltaic systems is dependent on effective fault detection and diagnosis, which are essential to maintain performance, efficiency, and operational safety. Traditional diagnosis methods can achieve good accuracy but often lack interpretability, which limits their use in real-time monitoring and decision-making. In this study, we propose a framework called Rule-based Classification Fault Detection and Diagnosis for Photovoltaic Systems (RCFDD-PVs). The framework incorporates a rule-based classification approach in which IF–THEN decision rules are intelligently generated using Discrete Tuna Swarm Optimization. This hybrid white-box approach ensures that diagnosis decisions are not only accurate but also understandable, offering a clear alternative to black-box models. The framework was validated on a dataset including both normal and faulty PV operating states. Results show that RCFDD-PVs achieved 99% accuracy and a weighted F1-score of 0.99, using only nine rules with four antecedents to achieve complete dataset coverage. The generated rules are concise and interpretable, making them practical for integration into PV monitoring systems.**

***Index Terms*—Photovoltaic systems, Fault Detection and Diagnosis, Rule-based Classification Fault Detection and Diagnosis for Photovoltaic Systems, Discrete Tuna Swarm Optimization, Explainable Artificial Intelligence.**

I. Introduction

Photovoltaic systems (PVs) represent one of the most widely adopted renewable energy technologies. They operate without fuel consumption and produce no carbon emissions. Their flexibility has positioned them as a preferred solution for electricity generation, ranging from small residential installations to large-scale solar power plants [1]. Faults such as potential-induced degradation, microcracks, delamination, and electrical contact deterioration can significantly reduce energy yield and increase maintenance costs [2], [3]. However, to ensure sustained performance and operational reliability, these systems require robust fault detection and diagnosis (FDD) mechanisms [4], [5].

Conventional diagnostic techniques, such as electrical measurements, infrared thermography, and thermal imaging, have long been used to detect early faults in PVs. While effective in certain cases, these methods are often limited by high costs, slow response, and poor scalability. As photovoltaic (PV) deployment grows, the need for automated, accurate, and reliable diagnostic tools has become more pressing. In this context, machine learning (ML) has emerged as a powerful alternative. By analyzing PV datasets, ML models can classify operating states, distinguish between normal and faulty conditions, and achieve higher accuracy with minimal human intervention, thereby overcoming many of the shortcomings of traditional approaches [3], [6]. Despite these advantages, many ML-based diagnosis models rely on black-box methods, such as neural networks or ensemble techniques. Although these models often deliver high accuracy, they provide little insight into the reasoning behind their predictions. This lack of interpretability reduces user trust and complicates the adoption of such models in safety-critical applications like PV monitoring [7]–[13]. In addition, they require large, high-quality datasets, intensive computation, and careful parameter tuning, which limit their deployment in real-world PV monitoring [14], [15]. This is the reason why there is an increasing interest in white-box methods, which provide both accuracy and transparency. Rule-based Classification (RC) is especially suitable for PV diagnosis, as explicit IF–THEN rules can be easily understood, verified, and deployed [16], [17]. At the same time, the growing application of artificial intelligence (AI) in energy management highlights the importance of explainable AI (XAI), particularly in safety-critical domains such as PVs [18]. Researchers increasingly advocate for interpretable machine learning frameworks that allow decision-makers and operators to better understand and trust AI outputs [19].

This combination of interpretability and reliability makes rule-based white-box models a promising direction for advancing FDD in PVs. Building on this motivation, this work introduces the RCFDD-PVs framework. The method integrates RC with DTSO to generate compact, interpretable, and accurate rules from solar datasets. In this way, the framework balances

Manuscript received February 1, 2026; revised March 9, 2026. Date of publication July 13, 2026. Date of current version July 13, 2026.

S. Sif is with the Laboratory of SATIT, Department of Electrical Engineering, Faculty of Science and Technology, Abbes Laghrour University, Khenchela, Algeria (e-mail: sif.souria@univ-khenchela.dz).

O. Chouhal, A. Ledmi, R. Mahdaoui, and H. Haouassi are with the ICOSI Laboratory, Department of Computer Science, Abbes Laghrour University, Khenchela, Algeria (e-mails: {chouhal.ouahiba, abdeldjalil.ledmi, mahdaoui.rafik, haouassi.hichem}@univ-khenchela.dz).

Y. Beddiaf is with the LTIMER Laboratory, Department of Electrical Engineering, Faculty of Science and Technology, Abbes Laghrour University, Khenchela, Algeria (e-mail: beddiaf.yassine@univ-khenchela.dz).

L. Chrifi-Alaoui is with the Laboratoire des Technologies Innovantes, University of Picardie Jules Verne, France (e-mail: larbi.alaoui@u-picardie.fr).

Digital Object Identifier (DOI): 10.24138/jcomss-2026-0009

diagnostic precision with transparency, making it both effective and practical for PV fault detection and monitoring.

The main contributions of this paper are summarized as follows:

- Development of a white-box framework for fault detection and diagnosis in PVs.
- Integration of rule-based classification with Discrete Tuna Swarm Optimization to generate compact and interpretable IF–THEN rules.
- Achievement of high diagnostic performance with a small rule base, thereby preserving both accuracy and transparency.
- Provision of a practical and deployable diagnostic approach for PV monitoring and operational decision support.

The remainder of this paper is organized as follows. Section II reviews related work. Section III summarizes the background concepts. Section IV presents the proposed approach. Section V defines the evaluation metrics. Section VI reports the experimental results. Section VII discusses the comparative analysis and computational complexity. Finally, Section VIII concludes the paper.

II. Related Work

A. Traditional Statistical Approaches for Photovoltaic Fault Detection and Diagnosis

Early methods of Photovoltaic Fault Detection and Diagnosis (PV-FDD) are founded on classical statistical methods. These methods consider the electrical output of PVs as a time-varying process and detect faults by observing variations in behavior. This has been particularly significant in time-series models. Predicting the power output of PVs and detecting unnatural variations in the forecasted profile are common efforts, typically done using approaches like ARIMA, ARMA, and ARMAX. Relative analysis reveals that forecasting with ARIMA can perform better than simple models by more than 50% and is a useful method for detecting early anomalies [20]–[22]. Linear and logistic regression-based methods have also been used to model statistical correlations between PV variables, as well as to identify normal and faulty operational states. Moreover, descriptive statistical analysis—based on measures such as mean, median, and standard deviation, or tools like boxplots and histograms—helps identify anomalies by summarizing data patterns [23].

However, despite their usefulness, these traditional methods have significant limitations. They are typically most successful in linear or weakly nonlinear systems, but real PV systems tend to exhibit strongly nonlinear behavior, particularly when environmental conditions change. Moreover, they normally require large amounts of manual modeling and have little robustness in large-scale or highly variable PV plants.

B. Machine Learning for PV-FDD

PV-FDD has benefited greatly from ML, which allows models to learn from historical and real-time data. The common use of ML techniques nowadays predicts PV output, system performance, and fault detection prior to significant power losses [24].

1) Classical ML Algorithms: Support Vector Machines (SVM), Random Forests (RF), and ensemble-based models represent some of the most effective approaches to PV diagnosis in various configurations [21], [24]. These methods assist in minimizing bias and variance by merging several predictors, enhancing reliability. Classical ML models are particularly useful in identifying early abnormalities in system functioning, which can result in substantial savings in maintenance costs and system downtime [24].

2) Deep Learning Approaches: Deep learning (DL) builds on the benefits of classical ML by implementing multi-layer neural networks that automatically extract high-level features from raw inputs, such as electrical measurements and images [25]. These approaches are particularly effective with complex, high-dimensional, and nonlinear PV datasets.

C. Black-box Models for PV-FDD

Another significant paradigm in PV-FDD is black-box machine learning models. Methods such as artificial neural networks (ANNs) [26], DL models like convolutional neural networks (CNNs) [25] and long short-term memory networks (LSTMs) [27], in addition to SVM and ensemble algorithms like gradient boosting, are appreciated because they can directly learn complicated, nonlinear relationships between data. Unlike conventional methods, they are not detailed physical models of PVs. They automatically uncover diagnostic patterns from extensive datasets and are very effective in detecting faults in contemporary PVs [28].

These models, though highly predictive, have a serious drawback: a lack of interpretability. As black boxes, system operators find it hard to understand the logic behind predictions. This limitation may impede their application in real-world PV diagnosis, where both accuracy and transparency are crucial.

D. White-Box Models for PV-FDD

White-box methods allow verification of the reasoning behind every diagnostic decision, which is vital for developing trust and reliability in safety-critical PV applications.

Some of the most popular white-box techniques include Decision Trees (DTs), which run as structured flowcharts where nodes test PV parameters and edges represent branches. Their transparent structure allows users to follow the logic precisely to arrive at a classification. In PV fault detection, DTs have demonstrated good performance, with accuracy reported up to 98%. Another common white-box model is the K-Nearest Neighbors (KNN) classifier. This method compares a new instance to the most similar cases in the training data. KNN involves no complex training phase, is easy to use, and can be easily scaled to other datasets. Studies report PV fault detection accuracy levels of up to 99% [29].

Fuzzy logic methods, beyond these classical interpretable models, have been central to PV-FDD because of their transparent, rule-based reasoning. Fuzzy Logic Systems (FLS) are prevalent in PV control and fault diagnosis, particularly in Maximum Power Point Tracking (MPPT), where adaptive fuzzy controllers have been shown to perform better than traditional Proportional-Integral (PI) controllers by converging faster, achieving higher tracking precision, and reducing oscillations. Reports indicate that adaptive fuzzy control can enhance PV output power by approximately 20% compared to traditional fuzzy controllers [30].

In diagnostic tasks, Type-1 FLS rely on human-interpretable IF–THEN rules to identify operating conditions based on variables including Direct Current (DC) voltage, current and irradiance [31]. Type-2 FLS extend these systems by adding an additional level of uncertainty modeling to better adapt to changing environmental conditions. Studies report that Type-2 FLS can match or outperform ANNs in fault detection in some scenarios [32], [33], and hybrid Type-2 schemes combined with optimization algorithms have also been explored [33].

Although white-box and fuzzy models are interpretable and have good diagnostic characteristics, they face scaling challenges in large-scale or real-time PV monitoring. They do not always provide efficient computation for high-volume or high-dimensional data streams, limiting their applicability in very large PV plants.

This demonstrates a challenge inherent in existing PV-FDD methods: traditional statistical and fuzzy methods are difficult to scale to complex systems, black-box ML models are difficult to interpret, and current white-box methods tend to lose accuracy under dynamically changing operating conditions. Our new hybrid framework, termed RCFDD-PVs, overcomes these limitations by integrating RC with DTSO. This combined approach generates compact, human-understandable IF–THEN rules while improving diagnostic accuracy, providing a scalable, interpretable, and high-performing solution for PV-FDD.

III. Background

A. Rule-Based Classification

A rule-based classifier consists of a set of IF–THEN rules that have been generated algorithmically from the training dataset. Each rule consists of two primary components: the antecedent and the consequent. The antecedent determines the conditions in which the rule is applicable and is composed of one or more terms. Each term has a variable, a relational operator, and an associated value. When multiple terms are found in the antecedent, they are typically combined using the logical operator AND. The consequent specifies the class label assigned when all the antecedent conditions are satisfied. These generic classification rules have the following key elements, as depicted in Figure 1. The rules are designed by system operators or generated by analysis of the training dataset, with each rule consisting of one or more antecedent conditions that result in a particular classification outcome [34].

B. Human-Readable Rules

The use of decision rules in practical situations must be comprehensible to expert operators. One refinement is to re-establish the semantic clarity of the rules, making them easier for experts to examine and use for decision-making [35]. However, in high-dimensional feature spaces and big data settings, generating rules may be complex and may lead to overfitting and duplication. To address this, rule sets are simplified using simplification techniques. The goals are to maintain interpretability and predictive performance and to eliminate redundant or unnecessary elements [36].

C. Entropy-Based Discretization

Entropy-Based Discretization (EBD) is a data preprocessing method that converts continuous data into discrete values while preserving information important for solving classification problems. It applies Shannon entropy to measure the homogeneity of classes following a candidate split point T.

The conditional entropy of a dataset S resulting from partitioning based on feature F at level T is:

$$E(F,T;S) = \frac{|S_1|}{|S|} \cdot Ent(S_1) + \frac{|S_2|}{|S|} \cdot Ent(S_2) \tag{1}$$

where S_1 and S_2 are subsets generated by splitting the dataset.

The entropy of each subset S_i is:

$$Ent(S_i) = -\sum_{j=1}^{Z} p(C_j, S_i) \cdot \log_2 \left(p(C_j, S_i)\right) \tag{2}$$

where $p(C_j, S_i)$ is the proportion of samples in class C_j, and Z is the total number of classes.

Information gain is defined as:

$$Gain(T) = Ent(S) - E(F,T;S) \tag{3}$$

This recursive process continues until the gain is statistically significant, resulting in optimized discretization that improves accuracy, interpretability, and the performance of supervised models [37]. Figure 2 shows the EBD process flowchart, in which the dataset is divided according to entropy minimization.

D. Discrete Tuna Swarm Optimization

1) Fundamentals of Tuna Swarm Optimization: Tuna Swarm Optimization (TSO) is a nature-inspired metaheuristic algorithm modeled after the cooperative hunting behaviors of tuna species (genus Thunnini). Tuna exhibit spiral and parabolic hunting formations. Each individual in the population represents a candidate solution, initialized as:

$$X_i^{int} = lb + rand \cdot (ub - lb), \quad i = 1, 2, 3, \ldots, NP \tag{4}$$

where X_i^{int} denotes the initial position of the i^{th} individual, (ub, lb) are the upper and lower bounds, $rand \in [0, 1]$, and NP is the population size.

2) Foraging Phases in TSO: Spiral Foraging: Spiral formation is modeled by:

$$X_i^{t+1} = \begin{cases} \alpha_1 \cdot (X_{\text{rand}}^t + \beta \cdot |X_{\text{rand}}^t - X_i^t|) \\ \quad +\alpha_2 \cdot X_i^t, \qquad\qquad i = 1 \\ \\ \alpha_1 \cdot (X_{\text{rand}}^t + \beta \cdot |X_{\text{rand}}^t - X_i^t|) \\ \quad +\alpha_2 \cdot X_{i-1}^t, \, i = 2, \ldots, NP \end{cases} \tag{5}$$

if $rand < \frac{t}{t_{\max}}$.

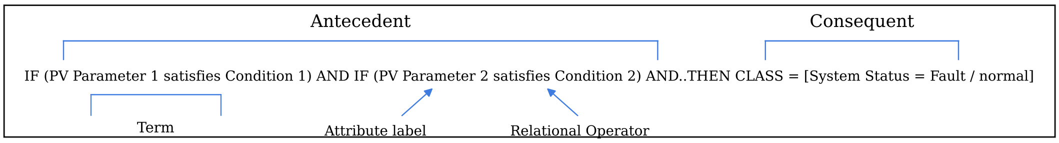

Fig. 1. The structure of a generic classification rule.

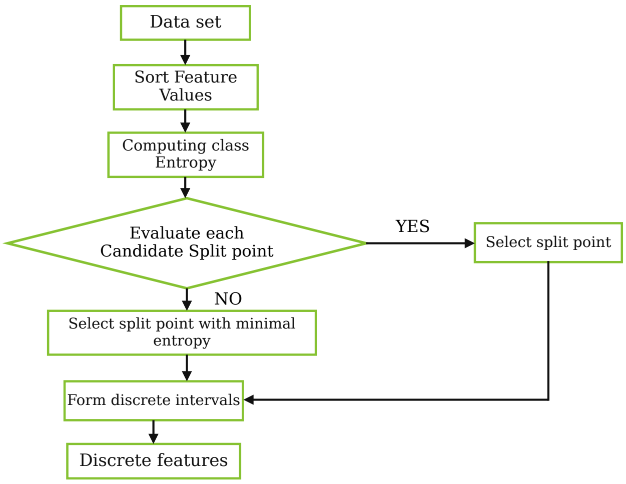

Fig. 2. The flowchart of the entropy-based discretization process.

Also,

$$X_i^{t+1} = \begin{cases} \alpha_1 \cdot (X_{best}^t + \beta \cdot |X_{best}^t - X_i^t|) & i = 1 \\ + \alpha_2 \cdot X_i^t, & \\ \alpha_1 \cdot (X_{best}^t + \beta \cdot |X_{best}^t - X_i^t|) & i = 2, \ldots, NP \\ + \alpha_2 \cdot X_{i-1}^t, & \end{cases} \tag{6}$$

if $rand \geq \frac{t}{t_{\max}}$. where:

- X_{best}^t: global best position.
- X_{rand}^t: random reference position.
- $\alpha_1 = a + (1 - a) \cdot \frac{t}{t_{\max}}$ controls attraction.
- $\alpha_2 = (1 - a) - (1 - a) \cdot \frac{t}{t_{\max}}$ controls inertia.
- $\beta = e^{bl} \cdot \cos(2\pi b)$ introduces oscillation.
- $l = e^{3\cos(((t_{\max} + \frac{1}{t}) - 1)\pi)}$ modulates oscillation magnitude.
- $b \in [0, 1]$ is random, and a is a tuning parameter.

3) Parabolic Foraging:

$$X_i^{t+1} = \begin{cases} X_{best}^t + rand \cdot (X_{best}^t - X_i^t) & \\ \qquad + (TF \cdot p^2) \cdot (X_{best}^t - X_i^t), & rand < 0.5, \\ TF \cdot p^2 \cdot X_i^t, & rand \geq 0.5. \end{cases} \tag{7}$$

where $TF \in \{1, -1\}$ and $p = 1 - \frac{t}{t_{\max}}$.

4) Discrete Tuna Swarm Optimization: Discrete Tuna Swarm Optimization (DTSO) adapts TSO for discrete optimization by introducing discrete vector-based operators:

- Swapping: randomly swap a percentage of elements within a solution vector.
- Fusion: stochastically mix elements between two solution vectors.
- Crossover: create a new vector by combining parts of two parents [38].

E. Sequential Covering Strategy

The Sequential Covering Strategy (SC) iteratively learns one rule at a time. At each iteration, a rule is induced to cover a subset of the training data, which is then added to the rule base; covered examples are removed, and the process repeats [39].

IV. Proposed Approach

A. Research Problem and Objective

The main objective of this work is to develop an accurate and interpretable model for fault detection and diagnosis in PV systems. Although many existing methods can detect faults effectively, most of the strongest performers operate as black boxes and provide limited insight into how diagnostic decisions are made. This lack of transparency makes it difficult for engineers to trust, validate, and deploy such models in real PV plants. The proposed approach, called RCFDD-PVs, addresses this limitation by combining entropy-based discretization, Sequential Covering (SC) and the DTSO algorithm to generate compact IF–THEN rules that describe system behavior clearly and accurately.

B. Dataset

The dataset used in this study was obtained from Kaggle and is based on simulations of a 250 kW PV power plant. It contains both normal operating conditions and three fault categories, namely string faults, string-to-ground faults and string-to-string faults. In total, the dataset includes 600 samples described by 30 features derived from electrical and environmental measurements, including DC current, voltage, power, temperature and solar irradiance, together with statistical descriptors such as the means, maxima, minima and variances of string currents. Additional features (Range 1 to Range 4) were computed from differences in current values measured by ammeters. The most informative variables identified by the feature-ranking analysis were Range 3, Range 4, Range 2 and Range 1. Temperature and irradiance ranged between 10–35°C and 100–1000 W/m^2, fault resistances ranged from 1Ω to 2000Ω and the simulated faults were introduced at 0.2 s during a 0.4 s simulation window [40]. Table I summarizes the class-wise distribution used for photovoltaic fault detection.

TABLE I
CLASS-WISE DATA DISTRIBUTION FOR PV FAULT DETECTION

Description	Record percentage	Count	Class
Fault-free system	16.97%	100	0
String fault	25.5%	153	1
String-to-ground fault	24.83%	149	2
String-to-string fault	33%	198	3

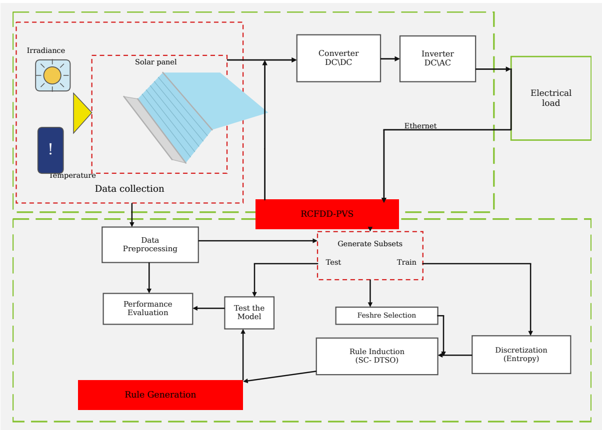

Fig. 3. Global Structure of the Proposed RCFDD-PVs Framework.

C. Architecture of the RCFDD-PVs Framework

The proposed RCFDD-PVs framework is an efficient and explainable ML-based diagnosis model that detects and classifies faults in PVs. The designed architecture of RCFDD-PVs system is depicted in Figure 3. The process is composed of several phases: data collection, preprocessing, feature selection, rule induction (SC-DTSO) and evaluation. Each component plays a distinct role in enhancing diagnosis accuracy and interpretability. The method is initialized by data collection from simulated or real PVs during the presence of different types of faults and normal states. Parameters such as voltage, current, irradiance, and temperature are recorded to obtain a full coverage of potential faults. Once the data has been assembled, the preprocessing phase applies all necessary transformations to make the data suitable for analysis. The selected features are then used in the rule generation stage, where a decision-tree-based learning algorithm extracts interpretable classification rules from the training data. Then, the rules are refined and accuracy is improved using rule induction (SC-DTSO). The developed model is tested and evaluated against unseen data in terms of accuracy and weighted F1-score. The output of the RCFDD-PVs is an interpretable set of IF–THEN rules for the diagnosis that describes how a set of PVs parameters (faulty and normal) relates to a fault class. In other words, which PV parameters are relevant and represent the fault class under consideration. These rules indicate an explainable and clear representation of the decision process, facilitating real-time diagnosis applications. The following section provides a complete explanation of the internal structure of the RCFDD-PVs framework and how it generates rules.

D. Detailed Process of the RCFDD-PVs Framework

The RCFDD-PVs framework produces a set of human-readable rules that describe operational conditions of the PVs. The process consists of three main stages, ranging from data preprocessing to rule generation.

1) Data Preprocessing and Feature Selection: The original dataset was imported and analyzed using Python libraries (Pandas, NumPy, Scikit-learn). Incomplete and/or duplicate data points were excluded. Remaining data were transformed into numerical format and extraneous columns deleted. Since the approach is rule-based, an EBD algorithm transformed continuous variables into categorical intervals. Each variable was discretized into three ranges determined by threshold values that minimized class entropy. Feature selection was then performed based on importance scores derived from a decision tree algorithm. Finally, the dataset was split into 80% training and 20% testing subsets using train-test-split in Scikit-learn.

2) Rule Induction : The procedure of Rule Induction (SC-DTSO) represents the most important part of the RCFDD-PVs process. In particular, Steps 3 and 4 describe the rule induction process in detail, explaining how classification rules are progressively generated and optimized.The Sequential covering strategy algorithm serves as the main process guiding the overall rule-learning procedure, working at a higher level to discover a complete set of classification rules for each fault type. Furthermore, the DTSO algorithm is employed as the principal optimization tool, working inside the SC loop to optimize individual rules. In other words, SC defines when and where to create a new rule, while DTSO decides how to build the best possible rule for the current class. This interaction allows the RCFDD-PVs model to progressively construct a compact and highly accurate rule base for PV fault diagnosis.

3) Sequential covering strategy : In this stage, the system employs the SC for each class to iteratively learn classification rules. Key steps:

1) Initialization: The rule base RS is initialized as empty. Hyper parameters such as population size N and maximum iteration count $t_{\max}$ are defined.
2) class-dependent iteration: For each fault class C_i, the algorithm isolates the subset of examples D_{C_i}.
3) Rule generation: A candidate rule is generated using the DTSO optimizer (`DTSO_LearnRule`), which searches for the rule that best describes D_{C_i} with high accuracy and coverage.
4) Rule evaluation and storage: Once validated, the rule is added to the rule set:
$$RS \leftarrow RS \cup \{rule\}.$$
5) Dataset reduction: The data samples covered by the induced rule are removed from the class subset:
$$D_{C_i} \leftarrow D_{C_i} \setminus \{x \in D_{C_i} \mid rule(x) = \text{true}\}.$$
The process repeats with the remaining subset until fewer than 90% of instances are uncovered.
6) Completion: After all classes have been processed, the final rule base (RS) includes the complete set of optimized rules for the fault diagnosis model. The SC

Algorithm 1: Sequential Covering Rule Induction
Input: Dataset D with classes $\{C_1, C_2, \ldots, C_K\}$
Output: Final rule set RS

1. Initialization
 $RS \leftarrow \emptyset$ $\leftarrow$ empty rule set
 Define max iterationsT_{max}, population size N
2. For each class $C_i \in \{c_1, c_2, \ldots\}$:
 Extract subset: D_{C_i} = all instances of class c_i
while D_{C_i} not empty & coverage ($\geq$ 90% remaining) do
 3. Learn one rule using DTSO:
 $rule \leftarrow$ DTSO Learn Rule(Dc_i)
 4. Add this rule to rule set:
 $RS \leftarrow RS \cup \{rule\}$
 5. Remove all instances covered by the new rule from S_c
 Continue until class subset nearly empty.
end while
6. Return the final rule set
 return RS
 $RS \subseteq R_{total}$

ensures that each rule makes a unique contribution to the overall classification, avoiding redundancy and maintaining interpretability. The SC strategy is summarized in Algorithm 1.

4) Integration of DTSO within the SC strategy: At the heart of SC, the Discrete Tuna Swarm Optimization algorithm acts as the optimizer responsible for generating high-quality rules. Each time DTSO is invoked, it fine-tunes a candidate rule using swarm-based search strategies, ensuring the rule covers the data effectively. By integrating DTSO within SC, the RCFDD-PVs model combines broad coverage learning with local rule refinement, producing a compact and easy-to-understand diagnosis rule set. When the SC loop calls: $rule \leftarrow DTSOLearnRule$, the DTSO procedure begins a population-based search among potential rule structures (encoded as discrete feature–value vectors). Each candidate represents a possible IF–THEN rule of the form V_i:

$$V_i = \text{IF}\left\{\left[A_1 = a_1\right], \left[A_2 = a_1\right], \ldots, \left[A_j = a_j\right],\right. \\ \left. \text{Then Class } = c \right. \tag{8}$$

where A_j represents a feature, a_j is a discretized value (via entropy-based binning), and c is the target class.

a) Discrete Operators: The DTSO employs the following operators:

- Spiral Movement ($\circledast$): *Random Swap*. Simulates exploration through random permutations. A portion ($\rho \times$ length) of the rule vector is shuffled randomly during this process.
$$V_i^{t+1} = V_i^t \circledast \lambda \tag{9}$$
- Parabolic Mixing ($\ominus$): *Bit-level Fusion*. Emulates tuna's parabolic attack by combining segments of two rules. The operator selects bits from both parents based on a stochastic mask.
$$V_i^{t+1} = V_i^t \ominus V_j^t \tag{10}$$
- Circular Exchange ($\oplus$): *Structured Crossover*. Models circular cooperation behavior. A one-point or uniform crossover swaps rule segments between individuals.
$$V_i^{t+1} = V_i^t \oplus V_j^t \tag{11}$$

The goal of DTSO is to optimize these candidate rules with respect to rule coverage:

$$F\left(V_i\right) = \text{ Coverage } (Xi) \tag{12}$$

This fitness function ensures that rules are general. The detailed procedure is outlined in Algorithm 2.

b) Within the DTSO core: Step 1: Initialization: Initialize a population of tuna individuals, where each represents a candidate rule set encoded as a discrete position vector. In addition, set key control parameters, including the exploration–exploitation balance coefficient, switching threshold and maximum iteration limit.

Step 2: Rule Vector Generation: Each individual's position vector is translated into a rule set by assigning discrete feature value pairs that define the classification conditions. This encoding ensures that only valid feature value combinations are used, following the representation scheme shown in Equation (4).

Step 3: Fitness Evaluation: The fitness of each individual is evaluated using a composite objective function. This function combines classification performance with rule compactness and interpretability. The fitness function is often structured, as shown in Equation (12). An adaptive time-varying component, such as

$$l = e^{3\cos(((t_{\max}+1/t)-1)\pi)} \tag{13}$$

Step 4: Iterative Optimization Loop: The following sub-steps are repeated until the stopping criterion is satisfied:

- 4.1: The fitness of each individual is recomputed.
- 4.2: Identify the current global best rule set X_{best} in the population.
- 4.3: For each individual:
  - 1. Update exploration–exploitation control parameters, α_1, α_2 and probability p.
  - 2. If $rand < z$, random exploration is applied using Equation (4).
  - 3. Else, perform either:rand $\geq z$
    - $*$ Exploratory update using operator $\otimes$ when rand $< t_{max}$ governed by:

$$X_i^{t+1} = \begin{cases} \alpha_1 \circledast \left(X_{\text{rand}}^t \oplus \beta \circledast \right. \\ \quad \left. \left|X_{\text{rand}}^t \ominus X_i^t\right|\right) + \alpha_2 X_i^t, \\ \qquad\qquad i = 1, \\ \alpha_1 \circledast \left(X_{\text{rand}}^t \oplus \beta \circledast \right. \\ \quad \left. \left|X_{\text{rand}}^t \ominus X_i^t\right|\right) + \alpha_2 X_{i-1}^t, \\ \qquad\qquad i = 2, \ldots, NP. \end{cases} \tag{14}$$

    - $*$ Exploitation update (leader-following) if ${}^t/t_{\max} \geq$

$rand$, guided by:

$$X_i^{t+1} = \begin{cases} \alpha_1 \circledast \left(X_{\text{best}}^t \oplus \beta \circledast \left| X_{\text{best}}^t \ominus X_i^t \right| \right) \oplus \alpha_2 X_i^t, & i = 1, \\ \alpha_1 \circledast \left(X_{\text{best}}^t \oplus \beta \circledast \left| X_{\text{best}}^t \ominus X_i^t \right| \right) \oplus \alpha_2 X_{i-1}^t, & i = 2, \ldots, NP. \end{cases} \tag{15}$$

Additionally, if $rand \geq 0.5$, apply a convergence-driven update:

$$X_i^{t+1} = X_{\text{best}}^t \oplus \text{rand} \circledast \left(X_{\text{best}}^t \ominus X_i^t \right) \oplus \left(TF \cdot p^2 \right) \circledast \left(X_{\text{best}}^t \ominus X_i^t \right). \tag{16}$$

Step 5: Rule Reconstruction: Based on the updated positions, the rule vectors for each tuna fish are reconstructed. This ensures that the new feature–value combinations represent valid and meaningful rules.

Step 6: Rule Pruning: Rule pruning strategies are applied to each updated rule set to remove redundant, irrelevant, or low-information conditions. This enhances the simplicity and generalizability of rules.

Step 7: Re-Evaluation: The fitness of the pruned and updated rule vectors is recomputed using the same composite objective function as in Step 3.

Step 8: Memory Update: For each individual, if the new fitness is better than the previous best, update its personal best position. In addition, the global best rule set is updated if a superior solution is obtained.

Step 9: Iteration Update: Increment the iteration counter $t \leftarrow t + 1$ and repeat the optimization process until $t = t_{\max}$.

Step 10: Final Rule Set Extraction: Once the maximum number of iterations is reached, return the global best rule vector as the final output.

Finally, SC defines when and where to create a new rule, while DTSO decides how to build the best possible rule for the current class. This interaction allows the model to progressively construct a compact and highly accurate rule base for photovoltaic fault diagnosis.

E. Rule Generation Output

The final output of the RCFDD-PVs framework is a comprehensive rule-based classifier composed of interpretable IF–THEN rules. Each rule captures the relationship between selected PV parameters and a fault condition. For example: Rule:

$$IF range4 \in [0.0311, +\infty[\ \wedge \ range3 \in]-\infty, -0.0311] \ \wedge \ range2 \in]0.0133, 0.1542[\ \wedge \ I4 \in]-\infty, 5.4795]$$
$$THEN \text{ Class} = 3.$$

V. Evaluation Metrics

To evaluate performance, accuracy and weighted F1-score were used. Accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

Algorithm 2: Discrete Tuna Swarm Optimization

1. Initialize the population of candidate solutions (rule sets) and set free parameters a and z.
2. Generate an initial rule vector for each candidate by assigning feature values (as per Eq. 4).
3. Evaluate each rule vector using the fitness function (Eq. 13).
4. Repeat the following steps until the maximum number of iterations t_{max} is reached:
   4.1. Recalculate fitness scores for the current population.
   4.2. Identify and update the best-performing solution X_{best}.
   4.3. For each individual (tuna) in the population:
      1. Update the control parameters α_1, $\alpha_2$and p.
      2. If a rand number $< z$:
         Apply random exploration (based on Eq. 4).
      3. Otherwise, if rand $\geq z$:
         If another rand < 0.5:
            If rand $< t_{max}$, perform an exploratory update ($\circledast$, Eq. 14).
            Else, apply a leader-based update ($\ominus$, Eq. 15).
         Else (rand ≥ 0.5):
            Execute an exploitation-based update ($\oplus$, Eq. 16).
5. Update rule vectors based on the adjusted rule sets.
6. Simplify the rules by removing redundant or low-information conditions.
7. Recalculate the fitness of each new rule vector.
8. Retain the improved solutions by updating X_{best} when necessary.
9. Increment the iteration count.
10. Output the optimal rule vector X_{best} and its corresponding fitness value.

---

where TP is true positives, TN true negatives, FP false positives, FN false negatives. F1-score:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{18}$$

Weighted F1-score:

$$Weighted\ F1 = \sum_{i=1}^{C} w_i \cdot F1_i, \quad w_i = \frac{n_i}{N} \tag{19}$$

where $F1_i$ is the F1-score of class i, n_i is number of samples in class i, N is total samples and C is number of classes. A confusion matrix was examined for class dependent diagnosis performance.

VI. Results

A. Discretization and Feature Rankings

In order to make continuous variables easier to handle, the data was discretized into clusters, or levels, and simplified into three intervals. This step helped simplify the data, making the generated rules more interpretable. The Table II contains the thresholds that were applied for each feature. Figure 4 displays the results of the feature ranking analysis. With the study ranking each feature's influence according to their scores, range 4, range 3, range 2 and I4 were among the most influential. Their high frequency and repetition across the rules demonstrate their importance in differentiating among fault classes.

B. Rule Generation and Coverage Analysis

Using the RCFDD-PVs model, we extracted nine classification rules that are easy to interpret and cover all four fault classes. Together, these cover the entire dataset. As shown in Table III, the rules form a compact and easily interpretable structure while maintaining strong predictive performance.

TABLE II
FEATURE DISCRETIZATION INTERVALS (AS PROVIDED)

Feature	interval 1 $]-\infty$, Threshold 1]	interval 2]Threshold 1, Threshold 2[	interval 3 [Threshold 2, ∞[
I1	$\leq 0.5493]$	$]0.5493, 5.3306[$	$[5.3306 \geq$
I2	$\leq 0.6999]$	$]0.6999, 4.7657[$	$[4.7657 \geq$
I1MAX	$\leq 2.6962]$	$]2.6962, 2.7850[$	$[2.7850 \geq$
I1MIN	$\leq 0.5944]$	$]0.5944, 5.2983[$	$[5.2983 \geq$
I1VAR	$\leq 2.21e-06]$	$]2.21e-06, 1.57e-05[$	$[1.57e-05 \geq$
I2MAX	$\leq 2.7900]$	$]2.7900, 2.8112[$	$[2.8112 \geq$
I2MIN	$\leq 0.7198]$	$]0.7198, 4.7248[$	$[4.7248 \geq$
I2VAR	$\leq 3.10e-06]$	$]3.10e-06, 1.81e-04[$	$[1.81e-04 \geq$
I3	$\leq 1.2763]$	$]1.2763, 5.6916[$	$[5.6916 \geq$
I4	$\leq 5.4795]$	$]5.4795, 5.5427[$	$[5.5427 \geq$
I3max	$\leq 1.2954]$	$]1.2954, 5.7060[$	$[5.7060 \geq$
I3min	$\leq 2.7066]$	$]2.7066, 2.7434[$	$[2.7434 \geq$
I3var	$\leq 0.0002853]$	$]0.0002853, 0.0003940[$	$[0.0003940 \geq$
I4MAX	$\leq 2.6981]$	$]2.6981, 2.7867[$	$[2.7867 \geq$
I4MIN	$\leq 5.4701]$	$]5.4701, 5.5216[$	$[5.5216 \geq$
I5	$\leq 2.6853]$	$]2.6853, 2.7460[$	$[2.7460 \geq$
I6	$\leq 2.6853]$	$]2.6853, 2.7460[$	$[2.7460 \geq$
Itotal1	$\leq 426.9594]$	$]426.9594, 426.9695[$	$[426.9695 \geq$
Itotalmax1	$\leq 456.2216]$	$]456.2216, 456.4591[$	$[456.4591 \geq$
Itotalmin1	$\leq 394.6606]$	$]394.6606, 394.6790[$	$[394.6790 \geq$
Vdcmean1	$\leq 518.0960]$	$]518.0960, 518.1960[$	$[518.1960 \geq$
Vdcmax1	$\leq 507.7869]$	$]507.7869, 517.1067[$	$[517.1067 \geq$
Vdcmin1	$\leq 514.3181]$	$]514.3181, 517.6640[$	$[517.6640 \geq$
Pdcmean1	$\leq 228.6653]$	$]228.6653, 229.6100[$	$[229.6100 \geq$
IR	$\leq 471.0]$	$]471.0, 484.0[$	$[484.0 \geq$
T	$\leq 13.5]$	$]13.5, 34.5[$	$[34.5 \geq$
range 1	$\leq 0.0503]$	$]0.0503, 0.1358[$	$[0.1358 \geq$
range 2	$\leq 0.0133]$	$]0.0133, 0.1542[$	$[0.1542 \geq$
range 3	$\leq -0.0311]$	$]-0.0311, 1.14e-15[$	$[1.14e-15 \geq$
range 4	$\leq 6.66e-16]$	$]6.66e-16, 0.03105[$	$[0.03105 \geq$

TABLE III
CLASSIFICATION RULES OF DISCRETIZED CONTINUOUS FEATURE VALUES AND PREDICTION OF TARGET CLASS

Rule No.	IF (Discretized intervals)	IF (Continuous intervals)	THEN Class
1	range4=2, range3=0, range2=1, I4=0	IF range4 $\in [0.0311, +\infty[$ AND range3 $\in]-\infty, 0.0311]$ AND range2 $\in]0.0133, 0.1542[$ AND I4 $\in]-\infty, 5.4795]$	3
2	range4=1, range3=2, range2=2, I4=0	IF range4 $\in]0.0000, 0.0311[$ AND range3 $\in [0.0000, +\infty[$ AND range2 $\in [0.1542, +\infty[$ AND I4 $\in]-\infty, 5.4795]$	1
3	range4=0, range3=0, range2=1, I4=0	IF range4 $\in]-\infty, 0.0000]$ AND range3 $\in]-\infty, 0.0311]$ AND range2 $\in]0.0133, 0.1542[$ AND I4 $\in]-\infty, 5.4795]$	2
4	range4=0, range3=1, I1=1, I4=0	IF range4 $\in]-\infty, 0.0000]$ AND range3 $\in]0.0311, 0.0000[$ AND I1 $\in]0.5493, 5.3306[$ AND I4 $\in]-\infty, 5.4795]$	0
5	range2=2, I1=1, I2MIN=1, I3=1	IF range2 $\in [0.1542, +\infty[$ AND I1 $\in]0.5493, 5.3306[$ AND I2MIN $\in]0.7198, 4.7248[$ AND I3 $\in]1.2763, 5.6916[$	1
6	range4=0, range3=0, I1MAX=2, I3=1	IF range4 $\in]-\infty, 0.0000]$ AND range3 $\in]-\infty, 0.0311]$ AND I1MAX $\in [2.7850, +\infty[$ AND I3 $\in]1.2763, 5.6916[$	2
7	range4=0, range2=1, I1=2, I3=1	IF range4 $\in]-\infty, 0.0000]$ AND range2 $\in]0.0133, 0.1542[$ AND I1 $\in [5.3306, +\infty[$ AND I3 $\in]1.2763, 5.6916[$	0
8	range3=0, I1MAX=0, I2MIN=1, I2VAR=1	IF range3 $\in]-\infty, 0.0311]$ AND I1MAX $\in]-\infty, 2.6962]$ AND I2MIN $\in]0.7198, 4.7248[$ AND I2VAR $\in]0.0000, 0.0002[$	3
9	range4=1, range2=2, I1MAX=2, I3=1	IF range4 $\in]0.0000, 0.0311[$ AND range2 $\in [0.1542, +\infty[$ AND I1MAX $\in [2.7850, +\infty[$ AND I3 $\in]1.2763, 5.6916[$	1

C. Rule Coverage Statistics

The statistics for individual rules and each class from the coverage analysis have been summarized in Table IV. All rules have an equal length of four conditions each and achieved a high and similar accuracy along with an equal coverage for each class. Additionally, the remaining instances represent the samples that were not covered in each of the iterations.

D. Rule Coverage Analysis

The data represented in rule coverage histogram is a measure of rule coverage related to each fault class in the data set which is shown in Figure 5. The coverage analysis shows

TABLE IV
RULE COVERAGE AND ACCURACY STATISTICS (AS EXTRACTED TEXT)

Rule	Target	Acc(%)	Len	Coverage	Total (class)	Remaining
1	Class 3	97.47	4	193	198	5
2	Class 1	86.27	4	132	153	21
3	Class 2	89.26	4	133	149	16
4	Class 0	91.00	4	91	100	9
5	Class 1	11.11	4	17	153	4
6	Class 2	10.74	4	16	149	0
7	Class 0	9.00	4	9	100	0
8	Class 3	2.53	4	5	198	0
9	Class 1	2.61	4	4	153	0

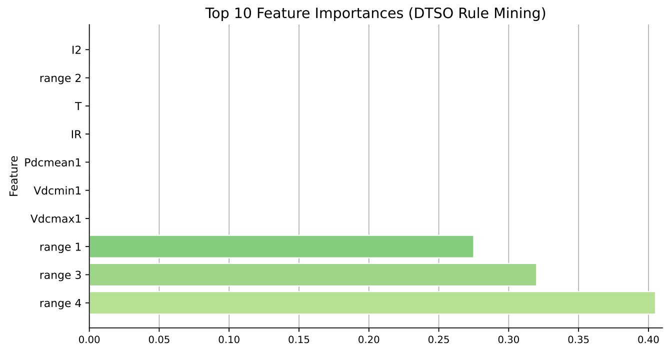

Fig. 4. Relative Importance of Discretized Features in RC.

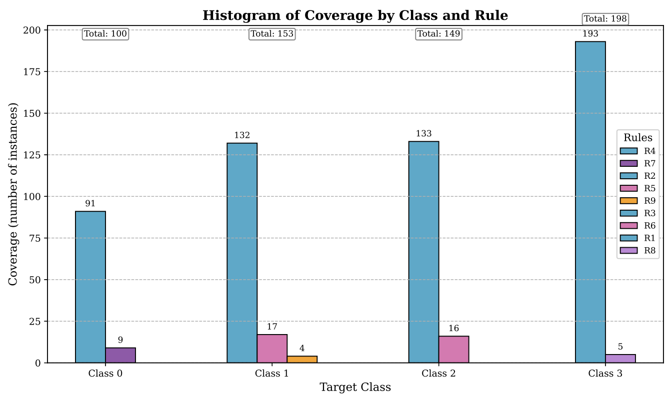

Fig. 5. Rule Coverage Histogram for the RCFDD-PVs Model.

that each partition of the rule set assigned to each class captures a significant majority of its corresponding instances, with a negligible coverage for members in other classes. For class 0, the main rule group covers 91 instances of the class while 9 other instances fall in the coverage of secondary rule assignments. Class 1 is represented by 132 instances that fall in the coverage of the main rule and a small coverage of 17 and 4 instances to other rules. Class 2 also shows a good consistency between the instances of the class and the rules assigned, with 133 main rule instances and 16 other secondary associations. Finally, class 3 achieves the highest level of coverage, with 193 instances accurately captured and only 5 cases showing minor overlap. As seen in Figure 7, there is a histogram providing more insight about the coverage and precision of the rules that were extracted. From this histogram, the first four rules clearly have high coverage (15% to 32%)

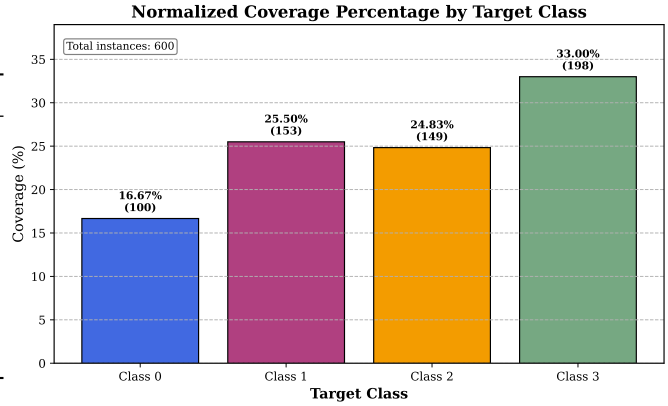

Fig. 6. Normalized coverage percentage by target class.

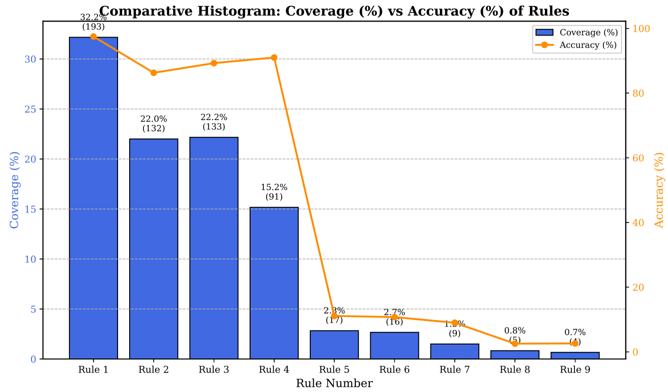

Fig. 7. Histogram comparing rule coverage and precision.

and high precision at the same time, highlighting the most important rules to find the primary patterns in the dataset. On the contrary, Rules 5 to 9 have much lower coverage, all of them are below 3%, reflecting their role as supporting rules that handle less frequent patterns and cover remaining instances. All of this goes to show that the rules optimized by the DTSO are efficient and compact, as there are a few rules that have high coverage that explain most of the data, while the smaller rules ensure complete coverage without reducing clarity. In addition, the class level coverage is illustrated in Figure 6. Class 3 contains the highest representation (33%) while class 1 follows (25.5%) along with class 2 (24.83%) with class 0 covering 16.67%. These class proportions reflect the original class distribution, demonstrating that the model representation is unbiased across all fault types. Overall, the generated rules offer complete coverage with respect to all other classes balanced representation while confirming the reliability in the RCFDD-PVs diagnosis framework.

E. Classification Performance

The dataset was tested using the RCFDD-PVs model, which achieved an overall accuracy of 99%, along with a weighted F1-score of 0.99. As seen in Figure 8, which illustrates confusion matrix, the system achieves near-perfect classification: classes 0, 1 and 3 were detected with 100% accuracy, while class 2 showed only six misclassified instances. These results show that the proposed method offers both outstanding

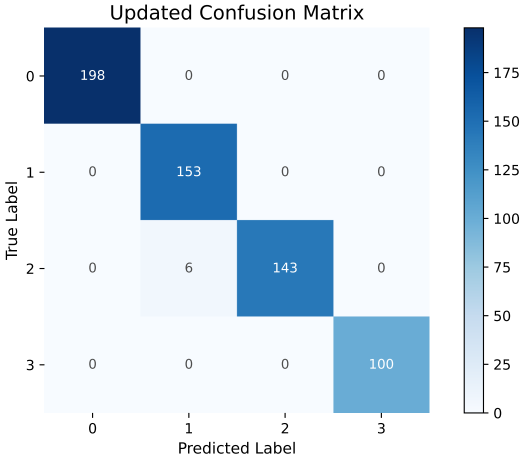

Fig. 8. Matrix of Confusion Showing the classification Performance of the Rule-Based Model.

TABLE V
COMPARATIVE PERFORMANCE OF RCFDD-PVS AND BENCHMARK MODELS (AS IN DOCX TABLE)

Model	CA(%)	F1(%)	Prec.(%)	Rec.(%)	Spec.(%)	Rules
RCFDD-PVs	99	99	99	99	99.48	YES
AdaBoost	95	94.9	95.8	95	98.3	NO
Random Forest	81	80.6	83.4	81	93.7	NO
Lasso Regression	76	75.3	79.9	76	92	NO
Ridge Regression	71	69.4	73.3	71	90.3	NO
Naïve Bayes	60	58.9	58.4	60	86.7	NO
CN2 rule induction	50	48.5	51.8	50	83.3	YES

diagnosis capability, confirming its effectiveness for reliable fault detection in PVs.

VII. Discussion and Comparative Analysis

A comparative study was conducted against both statistical learning models and rule-based alternatives. The benchmark set included AdaBoost, Random Forest, Lasso Regression, Ridge Regression, Naïve Bayes, and CN2 rule induction. Table V summarizes the results and shows that RCFDD-PVs offers the best overall balance between predictive performance and interpretability.

A. Computational Complexity Analysis

Let N_s denote the number of training samples, m the number of selected features, R the number of induced rules, L the maximum rule length, P the DTSO population size, and T_{max} the maximum number of iterations. For one SC call, evaluating a population of candidate rules requires checking up to L antecedents over the current uncovered subset, so the dominant training cost is bounded by $\mathcal{O}(P\,T_{max}\,N_s\,L)$ in the worst case. Because sequential covering induces R rules, the overall training cost can be upper-bounded by $\mathcal{O}(R\,P\,T_{max}\,N_s\,L)$. In contrast, prediction is inexpensive: each sample is tested against at most R rules, each of length L, which yields a per-sample inference cost of $\mathcal{O}(R\,L)$. In the present study, $R = 9$ and $L = 4$, so inference requires at most 36 antecedent evaluations per sample.

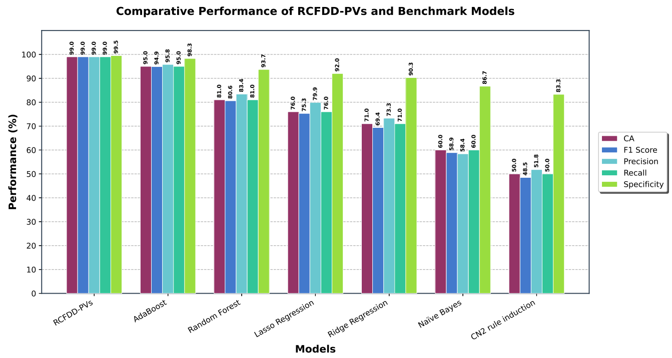

Fig. 9. Comparative Model Performance.

Compared with the benchmark methods, Naïve Bayes has roughly linear training complexity in the number of samples and features, linear models such as Ridge and Lasso depend on the number of optimization iterations. Ensemble methods such as AdaBoost and Random Forest add the cost of repeatedly constructing decision trees and evaluating splits, while CN2 may become expensive because the search space of candidate rule selectors grows combinatorially. Therefore, although SC-DTSO performs an iterative metaheuristic search during training, it preserves low-complexity inference and yields an explicit rule base that is particularly attractive for interpretable PV fault diagnosis.

B. Comparison with Benchmark Methods

As reported in Table V, RCFDD-PVs achieved 99% for classification accuracy, F1-score, precision, and recall, together with a specificity of 99.48%. AdaBoost provided the closest competing performance, with 95% accuracy and a 94.9% F1-score, but it remains a black-box ensemble method. Random Forest reached 81% accuracy, while the linear baselines and Naïve Bayes showed substantially lower performance. The CN2 rule-based model also performed markedly worse, suggesting that the proposed SC-DTSO optimization provides a more effective rule-search mechanism for this PV dataset.

These results indicate that the proposed framework is not only highly effective but also practically attractive because it produces explicit and human-readable IF–THEN rules. Figure 9 provides a visual summary of the evaluation metrics and highlights that RCFDD-PVs is the only tested model that combines near-optimal predictive performance with full rule-based interpretability.

VIII. Conclusion

This paper presented RCFDD-PVs, an interpretable framework for photovoltaic fault detection and diagnosis that combines entropy-based discretization, Sequential Covering and Discrete Tuna Swarm Optimization. The method transforms continuous variables into informative intervals and generates compact IF–THEN rules capable of distinguishing normal and faulty operating states in PV systems.

Validation on a four-class PV dataset showed that the proposed framework achieved 99% accuracy and a weighted

F1-score of 0.99 using only nine optimized rules. The comparative analysis confirmed that RCFDD-PVs outperformed the benchmark methods considered in this study while preserving the transparency of a white-box classifier. These results demonstrate that the proposed framework can provide both strong diagnostic reliability and clear decision support for PV monitoring applications.

Future work will focus on validating the approach on real field measurements, studying its robustness under broader operating variability and extending the framework toward online deployment for real-time PV monitoring and decision support.

References

[1] C. Soumya, B. Deepanraj, and J. Ranjitha, "A review on solar photovoltaic systems and its application in electricity generation," Thrissur, India, 2021, p. 020011. [Online]. Available: https://pubs.aip.org/aip/acp/article/749536

[2] G. R. Venkatakrishnan, R. Rengaraj, S. Tamilselvi, J. Harshini, A. Sahoo, C. A. Saleel, M. Abbas, E. Cuce, C. Jazlyn, S. Shaik, P. M. Cuce, and S. Riffat, "Detection, location, and diagnosis of different faults in large solar PV system—a review," *International Journal of Low-Carbon Technologies*, vol. 18, pp. 659–674, Feb. 2023. [Online]. Available: https://academic.oup.com/ijlct/article/doi/10.1093/ijlct/ctad018/7083000

[3] M. A. Almasoum, "Evaluating faults detection and their impact on Photovoltaic (PV) Modules."

[4] K. Osmani, A. Haddad, T. Lemenand, B. Castanier, M. Alkhedher, and M. Ramadan, "A critical review of PV systems' faults with the relevant detection methods," *Energy Nexus*, vol. 12, p. 100257, Dec. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2772427123000876

[5] B. Long-Dong, Y.-K. Wu, and M.-H. Pham, "Fault identification and diagnosis methods for photovoltaic system: A review," in *2021 7th International Conference on Applied System Innovation (ICASI)*. Chiayi, Taiwan: IEEE, Sep. 2021, pp. 126–129. [Online]. Available: https://ieeexplore.ieee.org/document/9568414/

[6] G. M. El-Banby, N. M. Moawad, B. A. Abouzalm, W. F. Abouzaid, and E. A. Ramadan, "Photovoltaic system fault detection techniques: a review," *Neural Computing and Applications*, vol. 35, no. 35, pp. 24 829–24 842, Dec. 2023. [Online]. Available: https://link.springer.com/10.1007/s00521-023-09041-7

[7] D. J. Benavides, P. Arévalo-Cordero, L. G. González, L. Hernández-Callejo, F. Jurado, and J. A. Aguado, "Method of monitoring and detection of failures in PV system based on machine learning," *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 102, pp. 26–43, Aug. 2021. [Online]. Available: https://revistas.udea.edu.co/index.php/ingenieria/article/view/340550

[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 Computing and Applications*, vol. 37, no. 15, pp. 8825–8844, May 2025. [Online]. Available: https://link.springer.com/10.1007/s00521-025-11035-6

[9] R. Ebead, B. Abo-Zalam, and E. Nabil, "System identification of photovoltaic system based on fractional-order model," *Journal of Computational Electronics*, vol. 22, no. 1, pp. 471–484, Feb. 2023. [Online]. Available: https://link.springer.com/10.1007/s10825-022-01996-5

[10] C. Utama, C. Meske, J. Schneider, R. Schlatmann, and C. Ulbrich, "Explainable artificial intelligence for photovoltaic fault detection: A comparison of instruments," *Solar Energy*, vol. 249, pp. 139–151, Jan. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0038092X22008301

[11] H. Slimani, O. Chouhal, Y. Beddiaf, R. Mahdaoui, R. Hamdi, and S. Djahfa, "Fault prediction in the photovoltaic system through machine learning approaches," in *Proceedings of the 6th International Conference on Electrical Engineering and Control Applications – Volume 1: ICEECA 2024, 19–21 November, Khenchela, Algeria*, ser. Lecture Notes in Electrical Engineering, vol. 1471. Springer Nature Singapore, 2025, pp. 198–206.

[12] O. Chouhal, R. Mahdaoui, A. Lahmari, A. R. Benelhabes, H. Haouassi, H. Rahab, and A. Bakhouche, "Explainable fault diagnosis: Rule-based approach with improved discrete grey wolf optimization algorithm," *International Journal of Information Technology & Decision Making*, 2025.

[13] S. Hassina, C. Ouahiba, B. Yassine, M. Rafik, H. Hichem, and H. Roumaissa, "Explainable fault diagnosis using discrete grey wolf optimization algorithm for photovoltaic system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 15, no. 6, pp. 5286–5296, 2025.

[14] D. Leite, E. Andrade, D. Rativa, and A. M. A. Maciel, "Fault Detection and Diagnosis in Industry 4.0: A Review on Challenges and Opportunities," *Sensors*, vol. 25, no. 1, p. 60, Dec. 2024. [Online]. Available: https://www.mdpi.com/1424-8220/25/1/60

[15] Z. Chen, F. Xiao, F. Guo, and J. Yan, "Interpretable machine learning for building energy management: A state-of-the-art review," *Advances in Applied Energy*, vol. 9, p. 100123, Feb. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2666792423000021

[16] G. Mitrentsis and H. Lens, "An interpretable probabilistic model for short-term solar power forecasting using natural gradient boosting," *Applied Energy*, vol. 309, p. 118473, Mar. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306261921016950

[17] G. Kasmi, L. Dubus, Y.-M. Saint-Drenan, and P. Blanc, "Space-scale exploration of the poor reliability of deep learning models: the case of the remote sensing of rooftop photovoltaic systems," *Environmental Data Science*, vol. 4, p. e22, 2025. [Online]. Available: https://www.cambridge.org/core/product/identifier/S2634460225000135/type/journal_article

[18] M. Mersha, K. Lam, J. Wood, A. K. AlShami, and J. Kalita, "Explainable artificial intelligence: A survey of needs, techniques, applications, and future direction," *Neurocomputing*, vol. 599, p. 128111, Sep. 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231224008828

[19] J. Wiggerthale and C. Reich, "Explainable Machine Learning in Critical Decision Systems: Ensuring Safe Application and Correctness," *AI*, vol. 5, no. 4, pp. 2864–2896, Dec. 2024. [Online]. Available: https://www.mdpi.com/2673-2688/5/4/138

[20] S. Tajjour, S. S. Chandel, H. Malik, M. A. Alotaibi, and T. S. Ustun, "A Novel Metaheuristic Approach for Solar Photovoltaic Parameter Extraction Using Manufacturer Data," *Photonics*, vol. 9, no. 11, p. 858, Nov. 2022. [Online]. Available: https://www.mdpi.com/2304-6732/9/11/858

[21] Y. Dou, S. Tan, and D. Xie, "Comparison of machine learning and statistical methods in the field of renewable energy power generation forecasting: a mini review," *Frontiers in Energy Research*, vol. 11, p. 1218603, Jul. 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenrg.2023.1218603/full

[22] Y. Liu and Y. Wu, "Fault diagnosis of photovoltaic modules: A review," *Solar Energy*, vol. 293, p. 113489, Jun. 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0038092X2500252X

[23] A. Bilal Zorić, "Applied Statistics: Basic Principles and Application," *International Journal of Innovation and Economic Development*, vol. 7, no. 3, pp. 27–33, Aug. 2021. [Online]. Available: https://researchleap.com/applied-statistics-basic-principles-and-application/

[24] H. N. Nguyen, Q. T. Tran, C. T. Ngo, D. D. Nguyen, and V. Q. Tran, "Solar energy prediction through machine learning models: A comparative analysis of regressor algorithms," *PLOS ONE*, vol. 20, no. 1, p. e0315955, Jan. 2025. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0315955

[25] H. Bakır, F. A. Kuzhippallil, and A. Merabet, "Automatic detection of deteriorated photovoltaic modules using IRT images and deep learning (CNN, LSTM) strategies," *Engineering Failure Analysis*, vol. 146, p. 107132, Apr. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1350630723000869

[26] G. Nassreddine, A. El 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, p. e115, Apr. 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/ail2.115

[27] B. Ladjal, M. Nadour, M. Bechouat, N. Hadroug, M. Sedraoui, A. Rabehi, M. Guermoui, and T. F. Agajie, "Hybrid deep learning CNN-LSTM model for forecasting direct normal irradiance: a study on solar potential in Ghardaia, Algeria," *Scientific Reports*, vol. 15, no. 1, p. 15404, May 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-94239-z

[28] Levent, G. Şahin, G. Işık, and W. G. J. H. M. Van Sark, "Comparative Analysis of Advanced Machine Learning Regression Models with Advanced Artificial Intelligence Techniques to Predict Rooftop PV Solar Power Plant Efficiency Using Indoor Solar Panel Parameters,"

Applied Sciences, vol. 15, no. 6, p. 3320, Mar. 2025. [Online]. Available: https://www.mdpi.com/2076-3417/15/6/3320

[29] G. S. Eldeghady, H. A. Kamal, and M. A. Moustafa Hassan, "Comparative analysis of the performance of supervised learning algorithms for photovoltaic system fault diagnosis," *Science and Technology for Energy Transition*, vol. 79, p. 27, 2024. [Online]. Available: https://www.stet-review.org/10.2516/stet/2024024

[30] B. E. Elnaghi, A. M. Ismaiel, M. M. Ismail, H. A. Zedan, and A. A. Salem, "Experimental validation of an adaptive fuzzy logic controller for MPPT of grid connected PV system," *Scientific Reports*, vol. 15, no. 1, p. 27173, Jul. 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-10188-7

[31] A. Et-taleby, Y. Chaibi, N. Ayadi, B. Elkari, M. Benslimane, and Z. Chalh, "Enhancing fault detection and classification in photovoltaic systems based on a hybrid approach using fuzzy logic algorithm and thermal image processing," *Scientific African*, vol. 28, p. e02684, Jun. 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2468227625001541

[32] R. Janarthanan, R. U. Maheshwari, P. K. Shukla, P. K. Shukla, S. Mirjalili, and M. Kumar, "Intelligent Detection of the PV Faults Based on Artificial Neural Network and Type 2 Fuzzy Systems," *Energies*, vol. 14, no. 20, p. 6584, Oct. 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/20/6584

[33] S. Rinesh, M. Arun, S. N. Kumar, C. Prajitha, and A. P. S. Kumar, "Evaluating the type 2 fuzzy logic controller with multilayer perceptrons for optimal tracking of solar photovoltaic systems," *International Journal of Low-Carbon Technologies*, vol. 20, pp. 394–403, Jan. 2025. [Online]. Available: https://academic.oup.com/ijlct/article/doi/10.1093/ijlct/ctaf016/8008641

[34] S. K. M. Hossain, S. A. Ema, and H. Sohn, "Rule-Based Classification Based on Ant Colony Optimization: A Comprehensive Review," *Applied Computational Intelligence and Soft Computing*, vol. 2022, pp. 1–17, Apr. 2022. [Online]. Available: https://www.hindawi.com/journals/acisc/2022/2232000/

[35] E. Shnarch, L. Choshen, G. Moshkowich, N. Slonim, and R. Aharonov, "Unsupervised Expressive Rules Provide Explainability and Assist Human Experts Grasping New Domains," Oct. 2020, arXiv:2010.09459 [cs]. [Online]. Available: http://arxiv.org/abs/2010.09459

[36] A. Wroblewska, P. Kaplanski, P. Zarzycki, and I. Lugowska, "Semantic rules representation in controlled natural language in FluentEditor," in *2013 6th International Conference on Human System Interactions (HSI)*. Sopot, Poland: IEEE, Jun. 2013, pp. 90–96. [Online]. Available: http://ieeexplore.ieee.org/document/6577807/

[37] A. Kumar and D. Zhang, "Biometric Recognition using Entropy-Based Discretization," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. Honolulu, HI, USA: IEEE, 2007, pp. II–125–II–128. [Online]. Available: http://ieeexplore.ieee.org/document/4217361/

[38] A. Ledmi, M. Ledmi, M. E. H. Souidi, H. Haouassi, and D. Bardou, "Optimizing Task Scheduling in Cloud Computing Using Discrete Tuna Swarm Optimization," *Ingénierie des systèmes d information*, vol. 29, no. 1, pp. 323–335, Feb. 2024. [Online]. Available: https://iieta.org/journals/isi/paper/10.18280/isi.290132

[39] D. Garcia, J. C. Gamez, A. Gonzalez, and R. Perez, "Using a sequential covering strategy for discovering fuzzy rules incrementally," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Istanbul, Turkey: IEEE, Aug. 2015, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/7337924/

[40] S. S. M. Ghoneim, A. E. Rashed, and N. I. Elkalashy, "Fault Detection Algorithms for Achieving Service Continuity in Photovoltaic Farms," *Intelligent Automation & Soft Computing*, vol. 29, no. 3, pp. 467–479, 2021. [Online]. Available: https://www.techscience.com/iasc/v30n2/44023

Souria Sif obtained her Bachelor's degree (Licence) in Electrotechnics from Abbes Laghrour University, Khenchela, Algeria, in 2019. She then earned her Master's degree in Electrical Control (Commandes Électriques) from the same university in 2021. She is currently a fourth-year Ph.D. student in Electrical Control at Abbes Laghrour University, Khenchela, Algeria. Her research interests are focused on fault diagnosis in photovoltaic systems using data mining techniques.

Ouahiba Chouhal received her PhD from the Department of Industrial Engineering at the University of Batna, Algeria, in 2018 and her HDR (Habilitation universitaire) from the University of Batna 2 in 2020. She is currently an Assistant Professor in the Department of Computer Science at the University of Khenchela, Algeria. She received her Magister degree from the Department of Computer Science at the University of Tebessa in 2004, Algeria. Her current research interests include evolutionary computation, artificial intelligence, industrial diagnosis and prognosis, multi-agent systems and service-oriented architecture (SOA).

Yassine Beddiaf obtained bachelor's degree in 1989. MSc and PhD all in Electrical engineering, from the University of Batna-2, Algeria in 1994, 2008 and 2016, respectively. He has held a teaching position in industrial engineering department in Khenchela University. Algeria. Since 2014. Nominated Head of the Department of Industrial Engineering from 2012 to 2016. Nominated master of conference class (A) since 2021. Field of research related mainly to the robust control of electrical systems, with applications of artificial intelligence.

Abdeldjalil Ledmi received his engineer's degree in computer science from the University of Khenchela (Algeria) in 2010. He also received his Master's degree in Computer Science from the same university in 2014 and his Ph.D in Computer Science from Larbi Tebessi University (Algeria) in 2019. He is working as a Lecturer in the Department of Mathematics and Computer Science at the University of Khenchela (Algeria). Moreover, he is affiliated as a researcher in the ICOSI Lab. His research interests include artificial intelligence, metaheuristics and swarm-based optimization, Cloud Computing.

Rafik Mahdaoui received the Ph.D. degree in Industrial Engineering Sciences from Batna University in 2013 for his thesis titled "Contribution to the dynamic monitoring production Temporal of systems evolving using Neuro-Fuzzy systems". He is currently a Professor in mathematics and computer science department and an IT security team leader in ICOSI laboratory at the Abbes Laghrour University, Khenchela, Algeria. His research interests include computer science, software security and Industrial systems.

Hichem Haouassi received his Ph.D. degree in computer science from the University of Batna, Algeria, in 2012. He is currently a full professor at Abbes Laghrour University, Khenchela, Algeria. His research interests include artificial intelligence, data mining, metaheuristics and swarm-based optimization, feature selection and classification.

L. Chrifi-Alaoui received the Ph.D. in Automatic Control from the Ecole Centrale de Lyon. Since 1999 he has a teaching position in automatic control in the Aisne University Institute of Technology, UPJV, Cuffies-Soissons, France. From 2004 to 2010 and 2014 to 2018 he was the Head of the Department of Electrical Engineering and industrial informatics. His research interests are mainly related to linear and non-linear control including sliding mode control, adaptive control and robust control, with applications to electric drive and mechatronic systems.