

Research paper

Open and reproducible estimation of PV single-diode parameters from datasheet data

Valerio Lo Brano^{*}

Department of Engineering, University of Palermo, Palermo, Italy

ARTICLE INFO

Keywords:

Photovoltaic modules
Single-diode model
Parameter estimation
Genetic algorithm
Open-source modelling
PVLlib
Energy system simulation

ABSTRACT

Photovoltaic (PV) module datasheets typically provide only five key electrical parameters under Standard Test Conditions: open-circuit voltage, short-circuit current, voltage and current at maximum power, and nominal peak power. Although these data are routinely available, they are not sufficient to derive the complete Single-Diode Model (SDM) representation required for accurate performance simulations. This study addresses this limitation by proposing a fully open-source and reproducible methodology to extract the full set of SDM parameters using only manufacturer datasheet key-points, without requiring measured I - V curves. The approach employs a Genetic Algorithm to minimize a composite relative-error index over the datasheet points, allowing all five SDM parameters to vary freely within physical constraints. The complete implementation and all associated scripts are publicly released under a permissive open-source license, ensuring unrestricted access and transparency. Validation is performed on an extensive dataset of 20,000 commercial PV modules from public repositories (SAM/CEC). Results demonstrate high repeatability (median RMSE ≈ 0.003), computational efficiency (median runtime < 6 s), and excellent agreement with reference data, confirming that the proposed framework provides a transparent and reproducible tool for parameter extraction at STC, suitable for both academic research and large-scale industrial database analyses.

1. Introduction

Modelling a PV device (cell, module, or array) means predicting its current–voltage (I - V) behaviour under given conditions of irradiance and temperature, and an accurate model of this curve is fundamental for system design and energy production estimation. Among available equivalent-circuit representations, the single-diode model (SDM) has become the de facto standard due to its balance between physical fidelity and computational simplicity, idealising the PV device as a circuit comprising a current source (photocurrent I_L), one diode, a series resistance (R_s), and a shunt resistance (R_{sh}), governed by an implicit transcendental relationship that encodes the diode saturation current (I_0) and the cell ideality factor (n). Despite this apparent simplicity, the SDM can reproduce a real module's I - V curve with very good accuracy across a wide range of operating conditions, which explains its widespread adoption in both research and industry — it underpins most PV simulation software (e.g., PVsyst's one-diode model) and appears extensively in studies on PV performance modelling (De Blas et al., 2002; De Soto et al., 2006; Celik and Acikgoz, 2007; Lo Brano et al., 2012).

The five independent parameters of the SDM (I_L , I_0 , n , R_s , R_{sh}) together fully determine the I - V curve at Standard Test Conditions (STC) as depicted in Fig. 1 and, combined with established temperature and irradiance correction formulas, enable performance prediction under arbitrary field conditions, making the SDM the cornerstone of virtually all PV performance simulation workflows.

A practical difficulty, however, is that PV manufacturers do not report these five model parameters on their datasheets; instead, they typically provide only five key electrical quantities at STC, open-circuit voltage (V_{oc}), short-circuit current (I_{sc}), voltage and current at maximum power (V_{mp} and I_{mp}), and nominal peak power (P_{mp}) together with temperature coefficients, but without the diode saturation current, ideality factor, or explicit resistance values (Baqir and Channi, 2022). Recovering the five SDM parameters from these five datasheet points constitutes a non-trivial inverse problem (Lo Brano and Ciulla, 2013): the governing equations are coupled, implicit, and transcendental (due to the exponential diode term), making traditional linear regression inapplicable and requiring instead iterative numerical techniques or analytical approximations, with the further complication that multiple parameter combinations may yield similar I - V characteristics while solutions must simultaneously satisfy physical plausibility constraints

^{*} <https://www.unipa.it/persona/docenti/1/valerio.lobrano>
E-mail address: valerio.lobrano@unipa.it.

<https://doi.org/10.1016/j.egy.2026.109280>

Received 10 November 2025; Received in revised form 3 March 2026; Accepted 6 April 2026

Available online 16 April 2026

2352-4847/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Nomenclature			
Acronyms			
GA	Genetic Algorithm	P_{mp}	maximum electrical power [W]
PVLib	PhotoVoltaics Library Python	R_s	series resistance [Ω]
DEAP	Distributes Evolutionary Algorithms in Python	R_{sh}	Shunt resistance [Ω]
BSD-3	Berkeley Software Distribution	T	Temperature of the PV cell [$^{\circ}\text{K}$]
RMSE	Root Mean Squared Error	T_{ref}	Temperature of the PV cell at STC (25 $^{\circ}\text{C}$ – 298.15 $^{\circ}\text{K}$)
STC	Standard test Conditions	k	Boltzmann constant; Tournament size
SDM	Single Diode Model	V	Voltage generated by the PV panel [V]
SAM/CEC	System Advisor Model/California Energy Commission PV Module Database	V_T	Thermal voltage [V]
I-V	Current-Voltage	V_{mp}	Voltage at the maximum power point [V]
PSO	Particle Swarm Optimization	V_{oc}	Open circuit voltage of the panel [V]
DE	Differential Evolution	μ_{Isc}	Short circuit current thermal coefficient [$\text{A}/^{\circ}\text{C}$]
DE	Differential Evolution	N_s	Number of cells in series
ABC	Artificial Bee Colony	q	Electron charge [C]
SCSO	Sine-Cosine Algorithm	n	Ideality factor of diode
NREL	National Renewable Energy Laboratory	$E_{g,ref}$	Bandgap energy
DE	Differential Evolution	E	Objective function
AM	Air Mass	$e_{V_{oc}}$	Relative error of open circuit voltage
MPP	Maximum Power Point	$w_{V_{oc}}$	Weight of relative error of open circuit voltage
SAM	System Advisory Model	$e_{I_{sc}}$	Relative error of short circuit current
API	Application Programming Interface	$w_{I_{sc}}$	Weight of relative error of short circuit current
GUI	Graphical User Interface	$e_{V_{mp}}$	Relative error of maximum power voltage
HPC	High-Performance Computing	$w_{V_{mp}}$	Weight of relative error of maximum power voltage
IQR	Interquartile Range method	$e_{I_{mp}}$	Relative error of maximum power current
PDF	Probability Density Function	$w_{I_{mp}}$	Weight of relative error of maximum power current
CDF	Cumulative Distribution Function	e_{stat}	Relative error of stationarity term
RMSE	Root Mean Square Error	w_{stat}	Weight of relative error of stationarity term
Symbols		$\mathbf{x}$	Vector of five model parameters
I	current generated by the panel [A]	$E(\mathbf{x}_i)$	objective function for each vector
I_L	photocurrent [A]	P_{sel}	Probability of selection of the best individual
I_{mp}	current at the maximum power point [A]	p_b	Probability that any individual in the population is the best among the sampled candidates
I_{sc}	short circuit current [A]	p_c	Probability of crossover
I_0	reverse saturation current [A]	p_m	Probability of mutation
a	ideality factor	p_{ind}	Probability of mutation of each parameter
P	electrical power [W]	r	Random number
		δ_q	Perturbation term
		η	Distribution index
		ρ	Fraction of replacement index

such as positive resistances and realistic ideality factors. Even slight inaccuracies in the extraction can produce a model that correctly

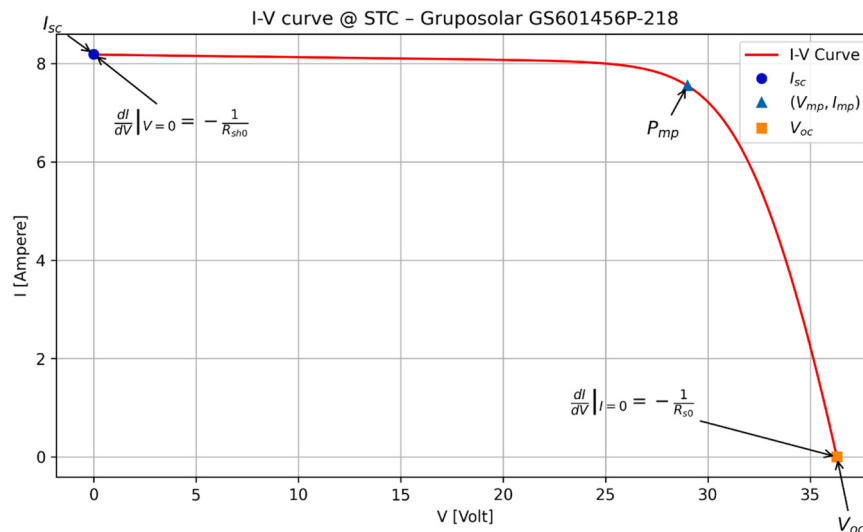


Fig. 1. annotated I-V curve of a PV module with physical meanings of the slopes in short circuit and open circuit points.

reproduces V_{oc} and I_{sc} but yields the wrong fill factor or maximum power, and the sparsity of the available data makes the solution sensitive to measurement tolerances on those key values, so that developing reliable extraction methods from limited datasheet information has been an active area of research for decades (Deotti and da da Silva Junior, 2023).

Several classes of methods have been developed to address this challenge (Deaconu et al., 2024; Beşkirli, Dag, 2023; Devarapalli et al., 2022). These methods can be categorized into analytical approaches, numerical optimization (metaheuristic) approaches (Omer, 2008), and hybrid methods combining elements of both (Chermite et al., 2025). Analytical and explicit approaches attempt to derive the parameters algebraically from the governing equations, often exploiting approximations valid at characteristic curve points or employing the Lambert W special function (Qin et al., 2024; Orioli and Di Gangi, 2019; Nunes et al., 2022) to obtain a closed-form resolution of the implicit exponential relationship; while these methods are typically fast and deterministic, they frequently rely on restrictive assumptions (e.g., $R_{sh} \rightarrow \infty$) or require precise derivatives of the I - V curve at open-circuit and short-circuit (Photovoltaic Solar Cell Models and Parameters Estimation Methods, 2025; Chan and Phang, 2005; Orioli and Di Gangi, 2019) (data that are not directly available from sparse datasheet values) making them sensitive to measurement noise and of limited practical applicability when only five key-points are known. Metaheuristic optimization methods, by contrast, cast parameter extraction as a minimization problem in which an objective function quantifying the mismatch between the model's predicted I - V values and the known data points is minimized by a stochastic search algorithm (Gupta et al., 2024; Chaib et al., 2024); bio-inspired and evolutionary techniques including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Artificial Bee Colony (ABC) (Oliva et al., 2014) have all been successfully applied, and recent comprehensive reviews confirm that metaheuristics have become the dominant approach, with numerous studies reporting very low fitting errors across benchmark cells and modules (Oliva et al., 2014; Gupta et al., 2024; Chaib et al., 2024). The appeal of these methods lies in their flexibility and robustness: they make few assumptions about the functional form of the solution, can accommodate any error definition or constraint, and perform a global search that reduces the risk of convergence on local minima — while, given the small number of points in a datasheet, their computational cost typically remains on the order of a few seconds per module on modern hardware. Hybrid methods combine elements of both paradigms, typically using an analytical derivation to reduce the search space or provide initial parameter estimates that a metaheuristic subsequently refines; machine learning approaches have also been explored as an alternative or complement to traditional iterative methods (Qin et al., 2024). The absence of a universally optimal solution has thus generated a diverse ecosystem of techniques, with the suitability of any given method dictated by a trade-off between computational efficiency, the degree of specialised knowledge required, and the availability of dedicated computational tools.

Notwithstanding the profusion of published techniques, validation in the literature is consistently limited to small benchmark datasets, typically comprising a few dozen to at most a few hundred modules drawn from a restricted set of technologies and manufacturers. While excellent accuracy on such benchmarks is well established (and is indeed expected under the well-controlled operating conditions of STC) the statistical behaviour of extraction methods across the full diversity of commercial PV products, spanning a wide range of manufacturers, technologies, power classes, and design configurations, has not been systematically characterised at large scale. This distinction is not merely quantitative: the robustness of an extraction algorithm, its failure modes, the distribution of residual errors, and the variability of extracted parameters across a heterogeneous commercial population cannot be reliably inferred from narrow validation sets, yet precisely this statistical characterisation is needed to assess whether a given method is suitable for

large-scale applications such as the automated parameterisation of entire module databases for system simulation, energy yield assessment, or bankability analysis. Furthermore, neither academic implementations nor commercial tools (Benghanem, 2009; Saadaoui et al., 2021; Gao et al., 2021) have been subjected to large-scale, systematic validation permitting quantitative statements about repeatability and accuracy distributions across thousands of commercially diverse modules (Bai et al., 2014; Muhsen et al., 2015), a statistical characterisation that is essential for assessing whether a given method is suitable for automated, large-scale parameterisation workflows.

The present study addresses this validation gap by introducing a fully open-source Python implementation of a Genetic Algorithm for SDM parameter extraction from standard datasheet values, validated on an extensive dataset of 20,000 commercial PV modules drawn from the publicly available NREL SAM/CEC database, with the explicit aim of providing a statistically rigorous characterisation of the method's performance across a commercially representative and diverse module population, an analysis made possible precisely by the batch automation capabilities of the open-source implementation. The algorithm requires only the five standard datasheet key-points at STC without measured I - V curves, experimental equipment, or proprietary software, and minimises a composite relative-error index over these five points while allowing all five SDM parameters to vary freely within physical bounds; the implementation leverages PVLlib for I - V modelling (ensuring consistency with industry-accepted models) and the DEAP framework for the evolutionary optimisation core, with support for parallel processing via Python's multiprocessing library to enable practical batch execution across large module sets. The complete source code is released under a BSD-3-Clause licence, implemented in Python 3.12, and thoroughly documented to enable immediate adoption and independent verification without proprietary software dependencies. Across the 20,000-module validation dataset, the algorithm demonstrates high repeatability across independent runs, converging to essentially identical parameter sets for any given module with median RMSE values on the order of 0.003 and median runtimes below 6 s on standard hardware, confirming both the accuracy and the practical computational feasibility of large-scale automated parameterisation.

The central contribution of this work is not a novel optimisation algorithm, but rather a rigorous, large-scale empirical validation of GA-based parameter extraction at a scale previously unreported in the literature, establishing, for the first time, a statistically robust characterisation of repeatability, accuracy, and computational efficiency across thousands of diverse commercial modules, together with a transparent, thoroughly documented, and immediately usable open-source implementation designed to integrate seamlessly into custom workflows, enabling use cases such as generating initial parameter estimates for more complex models, independently verifying manufacturer-supplied parameters in bankability assessments, or populating large simulation databases, all without the need for proprietary software or algorithmic re-implementation. The dissemination of this tool is intended to provide the research community and the photovoltaic industry with a validated resource for systematic model parameterisation in simulation and forecasting applications, and to contribute to enhanced reproducibility and transparency in PV performance modelling. Subsequent sections describe the optimisation problem formulation and genetic algorithm configuration, the computational libraries and dataset employed, and a comprehensive performance evaluation against the reference data.

2. Materials and methods

Unlike most published metaheuristic implementations that remain closed-source or require proprietary MATLAB toolboxes, this work provides a complete, executable implementation using only open-source libraries (PVLlib and DEAP). The technical framework differs from existing approaches in three key aspects:

- exclusive reliance on freely available, community-maintained libraries rather than commercial software;
- comprehensive validation on 20,000 modules rather than typical benchmark sets of < 100 modules;
- complete algorithmic transparency with all parameters, operators, and stopping criteria explicitly documented to enable independent verification and replication.

The following sections detail the mathematical formulation, optimization strategy, and validation methodology.

2.1. Single-diode model equations and assumptions (STC only)

Photovoltaic modules are modelled by the single-diode equivalent circuit, which uses five parameters to reproduce the I - V curve under a given condition. The model is defined by the implicit Shockley diode equation relating output current and voltage modified for an entire PV module:

$$I = I_L - I_0 \left(\frac{V + I R_s}{e^{n \cdot N_s \cdot V_T} - 1} \right) - \frac{V + I R_s}{R_{sh}} \quad (3)$$

This investigation is conducted under Standard Test Conditions (STC: 1000 W/m² irradiance, 25 °C cell temperature, AM1.5 spectrum), utilizing manufacturer datasheet values—namely, open-circuit voltage V_{oc} , short-circuit current I_{sc} , and maximum power point coordinates (V_{mp} , I_{mp}) as primary inputs. At STC, temperature-dependent effects are held constant, and the extracted five parameters explicitly characterize module performance at 25 °C. The model adopts a constant bandgap energy $E_{g,ref}$ and its temperature coefficient $\frac{dE_g}{dT}$, consistent with the California Energy Commission (CEC) model specifications (e.g. $E_{g,ref} \approx 1.121$ for silicon). The implicit single-diode equation is resolved analytically via the Lambert W function implementation within the PVLlib library (Jain and Kapoor, 2004; Jain and Kapoor, 2005; Hansen, 2015). This methodology yields a robust, exact solution for the current-voltage characteristic, circumventing the numerical instabilities associated with iterative solvers and thereby enhancing the reliability of the extraction process. All inputs are structured in accordance with the CEC five-parameter model conventions, and the `pvl-lib.pvsystem.calcparams_cec` function is employed internally to derive temperature-corrected parameters at the reference condition.

2.2. Definition of the objective function: a synthetic relative error index

The parameter identification is formulated as a numerical optimization problem, wherein a GA is employed to minimize a composite objective function. This function, denoted as E , constitutes a synthetic error metric that quantifies the aggregate discrepancy between model-predicted key electrical characteristics and their corresponding datasheet values. The metric incorporates relative errors for the open-circuit voltage, short-circuit current, voltage at maximum power, and current at maximum power, V_{oc} , I_{sc} , V_{mp} , I_{mp} , P_{mp} . Additionally, a stationarity term is included to enforce the zero-derivative condition of the power-voltage curve at the maximum power point. The relative error for any given quantity X is defined as $e_X = |X^{sim} - X^*|$ where X^* is the datasheet value and X^{sim} is the simulated value. The composite objective function is expressed as the following weighted sum:

$$E = w_{V_{oc}} \cdot e_{V_{oc}} + w_{I_{sc}} \cdot e_{I_{sc}} + w_{V_{mp}} \cdot e_{V_{mp}} + w_{I_{mp}} \cdot e_{I_{mp}} + w_{stat} \cdot e_{stat} \quad (4)$$

The stationarity term, e_{stat} , is defined as the absolute value of the derivative of power with respect to voltage at the reported maximum power point, normalized by the datasheet current I_{mp}

$$e_{stat} = \frac{\left| I \left(V_{mp}^* + V_{mp}^* \frac{dI}{dV} \right) \right|}{I_{mp}^*} \quad (5)$$

which would be zero if the model's V_{mp}^{sim} indeed yields a maximum power in which $\frac{dP}{dV} = 0$. In practice, $\frac{dI}{dV}$ at V_{mp} is computed via implicit differentiation of the single-diode equation.

The weights w are chosen to balance the influence of each term:

$$\begin{cases} w_{V_{oc}} = 5 \\ w_{I_{sc}} = 1 \\ w_{V_{mp}} = 1.5 \\ w_{I_{mp}} = 1.5 \\ w_{stat} = 0.3 \end{cases} \quad (6)$$

These values give highest priority to accurately matching V_{oc} (which is particularly sensitive to $n \cdot N_s \cdot V_T$ and I_0) and significant priority to matching the shape at the Maximum Power Point (MPP), while still ensuring I_{sc} is matched within reasonable error. The net effect is a single scalar error index E that heavily penalizes deviations in key performance points of the I - V curve. This formulation, focusing on relative errors, avoids dominance of any one quantity due to units or magnitude differences (e.g., errors in voltage vs. current), and the inclusion of the stationarity condition helps ensure the extracted parameters not only match the numeric values of V_{mp} and I_{mp} but also yield the correct curvature (slope zero) at the MPP. By minimizing E , the GA effectively seeks a best-fit I - V curve that aligns with the datasheet key parameters.

2.3. Genetic Algorithm implementation (population, mutation, elitism, stopping condition)

The optimization of the single-diode model parameters was carried out by means of a GA implemented through the open-source DEAP v1.4.2 framework. Each candidate solution, or individual, represents a possible combination of the five model parameters, grouped in the vector

$$\mathbf{x} = [I_L, I_0, n \cdot N_s \cdot V_{th}, R_s, R_{sh}]^T \quad (7)$$

A population of $N_p = 500$ individuals is initialized by sampling each parameter uniformly within physically realistic bounds (see Section 2.6). Each individual $\mathbf{x}_i$ is evaluated through the composite objective function $E(\mathbf{x}_i)$ defined in Section 2.2, which quantifies the relative deviation between the simulated and datasheet values of the characteristic key points. The optimization goal is therefore to minimize the scalar fitness value $f_i = E(\mathbf{x}_i)$. The GA proceeds by iteratively evolving this population through selection, crossover, mutation, and elitism, gradually improving the overall fitness across generations.

At each generation t , individuals are selected for reproduction using a tournament selection mechanism with tournament size $k = 3$. In each tournament, three candidates are chosen at random, and the individual with the lowest fitness (best solution) is retained as a parent. This mechanism introduces a moderate selection pressure, balancing diversity and convergence. The probability of selecting the best individual within a tournament can be expressed as

$$P_{sel} = 1 - (1 - p_b)^k \quad (8)$$

where p_b represents the probability that any individual in the population is the best among the sampled candidates. This probabilistic formulation ensures that high-quality solutions are more likely to contribute genetic material to subsequent generations, while still preserving the exploration of sub-optimal regions of the search space.

Pairs of selected parents undergo uniform crossover with a probability $p_c = 0.8$. In this operator, each gene (parameter) of the parent individuals is exchanged independently with a 50% probability, producing two new offspring that combine the genetic information of both

parents. Formally, for each gene j ,

$$x_{j,t+1}^{(1)} = \begin{cases} x_{j,t}^{(2)}, & \text{if } r < 0.5, \\ x_{j,t}^{(1)}, & \text{otherwise.} \end{cases} \quad x_{j,t+1}^{(2)} \text{ defined analogously} \quad (9)$$

where r is a uniform random number in $[0, 1]$.

This “swap-based” recombination guarantees that all offspring values remain within their feasible physical ranges, as only existing parameter values are exchanged between parents. The effect of the crossover operator is to promote genetic diversity and accelerate the convergence towards promising regions of the search space by mixing information from well-performing individuals.

After crossover, each offspring undergoes polynomial bounded mutation, which perturbs individual parameters to explore neighbouring regions of the parameter space. Mutation is applied to each gene with a global probability $p_m = 0.2$ per individual and an independent per-gene probability $p_g = 0.1$. The mutated value x'_j of a gene is obtained as

$$x'_j = x_j + \delta_q (x_u - x_l) \quad (10)$$

Where x_l and x_u are the lower and upper bounds of that parameter, and δ_q is a perturbation term drawn from the Deb polynomial distribution, defined as

$$\delta_q = \begin{cases} (2r)^{1/(1+\eta)} - 1, & r < 0.5, \\ 1 - [2(1-r)]^{1/(1+\eta)}, & r \geq 0.5, \end{cases} \quad (11)$$

with r again a random number in $[0, 1]$ and $\eta = 20$ the distribution index controlling the mutation step size.

This formulation ensures small, smooth local variations with occasional larger jumps, thereby maintaining a healthy balance between exploration (searching new solution regions) and refining existing good solutions. Any mutated gene is subsequently clipped to its admissible range to prevent non-physical values.

To prevent loss of good solutions due to random genetic operations, an elitism strategy is adopted: the best 2% of individuals (10 out of 500) are copied unchanged into the next generation. After the variation operators (selection, crossover, and mutation) have been applied, the new population is formed as

$$X_{t+1} = \text{elite}(X_t) \cup \text{offspring}(X_t) \quad (12)$$

ensuring that the global best solution can only improve or remain constant across generations. This elitist mechanism, combined with tournament selection, enhances convergence stability and provides a monotonic reduction of the best error in the population.

The evolutionary process continues for a maximum of 1500 generations or until one of two stopping criteria is satisfied. The first is the attainment of an absolute threshold on the total error:

$$E_{\text{best}} \leq 10^{-4} \quad (13)$$

which represents a negligible mismatch between simulated and data-sheet parameters. The second is the detection of generational stagnation, i.e., no improvement of the best fitness value over a prolonged interval.

In addition to the absolute threshold on the total error, the evolution process is monitored to detect stagnation phases of the best individual. Let E_t denote the minimum value of the objective function at generation t , and let W be a moving monitoring window (here $W = 200$ generations). The incremental variation of the best-fit value over this window is defined as:

$$\Delta E_t = \min_{\tau \in [t-W, t]} E_\tau - E_t \quad (14)$$

If the variation remains below a tolerance ε (in our case $\varepsilon = 10^{-6}$) for an entire window, the process is considered stagnant. In this case, a partial genetic restart (reseed) is triggered to restore diversity: a fraction $\rho = 0.2$ of the population is replaced by newly generated individuals

sampled within the physical bounds, while the remaining portion of the population (including the elite) is retained:

$$X_{t+1} = \text{elite}(X_t) \cup \text{offspring}(X_t) \cup \text{reseed}(\rho) \quad (15)$$

This mechanism preserves the best solutions (elitism) while simultaneously reigniting exploration in previously unvisited regions, thereby reducing the risk of being trapped in local minima. In addition, a minimum generation constraint is imposed to prevent premature termination caused by stochastic fluctuations during the early stages of evolution (here, at least 100 effective generations). Finally, a long-term stagnation threshold halts the algorithm if no significant improvement of the best-fit value is observed over a very large number of monitoring generations (configured as an extreme safeguard). In summary, the termination criterion is:

$$\text{stopif} \begin{cases} E_{\text{best}} \leq 10^{-4}, \\ \Delta E_t < \varepsilon \text{ for } \text{window} W \text{ (apply reseed)}, \\ t \geq t_{\text{max}}. \end{cases} \quad (16)$$

This combination of objective thresholds, stagnation control with partial reseeding, and a multi-term criterion applied to the best individual ensures stable and repeatable convergence, maintaining the proper balance between global exploration and local refinement of the single-diode model parameters.

When either criterion is met, the algorithm terminates early, returning the best parameter vector. The GA configuration parameters were selected based on established guidelines from evolutionary computation literature and preliminary tuning experiments. Population size (500 individuals) provides sufficient diversity for a 5-dimensional search space while maintaining computational tractability—typical recommendations suggest $10\text{--}100 \times$ the number of parameters. Tournament size ($k = 3$) balances selection pressure with diversity preservation. Crossover probability (0.8) represents a standard value proven effective for real-valued optimization. Mutation probability (0.2 per individual, 0.1 per gene) allows adequate exploration without disrupting good solutions. Elite size (2%, 10 individuals) ensures monotonic convergence by preserving best solutions across generations. The distribution index for polynomial mutation ($\eta = 20$) favours small perturbations for local refinement while occasionally permitting larger jumps. These values constitute a well-tested configuration for continuous parameter optimization in constrained search spaces.

All genetic algorithm parameters, including population size, crossover, and mutation probabilities, were calibrated through preliminary trials informed by established literature to ensure robust convergence without incurring prohibitive computational cost. This class of meta-heuristic is well-established in PV parameter identification, effectively balancing global exploration of the parameter space with local refinement of promising solutions.

2.4. Freeware environment

All simulations and optimizations were conducted in a free, open-source software environment to ensure reproducibility. The code was written in Python 3.12, taking advantage of several libraries. In particular, PVLib-python 0.13 (Anderson et al., 2023) was used for all photovoltaic model calculations and DEAP 1.4 for the genetic algorithm (Kim and Yoo, 2019). The PVLib library is a well-validated toolkit for PV systems modelling (Holmgren et al., 2018) and provides a convenient implementation of the single-diode model, including an analytic solver based on the Lambert W function (`pvlb.pvsystem.singlediode` with `method="lambertw"`). By using PVLib's `calcpars.cec` and `singlediode` functions, leverage industry-standard formulations for the PV module's equations were used, reducing the chance of implementation errors and ensuring consistency with known PV performance models. The DEAP library was chosen for its flexibility in constructing evolutionary algorithms; it provides readily customizable GA operators and built-in support for parallel evaluation. Both PVLib and DEAP are open-source and

widely used in the research community, which aligns with our goal of a transparent and reproducible methodology. Additional Python packages used include NumPy (Harris et al., 2020) (for numerical arrays), Matplotlib (Hunter, 2007) (for plotting the I - V curves), and tqdm (da Costa-Luis et al., 2022) (for progress tracking), all in their latest versions at the time of development. The entire environment is freeware, meaning that anyone can replicate or extend this work without proprietary software. This choice is particularly important for energy engineering research, as it lowers barriers for validation and technology transfer of the proposed method.

2.5. Output parameters

The outcome of the optimization procedure is a set of five parameters for the single-diode model that optimally replicate the photovoltaic module's performance at Standard Test Conditions (STC), in addition to a pre-defined temperature coefficient. Virtually, the entire PV module is treated as a single cell. Each parameter possesses a distinct physical significance, as delineated below:

- **Light-Generated Current I_L :** Representing the photocurrent at STC (in amperes), this parameter is directly proportional to incident irradiance. Its value closely approximates the module's short-circuit current I_{sc} under STC, assuming negligible resistive losses, and quantifies the current generated by the photovoltaic effect.
- **Diode Saturation Current I_0 :** This parameter denotes the dark saturation current of the diode (in amperes) at STC, corresponding to the recombination current within the cell in the absence of illumination. As a minuscule current opposing I_L , it critically influences the open-circuit voltage and the exponential region of the I - V characteristic. Physically, I_0 exhibits strong temperature dependence, governed by semiconductor recombination dynamics.
- **Composite Ideality Factor $a = n \cdot N_s \cdot V_T$:** Expressed in volts, this composite parameter is the product of the diode ideality factor, the number of series-connected cells and the thermal voltage. It dictates the slope of the exponential I - V relationship. Elevated values result in a more gradual "knee" in the curve, whereas lower values produce a sharper transition. For silicon cells at 25°C, the ideality factor typically slightly exceeds unity.
- **Series Resistance R_s :** This resistance (in ohms) models internal resistive losses arising from cell materials, metallization, and interconnections. It induces a linear voltage drop proportional to current, thereby diminishing the fill factor. Minimizing R_s is desirable to reduce parasitic power dissipation.
- **Shunt Resistance R_{sh} :** Representing leakage currents that bypass the p-n junction due to material imperfections, this parallel resistance (in ohms) primarily affects the I - V characteristic near the short-circuit condition. A reduced R_{sh} decreases the current output at low voltages, while an ideally infinite value indicates the absence of shunt paths.
- **Short-Circuit Current Temperature Coefficient α_{sc} :** This coefficient (typically in A/°C or %/°C), quantifying the increase in I_{sc} with cell temperature, is not a fitted parameter but a fixed input sourced from manufacturer datasheets. As the extraction is performed isothermally at STC, α_{sc} cannot be empirically inferred and is instead utilized by the underlying CEC model for temperature corrections beyond the reference condition.

In summary, the GA optimization yields $I_L, I_0, n \cdot N_s \cdot V_T, R_s, R_{sh}$ that, when plugged into the single-diode model, accurately reproduce the module's I - V curve at STC. The parameter α_{sc} is provided by the user to complete the model's inputs (especially if one were to simulate performance at other temperatures), but it remains fixed during the extraction process. Each extracted parameter can be related back to physical characteristics of the PV module (light generation, diode quality, resistive losses), which aids in interpreting the results beyond mere curve-

fitting.

2.6. Pipeline Overview

The overall fitting procedure can be summarized as a sequence of steps, illustrated in a schematic flow diagram (see Fig. 3 for a conceptual outline of the pipeline from input to output):

1. **User Inputs** – The process begins with the user providing the PV module's datasheet specifications at STC and the GA settings. Required inputs include the five key electrical points $V_{oc}, I_{sc}, V_{mp}, I_{mp}, P_{mp}$. The temperature coefficient is preset but it can be changed by the user as well as a target error threshold for stopping (e.g. $E_{target} = 10^{-4}$). These define the optimization targets and constraints.
2. **Initial Parameter Ranges** – Before optimization, each unknown parameter is given a plausible range based on physical considerations and the provided datasheet values. For example, an initial estimate for series resistance is $R_{s, approx} = \frac{V_{oc}}{I_{sc}}$, and for shunt resistance $R_{sh, approx} = \frac{V_{mp}}{0.2 \cdot (I_{sc} - I_{mp})}$. Bounds were set around such estimates (see Table 1) to allow flexibility. Likewise, I_L is bounded near I_{sc} (within $\pm 1\%$), I_0 is bounded to a typical small range, and $n \cdot N_s \cdot V_T$ (often denoted a_{ref}) is bounded between about 0.8 and 3 to cover realistic diode ideality factors. These ranges define a 5-dimensional search space for the GA.
3. **GA Initialization** – A population of candidate solutions is randomly generated. Each individual consists of a tuple $[I_L, I_0, n \cdot N_s \cdot V_T, R_s, R_{sh}]$, with each gene sampled uniformly from the respective range defined in the previous step. The population size (here 500) is chosen to ensure a diverse sampling of the search space. An initial fitness evaluation is done for the population to assess the error E of each individual's parameters.
4. **Evaluation (SDM simulation)** – For each individual, the five parameters are passed to the single-diode model solver (PVLlib). Using `pvl-lib.pvsystem.singlediode` with the Lambert W method, the module's I - V characteristics were computed implied by that parameter set. From the result, the key outputs $V_{oc}^{sim}, I_{sc}^{sim}, V_{mp}^{sim}, I_{mp}^{sim}$, and P_{mp}^{sim} are obtained.

In addition to reproducing the three key datasheet landmarks V_{oc}, I_{sc} , and P_{mp} , the algorithm evaluates another condition enforcing the tangency of the power curve at the nominal maximum-power point. This ensures that the extracted parameters yield a realistic curve shape whose P - V profile exhibits a true maximum (zero slope) at the manufacturer's V_{mp} .

Specifically, the power derivative is computed from the single-diode model as:

$$\frac{dP}{dV} = I(V) + V \frac{dI}{dV} \quad (17)$$

and evaluated at $V = V_{mp}$, where $I(V_{mp})$ is obtained from the analytical Lambert- W solution and $\frac{dI}{dV}$ is derived via implicit differentiation of the single-diode equation:

Table 1
Definition of solutions space.

Variable	First estimation	Lower boundary	Upper boundary
R_s	$\frac{V_{oc}}{I_{sc}}$	$\frac{1}{80} \frac{V_{oc}}{I_{sc}}$	$\frac{1}{8} \frac{V_{oc}}{I_{sc}}$
R_{sh}	$\frac{V_{mp}}{0.2 \cdot (I_{sc} - I_{mp})}$	$\frac{1}{2 \cdot 0.2 \cdot (I_{sc} - I_{mp})}$	$2 \cdot \frac{V_{mp}}{0.2 \cdot (I_{sc} - I_{mp})}$
I_L	I_{sc}	$0.99 \cdot I_{sc}$	$1.01 \cdot I_{sc}$
I_0	-	10^{-12}	10^{-8}
a_{ref}	-	0.8	3

$$\frac{dI}{dV} = -\frac{\frac{\partial F}{\partial V}}{\frac{\partial F}{\partial I}} \quad \text{with} \quad \begin{cases} \frac{\partial F}{\partial V} = \frac{I_0}{a} \cdot e^{\frac{V+I \cdot R_s}{a}} + \frac{1}{R_{sh}} \\ \frac{\partial F}{\partial I} = 1 + \frac{I_0 \cdot R_s}{a} \cdot e^{\frac{V+I \cdot R_s}{a}} + \frac{R_s}{R_{sh}} \end{cases} \quad (18)$$

The stationarity residual is subsequently quantified as the normalized absolute deviation of this derivative from zero, as defined in Eq. (5).

The objective function, E , is then computed for the individual candidate solution according to the formulation detailed in Section 2.2. This evaluation constitutes the most computationally demanding component of the algorithm.

5. **GA Evolution (Selection, Crossover, Mutation)** – Following each evaluation cycle, the GA orchestrates the creation of a new population through a sequence of selection, crossover, and mutation operators, as parameterized by the GAConfig class. The evolutionary process utilizes a fixed population size of 500 individuals and is iterated for a maximum of 1500 generations, subject to premature termination upon meeting a predefined convergence criterion.

Parent selection uses a tournament selection scheme of size 3, in which three individuals are chosen at random and the fittest is selected as a parent. This moderate selection pressure promotes gradual convergence without premature loss of diversity.

Selected parents undergo variation through two stochastic operators:

Crossover (recombination): Executed with a probability of $p_c = 0.8$, the algorithm employs a custom uniform swap-based operator (safe_mate). This mechanism exchanges corresponding parameters between two parent individuals with a 50% probability for each, ensuring all generated offspring reside within a valid and feasible parameter space.

Mutation: Applied to each individual with a probability $p_m = 0.2$. Within a selected individual, each parameter (gene) is mutated independently with a probability of $p_{ind} = 0.1$. The mutation follows a polynomial bounded distribution (shape parameter $\eta = 20$), implemented via the safe_mutate function. This strategy predominantly introduces minor perturbations for local fine-tuning, while occasionally permitting larger modifications to preserve global exploratory capability.

A strict elitism strategy is implemented, whereby the ten highest-fitness individuals (elite_size=10, constituting 2% of the population) are preserved unchanged into the subsequent generation. This guarantees the monotonic non-degradation of the best-found solution across generations. The algorithm terminates upon satisfying one of the following conditions:

1. **Convergence Criterion:** The best individual achieves an objective function value $E < 10^{-4}$, signifying convergence to a solution of sufficient numerical accuracy and physical consistency.
2. **Generation Limit:** A hard cap of 1500 generations is reached.
3. **Stall Safeguard:** A fail-safe mechanism halts the process if no improvement in the best error is observed over an extended period (15,000 generations), preventing excessive computational expenditure in practice.

This configuration, which carefully balances exploitative (elitism, controlled mutation) and exploratory (large population size, tournament selection) forces, was designed to ensure robust convergence to a near-optimal solution within a computationally tractable timeframe.

6. **Final Outputs and Plotting** – Upon termination of the genetic algorithm—triggered either by convergence to the error threshold or by reaching the maximum generation limit—the parameter set of the highest-fitness individual is retrieved as the final solution. This solution comprises the five optimized parameters $[I_L, I_0, n \cdot N_s \cdot V_T, R_s, R_{sh}]$.

To validate the solution, a final current-voltage (I - V) characteristic is simulated using the single-diode model with the extracted parameters and compared against the manufacturer's datasheet values. A graphical representation is generated, depicting the fitted I - V curve alongside markers indicating the key datasheet points V_{oc}^{sim} , I_{sc}^{sim} , and P_{mp}^{sim} , providing immediate visual confirmation of the model's accuracy. The visualization is annotated with key performance metrics, such as the final aggregate relative error. A comprehensive numerical summary, detailing the extracted parameter values and a comparison between simulated and datasheet values is concurrently output. This conclusive validation step verifies that the identified parameters yield a model which faithfully reproduces the module's expected performance. All resultant data and graphical outputs are systematically saved for subsequent analysis and documentation. The algorithm's flowchart is summarised in Fig. 2.

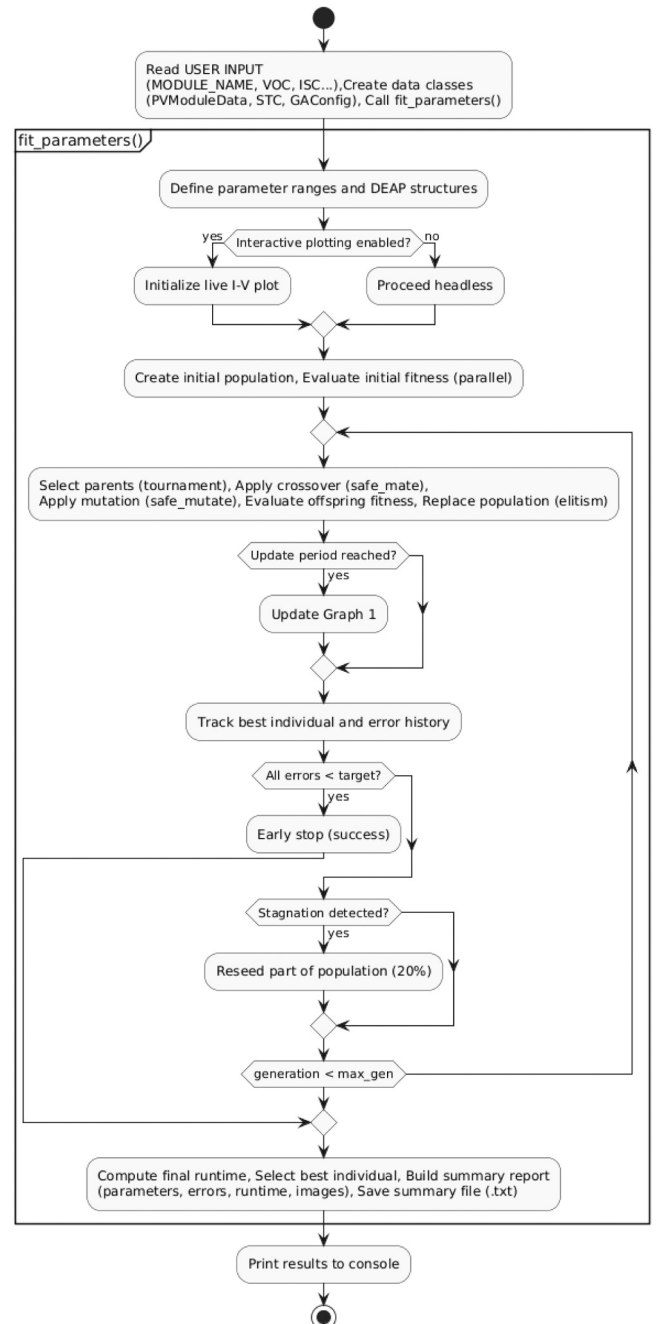


Fig. 2. working flow of the proposed algorithm.

3. Dataset and experimental setup

3.1. PV Module data and STC key-points extraction

The photovoltaic module specifications utilized in this investigation were sourced from the CEC PV Module Database, a comprehensive and authoritative repository integrated into the NREL System Advisor Model (SAM) and the PVLlib Python library. This database encompasses technical data for over 21,500 commercial PV modules, each characterized by a suite of 25 parameters, including nameplate ratings and key electrical performance indicators at Standard Test Conditions (STC). The five critical STC metrics—open-circuit voltage, short-circuit current, voltage at maximum power, current at maximum power, and peak power—were extracted for each module. These values, explicitly provided in dedicated database fields, serve as a reliable ground truth for the parameter fitting procedure. To ensure robust and automated data acquisition, a dedicated Python routine was developed leveraging the PVLlib Application Programming Interface (API). This routine incorporates a fallback strategy to handle nomenclature discrepancies: if a query using a module's primary name fails to return a record, the algorithm systematically attempts alternative known aliases and naming conventions to successfully locate the correct entry. This method guarantees the consistent retrieval of the target electrical parameters, irrespective of variations in manufacturer naming practices. The CEC database's extensive coverage, standardization, and provision of both datasheet values and derived single-diode parameters establish a robust foundation for this analysis, eliminating the need for manual data entry and its associated potential for error.

3.2. Random module sampling

The evaluation of the parameter extraction framework was conducted on a broad, statistically significant sample of photovoltaic modules, rather than on a limited selection of specific panels. From the comprehensive CEC database containing tens of thousands of entries, a large subset comprising several thousand modules was selected via a randomized sampling procedure. This methodology ensures that the experimental analysis encompasses a diverse array of manufacturers, cell technologies, and performance ratings, thereby mitigating selection bias and enhancing the generalizability of the findings. The sample size, on the order of 10^3 modules, represents a deliberate compromise between computational tractability and statistical representativeness; it is sufficiently extensive to facilitate a meaningful analysis of estimation accuracy across the photovoltaic technology landscape, while remaining feasible for large-scale batch processing. For each module in the sample, the known STC parameters—sourced directly from the database—served as the input for the parameter estimation algorithm. The randomized selection strategy precludes the ad hoc selection of modules and instead robustly demonstrates the framework's performance across a wide spectrum of commercially available products, spanning from high-efficiency monocrystalline silicon to older or less conventional designs. To guarantee the reproducibility of this study, the random seed governing the module selection process was fixed.

3.3. Batch automation

All selected modules were processed through an automated, headless batch workflow designed to estimate the five single-diode parameters utilizing the GA methodology, implemented via the DEAP (Fortin et al., 2012) library and integrated with PVLlib's CEC single-diode model. A custom Python script orchestrated the entire procedure without requiring manual intervention. To ensure stability and efficiency during large-scale execution, all graphical and interactive outputs, such as the dynamic plotting of I - V curves, were programmatically suppressed. This precaution was critical to prevent memory accumulation and avoid disruptive Graphical User Interface (GUI) pop-ups when processing

thousands of modules sequentially. The batch script systematically reads the list of sampled modules, retrieves their corresponding STC data, executes the GA-based parameter estimation for each module, and archives the results.

All intermediate and final results—comprising the optimized parameter sets and associated error metrics for each module—were automatically logged and aggregated. The outputs were consolidated into a structured summary file (e.g., an Excel spreadsheet), which catalogues each module alongside its extracted parameters and performance indicators. A representative excerpt of this summary is presented in Fig. 3 to illustrate the results format.

3.4. Computational setup

All computational simulations and optimization routines were performed on a standard workstation equipped with a i9-12900 CPU and on a more performant system characterized by 4 Intel Xeon Gold 6434 H. The first configuration utilized a 24-core processor, enabling the concurrent execution of up to 24 independent module analyses. The second configuration is capable of 64 concurrent processes. The implementation, developed in Python, was executed in a non-interactive, headless environment to ensure operational stability during extended processing periods. In conclusion, the deployment of the experimental workflow on a capable yet readily available computing platform underscores the practical viability and accessibility of the proposed framework for extensive photovoltaic module characterization.

4. Results

4.1. Fit accuracy and error statistics

The GA-based extraction algorithm was applied to the full dataset of 20,000 PV modules, and fit quality was assessed through the composite Root Mean Square Error (RMSE) metric defined as the sum of relative errors at V_{oc} , I_{sc} , V_{mp} , and I_{mp} . The resulting error distribution, summarised in Tables 2–3 and illustrated in Fig. 5, is markedly right-skewed:

```
1 == Single-Diode Parameter Extraction Summary ==
2 == Model: CEC ==
3 Start time      : 2025-10-22 08:47:37
4 End time        : 2025-10-22 08:47:52
5 Runtime (mm:ss) : 00:16
6 GA generations  : 101 / 1500 (early stop)
7 Population size : 500
8 Crossover prob  : 0.80
9 Mutation prob   : 0.20
10 Tournament size : 3
11 Workers (processes): 24
12
13 -- PV module (STC) --
14 Voc* = 36.30 V   Isc* = 8.19 A   Pmp* = 218.95 W
15 Vmp* = 29.00 V   Imp* = 7.55 A
16
17 -- Fitted parameters (best individual) --
18 alpha_sc (fixed) = 0.05   a_ref = 2.80539
19 I_L_ref = 8.26609 A   I_o_ref = 1.96248e-05 A
20 R_s     = 0.070035 Ω   R_sh = 453.125 Ω
21 nNsVth  = 2.80539 V
22
23 -- SDM simulation vs PV module --
24 Voc(sim) = 36.305 V | Voc* = 36.300 V
25 Isc(sim) = 8.265 A | Isc* = 8.190 A
26 Vmp(sim) = 29.000 V | Vmp* = 29.000 V
27 Imp(sim) = 7.471 A | Imp* = 7.550 A
28 Pmp(sim) = 216.663 W | Pmp* = 218.950 W
29
```

Fig. 3. Excerpt of solution output file.

Table 2
Statistical analysis on RMSE.

Figure	Value	Comment
Number of samples	20000	Large dataset of PV modules
Mean	0.006215	Low average error
Median	0.003020	Few high-error cases
Standard deviation	0.012506	Moderate dispersion
Minimum	0.000003	Excellent best-case accuracy
Maximum	0.520003	Rare extreme outlier
Number of outlier (IQR method)	1665	About 8% of cases exceed normal range, limited impact overall.

Table 3
Percentile analysis on RMSE.

Percentile	RMSE value	Comment
5th	0.000118	Excellent accuracy for the best 5% of predictions
25th	0.000772	Very high accuracy for the lower quartile
50th (median)	0.003020	Central measure of typical algorithm error
75th	0.007282	Errors become more noticeable for the upper quartile
95th	0.022539	Error for the worst 5% of predictions

the median RMSE of 0.003 lies well below the mean of 0.006, reflecting that a small subset of outlier cases (approximately 8% of the dataset by the IQR criterion) shifts the mean upward while leaving the bulk of the distribution unaffected, and over 95% of modules yield RMSE below 0.023. The near-complete overlap between the full-dataset and outlier-removed CDF curves confirms that these extreme cases carry negligible influence on the global performance metrics. It should be noted

that low RMSE values under STC are expected for any well-calibrated single-diode model, since the five optimisation targets coincide with the five datasheet constraints under the same operating conditions; the primary significance of these results therefore lies in the consistency and repeatability of the extraction across 20,000 commercially diverse modules, rather than in the absolute magnitude of the errors.

4.2. Statistical analysis across module type

Performance varies systematically across PV technologies, as shown in Table 4. Mono-crystalline and multi-crystalline silicon modules, which together constitute over 19,000 of the 20,000 samples, achieve median RMSE values around 3×10^{-3} and a 5th–95th percentile range below 0.02, confirming stable and consistent behaviour across a large and representative population. CdTe modules exhibit the highest errors (mean ≈ 0.11 , median ≈ 0.112) with a narrow distribution, suggesting a systematic offset rather than sporadic deviations, likely attributable to the different shape of the I - V curve in this technology relative to crystalline silicon. CIGS modules show moderately elevated errors (mean ≈ 0.038) with limited variability, consistent with the very small sample size ($n = 8$). The Thin-Film category presents the widest spread (std ≈ 0.05 , max = 0.52), reflecting its heterogeneous technological composition and the greater diversity of I - V curve characteristics within this group. Overall, crystalline-silicon technologies achieve the lowest and most stable RMSE, while non-silicon thin-film technologies show larger variability attributable to their broader physical and structural diversity.

Runtime statistics per technology (Table 5) indicate that computational performance is largely independent of module type. CIGS and CdTe modules converge with minimal runtime variability (mean ≈ 6.1 s, std ≤ 0.8 s). Mono-c-Si and Multi-c-Si modules exhibit median runtimes around 5.7 s, consistent with overall dataset behaviour, though their

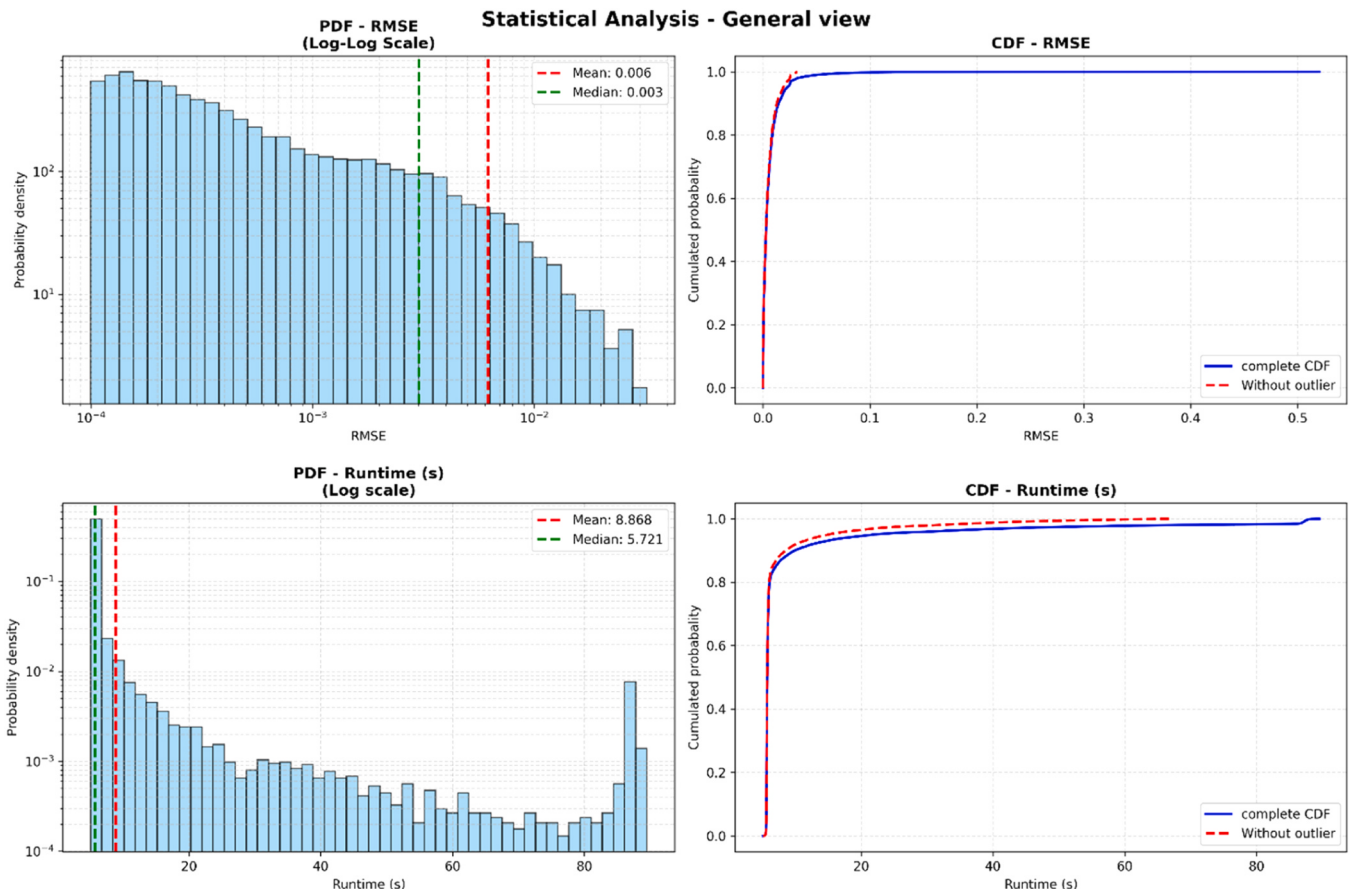


Fig. 5. Probability density function and cumulative density function for RMSE and runtime of proposed algorithm.

Table 4

RMSE statistical and percentile analysis per Module Type.

Module Type	CIGS	CdTe	Mono-c-Si	Multi-c-Si	Thin Film
Count	8	19	9053	10404	516
Mean	0.0376	0.1123	0.0050	0.0053	0.0415
Median	0.0367	0.1124	0.0028	0.0030	0.0298
Std	0.0090	0.0073	0.0062	0.0075	0.0505
Min	0.0272	0.1003	0.000003	0.000004	0.0001
Max	0.0535	0.1235	0.0509	0.1127	0.5200
p05	0.0273	0.1022	0.0001	0.0001	0.0026
p25	0.0308	0.1061	0.0008	0.0007	0.0171
p50	0.0367	0.1124	0.0028	0.0030	0.0298
p75	0.0429	0.1183	0.0067	0.0070	0.0566
p95	0.0504	0.1215	0.0178	0.0196	0.0873

Table 5

Runtime statistical and percentile analysis per Module Type.

Module Type	CIGS	CdTe	Mono-c-Si	Multi-c-Si	Thin Film
Count	8	19	9053	10404	516
Mean	6.11	6.11	9.05	8.76	7.98
Median	5.85	6.00	5.73	5.71	5.84
Std	0.76	0.28	12.92	12.34	9.19
Min	5.62	5.86	5.13	5.17	5.07
Max	7.93	6.96	89.50	89.22	88.95
p05	5.63	5.89	5.60	5.58	5.55
p25	5.72	5.94	5.66	5.65	5.74
p50	5.85	6.00	5.73	5.71	5.84
p75	6.11	6.12	5.93	5.90	6.00
p95	7.30	6.61	23.84	20.72	17.19

larger standard deviations (≈ 12 s) and occasional high-end outliers (maximum ≈ 89 s) slightly elevate the mean values to 9.0 s and 8.8 s respectively; these extended runs correspond to modules with unusual *I*–*V* characteristics requiring additional GA generations to meet convergence criteria, and represent challenging parameter landscapes rather than algorithm failures, as all cases ultimately converge to physically valid solutions. The Thin-Film category behaves similarly to crystalline silicon, with a median of 5.8 s and a comparable tail distribution. Across all technologies, over 90% of executions complete within approximately 6 s.

4.3. Repeatability and robustness of the fit

Given the stochastic nature of the GA, the reproducibility of results across independent runs was assessed by performing multiple extractions on the same module population, each initialised with a different random seed. The analysis confirms excellent repeatability: all independent runs converged to nearly identical final RMSE values, with no instances of divergent solutions or convergence to high-error local minima observed across the entire test set. In a representative example of five independent runs on the same module, the variation in final relative error was below 0.1% in absolute terms, and the resulting parameter sets were functionally interchangeable in terms of their *I*–*V* characteristics. These findings confirm that the combination of tournament selection, elitism, and early-stopping criteria effectively stabilises convergence and suppresses stochastic variability in the final output, providing reliable and consistent parameter extraction regardless of initialisation.

4.4. Computational efficiency

Runtime statistics are reported in Tables 6–7 and Fig. 5. The median processing time per module is 5.7 s on the 64-thread workstation used in this study, with over 90% of all extractions completing within approximately 6 s and only 5% exceeding 20 s. The mean runtime of 8.9 s slightly exceeds the median due to a tail of slower cases (maximum $\approx$

Table 6

Statistical analysis on runtime.

Figure	Value	Comment
Number of samples	20000	Large dataset of PV modules.
Mean	8.8	Average execution time is moderate.
Median	5.7	Typical run completes in under 6 s.
Standard deviation	12.5	High variability due to a few slow cases.
Minimum	5	Fastest runs complete almost instantly after initialization.
Maximum	89.5	Rare long executions caused by complex or poorly converging modules.
Number of outlier (IQR method)	3528	About 18% of runs show extended duration but limited influence overall.

Table 7

percentile analysis on runtime.

Percentile	RMSE value	Comment
5th	5.5	Nearly all runs are faster than 6 s.
25th	5.6	One-quarter of executions cluster around minimal runtime.
50th (median)	5.7	Typical runtime representative of steady algorithm behaviour.
75th	5.9	Three-quarters of all runs end below 6 s, confirming efficiency.
95th	21.7	Only 5% exceed 20 s, indicating very limited high-cost cases.

90 s), corresponding to modules with unusual *I*–*V* curve shapes — such as very low or very high fill factors, or atypical resistance ratios — that require a larger number of GA generations before meeting the convergence threshold; all such cases eventually converge to physically valid solutions, confirming that the extended runtimes reflect the intrinsic difficulty of specific parameter landscapes rather than algorithmic instability. Early stopping proved effective for the vast majority of modules, which reached satisfactory convergence within approximately 120 generations, well below the conservative maximum of 1500 generations. The complete 20,000-module dataset was processed in under 50 h, corresponding to a throughput of approximately 6–7 modules per minute on the 64-thread system; on a more typical workstation such as an i9-12900 with 24 simultaneous threads, throughput drops to approximately 2 modules per minute, indicating that the processing of tens of thousands of modules remains feasible within a few hours on commonly available hardware.

5. Code availability and reproducibility

The complete source code developed for this study has been archived in the journal's official repository and is distributed under the permissive BSD-3 license, which permits free use and modification subject to proper attribution. The repository includes two primary Python scripts: one for identifying the five single-diode model parameters for an individual photovoltaic module, featuring interactive graphical output, and a second for conducting automated batch analysis on multiple modules. Both scripts, along with exemplary output files, are fully available, providing all necessary resources to replicate the experiments described in this work.

To ensure full reproducibility, the code is implemented in Python (version 3.12) and relies exclusively on widely available open-source libraries. Key software dependencies include PVLib (v0.13.1), DEAP (v1.4.2), NumPy (v2.0.2), Matplotlib (v3.9.2), and tqdm (v4.66.5). The specific versions used are documented within the code to facilitate the reconstruction of the execution environment. The scripts are self-contained and include comprehensive inline comments detailing their intended use and configurable parameters; therefore, no further operational instructions are provided herein. Using the supplied programs, all

numerical and graphical results presented in this study can be replicated exactly. It should be noted that the genetic algorithm introduces stochasticity; to obtain results identical to those reported, the random number generator seed can be fixed within the script—a functionality already implemented—thereby guaranteeing the total repeatability of the analysis.

6. Conclusion

This study has presented a fully open-source, Python-based implementation of a Genetic Algorithm for the extraction of the five single-diode model parameters from standard PV module datasheet values, validated on an extensive dataset of 20,000 commercial modules drawn from the publicly available NREL SAM/CEC database and released under a BSD-3-Clause licence to ensure unrestricted transparency and reproducibility. The algorithm requires only the five key electrical quantities provided on virtually all datasheets — V_{oc} , I_{sc} , V_{mp} , I_{mp} , P_{mp} — without the need for measured I - V curves, user-supplied initial guesses, or proprietary software, and achieves consistent convergence to physically plausible parameter sets through the joint enforcement of physical bounds and a stationarity condition at the maximum power point.

The large-scale validation confirms that the method performs reliably and consistently across the full diversity of the commercial module population, with crystalline-silicon technologies, which represent the overwhelming majority of the dataset, achieving the most stable fits, while thin-film and non-silicon technologies exhibit greater variability attributable to their broader physical and structural diversity. It should be clearly acknowledged that low absolute errors under STC are expected for any well-calibrated single-diode model operating under the very conditions used for optimisation; the primary value of this validation therefore lies in its statistical scale and commercial representativeness, rather than in the absolute magnitude of individual fit errors. Repeatability across independent GA runs was confirmed to be excellent, with parameter sets converging to functionally interchangeable solutions regardless of random initialisation, demonstrating that the implemented combination of tournament selection, elitism, and early-stopping criteria effectively suppresses stochastic variability and renders the extraction robust in the context of automated, large-scale processing.

An important highlights of the present work is that parameter extraction is performed exclusively at Standard Test Conditions (1000 W/m², 25°C, AM1.5), so the extracted parameter set characterises module behaviour at this single reference point; translation to arbitrary field conditions under varying irradiance, temperature, shading, or module mismatch requires additional correction models (such as the CEC or De Soto frameworks already implemented in PVLlib) that are well established and outside the scope of this study. Field validation under real operating conditions would require coupling the extracted STC parameters with appropriate irradiance and temperature translation models is a distinct research question that the open-source architecture of this implementation is explicitly designed to facilitate.

Future research directions include extending the extraction framework to incorporate multiple operating conditions simultaneously — for example, additional I - V points at different irradiance or temperature levels when available from manufacturers — as well as adapting the algorithm to more complex model architectures, such as the two-diode model, to improve predictive performance for technologies whose physical behaviour is less well captured by the basic single-diode formulation and for operating conditions that deviate significantly from STC.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This study was conducted as part of the research activities within the Project “Network 4 Energy Sustainable Transition — NEST”, Spoke 1: SOLAR: PV, CSP & CST, Project code PE0000021, Concession Decree No. 1561 of 11.10.2022 adopted by Ministero dell’Università e della Ricerca (MUR), CUP. Project funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.3 - Call for tender No. 341 of 15.03.2022 of Ministero dell’Università e della Ricerca (MUR); funded by the European Union – NextGenerationEU. CUP B73C22001280006.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at [doi:10.1016/j.egy.2026.109280](https://doi.org/10.1016/j.egy.2026.109280). In addition, the code is also available at <https://github.com/valeriolobrano/pvfit5> and <https://pypi.org/project/pvfit5/>

Data availability

All data is available and issued with the paper.

References

- Anderson, K.S., Hansen, C.W., Holmgren, W.F., Jensen, A.R., Mikofski, M.A., Driesse, A., 2023. pvlib python: 2023 project update. *J. Open Source Softw.* 8, 5994.
- Bai, J., Liu, S., Hao, Y., Zhang, Z., Jiang, M., Zhang, Y., 2014. Development of a new compound method to extract the five parameters of PV modules. *Energy Convers. Manag.* 79, 294–303.
- Baqir, M., Channi, H.K., 2022. Analysis and design of solar PV system using Pvsyst software. *Mater. Today Proc.* 48, 1332–1338.
- Benghanem, M., 2009. Low cost management for photovoltaic systems in isolated site with new IV characterization model proposed. *Energy Convers. Manag.* 50, 748–755.
- Beşkiri, A., Dag, I., 2023. Parameter extraction for photovoltaic models with tree seed algorithm. *Energy Rep.* 9, 174–185.
- Celik, A.N., Acikgoz, N., 2007. Modelling and experimental verification of the operating current of mono-crystalline photovoltaic modules using four-and five-parameter models. *Appl. Energy* 84, 1–15.
- Chaib, L., Tadj, M., Choucha, A., Khemili, F.Z., EL-Fergany, A., 2024. Improved crayfish optimization algorithm for parameters estimation of photovoltaic models. *Energy Convers. Manag.* 313, 118627.
- Chan, D.S.H., Phang, J.C.H., 2005. Analytical methods for the extraction of solar-cell single-and double-diode model parameters from IV characteristics. *IEEE Trans. Electron Devices* 34, 286–293.
- Chermite, C., Douiri, M.R., Kraiem, H., Flah, A., 2025. Guided hybrid meta-intelligence for advanced photovoltaic parameter estimation. *Energy Rep.* 14, 2607–2626.
- da Costa-Luis R., Nolet C., Parente A.G., Zito G., Gohlke C., Vinícius J.L., et al. tqdm: A Fast, Extensible Progress Bar for Python and CLI 2022. [doi:10.5281/zenodo.595120](https://doi.org/10.5281/zenodo.595120).
- De Blas, M.A., Torres, J.L., Prieto, E., García, A., 2002. Selecting a suitable model for characterizing photovoltaic devices. *Renew. Energy* 25, 371–380.
- De Soto, W., Klein, S.A., Beckman, W.A., 2006. Improvement and validation of a model for photovoltaic array performance. *Sol. Energy* 80, 78–88.
- Deaconu, A.M., Cotfas, D.T., Cotfas, P.A., 2024. Extracting photovoltaic cells parameters for three diode model using HSDA algorithm. *Energy Rep.* 12, 5096–5109.
- Deotii, L.M.P., da Silva Junior, I.C., 2023. A survey on the parameter extraction problem of the photovoltaic single diode model from a current–voltage curve. *Sol. Energy* 263, 111930.
- Devarapalli, R., Rao, B.V., Al-Durra, A., 2022. Optimal parameter assessment of solar photovoltaic module equivalent circuit using a novel enhanced hybrid GWO-SCA algorithm. *Energy Rep.* 8, 12282–12301.
- Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., Gagné, C., 2012. {DEAP}: evolutionary algorithms made easy. *J. Mach. Learn. Res.* 13, 2171–2175.
- Gao, S., Wang, K., Tao, S., Jin, T., Dai, H., Cheng, J., 2021. A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models. *Energy Convers. Manag.* 230, 113784.
- Gupta, J., Beryozkina, S., Aljaidei, M., Singla, M.K., Safaraliev, M., Gupta, A., et al., 2024. Application of hybrid chaotic particle swarm optimization and slime mould algorithm to optimally estimate the parameter of fuel cell and solar PV system. *Int. J. Hydrogen Energy* 83, 1003–1023.
- Hansen, C., 2015. Parameter estimation for single diode models of photovoltaic modules. Harris, Millman, C.R., van der Walt, K.J., Gommers R, S.J., Virtanen, P., Cournapeau, D., et al., 2020. Array programming with {NumPy}. *Nature* 585, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>.
- Holmgren, W.F., Hansen, C.W., Mikofski, M.A., 2018. pvlib python: a python package for modeling solar energy systems. *J. Open Source Softw.* 3, 884.
- Hunter, J.D., 2007. Matplotlib: a 2D graphics environment. *Comput. Sci. \ Eng.* 9, 90–95. <https://doi.org/10.1109/MCSE.2007.55>.

- Jain, A., Kapoor, A., 2004. Exact analytical solutions of the parameters of real solar cells using Lambert W-function. *Sol. Energy Mater. Sol. Cells* 81, 269–277.
- Jain, A., Kapoor, A., 2005. A new method to determine the diode ideality factor of real solar cell using Lambert W-function. *Sol. Energy Mater. Sol. Cells* 85, 391–396.
- Kim, J., Yoo, S., 2019. Software review: Deap (distributed evolutionary algorithm in python) library. *Genet Program Evol. Mach.* 20, 139–142.
- Lo Brano, V., Ciulla, G., 2013. An efficient analytical approach for obtaining a five parameters model of photovoltaic modules using only reference data. *Appl. Energy* 111. <https://doi.org/10.1016/j.apenergy.2013.06.046>.
- Lo Brano, V., Orioli, A., Ciulla, G., 2012. On the experimental validation of an improved five-parameter model for silicon photovoltaic modules. *Sol. Energy Mater. Sol. Cells* 105. <https://doi.org/10.1016/j.solmat.2012.05.028>.
- Muhsen, D.H., Ghazali, A.B., Khatib, T., Abed, I.A., 2015. Parameters extraction of double diode photovoltaic module's model based on hybrid evolutionary algorithm. *Energy Convers. Manag.* 105, 552–561.
- Nunes H., Pombo J., Mariano S., do Rosário Calado M. Newton-Raphson method versus Lambert W function for photovoltaic parameter estimation. 2022 IEEE Int. Conf. Environ. Electr. Eng. 2022 IEEE Ind. Commer. Power Syst. Eur. (EEEIC/T&CPS Eur., 2022, p. 1–6.
- Oliva, D., Cuevas, E., Pajares, G., 2014. Parameter identification of solar cells using artificial bee colony optimization. *Energy* 72, 93–102.
- Omer, A.M., 2008. Energy, environment and sustainable development. *Renew. Sustain Energy Rev.* 12, 2265–2300. <https://doi.org/10.1016/J.RSER.2007.05.001>.
- Orioli, A., Di Gangi, A., 2019. A procedure to evaluate the seven parameters of the two-diode model for photovoltaic modules. *Renew. Energy* 139, 582–599.
- Photovoltaic Solar Cell Models & Parameters Estimation Methods 2025. (<https://g2vop.tics.com/photovoltaics-solar-cells/parameter-estimation-methods>).
- Qin, C., Li, J., Yang, C., Ai, B., Zhou, Y., 2024. Comparative study of parameter extraction from a solar cell or a photovoltaic module by combining metaheuristic algorithms with different simulation current calculation methods. *Energies* 17, 2284.
- Saadaoui, D., Elyaqouti, M., Assalaou, K., Lidaighbi, S., 2021. others. Parameters optimization of solar PV cell/module using genetic algorithm based on non-uniform mutation. *Energy Convers. Manag X* 12, 100129.