BLIDA 1 UNIVERSITY

Faculty of technology
Department of Automatics and Electrical Engineering

DOCTORAL THESIS

In Automatics

CONTRIBUTION TO THE DIAGNOSIS OF FAULTS IN PHOTOVOLTAIC SYSTEMS USING STATISTICAL APPROACHES AND META HEURISTIC ALGORITHMS

Defended by:

Youssouf MOULELOUED

Before a jury composed of:

A. FERDJOUNI	Professor, Blida 1 University	Chairman
S. TAHRAOUI	MCA, Hassiba Ben Bouali University	Examiner
M.L. FAS	MCA, Blida 1 University	Examiner
K. KARA	Professor, Blida 1 University	Supervisor
A. CHOUDER	Professor, M'sila University	Co-supervisor

Blida, 2024

بسم الله الرحمن الرحيم

قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ
أَنتَ الْعَلِيمُ الْحَكِيمُ

سورة البقرة , الآية 31

DEDICATION

In deep gratitude, I dedicate this work to my beloved parents, **Mouleloued M'hamed** and **Kherchaoui Fadhila**, whose unwavering love, sacrifice, and encouragement have been the guiding lights of my journey. Their support has been my strength, and their prayers my solace. May their lives be filled with health, happiness, and blessings.

To my family, whose boundless love and support have been the cornerstone of my achievements. I dedicate this work to my brother, sister, grandparents, uncles, and aunts, whose encouragement have inspired me to reach for the stars. Your belief in me has fueled my determination, and for that, I am forever grateful.

To my cherished friends and mentors, whose words of wisdom and encouragement have sustained me through the challenges of this journey. This work is a tribute to your unwavering support and belief in my abilities.

ACKNOWLEDGEMENT

First and foremost, I thank the good Allah, all powerful, to have given me the strength to survive, as well as the audacity and the patience to overcome all the difficulties.

The work presented in this thesis was carried out at the Electrical Systems and Remote Control Laboratory (LabSET), Department of Automation and Electrical Engineering, Faculty of Technology, University of Blida 1 Blida, Algeria, under the supervision of Mr. **KAMEL KARA**, Professor at the University of Blida 1, who supervised me throughout this thesis and who shared with me his brilliant intuitions. May he also be thanked for his availability and for the numerous encouragements that he lavished on me.I would also like to extend my gratitude to Professor **AISSA CHOUDER** from the University of Mssila, who co-supervised this thesis. His expertise and valuable feedback significantly contributed to the completion of this work.

I express my gratitude to Mr. **ABDELLAZIZ FERDJOUNI**, Professor at the University of Blida 1, agreed to chair the jury of this thesis.

My thanks also go to Madame. **SOUAAD TAHRAOUI**, MCA at the University of Chlef, and Mr. **MOHAMED LAMINE FAS**, MCA at the University of Blida 1, who agreed to be examiners.

In order not to forget anyone, my warm thanks are addressed to all those who helped me to realize this modest work.

ABSTRACT

This thesis undertook three main studies to detect and diagnose faults in photovoltaic systems. The first study involved simulating solar panels using a single-diode electrical circuit model and deriving the mathematical formula that characterizes the circuit's behavior.The GPC and EPC algorithms were used to determine the values of the five parameters in this model, and a database consisting of practical measurements was employed to assess the effectiveness of these algorithms. In the context of the second study, a novel method for fault detection and diagnosis in PV systems, based on the well-known GPC algorithm, was developed. This approach entails partitioning the training dataset into two hyper spheres, each representing a class, and only calculates the distances between a new data point and the center of each sphere. This eliminates the need to calculate distances across the entire dataset, as is required in classical KNN. In the last achievement of this thesis, another statistical algorithm for the detection and diagnosis of faults in photovoltaic systems was investigated. In contrast to the decision tree based on the Gini index, this algorithm computes Euclidean distances between a chosen point and the entire dataset. It extracts the minimum and maximum distances for each class, and arranging these distances in ascending order identifies one particular case among five. The faults are classified based on this identified case. To ensure the effective operation of both algorithms, four essential features are necessary: cell temperature, irradiance, as well as current and voltage at the maximum power point. Three distinct faults were taken into account: short circuit, open circuit, and partial shading. Finally, these methods were evaluated against a range of machine learning algorithms, such as SVM, DT, KNN, and RF. The obtained results using the developed algorithms demonstrated significant enhancements in accuracy, precision, and recall.

Keywords: single diode model, metaheuristic, GPC, EPC, fault detection and diagnosis, FDD, machine learning algorithms, statistical approaches.

RESUME

Dans le cadre de la réalisation de cette thèse, nous avons développé trois algorithmes principaux. Le premier algorithme concerne l'identification des paramètres du modèle à une seule diode d'un panneau solaire en utilisant les algorithmes GPC et EPC. Nous avons utilisé une base de données pratique pour évaluer les performances du modèle obtenu. Ce modèle est utilisé dans la génération des données d'un système photovoltaïque en fonctionnement. Dans le cadre du second travail, nous avons développé une nouvelle méthode de détection et de diagnostic des défauts dans les systèmes photovoltaïques, basée sur l'algorithme GPC. Dans cette méthode, l'ensemble de données d'apprentissage est divisé en deux hypersphères, chacune représentant une classe. Ensuite, seules les distances entre le point de données choisi et le centre de chaque sphère sont calculées. Cela élimine le besoin de calculer toutes les distances entre les points de l'ensemble de données, comme cela est requis dans l'algorithme KNN classique. Quant à la dernière contribution, nous avons développé un autre algorithme statistique pour la détection et le diagnostic de défauts dans les systèmes photovoltaïques. Contrairement à l'arbre de décision basé sur l'indice de Gini, cet algorithme, utilise le calcul des distances euclidiennes entre un point choisi et les autres points de l'ensemble de données. Ensuite, en se basant sur les distances minimales et maximales pour chaque classe, un cas particulier parmi cinq est identifié, et les défauts sont classés en fonction de ce cas identifié. Les algorithmes de détection et diagnostic que nous avons développé utilisent quatre attribues essentielles: la température des cellules, l'irradiance, ainsi que le courant et la tension au point de puissance maximale. Nous avons considéré trois défauts distincts: court-circuit, circuit ouvert et ombrage partiel. Enfin, nous avons évalué l'efficacité des méthodes que nous avons développées en considérant une gamme d'algorithmes d'apprentissage automatique, tels que SVM, DT, KNN et RF. Les résultats obtenus à l'aide des algorithmes développés ont démontré des améliorations significatives en termes d'exactitude, de précision et de rappel.

Mots clés : détection et diagnostic des défauts, modèle à une seule diode, GPC, EPC, algorithmes d'apprentissage automatique.

الملخص

بغية الكشف عن الأعطال وتشخيصها في الأنظمة الكهروضوئيّة، تمّ إنجاز ثلاثة أعمال رئيسية في هذه الأطروحة. بالنسبة للإنجاز الأول فهو عبارة عن نمذجة للألواح الشمسية بدارة كهربائية أحادية الصمام واستخراج المعادلة الرياضية التي تصف سلوك الدّارة. تتميّز المعادلة الرياضيّة بوجود خمسة معلَّمات تكشف حالة اشتغال الألواح الشمسية فيما إذا كانت طبيعية أو لا. تحديد قيم هذه المعلّمات مهم جدّاً، لذلك تمّ الاستعانة بخوارزميتي ال GPC و ال EPC لهذا الغرض. تمّ تقييم الخوارزميتان باستعمال قاعدة بيانات حقيقية.

فيما يتعلق بالإنجاز الثاني، يتمثّل في تطوير خوارزمية لاكتشاف الأخطاء وتشخيصها في الأنظمة الكهروضوئية، بحيث تقوم بتقسيم مجموعة بيانات التدريب إلى كرتين، كل كرة تمثّل قسم. تقوم هذه الطريقة، جنبًا إلى جنب مع خوارزمية GPC المعروفة، بحساب المسافات حصريًا بين نقطة بيانات جديدة ومركز كل كرة، مما يلغي الحاجة إلى حسابات المسافة عبر مجموعة البيانات بأكملها كما هو الحال في خوارزمية الKNN الكلاسيكية.

أما بالنسبة للمساهمة الأخيرة، فهي عبارة عن خوارزمية أخرى تمّ تطويرها لاكتشاف الأخطاء وتشخيصها في الأنظمة الكهروضوئية. تستخدم الخوارزمية المطورة شجرة القرار القائمة على حساب المسافات الإقليدية لتصنيف الأخطاء المختلفة. على النقيض من شجرة القرار المعتمدة على مؤشر جيني، تحسب هذه الخوارزمية المسافات بين نقطة و مجموعة البيانات بأكملها. وبعد ذلك يتمّ استخراج القيمة الأدنى والقيمة العليا للمسافات لكل قسم. ترتيب هذه المسافات تصاعديّا ينتج عنه ظهور حالة واحدة من جملة خمس حالات. بناء على الحالة الظاهرة، يتم تقسيم البيانات.

أربعة مداخل ضرورية لضمان تشغيل الخوارزميتين بصورة فعّالة هي: درجةحرارة الخلية، والإشعاع، وكذلك التيار والجهد عند أقصى نقطة طاقة. تمّ أخذ ثلاثة أخطاء بعين الاعتبار: الماس الكهربائى، والدّارة المفتوحة، والتظليل الجزئي. في الأخير، الخوارزميات المطوّرة تمّ مقارنتها مع مجموعة من خوارزميات تعلم الآلة هي: DT, KNN, RF, and SVM وقد أظهرت الخوارزميات المطوّرة أداءً أفضل عن باقي الخوارزميّات على مستوى الaccuracy, precision, recall.

الكلمات المفتاحية: نموذج أحادي الصمام، GPC، EPC،كشف الأعطال وتشخيصها، FDD، خوارزميات تعلم الآلة، الطرق الإحصائيّة.

NOMENCLATURE

A	The total surface area of the penguin.
C_g	The ECM value of a single PVM.
A	The total surface area of the penguin
C_x	The ECM value from the starting point to the cut-off.
D	The length of transmission line.
d	The block displacement.
$dist_i$	The distance between the datapoint and the entire dataset.
$dist0_i$	The distance between the 1_{st} hypersphere center and the entire dataset.
$dist1_i$	he distance between the second hypersphere center and the entire dataset.
$dist_{new0_i}$	The new distances for the 1_{st} hypersphere.
$dist_{new1_i}$	The new distances for the second hypersphere.
$dist_i^0$	The distances between the chosen point and all points of class 0.
$dist_i^1$	The distances between the chosen point and all points of class 1.
E_c	The effictive irradiance.
E_{dc}	The energy produced by the PVA.
E_0	The nominal irradiance.
$F_A(X)$	The probability density function.
$factor0$	The minimization factor of the 1_{st} hypersphere.
$factor1$	The minimization factor of the second hypersphere.
G	The irradiance.
g	The gravity.
G_{ref}	The reference irradiance.
H_i	The total in-plan irradiance of the PVA.
I	The current.
I_{in}	The output current of the inverter.
I_{meas}	The measured current.
I_{mpp}	The maximum power point of current.
I_{mp-STC}	The maximum current at STC.
I_{ph}	The light generated current.
I_{pmax0}	The nominal current at the MPP.
I_{pv}	The output current of PVM, PVA, or PV cell.
I_{sc}	The current of short circuit .
I_{sh}	The current passes through R_{sh} .

I_{sim}	The simulated current.
I_0	The diode initial current.
K	Boltzman coefficient.
L_{cm}	The miscellaneous capture losses.
L_{ct}	The thermal capture losses.
M	The number of PVMs in a serie.
max_0	The maximum distance for class 0.
max_1	The maximum distance for class 1.
min_0	The minimum distance for class 0.
min_1	The minimum distance for class 1.
n	The ideality factor of diode.
n_c	The number of classifiers.
n_m	The number of classes in the dataset.
N_p	The number of PVMs in parallel.
N_s	The number of PVMs in series.
P	The power.
P_{dc}	The maximum power obtained from the PVA.
P_i	The current position.
P_{mp}	The anticipated sandia model.
P_{pmax0}	The nominal power at the MPP.
P_{ref}	The MPP generated by the PVA.
q	The electric charge.
$Q_{penguin}$	The heat transmission from the penguin to the environement.
R_s	The series resistance.
R_{sh}	The shunt (paralell) resistance.
$R0$	The radius of the first hypersphere.
$R1$	The radius of the second hypersphere.
T	The temerature.
T_{amb}	The ambiant temerature.
T_c	The cell temerature.
T_{stc}	The temerature at STC.
T_s	The absolute temerature.
V	The voltage.
V_{in}	The output voltage of the inverter.
V_{meas}	The measured voltage.
V_{mpp}	The maximum power point of voltage.
V_{mp-STC}	The maximum voltage at STC.
V_{oc}	The voltage of open circuit .
V_{pmax0}	The nominal voltage at the MPP.
V_{pv}	The output voltage of PVM, PVA, or PV cell.
V_{sim}	The simulated voltage.
v_0	The initial velocity.

x	The distance.
x_k	The x coordinate for each penguin.
(x_0, y_0, z_0, t_0)	The center coordinate of the first hypersphere.
(x_1, y_1, z_1, t_1)	The center coordinate of the second hypersphere.
$(x_1, x_2, ...x_N)$	The coordinate of the chosen point.
Y_r	The measure reference yield.
Y_a	The array energy.
y_k	The y coordinate for each penguin.
Y_{real}	The real class labels.
Y_{label}	The predicted labels.
$\delta(T_c)$	The thermal voltage.
α_{mp}	The temperature coefficient for I_{mp}.
β_{mp}	The temperature coefficient for V_{mp}.
θ	The ramp angle with the horizon.
ϵ	The emissivity of bird's plumage.
μ_k	The kinetic coefficient of friction.
μ_{kmax}	The maximum value of friction.
μ_{kmin}	The minimum value of friction.
φ	The mutation factor.

LIST OF ABBREVIATIONS

AC	Alternative Current
AIT	Artificial Intelligence Techniques
ANN	Artificial Neural Network
CNN	Convolution Neural Network
DT	Decision Tree
DC	Direct Current
ECM	Earth Capacitance Measurement
ELI	Electroluminescence Imagining
EPC	Emperor Penguins Colony
FDD	Fault Detection and Diagnostics
FN	False Negative
FP	False Positive
GCPV	Grid Connected Photovoltaic
GPC	Giza Pyramids Construction
IMI	Infrared/Thermal Imagining
IR	Infrared
KNN	K-Nearest Neighbor
LCR	Inductance Capacitance and Resistance meter
LIT	Lock-In Thermography
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracker
NB	Naive Bayes
PCA	Principal Component Analysis
PELA	Power and Energy Losses Analysis
PNN	Probabilistic Neural Network
PV	Photovoltaic
PVM	Photovoltaic Module
PVA	Photovoltaic Array
PVS	Photovoltaic System
RF	Random Forest
RMSE	Root Mean Square Error
SAPV	Stand Alone Photovoltaic

SDM	Single Diode Model
SML	Supervised Machine Learning
SVM	Support Vector Machine
SSPA	Statistical and Signal Processing Approaches
TDR	Time Domain Reflectometry
TN	True Negative
TP	True Positive
VCM	Voltage and Current Measurement
VI	Visual Inspectation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Context:

The importance of photovoltaic systems (PVS) is to effectively respond to the greatest challenges currently facing the world. Providing energy to a growing world population is the first challenge [5–7]. The second challenge is to produce this energy without harming the environment or contributing to climate problems such as global warming [8–10]. The sun is in fact the main source of energy on our planet. Photovoltaic energy sources are also sustainable and renewable.

Problem statement:

Photovoltaic systems are valuable and important because they produce clean, sustainable electricity, but they are also prone to various faults. These faults would have a negative impact on photovoltaic systems because they would reduce the amount of electrical energy produced, deteriorate the performance of the systems and shorten their life cycle [11–15].

Various elements can influence the performance of PVS, including tilt angle [16–20], dust accumulation, humidity, air speed [21], and other types of faults [22, 23]. It is therefore crucial to focus on real-time detection and diagnosis of faults to minimize their impact. Dangerous faults such as open circuit, short circuit and partial shading on the DC side can have significant consequences.

The increasing emphasis on fault detection and safety in photovoltaic systems has stimulated the development of many fault detection and diagnosis (FDD) methods. Key attributes that can define these methods include their ability to quickly identify malfunctions, necessary input data (including climate and electrical data), and selectivity, which refers to their ability to differentiate between different types of faults. Fault detection and classification techniques can be classified into two main groups: visual and thermal methods (VTM) and electrical based methods (EBM)[24].

visual and thermal methods are employed for detecting a range of faults, including hot spots, browning, discoloration, surface soiling, and more [25, 26]. On the other hand, electrical based methods are utilized in various components of a PVS, whether

on the DC or AC side, to identify faults like grounding faults, arc faults, diodes faults, inverter faults, and so forth.

The implementation of visual and thermal methods is considered expensive compared to electrical based methods due to the need for additional expensive devices such as LCR meters, thermal cameras, etc., for fault detection and diagnosis [27, 28]. visual and thermal methods and electrical based methods can be subdivided into more specific categories. Subcategories of visual and thermal methods include infrared/thermal imaging (IMI), visual inspection (VI), electroluminescence imaging (EI), locked-in thermography (LIT), and hybrid methods. Concerning electrical based methods, there are five subcategories including statistical and signal processing approaches (SSPA), IV characteristics analysis, power and energy loss analysis (PELA), voltage and current measurement (VCM) and artificial intelligence techniques (AIT) [29].

Objectives:

Considering the previous discussion, this thesis will present two innovative algorithms for detecting and diagnosing faults in photovoltaic systems based on K-nearest neighbor (KNN) and decision tree (DT) methods.

KNN and DT algorithms are known for their simplicity, ease of understanding and interpretation, and easy implementation in real-world applications, as well as their commendable performance in many cases. The algorithms developed in this work aim to maintain these advantages and further improve their performance.

The modifications introduced to the used algorithms could contribute to a significant improvement in the efficiency of detecting and diagnosing faults on the direct current (DC) side of photovoltaic systems. Additionally, these modifications could expedite the fault diagnosis process, enabling real-time fault detection and diagnosis.

To conclude on the efficiency of the developed algorithms, a comparative study, considering other machine learning algorithms such as support vector machine (SVM), KNN, DT, and random forest (RF), must be carried out.

Thesis organization:

The overall thesis is divided into four distinct chapters:

The first chapter explores the topic by conducting a comprehensive examination of PVSs and their various fault types. Subsequently, a review of existing literature is undertaken to assess the methods that have previously been suggested for detecting and diagnosing faults in these systems. Finally, an overview of data acquisition and monitoring systems for PVS is provided.

The succeeding chapter exclusively focuses on the one-diode model of a solar cell and its parameter identification. Following this, the procedure to identify the five electrical

parameters of this model using two metaheuristic optimization algorithms, namely giza pyramid construction (GPC) and emperor penguins colony (EPC), is provided.

Chapter three is dedicated to presenting the first algorithm developed in this work for detecting and diagnosing faults in the DC side of photovoltaic systems. It presents and discusses the obtained results using this algorithm and several other machine learning algorithms considered for the sake of comparison.

Chapter four introduces the second algorithm developed in this study for detecting and diagnosing faults in the DC side of photovoltaic systems. This algorithm relies on the Decision Tree algorithm. It presents the main steps of the detection and diagnosis procedure design, along with the results of the carried-out comparative study, which are given and discussed.

In the conclusion chapter, some findings and future outlooks are pointed out.

List of publications :

- Y. Mouleloued, K. Kara, and A. Chouder, "A developed algorithm inspired from the classical knn for fault detection and diagnosis pv systems," Journal of Control, Automation and Electrical Systems, vol. 34, no. 5, pp. 1013–1027, 2023.
- Y. Mouleloued, K. Kara, and A. Chouder, "Parameters extraction of photovoltaic module using Giza pyramid construction optimization algorithm," in 2022 2nd International Conference on Advanced Electrical Engineering (ICAEE), pp. 1–6, IEEE, 2022.

CHAPTER 1

STATE OF THE ART IN FAULT DETECTION AND DIAGNOSIS IN PHOTOVOLTAIC SYSTEMS

1.1 Introduction

Detecting and diagnosing faults in photovoltaic systems are critical for delivering sustainable and efficient power. The efficiency of photovoltaic installations depends on the early detection and diagnosis of faults, as even a simple fault can significantly reduce the production of electrical power. The recent advancements in artificial intelligence techniques have enabled the development of high-quality fault detection and diagnosis algorithms, resulting in improved performance and the realization of the full potential of photovoltaic systems.

The aim of this chapter is to provide a review of significant methods for fault detection and diagnosis in photovoltaic systems available in the literature. To achieve this, the chapter begins by introducing the main components of photovoltaic systems and outlining the most common faults associated with these systems.

1.2 Fundamentals of photovoltaic systems

1.2.1 Photovoltaic system component

1.2.1.1 Photovoltaic generators

A photovoltaic generator is composed of interconnected photovoltaic modules (PVM), forming a unit that produces a given electrical power.

A PVM in figure 1.1 is a collection of photovoltaic cells connected both in parallel and series configurations to produce higher voltages, currents and power levels. Grouping cells in parallel increases output current, while grouping them in series increases output voltage. A PVM is the fundamental component of any PVS, converting sunlight directly into direct current through the photovoltaic effect.

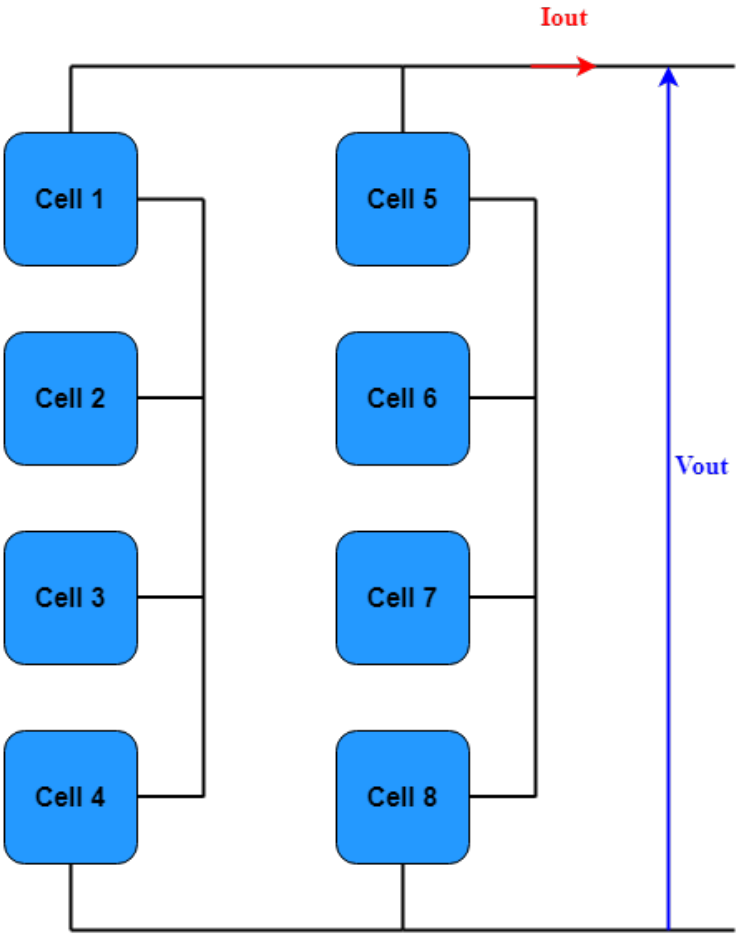

Figure. 1.1: Photovoltaic module of eight cells

We refer to a collection of PVM connected in series as a photovoltaic (PV) string. The desired output voltage that the user/customer wants to achieve determines the number of modules that should be used in series. A solar or photovoltaic panel comprises one or more pre-wired PVMs, ready for installation. Depending on the specific usage requirements, these modules/panels are arranged in a series-parallel configuration to form a photovoltaic array (PVA), ensuring the attainment of the desired voltage and current.

1.2.1.2 Solar charge controller

The solar charge controller or regulator serves two main purposes. Firstly, it prevents solar energy from overcharging and discharging batteries. During hot sunny days, when the PV generator produces more power than the batteries can store, overcharging issues may occur. In such cases, the solar charge controller is necessary to interrupt the power flow from the PV generator to the batteries. Conversely, at night when the PV generator cannot power appliances, discharge problems may arise. In this scenario, all loads draw their energy from batteries, potentially causing discharge if demand exceeds supply. A solar charge regulator is essential to disconnect the batteries from the appliances to prevent this problem[30, 31] .Another function, provided by certain charge controllers, incorporates a maximum power point tracker (MPPT) capable of swiftly and accurately tracking maximum power point (MPP), thus ensuring that the operating point of the PV panels remains at the MPP for the majority of the time[31].

1.2.1.3 Converters

DC-AC converter

The DC-AC power converter, also known as an inverter, plays a crucial role in converting DC power into alternating current (AC) power. This conversion is essential for two main purposes: firstly, to supply power to AC appliances, and secondly, to synchronize the AC signal generated by the inverter with the AC signal from the grid.

In the grid-connected photovoltaic (GCPV) system, inverters typically incorporate the MPP tracker. However, this feature is not present in the stand-alone photovoltaic (SAPV) system.

Four different topologies can be used to implement inverters in PVSs.

- Central inverter

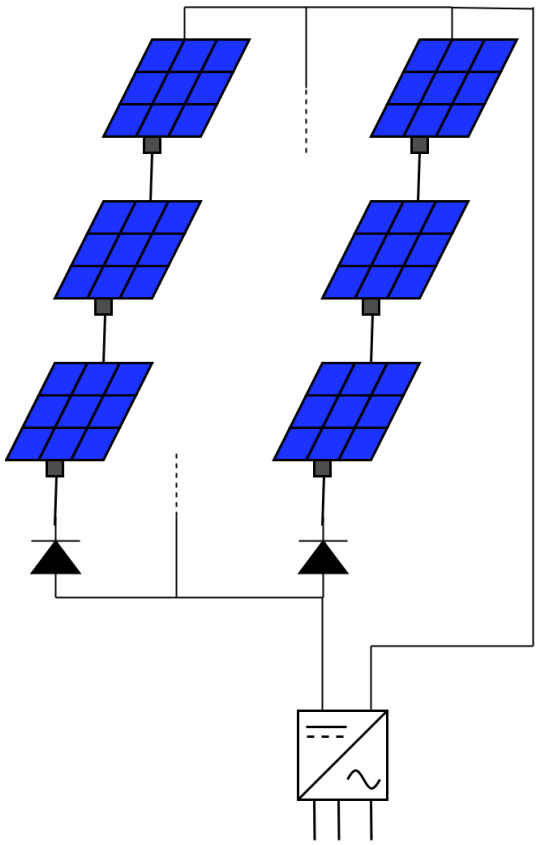

Figure. 1.2: Central inverter configuration

The central inverter configuration is the most commonly used topology in modern PVSs. As shown in figure 1.2, only one inverter is employed for multiple PVM. This makes it straightforward to design and deploy. Nonetheless, the central inverter topology has several drawbacks: As the system grows in size, more DC wires are required to reach the inverter, increasing the wiring cost while diminishing safety. Not to mention the inverter's inability to track the MPP in the case of shadowing.

- Module inverter

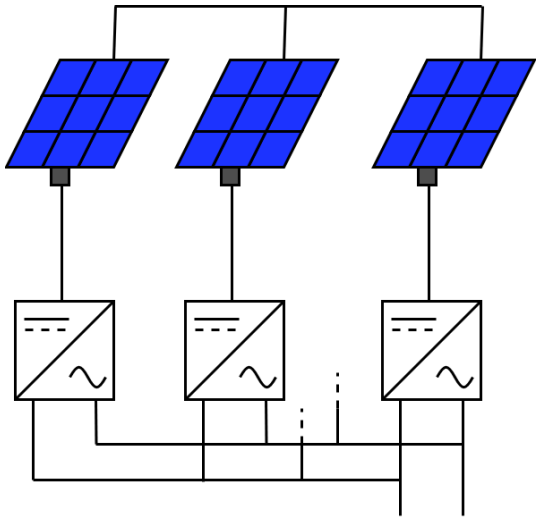

Figure. 1.3: Module inverter configuration

As is clear in figure 1.3, each module has its own dedicated inverter to efficiently invert the DC power and track the MPP. It is significantly easy to expand the system size while using this topology. However, as the system size grows, so does the implementation cost.

- String inverter

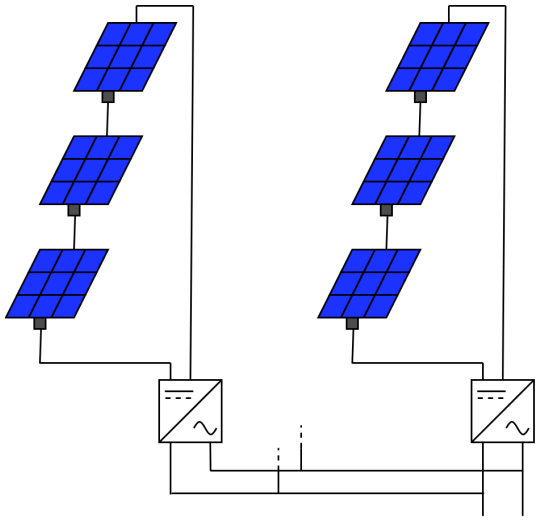

Figure. 1.4: String inverter configuration

The topology of figure 1.4 seeks to strike a balance between the two previously stated topologies. Because each PVS operates independently at its MPP, it

ensures higher energy yields. This means that the string inverters run at their MPP more precisely than the central inverter, but less precisely than the module inverter. The implementation is more difficult than that of the module inverter.

- Multi-string inverter

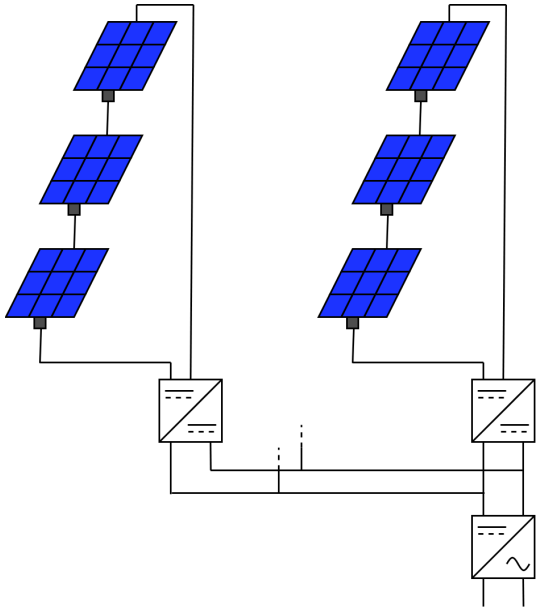

Figure. 1.5: Multi-string inverter configuration

This topology accomplishes two objectives:

1- Produce more energy as the same as string inverter topology.

2- Lower implementation cost when compared to the central inverter topology.

Each PV string pre-power is processed utilizing low DC-DC converters, as shown in figure 1.5. MPPTs are also deployed in conjunction with each converter.

DC-DC converter

A DC-DC converter, also known as a chopper, is a power electronic device linked to the MPP tracker [32]. It incorporates inverters and, in certain instances, a solar charge controller. A DC-DC converter accepts a variable DC signal as input and outputs a fixed DC signal.

1.2.1.4 Batteries

The most expensive part of the PVS is the battery. Particularly in the grid-off system, they are essential. In order to increase the total capacitance of PVSs, engineers typically use multiple batteries. It is known as a "bank of batteries". They serve as a form of energy storage, supplying electricity to loads at night or inclement weather.

1.2.1.5 Distribution panel

The distribution panel (distribution board)is a part of a grid-connected system that is used to determine whether the power coming from the inverter is enough to run the appliances or whether it is too much. In other words, the inverter's output is directed to power the devices by the distribution board, and the remaining energy is sent straight to the grid.

1.2.2 Photovoltaic system configuration

The photovoltaic system has three main configurations:

1.2.2.1 Grid connected photovoltaic system

This configuration allows the integration of a PVS with the electrical grid. It is commonly employed in regions where grid connectivity is available. Figure 1.6 provides a more detailed comprehensive illustration of the GCPV system. As observed, the balance of system plays a crucial role in regulating the energy flow

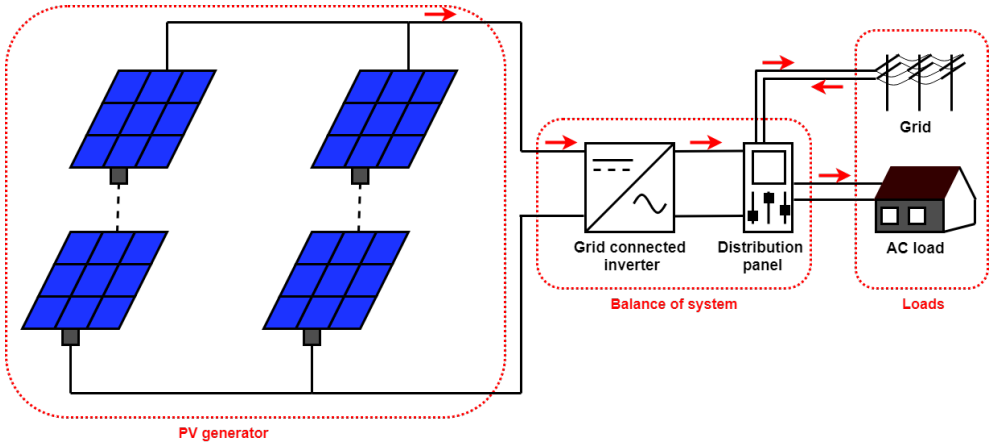

Figure. 1.6: Grid connected photovoltaic system configuration

1.2.2.2 Stand-alone photovoltaic system

A Stand-alone photovoltaic system operates independently of grid connection and is commonly utilized in areas lacking grid infrastructure, including remote villages, rural regions, and urban areas. This configuration is illustrated in figure 1.7.

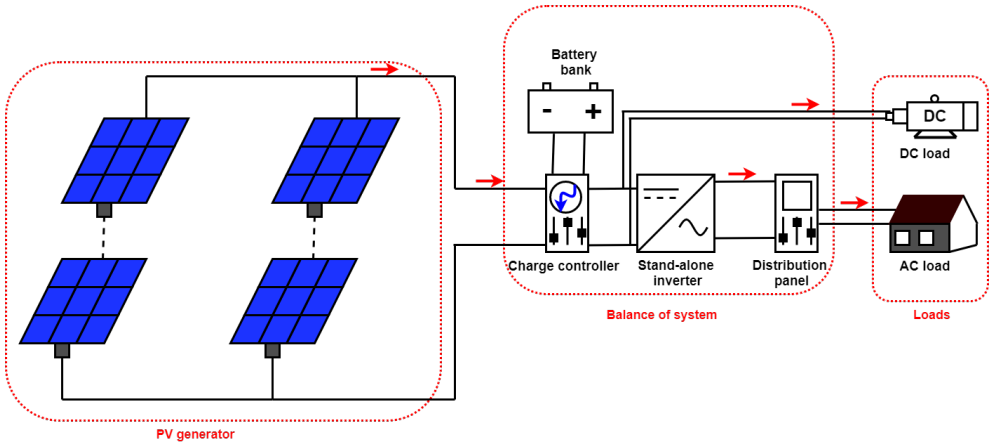

Figure. 1.7: Stand alone photovoltaic system configuration

1.2.2.3 The hybrid photovoltaic system

This configuration combines elements from the previous setups, incorporating both the SAPV system and the GCPV system, as depicted in figure 1.8.

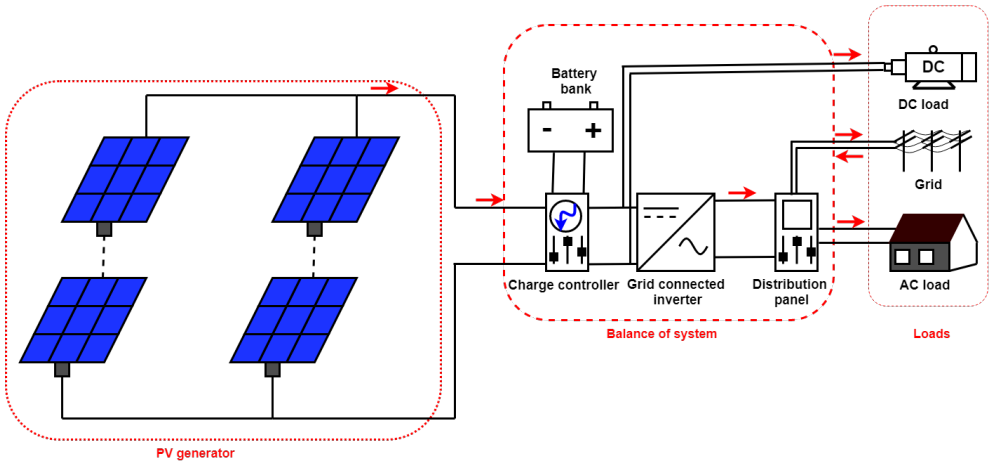

Figure. 1.8: Hybridization photovoltaic system configuration

1.2.3 Reliability of photovoltaic system

Before investing their money, investors always seek to assess the reliability of new technology provided by engineers. The field of renewable energy in general, and specifically solar energy, has seen massive investments in recent years as a result of growing confidence in these technologies and their consideration as a promising option for meeting sustainable energy needs.

The reliability of photovoltaic systems is primarily connected to the system's ability to generate electrical power permanently and continuously over its expected lifespan of 25 to 30 years. Among the factors that contribute to the long-term reliability of solar systems are the quality of the system components, as well as simple periodic maintenance, such as cleaning the solar panels and checking the electrical cable connection [33].

1.3 Common faults in photovoltaic systems

Similar to any other system, the performance of a photovoltaic system will degrade over time due to potential operational faults. Figure 1.9 illustrates the common faults in PVSs. These faults are classified into two main categories: DC side and AC side, as depicted in the figure. Faults colored by blue are extensively discussed.

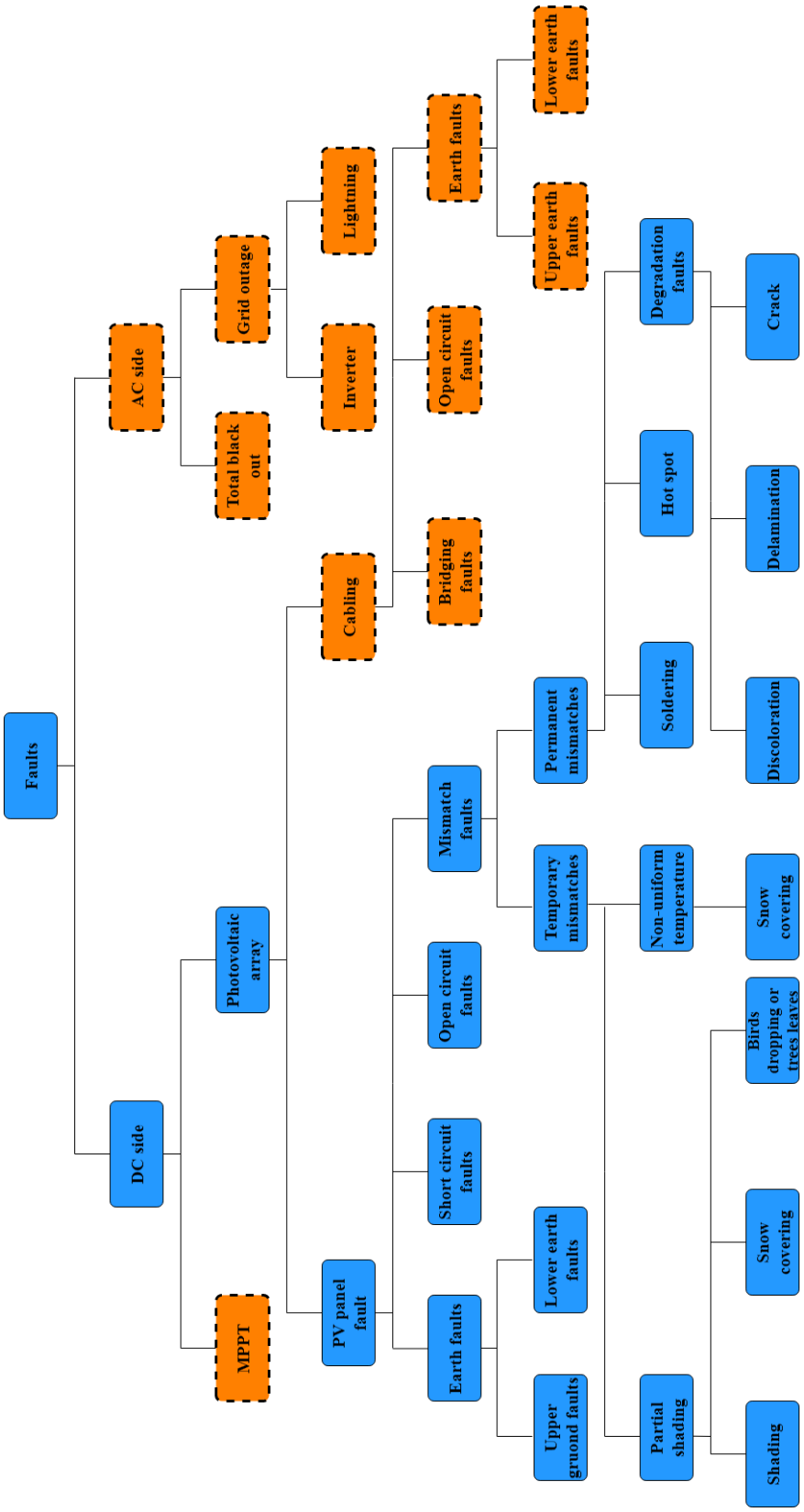

Figure. 1.9: DC and AC side faults in PV systems.

1.3.1 Faults in DC side

The common faults on the DC side include the missing maximum power tracking fault and a collection of faults at the PV generator level.

1.3.1.1 MPPT fault

Due to the ongoing changes in weather (temperature and radiation), PV generators provide variable DC power. However, appliances typically require a continuous power supply over time. In this case, a DC-DC converter comes in handy. The converter can deliver a constant amount of power at the highest level it can ever produce, thanks to an algorithm integrated into its operation. Every time the weather changes, this algorithm seeks to find the maximum power that a PV generator can deliver. There is an MPPT fault if the algorithm cannot accurately determine the maximum power.

1.3.1.2 photovoltaic generator faults

Figure 1.10 illustrates four distinct fault types that can occur at the PV generator level: mismatch fault, short circuit fault, open circuit fault, and ground fault.

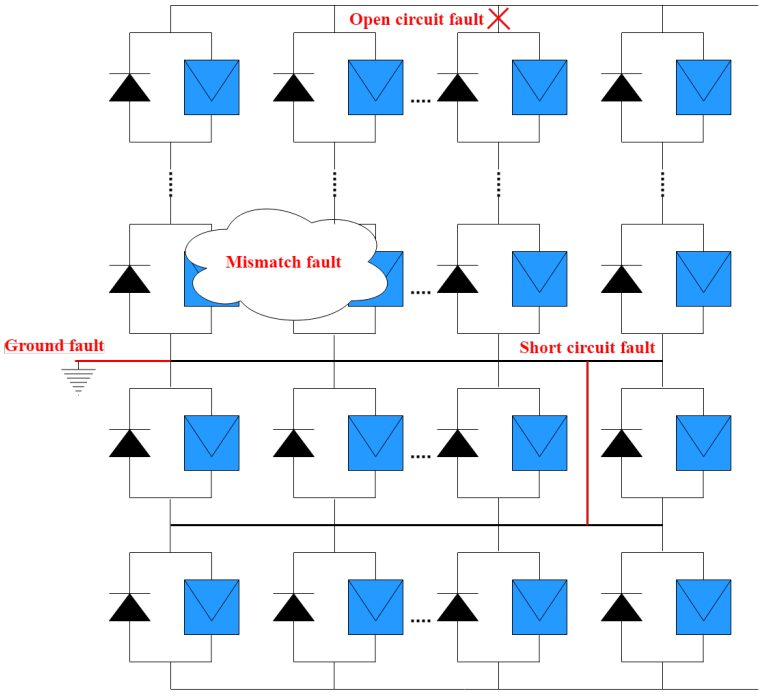

Figure. 1.10: PV generator faults

Mismatch fault

Two examples of the mismatch fault exist:

- When a PV generator comprises multiple cells or modules with varying electrical properties (I_{sc}, V_{oc}, ...etc).
- When a PV generator is made up of multiple cells or modules working in various conditions (temperature and radiation).

This type of fault is divided into two classes based on the varying weather conditions on the PV generator:

1- **Temporary mismatch:** When snow covers the PV generator, the temperature across its surface is not uniform. Additionally, a PV generator may occasionally be partially shaded by clouds, desert dust, bird droppings, tree leaves, etc.

2- **Permanent mismatch:** This type of fault includes soldering, hotspot, and degradation faults:

   a. **Soldering fault:** This kind of fault happens when assembling a PVM by soldering a collection of solar cells. Unprofessional soldering of solar cells may result in them becoming disconnected from each other.

   b. **Hotspot fault:** This type of fault is caused by the PVM's operation under partial shade. The solar cell generates power when it is in its normal state of operation, making it an active component. The solar cell turns into a passive component in cases of partial shading, absorbing and dissipating energy from other solar cells (in series). The result is that the shaded solar cell will heat up to the point of collapse. A parallel diode referred to as a "bypass diode" is needed to solve this problem.

   c. **Degradation faults:** PVM performance can degrade for two main reasons: when the PVM reaches the end of its life cycle, which can occur at 20, 25, 30 years, etc., and when some of the solar cells malfunction. the gradual reduction in output power brought on by raising or lowering the shunt resistor R_{sh}.

Short circuit fault

Figure 1.10 provides an illustration of a short circuit fault, also known as a line-line fault. It is essentially an accidental low-resistance connection occurring within a PVS between two points with differing potentials (excluding the ground point) [34].

Open circuit fault

The open circuit fault at the module level is shown in figure 1.10. It happens whenever a connection between two modules connected in series is harmed or broken. This error can also happen at the solar cell level if the connection holding two solar cells in series together is compromised.

Ground fault

In PVA , a ground fault is an unintentional electrical short circuit involving ground and one or more typically designated current-carrying conductors. It could be caused by a cable in a PV junction box accidentally contacting a grounded circuit [35].

Bridging fault

Bridging fault occurs when the connection between two PVMs obtain more resistivity [36].

1.3.2 Faults in AC side

As shown in figure 1.9, AC side faults can be classified as total blackouts or grid outages.

1.3.2.1 Total blackout

Total blackout fault in PVSs is the condition in which the system completely fails to produce electrical power due to natural disasters such as storms and lightning [37].

1.3.2.2 Inverter fault

One of the major problems that arises on the AC side level is improper inversion from DC to AC. There are several factors that can cause an inverter to fail, including insufficient heat dissipation in switch components and failure of capacitors [36].

1.3.2.3 Grid fault

A grid fault causes a loss of electrical power because the PVS is unable to draw power from the grid. This occurs when solar panels are unable to supply the necessary electricity to the devices. This type of fault also arises when extra energy is generated by the PVS but cannot be returned to the grid.

1.3.3 Impact of some faults on photovoltaic performance

The negative impact on the PVS performance under various fault conditions, such as short circuit, open circuit, and shading, has been demonstrated in [1].

1.3.3.1 Impact of short circuit fault on photovoltaic performance

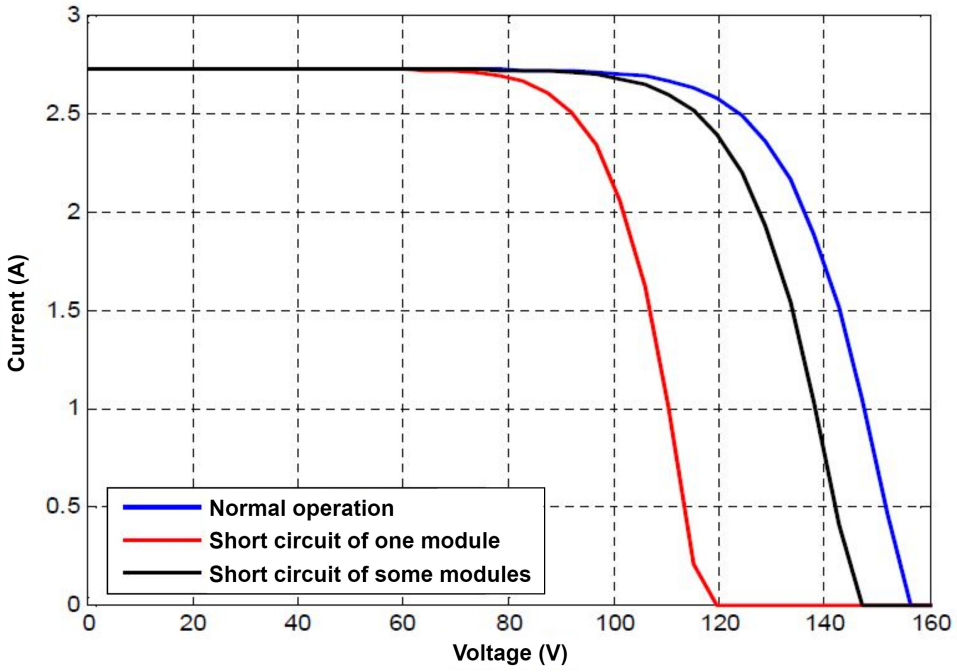

Figure. 1.11: I-V characteristic of PV generator under normal operation and short circuit of PVM and a group of PV cells [1]

Figure 1.11 depicts the I-V characteristics of solar panels in the absence of faults as well as in the presence of short circuit faults in cell and module levels.

The voltage of the solar panels is decreasing, yet the current remains steady. This is due to the fact that when solar panels are shortened, this leads to a lack of voltage across these panels, which explains the reduction in voltage. The more solar panels that are shorted out, the lower the voltage becomes.

1.3.3.2 Impact of open circuit fault on photovoltaic performance

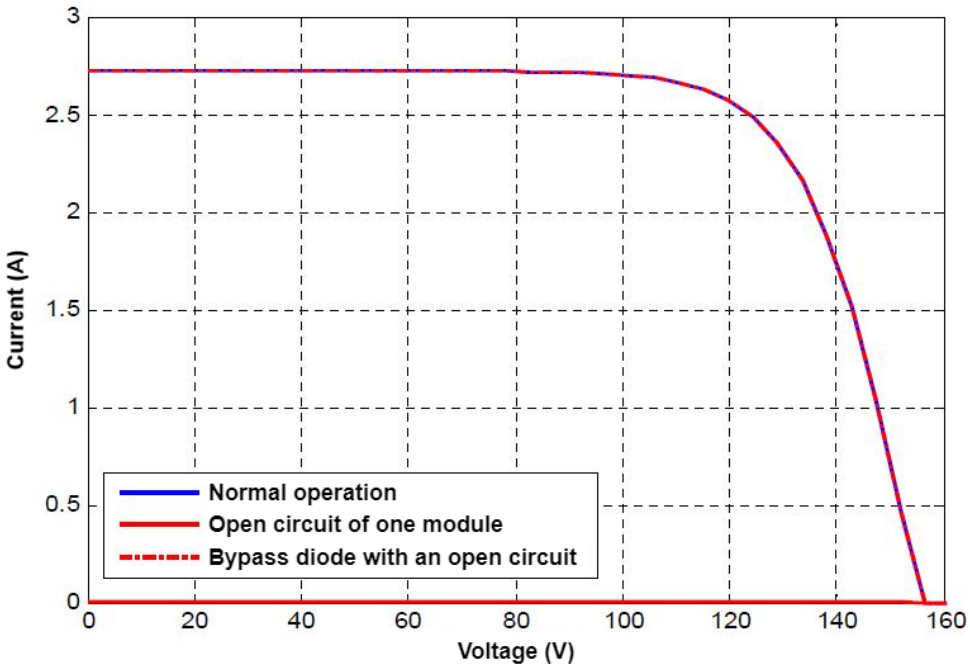

Figure. 1.12: I-V characteristic of PV generator under normal operation and open circuit of PVM with and without bypass diode [1].

Figure 1.12 displays the I-V characteristics of solar panels under normal conditions, as well as when an open circuit fault is present, both with and without the inclusion of a bypass diode. No current circulates through the circuit, regardless of the tension applied across the solar panels, because current cannot pass through an open circuit. Bypass diode has no role in the operation, unless the PV generator functions under shading issue.

1.3.3.3 Impact of shading fault on photovoltaic performance

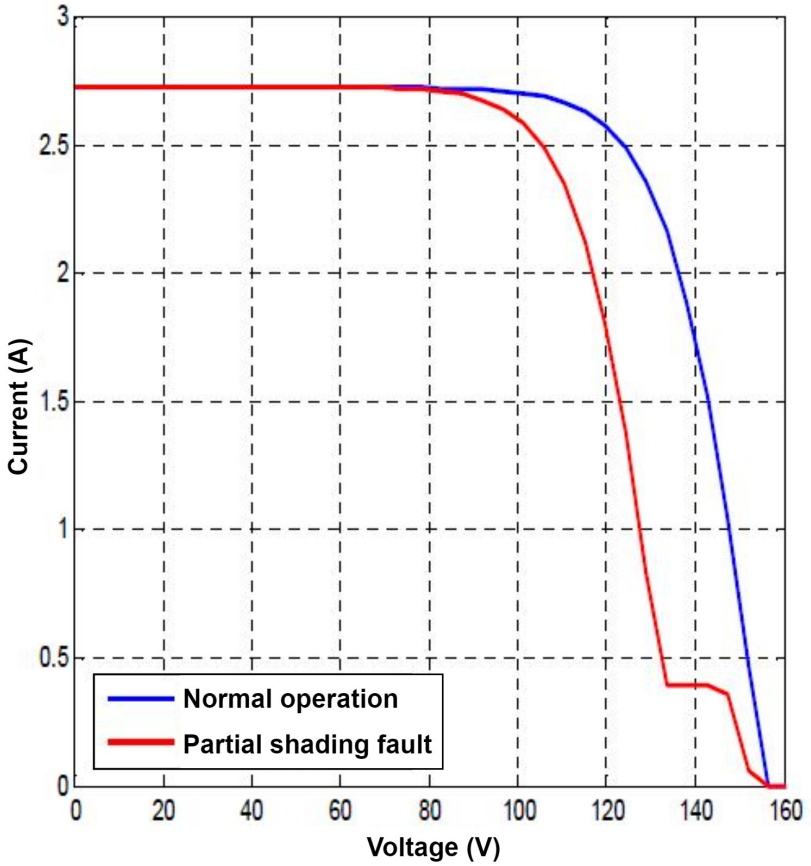

Figure. 1.13: I-V characteristic of PV generator under normal operation and partial shading of one module [1].

When PVMs are exposed to sunlight, they produce electrical power. PVM shading limits the amount of sunlight reaching the panels. The effect of shading on the performance of PVMs is manifested by a deterioration of the electrical power produced, resulting in a decrease in system efficiency and the possibility of hot spot damage. A parallel bypass diode is utilized with each PVM to mitigate the influence of shade on PVM performance.

Figure 1.13 depicts the evolution of the I-V characteristic of a shaded PVM. When a PVM is shaded, the bypass diode activates and creates a small resistive path for electrical current to pass through, causing the PVM to become short-circuited, resulting in an inflection point in the I-V characteristic [1].

The electrical power of a PVA is essentially the sum of the electrical power generated by each PVM. Typically, a PVA provides electrical power that closely aligns with the

desired electrical power. Any deviation from this expected power output is considered a 'fault,' signifying a reduction in the generated electrical power value. Identifying the source of this deterioration is crucial for enhancing the long-term reliability of the PVS and maximizing electrical power generation. Consequently, numerous techniques and algorithms have been developed to detect and diagnose faults early in the maintenance process. The upcoming section will explore many of these techniques, along with their respective benefits and drawbacks [37].

1.4 IV. Detection and diagnosis methods of faults in photovoltaic systems

A wide range of methods and algorithms for detecting and diagnosing faults in PVSs are presented in the literature. To operate effectively and make correct decisions, these algorithms require several measurements: temperature (T), irradiance (G), current and voltage at the MPP (I_{mpp}, V_{mpp}), output current and voltage of the PVM (I_{pv}, V_{pv}), inverter current and voltage (I_{in}, V_{in}), and the I-V characteristic. To compare and distinguish between these methods, several criteria are used such as:

To compare and distinguish between these methods, several criteria are used such as:

- The method ability in detecting faults and abnormal states during the system functionality.
- The method ability in accurately identifying faults.
- The robustness of the algorithm against noisy data.
- The number of faults can be detected and identified.

Detection and diagnosis methods for faults in PVSs can be categorized into two main global categories: electrical methods and visual & thermal methods [16, 17]. Each of these categories can be further subdivided. Figure 1.14 represents a summary classification of the most discussed detection and diagnosis methods in the literature.

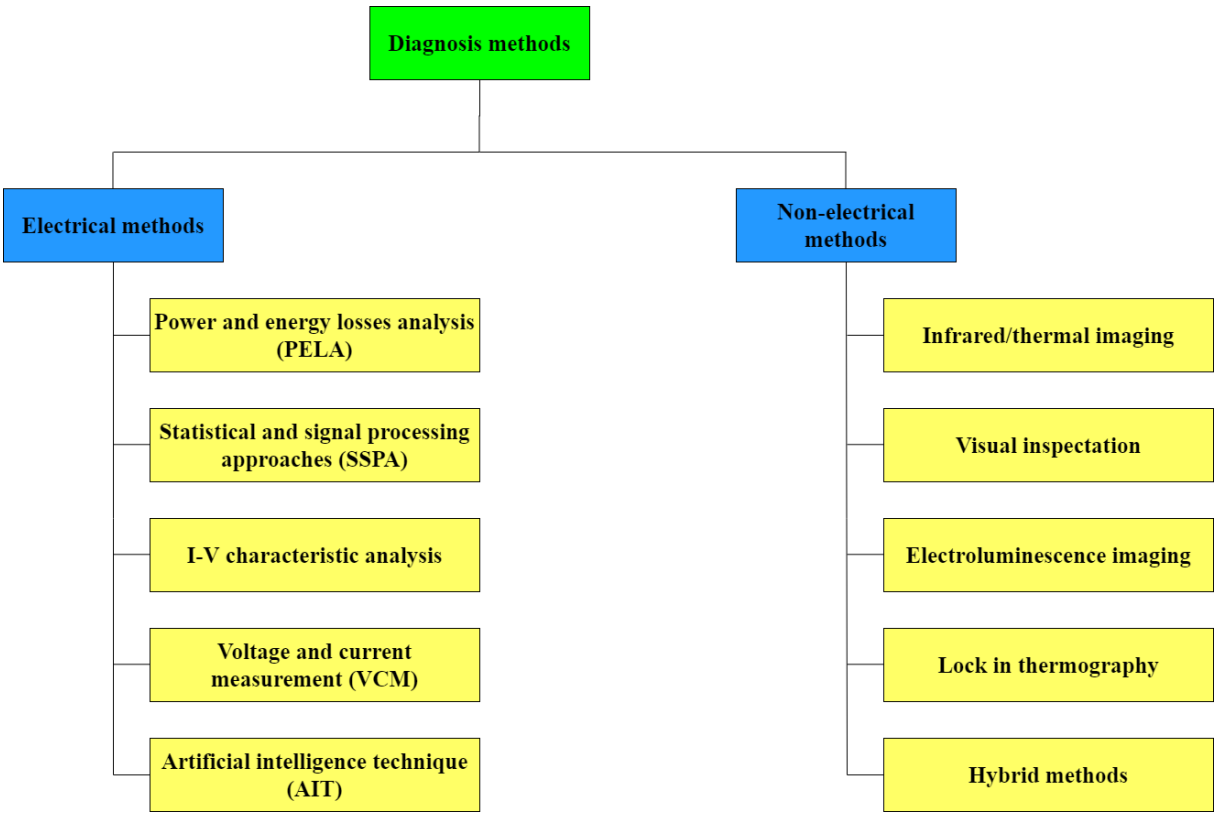

Figure. 1.14: categories of diagnosis methods

1.4.1 Visual & thermal methods (Non-electrical methods)

1.4.1.1 Infrared/ thermal Imaging

Infrared/thermal imaging is employed to detect and localize various faults in PVMs, including junction boxes, connectors, etc [18]. IMI utilizes an infrared (IR) camera to detect the heat generated by the PV generator during both normal and abnormal operations, such as shunted cells and short circuits[19].The advantage of IMI lies in its avoidance of sensor installation, making its implementation as a diagnostic method more cost-effective compared to other approaches[18].IMI can be applied, from small to large PVMs [18, 20].However, this method has not been adapted to detect all types of faults, or at least most of them, including open circuit faults.

1.4.1.2 Visual Inspectation

Visual Inspectation is conducted through human intervention to monitor the performance of PVMs. In this process, various symptoms that may occur in PVMs are

monitored, allowing the detection and diagnosis of certain faults such as bird droppings, degradation, cell cracking, etc[21].

1.4.1.3 Electroluminescence Imaging

Electroluminescence Imaging (ELI) is an effective method for detecting faults, especially micro cracks that are challenging to notice with the naked eye. This process involves capturing high-resolution images of PVMs, revealing faults through lines or black spots [22], such as cell cracks and poor connections. In the study referenced in [23], the authors used a drone for fault detection. ELI excels in detecting micro faults; however, it is not suitable for identifying other faults like shading and short-circuit faults.

1.4.1.4 Lock-in Thermography

Lock-in Thermography is commonly used in the failure analysis of integrated circuits [24]. A lower value of the shunt resistor in the solar cell can create an alternative current path for the photovoltaic current, allowing it to pass through as a leakage current. This leakage current produces an amount of heat that can be thermo-graphically monitored [25]. LIT can be used to detect small changes that occur in the PV generator parameters, such as ideality factors and series resistance of the solar cell, using an infrared camera[25].

1.4.1.5 Hybridization methods between visual & thermal methods and artificial intelligence techniques

Many techniques have been presented in the literature that rely on the combination of AIT and visual/thermal methods. For example, in [26], the authors proposed a technique that combines a parameter-based model with LIT for fault detection purposes. They successfully detected and diagnosed faults such as hot spots and cracking. In [27], a technique that combines convolution neural network (CNN) with LTI was proposed. Another hybrid approach, integrating IMI with intelligent edge detection, is presented in [28]. Additionally, [29] introduced a combination of Canny edge detection with image processing for detecting and diagnosing faults such as cracks in PVMs.

1.4.2 Electrical methods

1.4.2.1 Power and energy losses analysis

To facilitate the control of small GCPV system up to 5 KW, a procedure for the early detection of partial energy losses and system abnormal functions has been proposed in [2]. This procedure initially involves detecting failures in the photovoltaic system by comparing the measured energy with the simulated energy. To yield actual energy, a satellite is used to drive irradiance values (figure 1.15).

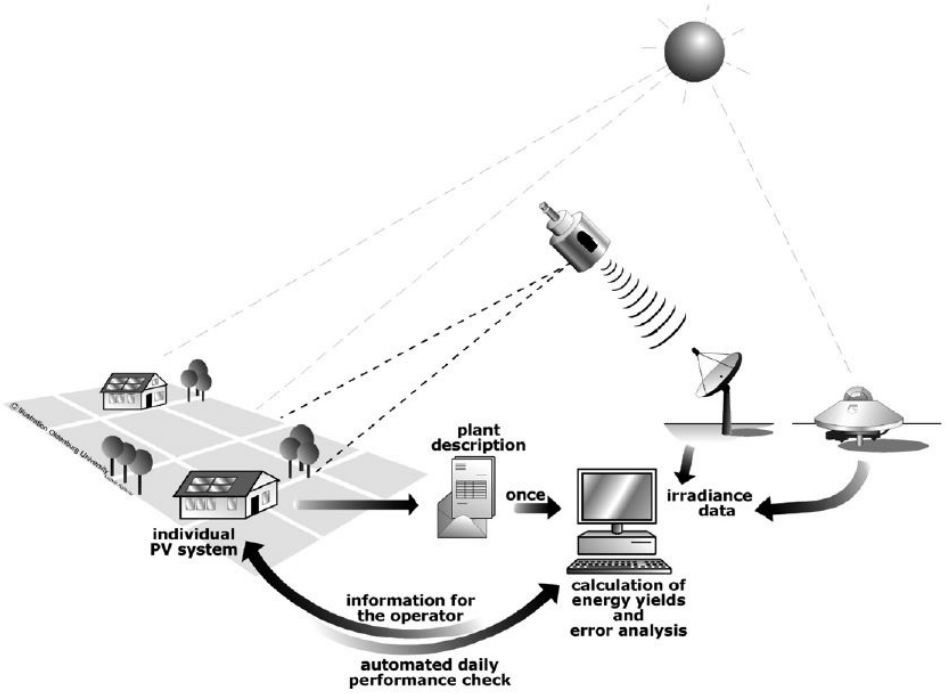

Figure. 1.15: Structure of the proposed procedure [2]

After detecting a fault in the system, the next obvious step is to identify the type of this fault. Table 1 presents faults along with their sources.

Climate data were collected every 5 minutes over two years to develop a model for monitoring energy losses (due to faults) in 27 domestic PVSs located in two different areas [38]. The model monitors the energy losses of the PVSs during normal operation and also when a number of faults occur. The authors developed data analysis techniques that enabled them to detect seven types of faults, including: shading, inverter shutdown, system isolation, inverter MPPT failure, and others. An intelligent model for fault detection in photovoltaic fields was presented in [39]. In this work, fuzzy logic was employed to compute the estimated power output from PV fields,

General failure type	Failure
Constant energy loss	Degradation, Soiling, Module defect, String defect
Changing energy loss	Shading, Grid outage, High losses at low power, Power limitation, MPP tracking, Hot inverter, High temperature
Snow cover	Snow cover
Total blackout	Defect inverter, Defect control devices

Table 1.1: Faults with their sources

and these estimates were compared to the measured power values. If the difference exceeded a predetermined threshold, it indicated a fault in the PV field. The authors utilized this intelligent model for both detecting and diagnosing faults. However, it's important to note that the scope of this work was limited to explaining the method of fault detection.

In [40], an algorithm is proposed for detecting and diagnosing faults in GCPV systems. The algorithm consists of two fundamental steps. The first step involves continuous tracking of energy losses at the solar module level. If the measured values exceed predefined threshold values, it indicates a malfunction in the photovoltaic system. In the second step, faults are diagnosed using voltage and current ratios Thermal capture losses (L_{ct}) and miscellaneous capture losses (L_{cm}) are two novel indicators of power losses defined for the detection phase. Thermal capture losses occur when solar panels operate at temperatures exceeding 25°C. The remaining capture losses manifest in various forms, including solar panel failures, MPP tracking failures, etc.

The following equations represent the basis for calculating losses in thermal capture and losses in other types of capture:

$$L_c = Y_r\left(G, T_c\right) - Y_a\left(G, T_c\right) = \frac{H_i\left(G, T_c\right)}{G_{ref}} - \frac{E_{dc}}{P_{ref}} \tag{1.1}$$

$$L_{ct} = Y_a\left(G, 25°C\right) - Y_a\left(G, T_c\right) \tag{1.2}$$

$$L_{cm} = L_c - L_{ct} \tag{1.3}$$

where:
L_c is the capture losses, L_{ct} is the thermal capture losses, $L_{cm}$is the miscellaneous capture losses, Y_r is the measured energy reference yield, Y_a is the array energy, G is the irradiance, T_c is the cell temperature, H_i is the total in-plan irradiance of the PVA,

E_{dc} is the energy produced by the PVA, P_{ref} is the maximum power output generated by the PVA and $G_{ref} = 100W/m^2$.

Figure 1.16 represents the flowchart of the fault detection procedure, while figure 1.17 shows the flowchart of the diagnosis model.

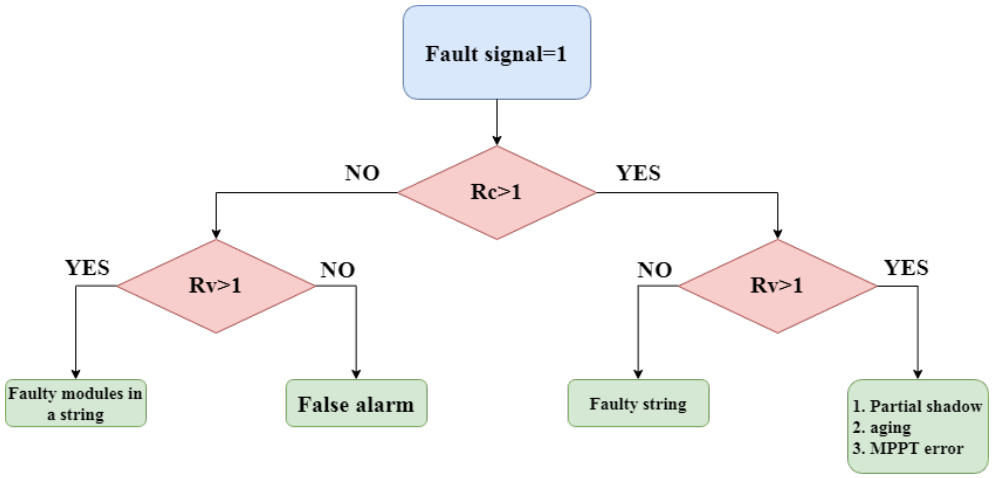

Figure. 1.16: Flowchart of fault detection procedure

R_c and R_v are current and voltage ratios respectively, where:

$$R_c = \frac{I_{PV}^{sim}}{I_{PV}^{meas}} \tag{1.4}$$

$$R_v = \frac{V_{PV}^{sim}}{V_{PV}^{meas}} \tag{1.5}$$

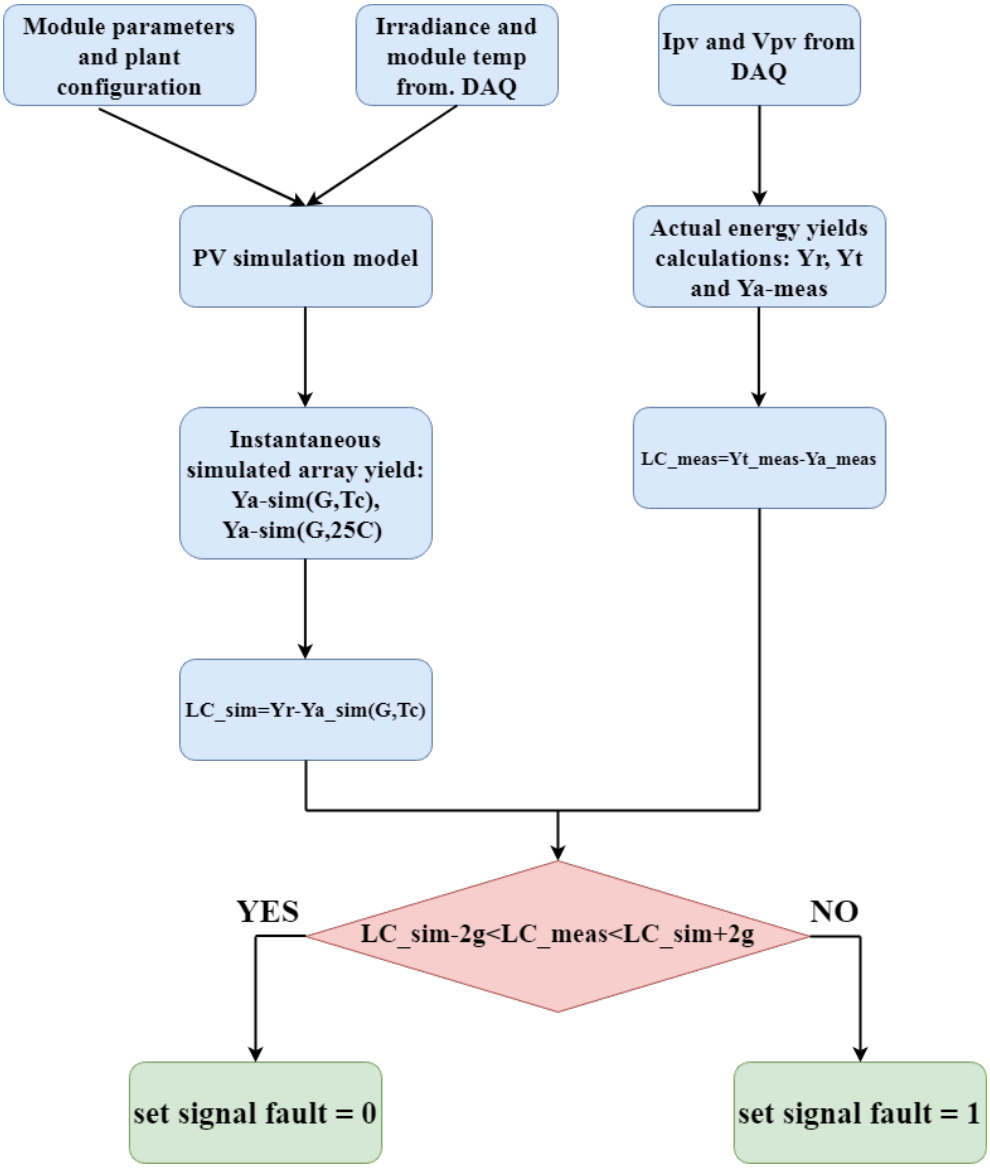

Figure. 1.17: Flowchart of the diagnose method

1.4.2.2 Statistical and signal processing approaches

A method for fault detection based on the Time Domain Reflectometry (TDR) technique was proposed by Takashima et al [41]. The implementation of this method in a PVS is illustrated in (figure 1.18).

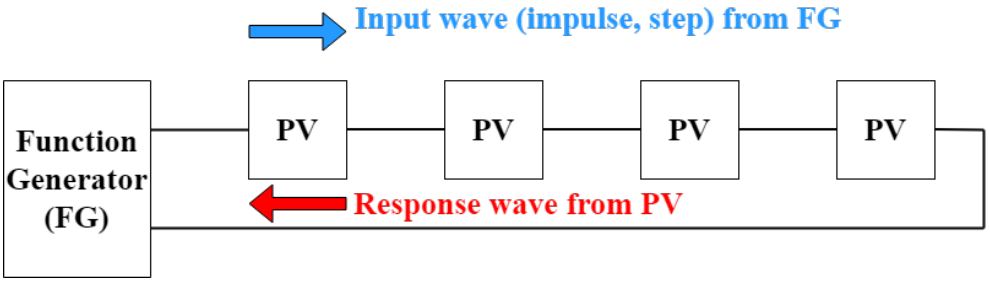

Figure. 1.18: Concept of TDR measurement in PVS

The main principle of this method involves sending an electrical signal (a step or pulse) and comparing it to the signal that is reflected back. The location and type of fault, such as open circuit and short circuit, are identified based on the signal's delay and shape. A number of solar panels were used in series to test the effectiveness of this method.

The disconnection between two PVMs in a PVS can be determined using a method based on the Earth Capacitance Measurement (ECM) technique [42].This method is primarily used to identify power line disconnections, as depicted in (figure 1.19).

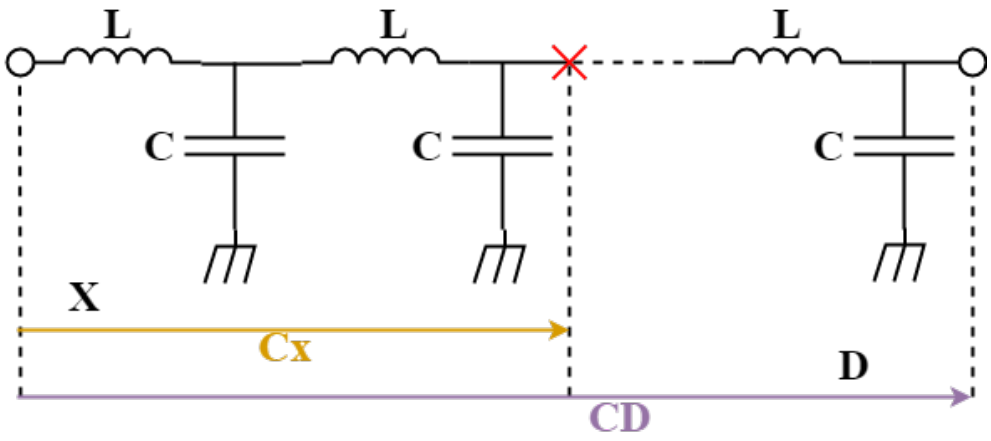

Figure. 1.19: Transmission line model.

The following equation is used by the ECM technique to calculate the distance between the starting point and the cut-off location:

$$x = \frac{C_x}{C_p} D \tag{1.6}$$

where: x is the distance from the starting point to the disconnecting point (m),C_x is the ECM value from the starting point to the cut-off site, C_p is the ECM value for the entire power line and D is the transmission line's length (m). The values of C_x and C_p

are determined using the LCR meter. In their work, the authors have considered the PV string like a transmission line figure 1.20.

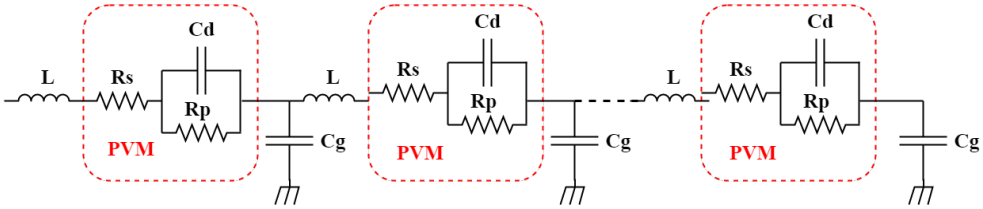

Figure. 1.20: Model of a PV string

When generating energy, the capacitor C_d can be neglected, and accordingly, the PVS can be considered as a transmission line, and therefore the ECM technique can be applied to determine the disconnection between the PVMs according to the following equations:

$$n = \frac{C_{\text{trouble}}}{C_{\text{good}}} M \tag{1.7}$$

$$C_{\text{trouble}} = C_x = n \cdot C_g \tag{1.8}$$

$$C_{\text{good}} = C_p = M \cdot C_g \tag{1.9}$$

where:
C_g is the EMC value of a single PVM, M is the number of PVMs in a given series and n is the number of PVMs between the start and the end of the PVM at which the disconnect occurs.

Two PV string sets have been used to evaluate the performance of this technique. The first set consists of ten PVMs, each with a 130 W power output, connected in series. The second set is made up of ten PVMs, each with a power of 80 W, connected in series. In [43], authors propose a method which integrates a single diode model (SDM) with the expanded capability of an exponentially weighted moving average (EWMA) control chart for early detection of changes in a PVS. The SDM, characterized by its few calibration parameters, is utilized for forecasting the optimal power coordinates of current, voltage, and power based on recorded temperatures and irradiances of the healthy PVA. Residuals, representing the disparities between measurements and SDM predictions, are computed and utilized as fault indicators. Subsequently, the EWMA monitoring chart is employed on the uncorrelated residuals derived from the SDM to detect and classify faults. Performance evaluation of this approach is conducted using real data from a GCPV system situated at the Renewable Energy Development Center

in Algeria. The findings demonstrate the successful monitoring of the DC side of PVSs and the detection of partial shading faults.

1.4.2.3 I-V characteristic analysis

In addition to detecting partial shade, the $\frac{dI}{dV} - V$ characteristic was used by Miwa et al to estimate how many PVMs would be exposed to partial shade [44]. The $\frac{dI}{dV} - V$ property is extracted from PVA after the $I - V$ characteristic has been taken out of it. In the $\frac{dI}{dV} - V$ characteristic, partial shading arises as a simple convexity.

The number of solar panels exposed to partial shade can be partly determined by the convexity's location on the $\frac{dI}{dV} - V$ characteristic. Fewer solar panels exposed to partial shade result in higher convexity tension, while more solar panels exposed to partial shade lead to lower convexity tension.

To identify and locate faults in a PV generator, the authors of [45] utilized three parameters from the I-V characteristic:

- Maximum power: A decrease in this parameter indicates a fault occurrence. To identify this reduction, a straightforward procedure has been employed: measuring the $I - V$ characteristic under any conditions, translating the result to standard circumstances, and then comparing it with the characteristic provided by the manufacturer in the data sheet.

- Series resistance: Based on the slope of the I-V characteristic measured close to the open circuit voltage, its value can be estimated. Poor contact between cells and modules is indicated by an increase in this parameter.

- Temperature: The open circuit voltage is the parameter most impacted by module temperature. The rise in module temperature indicates an insufficient connection between the modules.

1.4.2.4 Voltage and current measurement

A fault detection method for solar arrays based on VCM has been proposed in [46]. In this method, the Hall sensor collects the electric current and voltage signals of each series branch using microcomputer techniques. The difference between the maximum current value and the other values is used to determine the fault. The incorrect solar series branch can then be determined by calculating the deviation of each branch. It can determine the fault section in the branch based on the voltage signal. Finally, a fault notification will be generated for the immediate attention of the maintainer.

1.4.3 Artificial intelligence technique

Many research works have explored the use of Artificial Neural Networks (ANN) to develop efficient algorithms for fault detection and diagnosis in PVSs. In [47], a comparative study between two neural network approaches has been conducted. The two methods require four attributes (T, G, I_{mpp}, V_{mpp}) to make five different decisions (normal operation, short circuit of three modules fault, short circuit of ten modules fault, open circuit fault). The first approach employs a feed forward neural network for fault detection (healthy or faulty state), and a second feed forward neural network for diagnosing the occurred fault. The probabilistic neural network (PNN)-based approach is similar to the first, except it employs a PNN architecture instead of a feed forward ANN. According to the results of this study, the PNN-based method outperforms the ANN-based approach in both fault detection and diagnosis phases, regardless of the presence of noise in the data. Another research work utilizing ANN to propose a method for fault detection and diagnosis in PVSs is presented in [48]. In this work, a simulation model has been used to detect the occurrence of faults, and then two algorithms have been employed to diagnose eight different faults. The first algorithm identifies five faults using different attributes, while the second algorithm, based on ANN, identifies the remaining faults using the same attributes.

The principal component analysis (PCA) algorithm has been employed to extract suitable features for the purpose of fault detection and diagnosis in GCPV systems [49]. Then, various supervised algorithms, such as DT, RF, SVM, naive bayes (NB), and KNN, were utilized to detect and diagnose various faults (inverter fault and grid connection fault on the grid side, PV panel fault, sensor fault, and PV panel connection fault on the PV side).All the algorithms under consideration achieved an accuracy greater than 96%. The diagram below illustrates how the PCA and supervised machine learning (SML) algorithms collaborate to discover and diagnose faults in GCPV system.

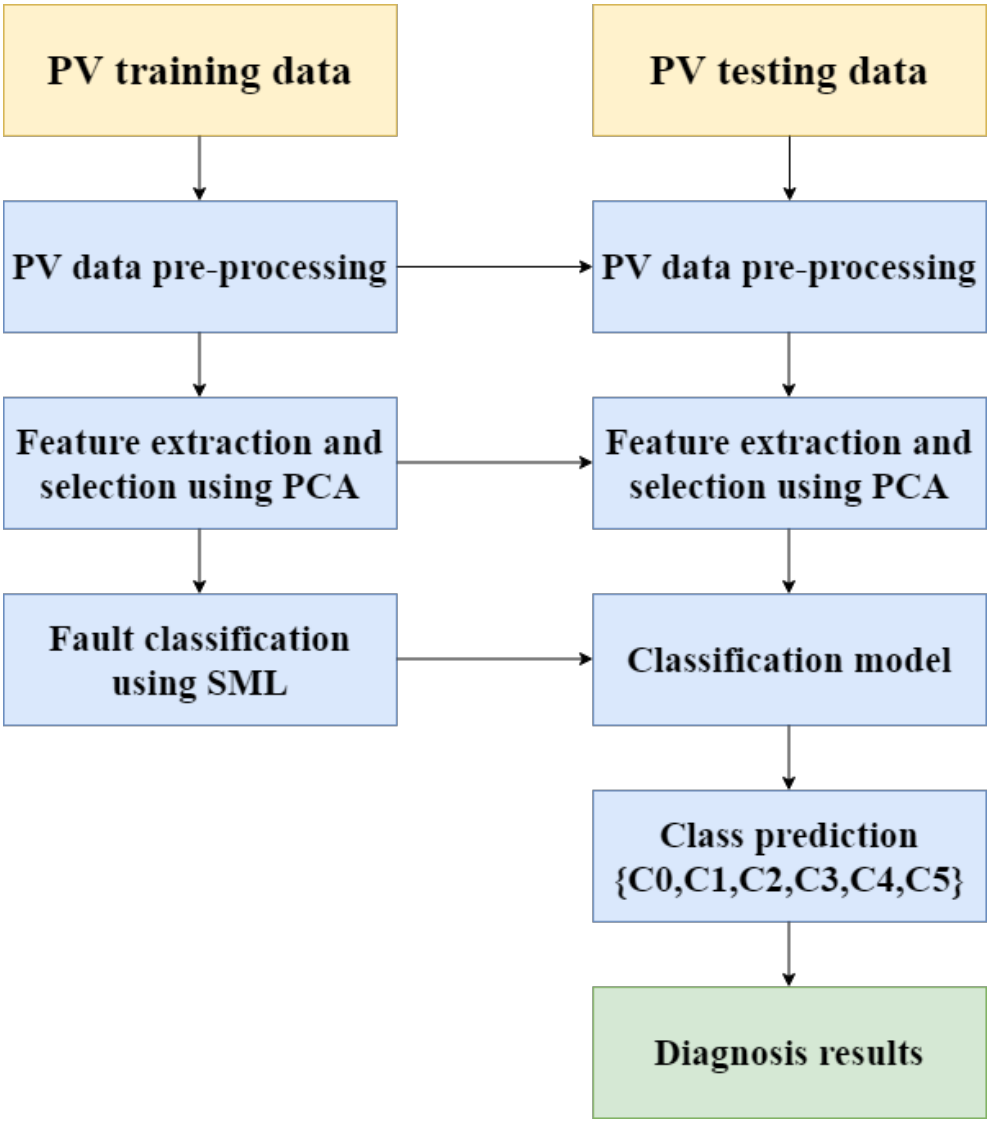

Figure. 1.21: GCPV fault detection and diagnose using PCA and SML

Recent research often combines the KNN method with other machine learning techniques or preprocessing methods to enhance fault detection accuracy. For instance, in [50], a modified version of the KNN algorithm based on the Ada-boost algorithm and the Markov chain has been introduced and employed to classify historical samples into four categories: sunny, cloudy, overcast, and rainy. Hybrid methods, incorporating KNN alongside algorithms such as decision trees, neural networks, or ensemble methods, aim to leverage the strengths of different approaches [51–53]. The RF algorithm exhibits several merits for fault detection and diagnosis in PVSs. It is known for its high accuracy, robustness, and ability to handle large datasets with numerous features. An example of RF-based approaches for fault detection and diagnosis in PVSs is illustrated

in [3]. The simplified flowchart of the algorithm developed in this work is given by (figure 1.22). This algorithm uses the current from each string in the PVA, along with the PVA voltage, to identify and diagnose four types of faults: degradation, partial shading, line-to-line faults, and open circuit faults.

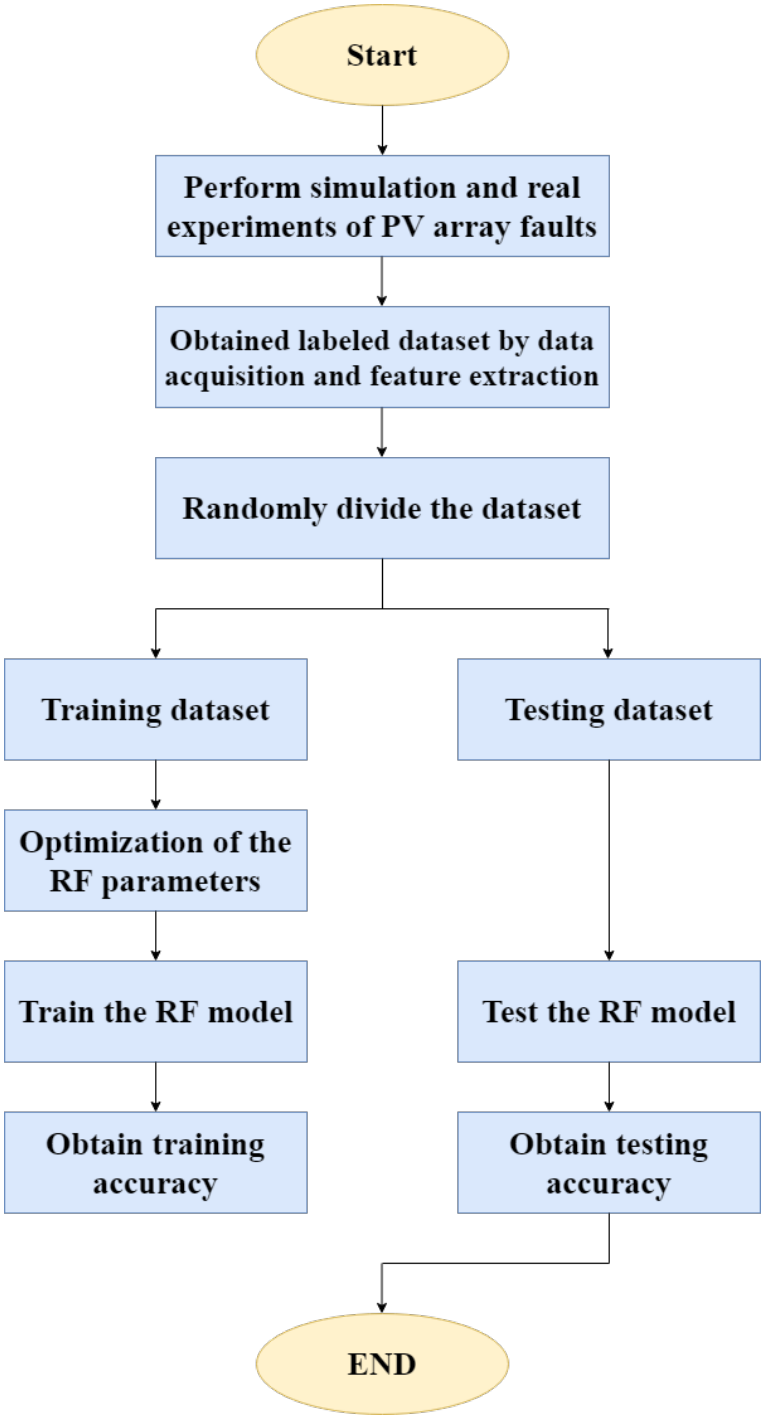

Figure. 1.22: Flowchart of the method used to diagnose faults in PVS based on RF[3].

DT, popular machine learning algorithms used for classification, model decisions based on a tree-like graph of decisions and their possible consequences. They have been successfully employed, along with their combinations with other techniques, in various recent research works to develop efficient algorithms for fault detection and diagnosis in PVSs [54, 55]. A distinctive technique based on the C4.5 DT algorithm has been introduced in [12].The method presented in this work utilizes one model, based on the Sandia model, for fault detection and another, consisting of two C4.5 DT, for diagnosing three types of faults: short circuit, line-line, and string or free faults.

1.5 Data acquisition and monitoring photovoltaic systems

To monitor the performance of PVSs in real-time, a so-called data acquisition system using sensors must be employed [56]. Temperature and sun irradiances are examples of what can be gathered. Voltage and current data can also be obtained from both the PV generator and the inverter. The data acquisition system is an electronic card that is connected to a group of sensors on one hand and to a computer on the other, where the latter interacts with the data by sketching the electrical characteristics I-V and P-V of the PVS in real-time.

The data acquisition and monitoring of the performance of PVSs can be facilitated through the use of a data logger, which collects and stores data in a format such as an Excel spreadsheet, for example. The data logger plays a crucial role in establishing a comprehensive database that can be utilized in various ways, including the development of algorithms for fault detection and diagnosis.

1.6 Cases studies

In this work, two algorithms have been developed for the detection and diagnosis of faults on the DC side of PVSs. The first algorithm employs the KNN method, while the second algorithm is based on the decision tree algorithm. The primary goal of these algorithms is to classify various faults selected for their significant impact on PVSs reliability, energy loss, and the potential for complete PVM corruption. Notably, the first algorithm is designed to categorize four classes, whereas the second algorithm extends its classification to seven classes.

The performance of these algorithms is assessed using precision, recall, and accuracy metrics, which are selected for their ability to evaluate the effectiveness of a given classification algorithm. The chosen fault types are: partial shading fault, short circuit fault, and open circuit fault.

In this thesis, the proposed algorithms are designed to detect and diagnose faults in a PVA consisting of two parallel PV strings. Each string comprises fifteen Isofoton 106/12 W modules figure 1.23.

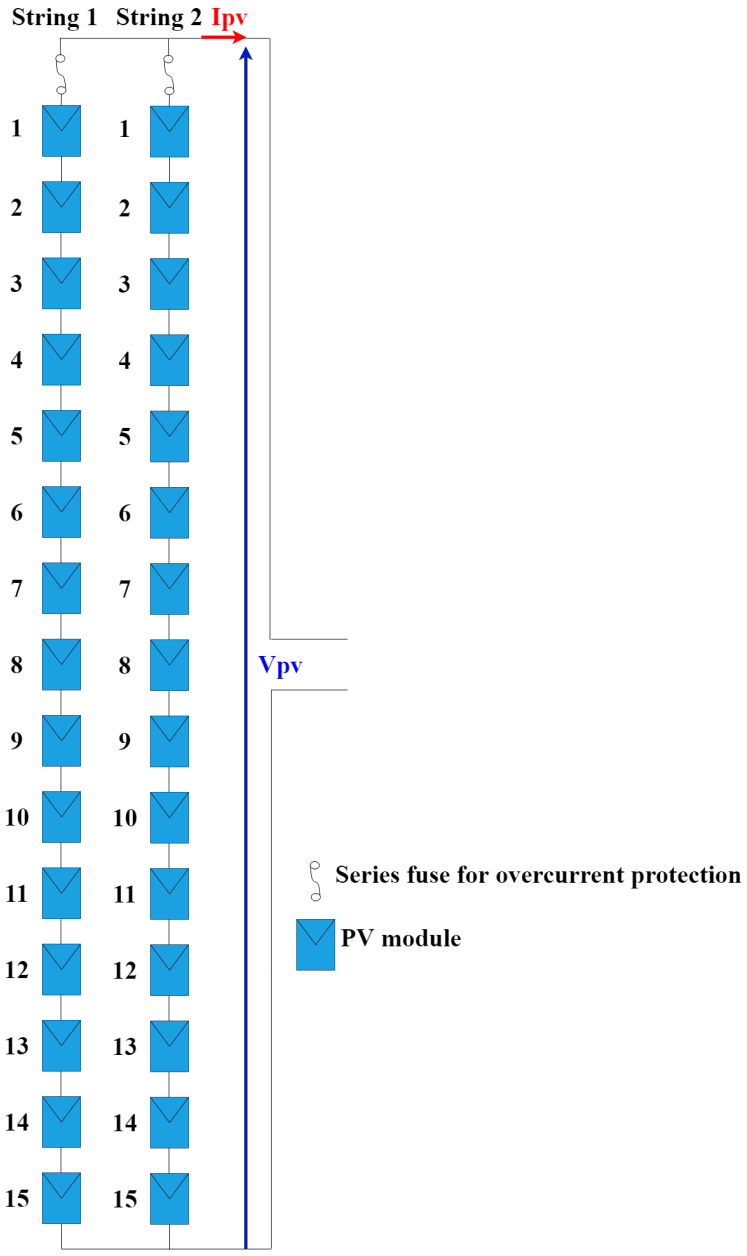

Figure. 1.23: PVA used in this work

1.7 Conclusion

This chapter has addressed three major points. Firstly, it presented a comprehensive examination of PVSs, exploring their possible configurations, and providing a brief discussion of their common faults. Secondly, the chapter reviewed prominent fault detection and diagnosis approaches proposed in the literature. The narrative primarily focused on the detection and diagnosis of faults in PV panels, categorizing them into electrical and non-electrical approaches. By analyzing these algorithms, we gained insights into the advantages, drawbacks, and limitations of each, paving the way for the development of specific algorithms that will be detailed in the coming chapters.

CHAPTER 2

PHOTOVOLTAIC ARRAY MODELING AND VALIDATION

2.1 Introduction

This chapter focuses on two key aspects: the modeling of photovoltaic systems and determining the parameters of their models. It begins by exploring the fundamental physical principles governing the operation of photocells and their electrical properties. Detailed procedures for establishing parameters of photovoltaic modules, using EPC and giza pyramid construction algorithms, are then outlined. Additionally, an effective approach for estimating the MPP based on these identified parameters is discussed.

The accuracy of the identified parameters is assessed using real static (I-V) curves, while the effectiveness of the MPP estimation strategy is validated through experimental measurements.

2.2 Photovoltaic modeling

An electronic component that is designed to convert photons (light particles) into direct current is called a photovoltaic cell or a solar cell. The term "photovoltaic effect" refers to this conversion as it shown in figure 2.1.

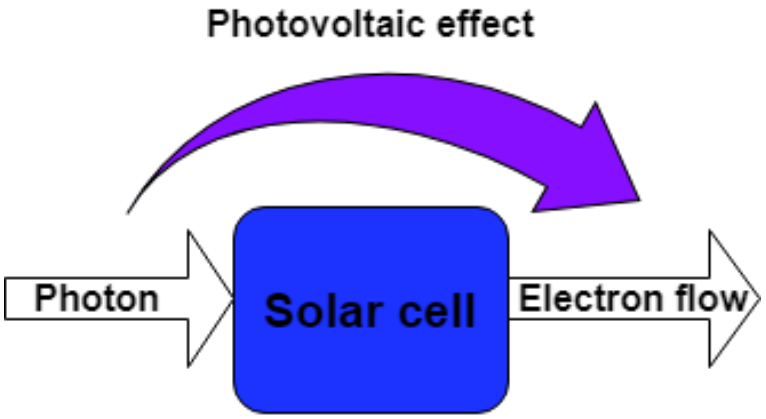

Figure. 2.1: Photovoltaic

As depicted in figure 2.2, the photovoltaic effect takes place in semiconductor materials when the energy absorbed from an outside source, such as photons from sunlight, is greater than the energy gap between the valence and conduction bands.

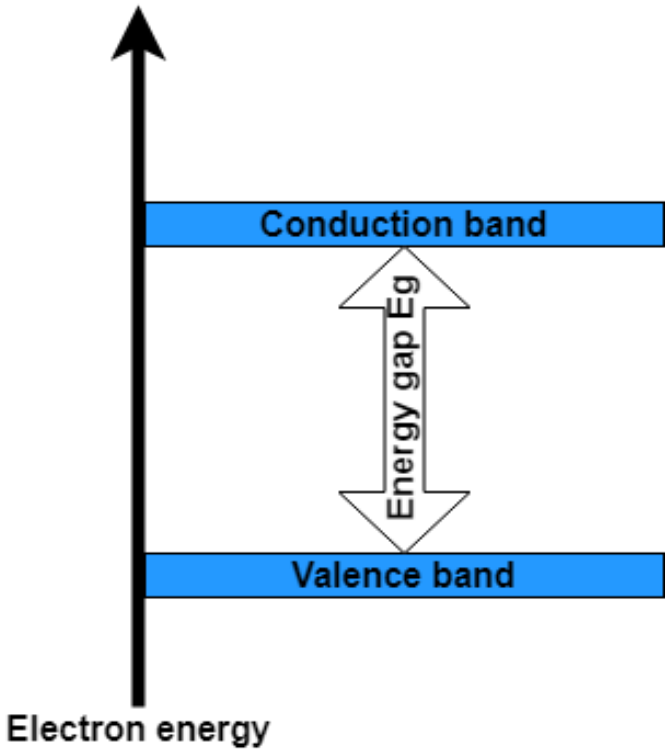

Figure. 2.2: Energy levels.

One of the characteristics of semiconductor materials like silicon (Si) is that when they take in an external energy that is greater than the energy gap (in our case, sunlight or photons), the energized electrons in the valence layer become free to move, leaving holes behind (electron vacant). These free electrons move irregularly in all directions. These electrons must move in one direction to generate an electric current, which can be achieved by combining two types of semiconductors, P-type and N-type, as shown in figure 2.3.

In order to create the N-type, atoms like phosphorus that have more electrons than silicon are added, whereas boron that has one fewer electrons than silicon is added to create the P-type. Doping is the term used to describe the process of producing N- and P-types.

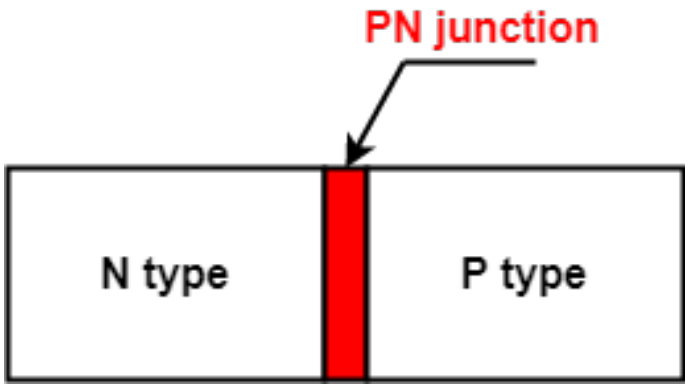

Figure. 2.3: PN junction diode.

When the N-type side is exposed to sunlight, the excited and energized electrons diffuse (move) to the P-type through the PN junction, while the excited and energized holes diffuse from the P-type to the N-type. In order to allow external electrons to pass in one direction and block them from doing so in the other, an electric field is created by the movement of electrons from N-type to P-type and holes from P-type to N-type [57–61]. The functioning of PV cells closely resembles that of semiconductor diodes. While the current in a diode is generated by applying an external voltage, in a PV cell, the current is induced by exposing it to light, as depicted in figure 2.4(a), while figure 2.4(b) shows solar cell components [4].

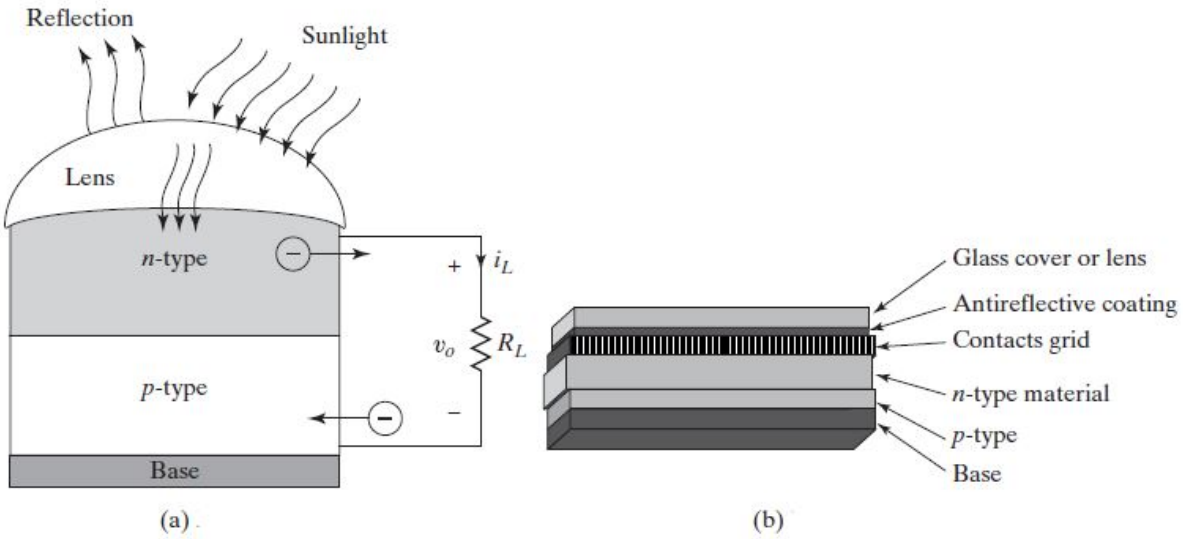

Figure. 2.4: PV cell [4]

The cell is symbolized by a diode, as illustrated in figure 2.5a. The current of the cell, denoted as I_c and corresponding to the reverse-biased current of the diode, is minimal. The cell's representation can be simplified as a reverse-biased diode alongside a current source, as depicted in figure 2.5b.

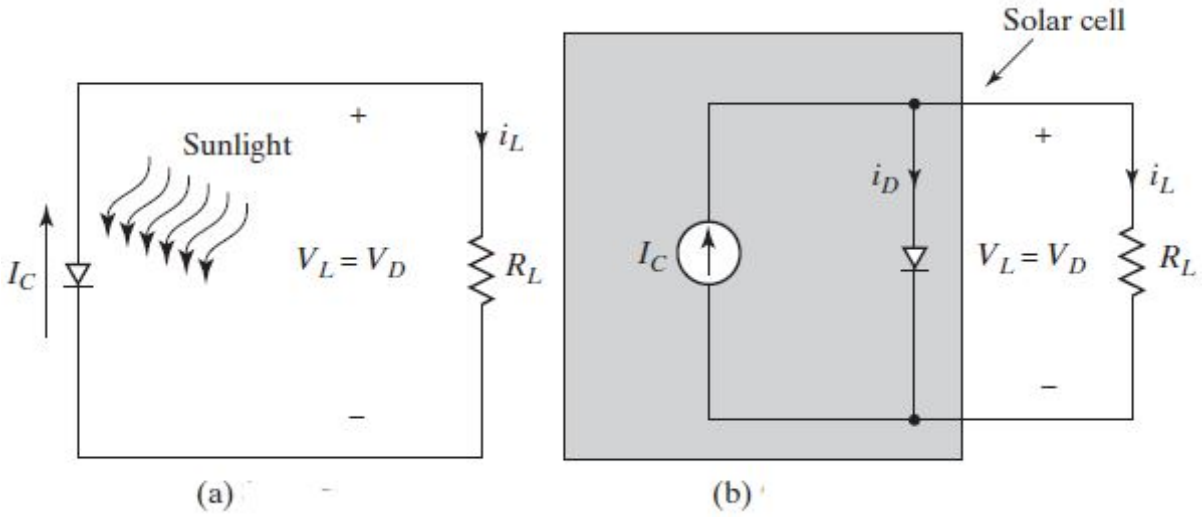

Figure. 2.5: Ideal PV cell

2.2.1 Effect of electrical losses

The most commonly used PVM equivalent circuit models are the single, double, and triple diode models [17–20]. The model that is used in this work is the SDM as it appears in figure 2.6.

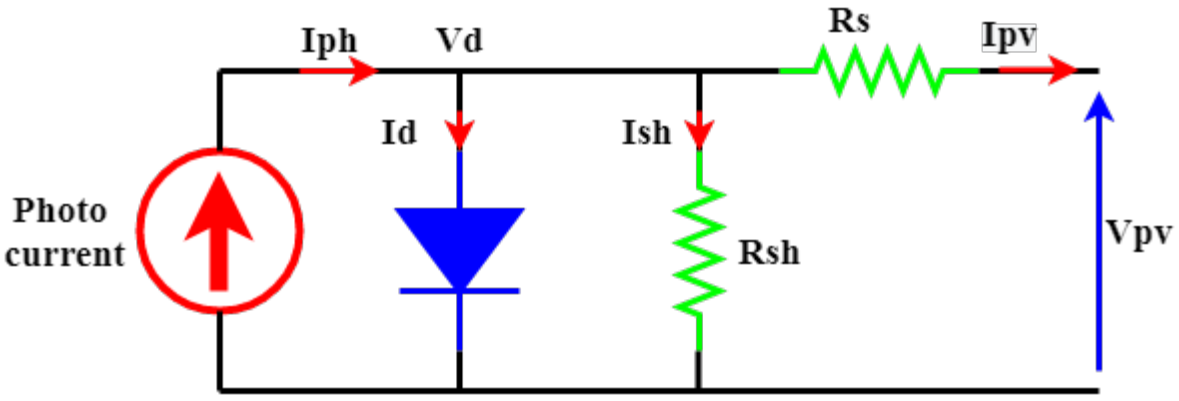

Figure. 2.6: Equivalent circuit of a practical SDM

The single diode model, given in figure 2.6, incorporates electrical losses that are attributed to collector traces and external wires, they are represented by a series resistance R_s. The value of R_s typically falls within the range of a few milliohms. Additionally, the internal resistance of the crystal is denoted by a parallel resistance R_{sh}, with a value typically falling within the range of a few kilohms.

From the circuit of figure 2.6 we have:

$$I_{pv} = I_{ph} - I_d - I_{sh} \tag{2.1}$$

$$I_{sh} = \frac{V_{Rsh}}{R_{sh}} \tag{2.2}$$

$$I_{sh} = \frac{V_{pv} + R_s I_{pv}}{R_{sh}} \tag{2.3}$$

The diode current is given as follow [4]:

$$I_D = I_0 \left[\exp\left(\frac{q\left(V_{pv} + R_S I_{pv}\right)}{nkT} \right) - 1 \right] \tag{2.4}$$

By substituting the aforementioned equations 2.2 to 2.4 into equation 2.1, we get the I-V characteristic of a solar cell as given by equation 2.5.

$$I_{pv} = I_{ph} - I_0 \left[\exp\left(\frac{q\left(V_{pv} + R_s I_{pv}\right)}{nkT} \right) - 1 \right] - \frac{V_{pv} + R_s I_{pv}}{R_{sh}} \tag{2.5}$$

The equation representing the P-V characteristics of a solar cell is provided as follows:

$$P_{pv} = I_{pv} V_{pv} \tag{2.6}$$

Where: I_{pv}, V_{pv} and P_{pv} are the generated output current, voltage and power from the PV cell, I_{ph} is the light-generated current, I_0 is the diode initial current, R_s and R_{sh} are series and shunt resistors respectively, q is the electron charge ($1.60 \times 10^{-19} C$), k is the Boltzmann constant ($1.38 \times 10^{-23} J/K$), n is the ideality factor of diode, and T is the cell temperature.

2.2.2 Effects of Irradiance and Temperature

Figure 2.7 depicts the influence of temperature on the I-V and P-V characteristics when the irradiance is $1 KW/m^2$, whereas figure 2.8 depicts the impact of solar irradiance on the I-V and P-V characteristics when the temperature is $25 C^0$.

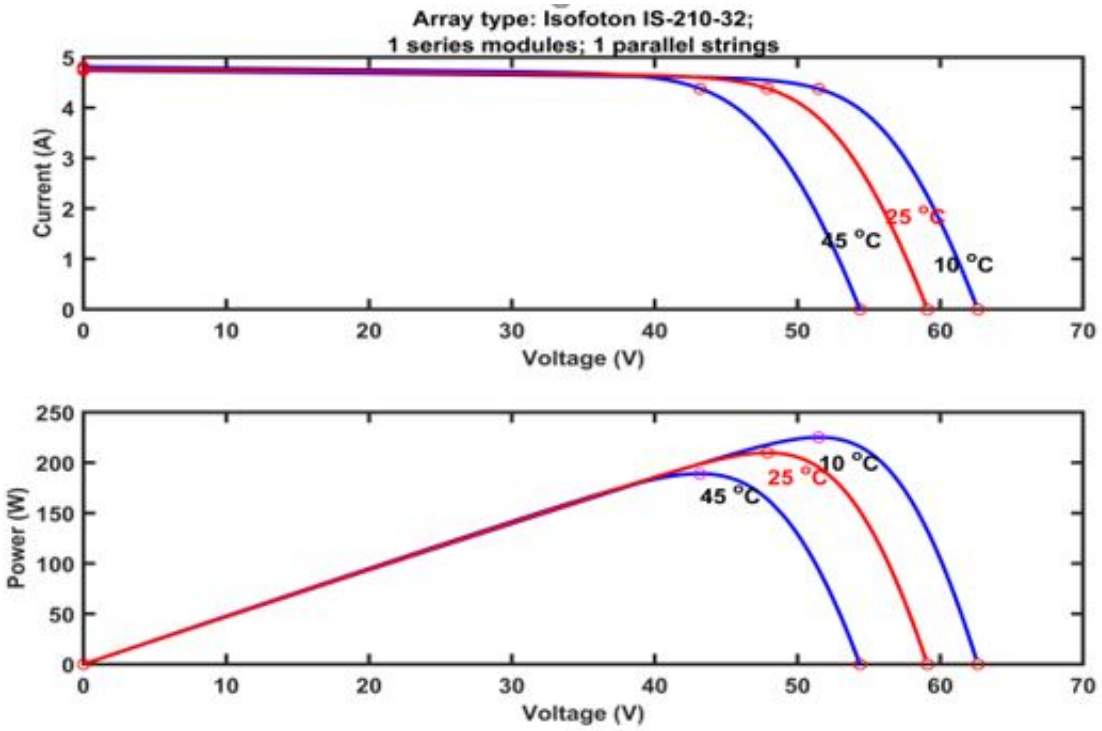

Figure. 2.7: Effect of temperature on the I-V and P-V characteristics

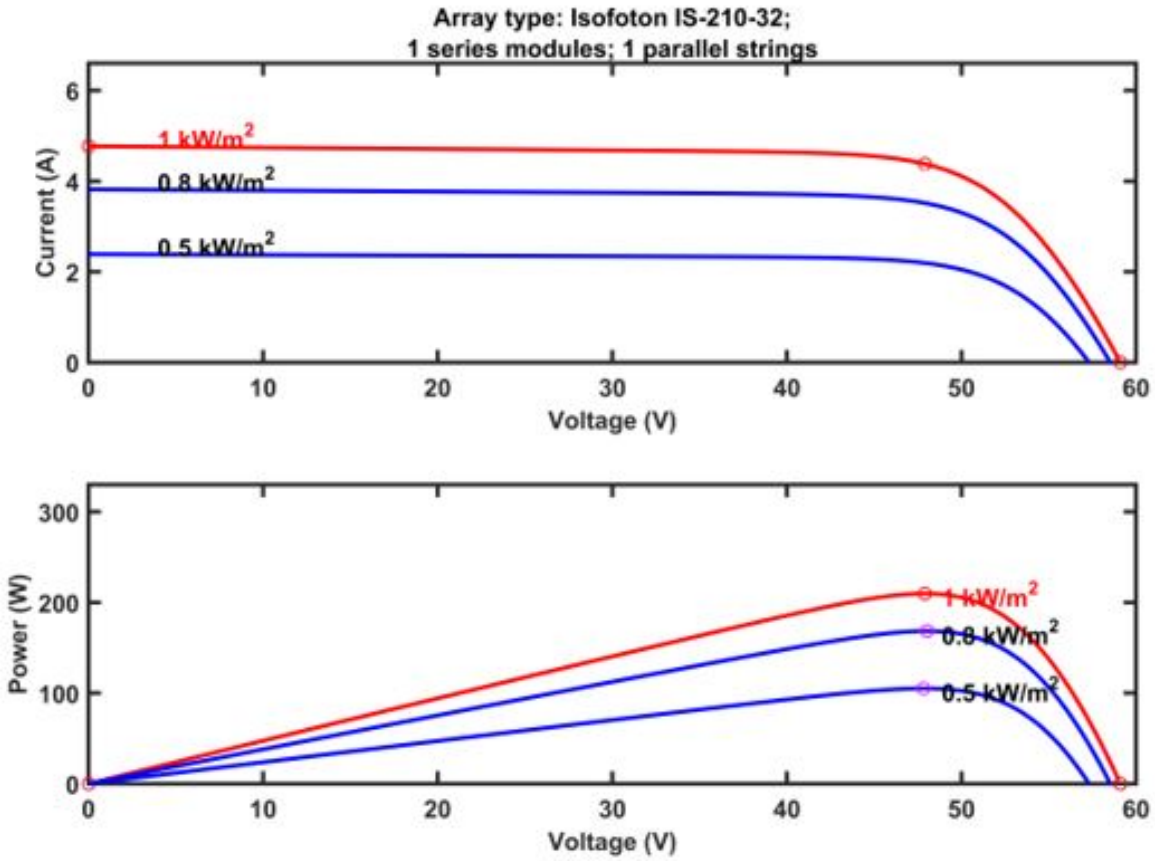

Figure. 2.8: Effect of irradiance on the I-V and P-V characteristics

2.3 Identification of the single diode model parameters

I_{ph}, I_d, R_s, R_{sh},and n are the five parameters that should be found. identifying these parameters is considered an optimization problem for which a given cost function must be minimized using an appropriate optimization algorithm. In this work, two metaheuristic optimization algorithms, namely the EPC and the GPC algorithms, are used to minimize the cost function given by:

$$RMSE = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N} f(V,I,\theta)^2\right)} \tag{2.7}$$

Where:

$$f(V,I,\theta) = I_{meas} - \left(I_{ph} - I_0\left[\exp\left(\frac{q\,(V+R_sI)}{nkT}\right) - 1\right] - \frac{V+R_sI}{R_{sh}}\right) \tag{2.8}$$

$\theta = [I_{ph}, I_0, R_s, R_{sh}, n]$ is the vector of parameters to be estimated and N is the data size. the parameters identification procedure is highlighted in figure 2.9.

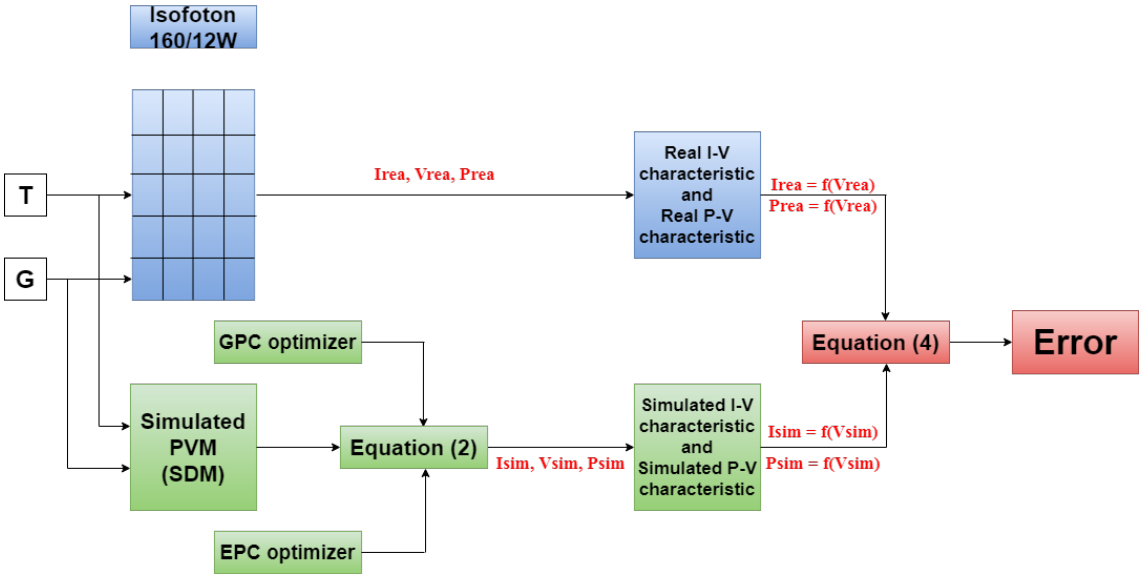

Figure. 2.9: Parameters identification procedure

2.3.1 Emperor Penguins Colony

The EPC algorithm [62–65] was inspired by the Emperor penguin's behavior. Penguins attempt to reach the right heat inside the colony based on their position in the colony, which causes them to strive against the extreme coldness in their environment and control their heat body temperature. To do this, penguins must cluster and maneuver in a spiral-like motion.

Huddling together allows penguins to produce a warm environment. The heat concentrates in the huddle's center. Penguins travel in a spiral-like motion to spread out the heat, giving each of them the opportunity to reach the core. The heat inside the huddle can reach any amount as much as penguins require due to the spiral movement. Penguins reduce their body temperature as they come closer to the center of the huddle.

Heat radiation emitted (heat transfer) from each penguin is defined as follow:

$$Q_{\text{pengui}} = A\varepsilon\sigma T_s^4 \tag{2.9}$$

Where:

Q_{pengui} : is the rate of heat transmission from the penguin to the environment in units of time (W). A: is the total surface area of the penguin which is calculated and is $0.56m^2$. ε: According to [66], is the emissivity of bird's plumage which is

considered 0.98. σ: is the Stefan-Boltzmann constant ($5.6703\ddot{O}10^{-}8W/m^2K^4$). T_s : is the absolute temperature in Kelvin (K) which is considered 35^0C equal to 308.15 K.

2.3.1.1 Heat attractiveness

- Each penguin is considered a linear source of heat.
- It's well known that the heat transfer from the warm source to the cold one. Which means that the cold penguin is attracted to the warm one.

$$Q = Q_{\text{penguin}}\, e^{-\mu x} \tag{2.10}$$

Where:
μ: is an attenuation coefficient (its role is to ensure that heat emitted by the penguin body is reduced). x: is the distance between two linear heat sources (two penguins).

2.3.1.2 Coordinate spiral-like movement

To compute new positions for each penguin, the following equations are used:

$$\begin{cases} x_k = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}}\cos\left\{\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\} \\ y_k = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}}\sin\left\{\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}}+Qe^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\} \end{cases} \tag{2.11}$$

The spiral-like movement may become a spiral monotonous due to the predetermined angle. To increase diversity and avoid being limited to a monotonous spiral path, a new coefficient is introduced, called the mutation factor φ, which is multiplied with a random vector ϵ. Then it's added with the equation 2.10. The equation can be considered as follow:

$$Eq\cdot(10)+\varphi\epsilon_i \tag{2.12}$$

this algorithm is controlled by the body heat radiation of penguins and their spiral-like movement in their colony.

The basic steps of the EPC algorithm are summarized as follows:

A. Generate the initial population array (colony size).

B. Compute the initial cost function for each penguin in the colony.

C. Compare between a single penguin cost function with the entire colony and do the following:

   – Compute heat radiation using equation 2.9.

   – Compute attractiveness using equation 2.10.

   – Compute coordinate spiral movement using equation 2.11.

   – Determine new position using equation 2.12.

D. Sort and find the best solution.

E. Decrease heat radiation, mutation coefficient, and increase heat absorption coefficient.

F. Go back to step C and do the same process until reach the maximum iteration value.

2.3.2 Giza Pyramids Construction

According to [67], workers such as slaves, masons, carpenters, etc are supervised by a pharaoh's special agent. Workers bring stone blocks from different places to the construction site then, the pharaoh's agent specify the exact location for each stone where should be. If a worker gets exhausted or injured, he will be substituted by another energetic worker. The pyramid was built using sloping roads. Workers must push stone blocks from its initial position to the location installation in the pyramid. Notice that the movement of the stone block is influenced by three factors which are friction force, ramp gradient, and initial velocity. Some rules are considered for this algorithm:

- Straight-on ramp is used to build the pyramids.
- Only one ramp is used.
- Ramp gradient (angle with the horizon) is less than 15.
- The solutions are extracted by combining between the position of the worker and the stone block.
- Friction of stone block is considered, while the worker's friction is ignored.

- Some workers are substituted and put them into new positions due to fatigueness.

The basic steps of the GPC algorithm are summarized as follows:

A. Randomly generate the initial population array of stone blocks (workers).

B. For each stone block, compute the amount of stone block displacement d as follows:

$$d = \frac{v_0^2}{2g\left(\sin\theta + \mu_k \cos\theta\right)} \tag{2.13}$$

Where:
g : is the gravity, $g = 9.8$. θ : is the ramp angle with the horizon $\theta < 15^0$..
v_0 : is the initial velocity of the stone block and is determined by a uniformly distributed random number in each iteration.

$$v_0 = \text{rand}(0, 1) \tag{2.14}$$

μ_k : is the kinetic coefficient of friction between the stone block and the ramp. $\mu_k = rand(\mu_{kmin}, \mu_{kmax})$. μ_{kmin} and μ_{kmax} are randomly predetermined.

C. Calculate the new position for workers (the movement of workers).

$$x = \frac{v_0^2}{2g\sin\theta} \tag{2.15}$$

The worker friction is ignored as it mentioned earlier in rule number five.

D. Estimate the new position (solution). The new position (solution) is obtained from the resultant of stone blocks and workers as follows:

$$P = (P_i + d) \times x\varepsilon_i \tag{2.16}$$

Where:
P_i: is the current position.
d : is the displacement value of the stone.
x : is the movement of workers value.
ε_i : is a random vector that follow the uniform distribution.

E. Investigating possibility of substituting workers for the next iteration.
Each worker has a possibility of 50% to lose his power. Thus, 50% to substitute each worker.

F. Return to step B and repeat the same process to reach the maximum iteration.

The flowchart of the GPC algorithm is given in figure 2.10.

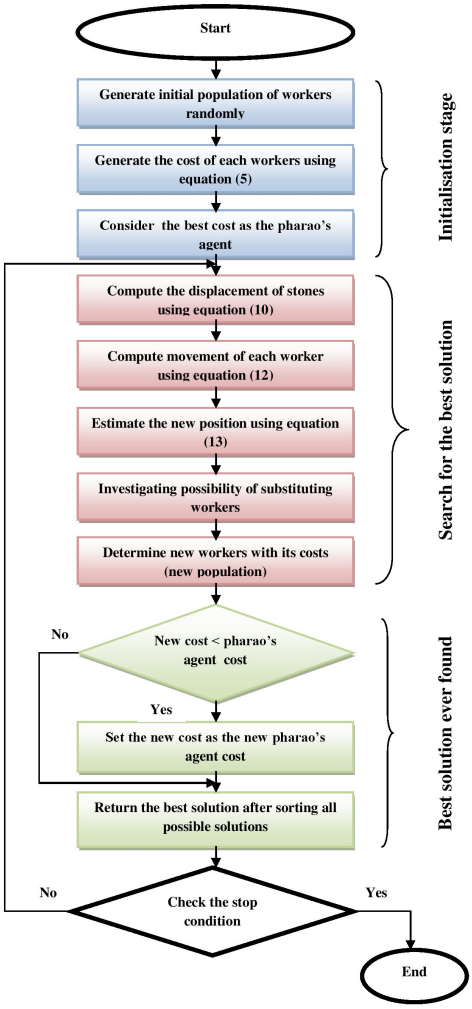

Figure. 2.10: Flowchart of GPC algorithm

2.4 MAXIMUM POWER POINT EXTRACTION

According to [68], the following equations can be used to estimate V_{mpp}, I_{mpp}, and P_{mpp} values.

$$V_{p\max 0} = \frac{V_{p\max}}{1 + C_T (T_J - T_{J0})} + V_T \frac{T_{J0}}{T_J} \ln\left(\frac{E_0}{E_{eff}}\right) - I_{p\max} R_S \left(\frac{E_0}{E_{eff}} - 1\right) \tag{2.17}$$

$$I_{p\max 0} = I_{p\max} \frac{E_0}{E_{eff}} \tag{2.18}$$

$$P_{p\max} = I_{p\max} \times V_{p\max} \tag{2.19}$$

Where:
$E_0 = 100w/m^2$ is the nominal irradiance, $T_{J0} = 25^0$ is the nominal temperature, R_s : is the series resistance, E_{eff} : is the effective irradiance, C_T : is the temperature coefficient of the power. $C_T = -0.0044K^{-1}$, and T_J : is the cell temperature and its given by:

$$T_J\left(E_{eff}, T_{amb}\right) = T_{amb} + (NOCT - T_{ambn}) \frac{E_{eff}}{E_N} \tag{2.20}$$

Where:
T_{ambn} : is the nominal ambient temperature, it's given in the datasheet, $T_{ambn} = 20^0$, $E_N = 800w/m^2$, and V_T : is the thermal voltage where :

$$V_T = \frac{nkT}{q} \tag{2.21}$$

$k = 1.38.10^{-23} m^2 kg K^{-1} s^{-2}$: is the constant of the Boltzmann, $q = 1.60Ö10^{-19} C$: is the electric charge value, I_{pmax0}: is the nominal current at the maximum power point, and V_{pmax0} is the nominal voltage at the maximum power point. I_{pmax0} and V_{pmax0} are both given in the datasheet.

2.5 Simulation results

To evaluate the EPC and the GPC optimizers performance, a Matlab environment with experimental measures of current I and voltage V of $ISOFOTON106/12W$

Electrical characteristic	ISOFOTON 106/12W
$P_{mp}(W)$	106
$V_{mp}(V)$	17.4
$I_{mp}(A)$	6.10
$V_{oc}(V)$	21.6
$I_{sc}(A)$	6.54
$\beta V_{oc}\left(\%/^0C\right)$	-0.36
$\alpha I_{sc}\left(\%/^0C\right)$	0.06

Table 2.1: Electrical characteristic of $ISOFOTON106/12W$.

Parameters	I_{ph}	I_d	R_S	R_{sh}	n
range	$[0-10]$	$\left[10^{-7}-10^{-4}\right]$	$[0-1]$	$[0-400]$	$[0-75]$

Table 2.2: Interval values of the five parameters.

module is used. Table 1 shows the characteristic of the used module, while table 2 gives the upper and the lower values of the parameters $(I_{ph}, I_0, R_s, R_{sh}, n)$.

Tables 3 and 4 show the algorithm specific parameters for the EPC and the GPC algorithms respectively.

Using the EPC and the GPC algorithms, the module parameters are extracted and gathered in table 5. We note that the two algorithms are executed 30 times. The root mean square error (RMSE) value given in table 5 is the mean of 30 times execution.

Due to its higher convergence to the global minimum, the next set of figures will only include the GPC algorithm.

The EPC parameters	Values
Colony size	150
Heat radiation damping ratio	0.95
Attenuation coefficient	1
Attenuation coefficient damping ratio	0.98
Mutation coefficient	0.2
Mutation coefficient damping ratio	0.8
Selected arbitrary for a	0.2
Selected arbitrary for b	0.5

Table 2.3: Specific parameters for the EPC algorithm.

The GPC parameters	Values
Population size	150
Gravity	9.80
Angle ramp	8
Initial velocity	rand$(0,1)$
Minimum friction	5
Maximum friction	10
Substitution probability	0.5

Table 2.4: Specific parameters for the GPC algorithm.

Module's parameters	EPC	GPC
$I_{ph}(A)$	4.94	4.99
$I_d(A)$	1.39×10^{-6}	4.73×10^{-5}
$R_s(\Omega)$	0.15	0.086
$R_{sh}(\Omega)$	275.02	380
n	53.07	69.32
RMSE	**0.088**	**0.033**

Table 2.5: Identified parameters values.

Figure 2.11 depicts the estimated I-V characteristic using the obtained parameters' values and the measured I-V characteristic, while figure 2.12 shows the estimated P-V characteristic using the obtained parameters values and the measured P-V characteristic.

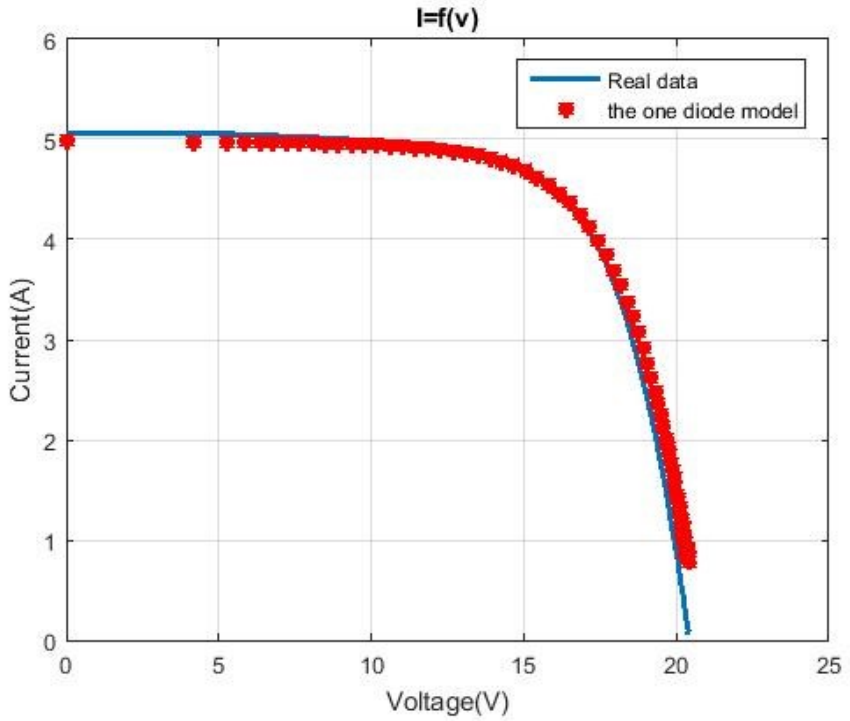

Figure. 2.11: Measured and estimated I-V characteristics

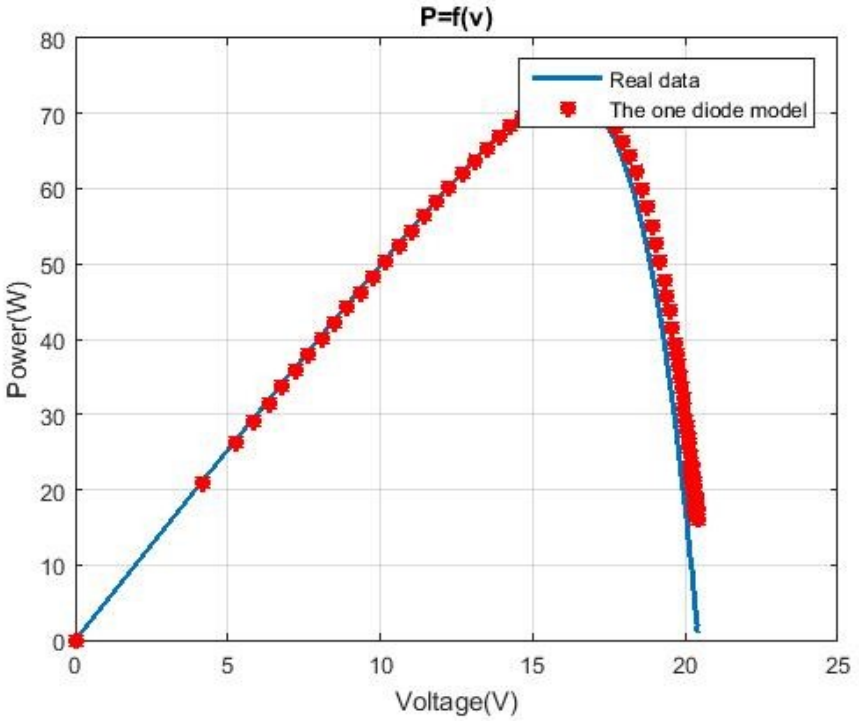

Figure. 2.12: Measured and estimated P-V characteristic

Figure 2.13 represents a comparison between $I_{mpp}^{measure}$ and $I_{mpp}^{estimate}$, figure 2.14 represents a comparison between $V_{mpp}^{measure}$ and $V_{mpp}^{estimate}$, and figure 2.15 represents a comparison between $P_{mpp}^{measure}$ and $P_{mpp}^{estimate}$.

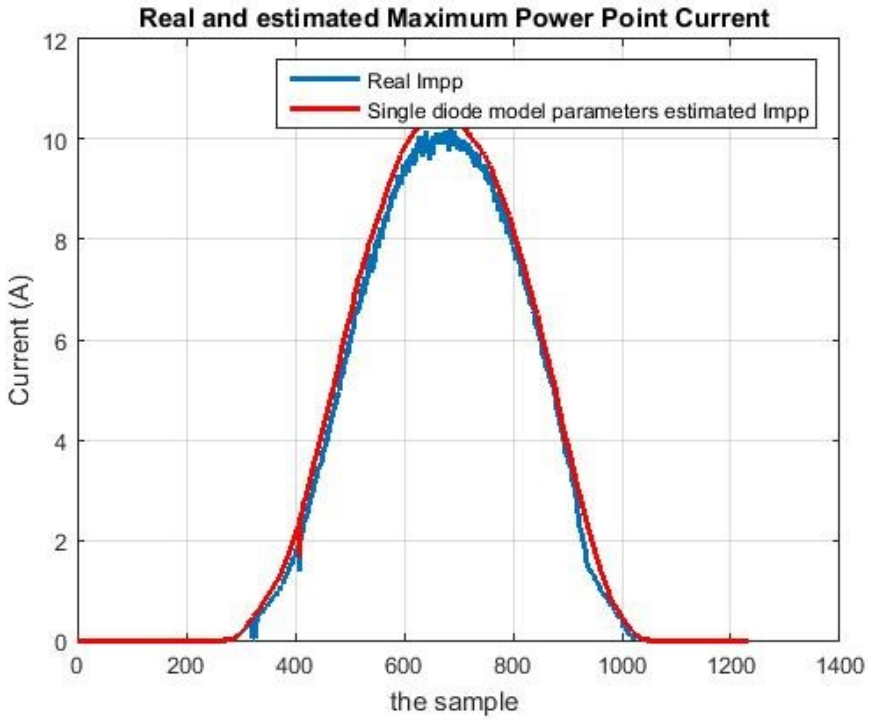

Figure. 2.13: measured and estimated I_{mpp}

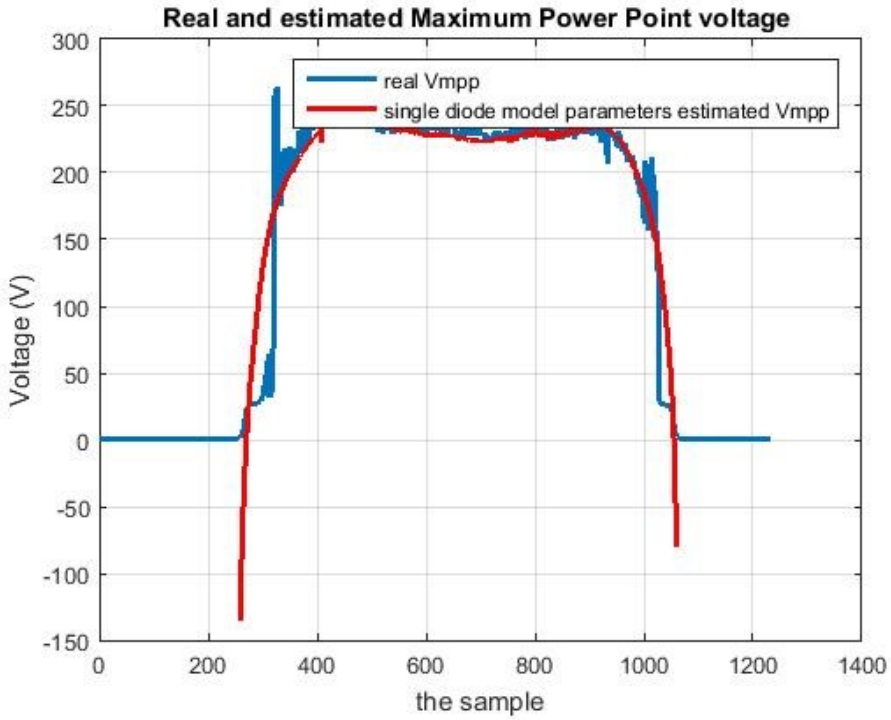

Figure. 2.14: measured and estimated V_{mpp}

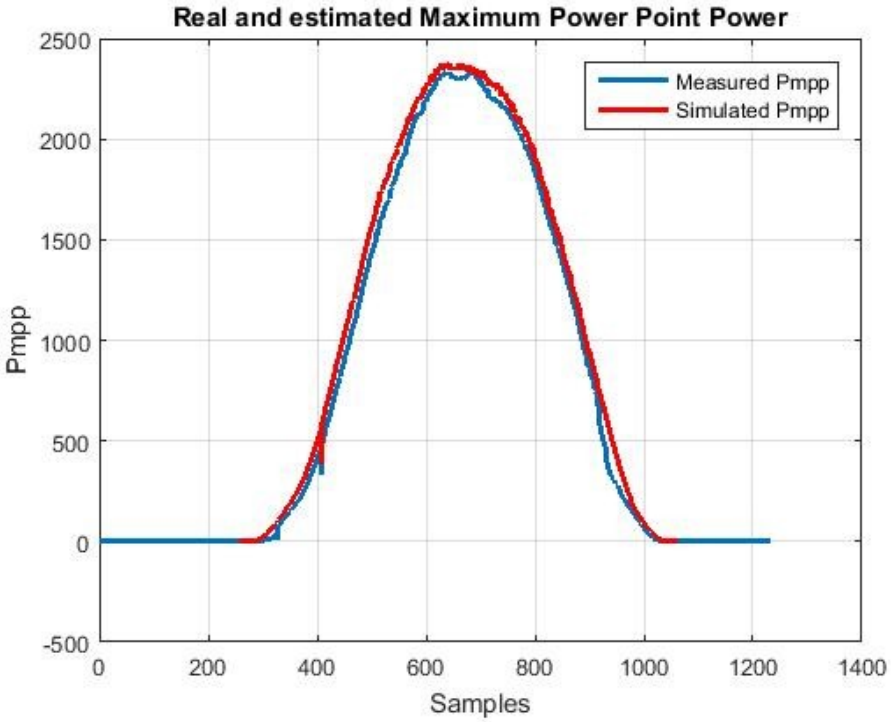

Figure. 2.15: measured and estimated P_{mpp}

It is clear how the two curves in figures 2.11 - 2.15 coincide, which indicate the accuracy of the obtained values and the effectiveness of the used algorithm.

Figures 2.16 and 2.17 show the evolution of the cost function value and the values of the different parameters during the optimization process, respectively.

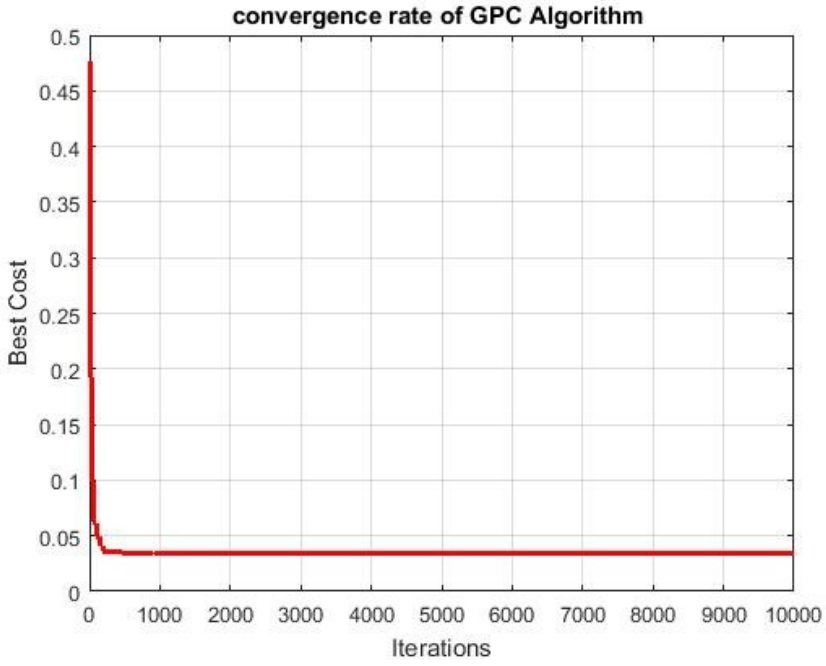

Figure. 2.16: Evolution of the value of the cost function during the optimization process

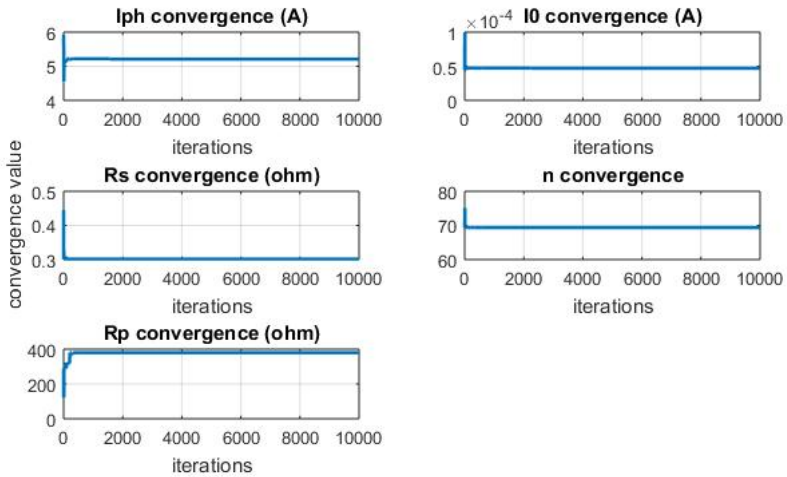

Figure. 2.17: Evolution of the values of the different parameters during the optimization process

2.6 Conclusion

This chapter is principally focused on elucidating the modeling of photovoltaic systems and the determination of their parameters. It provides an overview of the physical principles of solar cells and describes the single diode model. The chapter further discusses the identification of PVM parameters through the utilization of EPC and GPC algorithms. Finally, it presents a proposed approach for estimating the coordinates of the maximum power point.

The effectiveness of the proposed parameters identification algorithms was validated using the Isofoton 106/12W PVM. The parameters identified for the PVM were incorporated into the characteristic equation, and the model was then compared with actual measurements of I-V curves. The obtained results clearly indicate the success of the proposed algorithms in accurately extracting the PVM parameters. Additionally, an efficient approach for MPP estimation, based on the identified parameters, was developed and tested. The outcomes of the MPP estimation demonstrate the effectiveness of these algorithms.

CHAPTER 3

FAULTS DETECTION AND DIAGNOSIS OF PHOTOVOLTAIC SYSTEMS USING MODIFIED K-NEAREST NEIGHBORS ALGORITHM

3.1 Introduction

During the operation of photovoltaic systems, various faults can occur and result in serious problems, such as energy loss or system shutdown. Therefore, it is crucial to identify and diagnose these faults in order to improve system performance. The purpose of this work is to propose an efficient and simple procedure for the early detection and diagnosis of faults on the direct current side of photovoltaic systems using a modified version of the KNN algorithm and the metaheuristic GPC algorithm. These faults include the short circuit of three modules, short circuit of ten modules, and string disconnection.

3.2 Dataset description

The considered faults include the short circuit of three modules, short circuit of ten modules, and string disconnection.

The PVA used to generate dataset for both healthy and faulty states is composed of two parallel strings as it seen in figure 3.1. Each string comprises fifteen Isofoton PVMs (106W/12V) connected in series. This PVA is simulated using the Simulink/Matlab environment under both healthy conditions and the considered faults. The simulation is carried out for different values of the temperature and the irradiance. Each sample of the generated dataset is composed of four physical quantities: cell temperature, irradiance, current at the maximum power point, and voltage at the MPP (T, G, I_{mpp}, V_{mpp}).

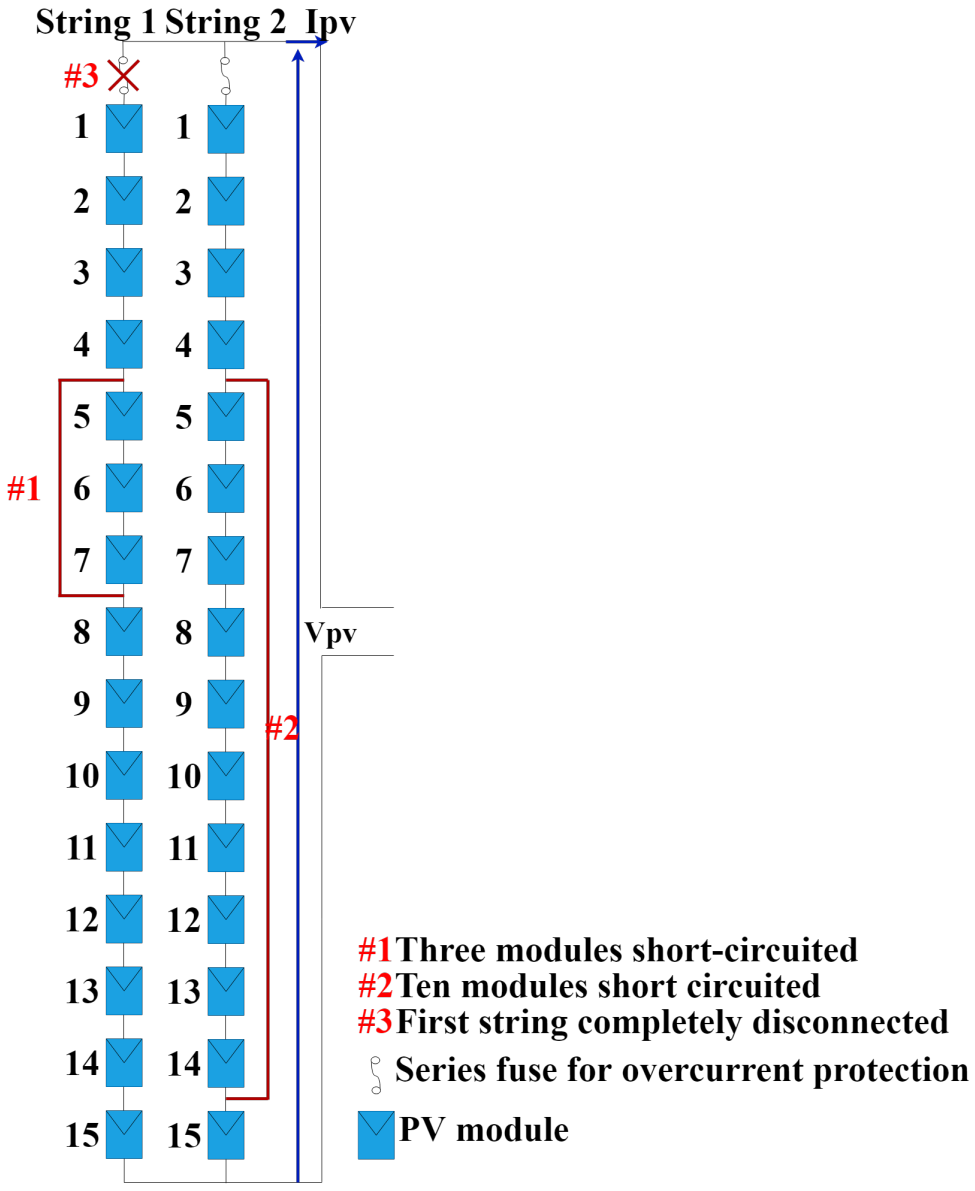

Figure. 3.1: Studied PVS with different considered faults.

Figures 3.2 and 3.3 depict the current I_{mpp} and voltage V_{mpp} for the considered four classes (healthy state and faulty states), respectively. It is clear from figure 3.2 that the fault of string disconnection affects the current I_{mpp}, while the fault of short circuit of modules affects the voltage V_{mpp}. The more modules that are short-circuited, the greater the impact on the current I_{mpp}. The class labels and the number of samples per class are given in table 1. The generated dataset is divided into a training set and a testing set.

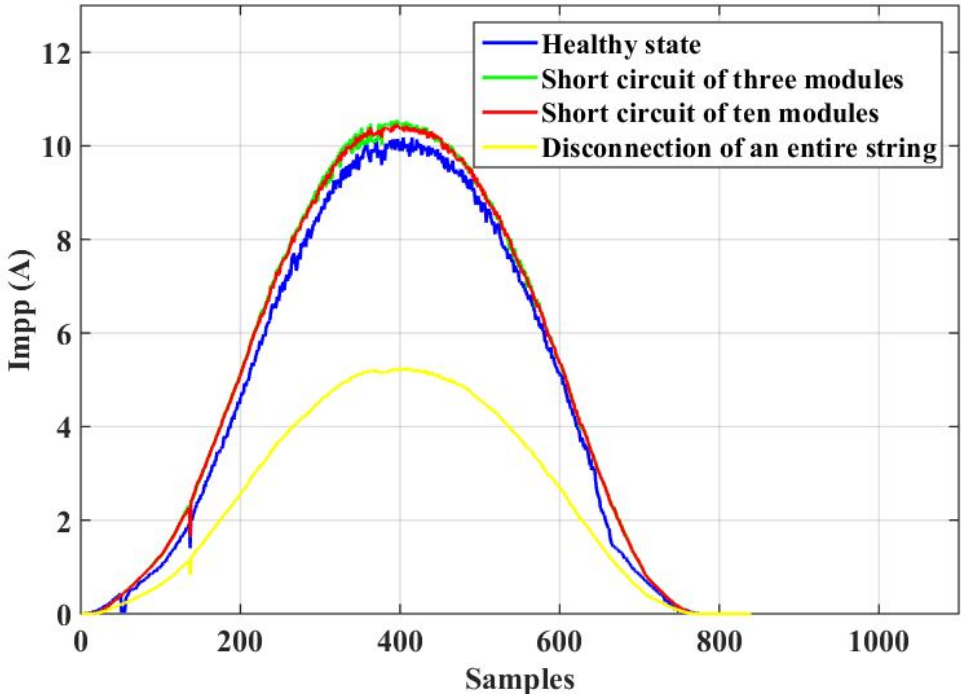

Figure. 3.2: I_{mpp} of the healthy state system and faulty states

	Class name	Data length	Label
Class 0	Normal operation	840	0
Class 1	Short circuit of 3 modules	840	1
Class 2	Short circuit of 10 modules	840	2
Class 3	Disconnect a string of 15 modules	840	3
Data samples	-	3360	-

Table 3.1: Dataset with its classes name, data length, and labels.

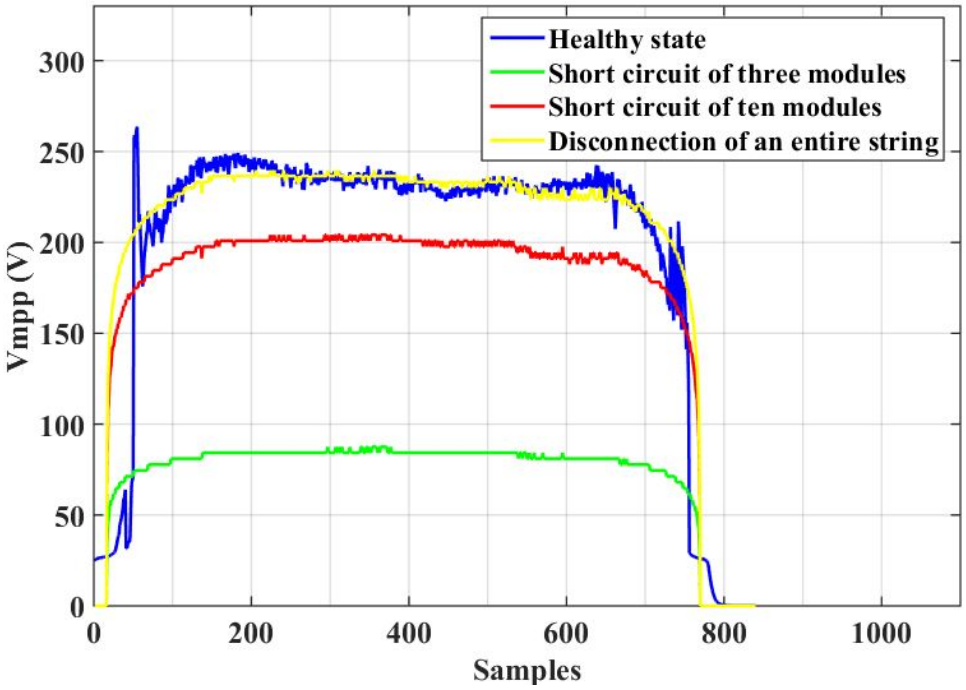

Figure. 3.3: V_{mpp} of the healthy state system and faulty states

The generated dataset has four classes with four attributes which are: cell temperature T, solar irradiance G, and MPP coordinates I_{mpp} and V_{mpp}. The detail of each class is given by table 1.

In order to train and evaluate the proposed algorithm, the dataset was divided into a training set and a testing set.

3.3 Faults detection and diagnosis strategy

3.3.1 Faults detection and diagnosis principle

The developed procedure must distinguish between four classes: the healthy class and three classes representing different types of considered faults. The idea of the proposed approach is to transform the multi-classification problem into a binary classification problem and utilize a modified version of the well-known KNN classifier. The training dataset is divided into two hyper-spheres, each representing a distinct class, then the giza pyramid construction algorithm is utilized to determine the optimal center coordinates of these hyper-spheres. To classify a new data-point using the proposed classifier, which combines the KNN classifier and the giza pyramid construction algorithm, distances are computed only between the new data point and the center of each sphere. Unlike the classical version of the KNN classifier, which involves computing distances between the new data point and the entire dataset.

In the proposed strategy, the number n_c of required classifiers to classify n_m classes is given by:

$$n_c = n_m - 1 \tag{3.1}$$

In this work, since there are four classes, it is necessary to design three classifiers, as is shown by the flowchart of figure 3.4. The first classifier separates class 3 (string disconnection fault) from the rest of the classes, the second classifier separates class 0 (normal operating) from classes 1 and 2, while the last classifier separates classes 1 (short circuit of three modules) and 2 (short circuit of ten modules).

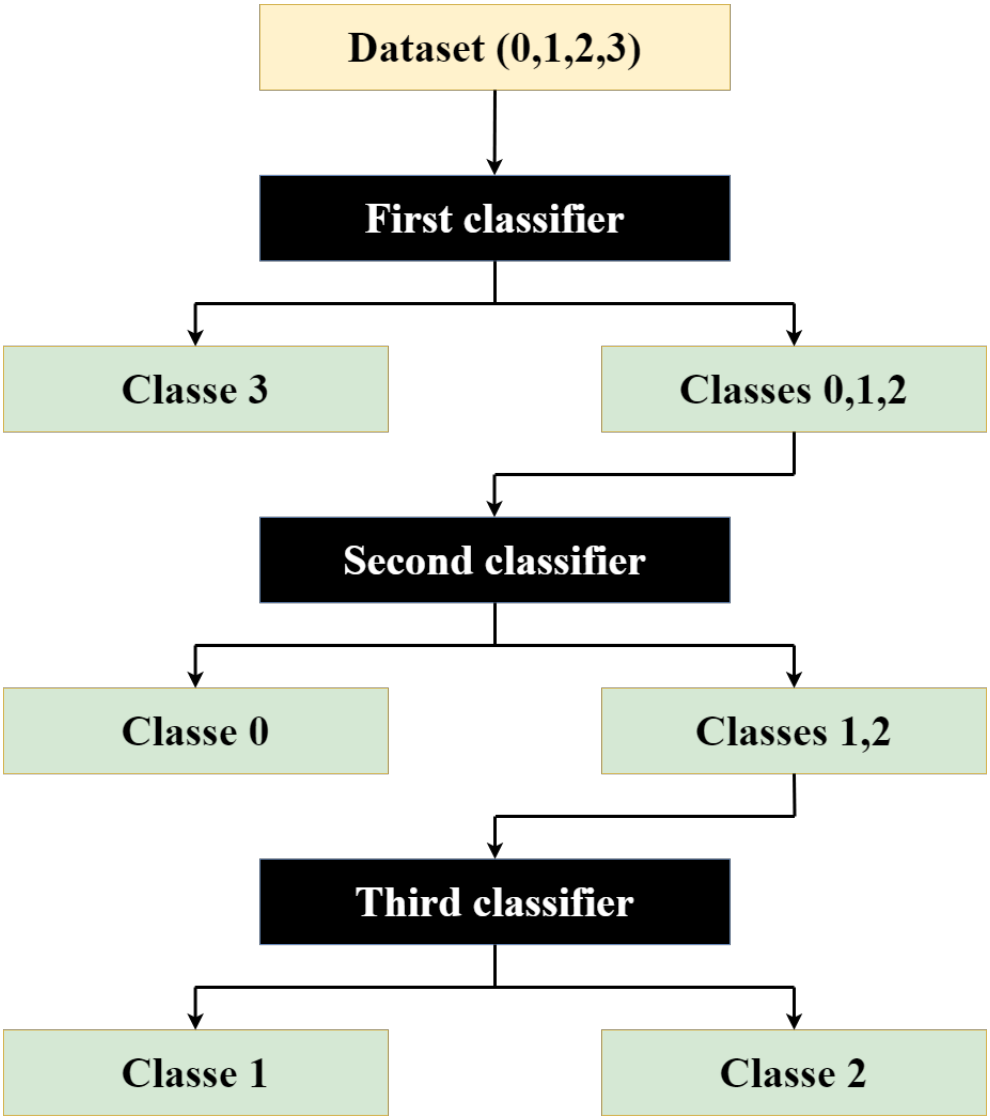

Figure. 3.4: Classification strategy

3.3.1.1 Classification algorithm

For each classifier of the flowchart of figure 3.4, a classification algorithm is required. The developed algorithm is inspired from the classical KNN algorithm. Two hyperspheres representing two different classes are used, and their optimal center coordinates are obtained using the GPC metaheuristic algorithm. In the proposed classification algorithm, only the distances between each new data and the center of each hypersphere computed. In the following, the coordinates (x_i, y_i, z_i, t_i) correspond to the used attributes $(T_i, G_i, I_{mpp_i}, V_{mpp_i})$, respectively, and n is the number of samples in the used dataset.

3.3.1.2 KNN algorithm

The main steps of the KNN algorithm in binary classification are given below:

- Randomly initialize the neighbors' number K.
- For each data-point (x_i, y_i, z_i, t_i), compute the distance between the data-point and the entire dataset using the Euclidean distance.

$$\text{dist}_{\,i} = \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2 + (z_1 - z_i)^2 + (t_1 - t_i)^2} \tag{3.2}$$

  Where:
  $i = 1, 2, \ldots, n$ and n is the dataset length.
  (x_1, y_1, z_1, t_1) : are the first data coordinates in the dataset.

- Sort the computed distances in an ascending order.
- Pick the first K data-points corresponding to the first distances and get their labels.
- Return the mode of the obtained labels.

3.3.1.3 Modified KNN algorithm

Unlike the classical version of the KNN classifier, which involves computing distances between the new data point and the entire dataset,the proposed modified version consists of five steps:

- Select the initial values for the centers of the two hyper-spheres (x_0, y_0, z_0, t_0) and (x_1, y_1, z_1, t_1).
  These values can be chosen from the dataset of class 0 and class 1, respectively. Randomly initialize two values for radius R0 and R1. For simplification reason, R0 is set to be equal to R1. Next, use the GPC algorithm to obtain the optimal value of the center of each hyper-sphere by minimizing a given cost function.
- Compute distances between each center and the entire data using:

$$\text{dist0}_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2 + (t_0 - t_i)^2} \tag{3.3}$$

$$\text{dist1}_i = \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2 + (z_1 - z_i)^2 + (t_1 - t_i)^2} \tag{3.4}$$

Where: $i = 1, 2, \ldots, n$.

- Select the longest distance from $dist0$ and $dist1$, then compute the factors using:

$$factor0 = \frac{R0}{\max(dist0)} \tag{3.5}$$

$$factor1 = \frac{R1}{\max(dist1)} \tag{3.6}$$

- Calculate the new distances $dist_{new0}$ and $dist_{new1}$ using the following equations:

$$dist_{new0_i} = factor0 * \text{dist}_i \tag{3.7}$$

$$dist_{new1_i} = factor1 * \text{dist}_i \tag{3.8}$$

Where: $i = 1, 2, \ldots, n$. By doing this, all data-points are located inside the hyper-sphere.

- if $dist_{new0_i} < dist_{new1_i}$, then the point belongs to class A. Otherwise, it belongs to class B.

The aforementioned steps are summarized by the flowchart of figure 3.5.

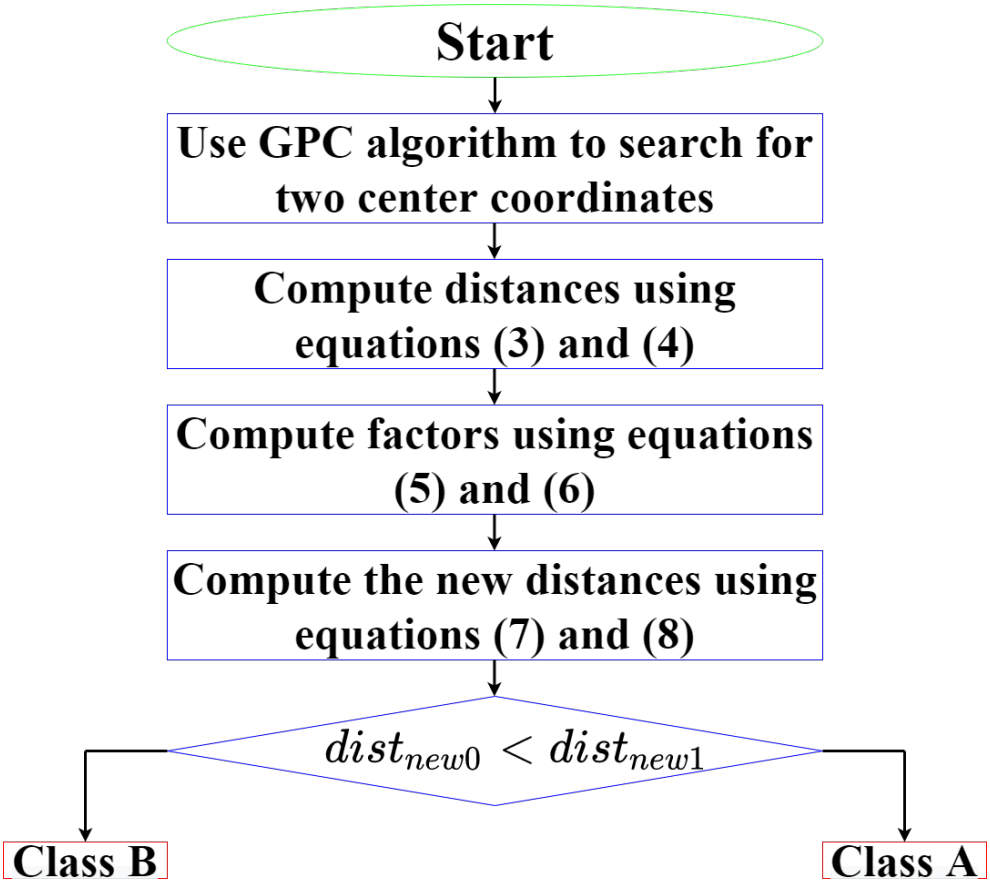

Figure. 3.5: Flowchart of the proposed algorithm.

3.4 Results and discussion

To assess the efficiency of the proposed approach, a comparative study is conducted, including the classical version of the KNN, support vector machine, decision tree, and random forest algorithms. The evaluation criteria considered are accuracy, precision, recall, and execution time.

3.4.1 Training and testing the proposed classifier

For each of the three classifiers in the flowchart of figure 3.4, the training stage aims to search for the appropriate values of the center coordinates of the first hyper-sphere (class A) and the center of the second hyper-sphere (class B) that minimize the following cost function:

$$\text{cost}_{\text{ function}} = \frac{1}{n}\sum_{i=1}^{n}(Y_{\text{real}} - Y_{\text{label}})^2 \tag{3.9}$$

where Y_{real} are real class labels and Y_{label} are predicted labels. The testing stage aims to evaluate the classifier's performance with testing data. To do this, a mathematical tool called confusion matrix is used. As is shown in table 2, this matrix is composed of two rows and two columns.

Real class labels	Predicted label: A	Predicted label: B
Class A	True Positive (TP)	False Negative (FN)
Class B	False Positive (FP)	True Negative (TN)

Table 3.2: Confusion matrix.

TP: denotes data that are in class A, and they are classified in class A by the classifier.
FN: denotes data that in class A, and the classifier classifies them in class B.
FP: denotes data that in class B, and the classifier classifies them in class A.
TN: denotes data that are in class B, and they are classified in class B by the classifier.

In addition to the confusion matrix, the following metrics concepts are usually used to evaluate the classification performance of a given classifier:

- The accuracy: this metric, given by equation 3.10, aims to answer the following question:
  Among the entire test data, what proportion of the data which the classifier correctly classified them?

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \times 100 \tag{3.10}$$

- The precision (positive predictivity): this metric, given by equation 3.11, aims to answer the following question:
  Among the data classified as positive (in class1), what proportion of the data which the classifier correctly classified them?

$$\text{Precision} = \frac{TP}{TP+FP} \times 100 \tag{3.11}$$

- The recall (sensitivity): this metric aims, given by equation 3.12, to answer the following question:
  Among the data which is actually positive (class 1 in the data set), what proportion of the data the classifier correctly classified them?

$$\text{Recall } = \frac{TP}{TP+FN} \times 100 \tag{3.12}$$

As is shown by figure 3.4, three classifiers are required to detect and diagnosis the considered faults (healthy operating, short circuit of three modules, short circuit of ten modules and string disconnection). Hence, to evaluate the performance of the proposed strategy of faults detection and diagnosis, the analysis of each of the three classifiers, using the average values of the aforementioned metrics and the average value of the execution time is necessary. The higher the values of accuracy, precision, and recall, the better the performance of the proposed strategy. All simulation are carried out using personnel computer with Intel(R)Core(TM)i3 processor (2.5 GHz).

3.4.2 Obtained results using the modified KNN algorithm

The first step, as shown in the flowchart of figure 3.5, is to use the GPC algorithm to find the centers of the first hyper-sphere (class A) and the second hypersphere (class B) for classifiers 1, 2 and 3. The parameters values of the GPC algorithm are listed in table 3. Following the completion of the remaining steps as shown in figure 3.5, the confusion matrix is extracted to determine the accuracy, precision, and recall for the first classifier (table 7), the second classifier (table 11), and the third classifier (table 15).

Population size	Gravity	Angle of Ramp	Minimum friction	Maximum friction	Substitution probability
60	9.8	10	5	10^5	0.5

Table 3.3: Parameters values of the GPC algorithm.

3.4.2.1 Classifier 1

Tables 4 and 5 give the number of samples in training and testing data for each class respectively. While table 6 represents the center coordinates for the first classifier.

Class 0	Class 1	Class 2	Class 3	Total
706	705	722	723	2856

Table 3.4: Training dataset for the first classifier.

Class 0	Class 1	Class 2	Class 3	Total
134	135	118	117	504

Table 3.5: Testing dataset for the first classifier.

	$T\,(°\mathrm{C})$	$G\left(W/m^2\right)$	$I_{mpp}(A)$	$V_{mpp}(V)$
Center (x_0, y_0, z_0, t_0)	$2.08e-6$	0.003	$5.71e-5$	$4.77e-5$
Center (x_1, y_1, z_1, t_1)	0.0011	$1.757e-5$	0.72	$8.857e-4$

Table 3.6: Center coordinates found using the GPC algorithm for the first classifier.

The confusion matrix of the first classifier is given below:

$$Confusion_{matrix1} = \begin{bmatrix} 372 & 15 \\ 14 & 103 \end{bmatrix}$$

Based on the obtained values of the $Confusion_{matrix1}$, classifier 1 successfully identified 475 of the 504 data points. 372 (TP) and 103 (TN) data points are successfully classified into classes 0 and 1, respectively. 29 data points are incorrectly classified by classifier 1, of which 15 (FN) data points should belong to Class 1 but are classified as Class 0, and 14 (FP) data points should belong to Class 0 but are classified as Class 1.

The corresponding values of the accuracy, the precision, and the recall are gathered in table 7. Figure 3.6 depicts the evolution of the cost and accuracy functions over iterations. It can be observed that accuracy and the cost function rapidly converge toward their higher and smallest values respectively.

Accuracy	Precision	Recall	Execution time (100)
94.24%	96.37%	96.12%	0.042(s)

Table 3.7: Metrics values of the first classifier.

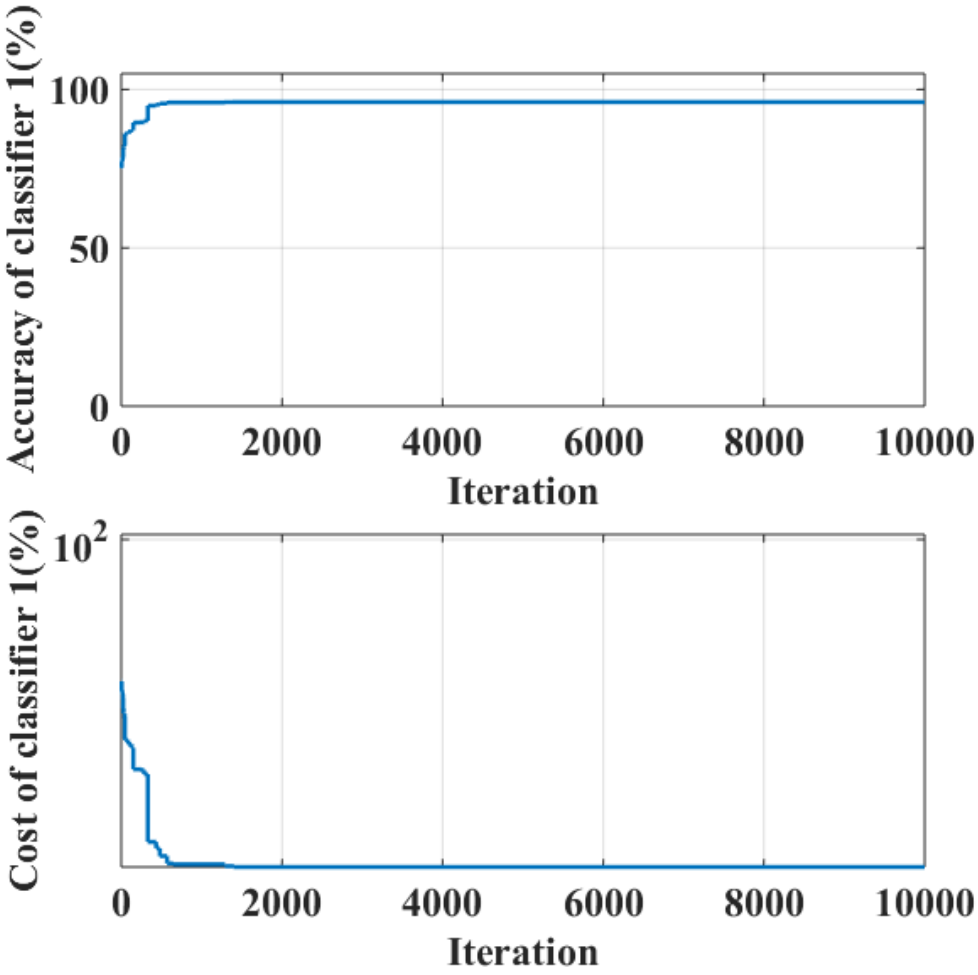

Figure. 3.6: Accuracy and the cost function of the first classifier in terms of iteration.

3.4.2.2 Classifier 2

Tables 8 and 9 give the number of samples in training and testing data for each class respectively, while table 10 represents the center coordinates for the second classifier.

Class 0	Class 1	Class 2	Class 3	Total
706	705	722	84	2217

Table 3.8: Training dataset for the second classifier.

Class 0	Class 1	Class 2	Class 3	Total
125	129	118	14	386

Table 3.9: Testing dataset for the second classifier.

	$T\,(°\mathrm{C})$	$G\left(W/m^2\right)$	$I_{mpp}(A)$	$V_{mpp}(V)$
Center (x_0, y_0, z_0, t_0)	400	400	$6.52369e+3$	$1.33088e+3$
Center (x_1, y_1, z_1, t_1)	400	400	400	$2.73569e+2$

Table 3.10: Center coordinates found using the GPC algorithm for the second classifier.

Based on the confusion matrix of the second classifier given below, the accuracy, the precision,and the recall are computed and given in table 11.

$$Confusion_{matrix2} = \begin{bmatrix} 259 & 2 \\ 29 & 96 \end{bmatrix}$$

Table 11 presents the second classifier performance and figure 3.7 represents the evolution of the accuracy and the cost function over iterations.

Accuracy	Precision	Recall	Execution time (100)
92%	90%	99.23%	0.041(s)

Table 3.11: Metrics values of the second classifier.

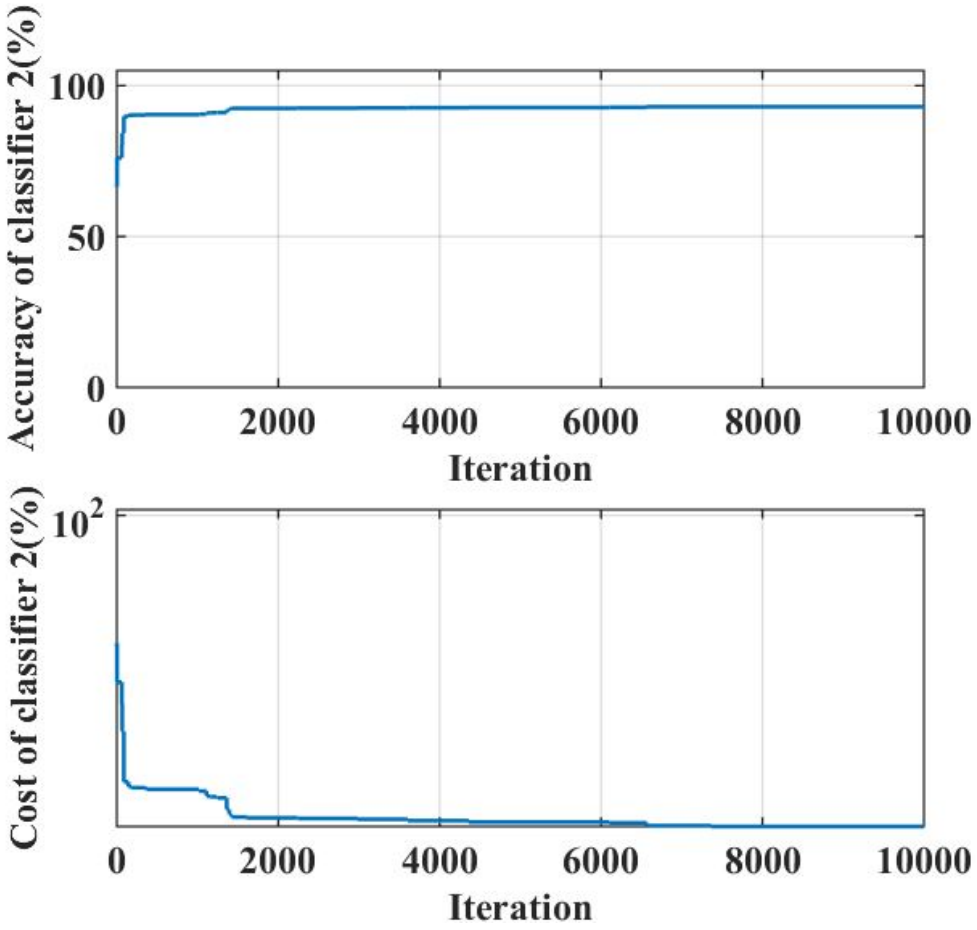

Figure. 3.7: Accuracy and the cost function of the second classifier in terms of iteration.

3.4.2.3 Classifier 3

Table 12 and 13 give the number of samples in training and testing data for each class respectively, while table 14 represents the center coordinates for the third classifier.

Class 0	Class 1	Class 2	Class 3	Total
195	705	722	84	1706

Table 3.12: Training dataset for the third classifier.

Class 0	Class 1	Class 2	Class 3	Total
29	127	118	14	288

Table 3.13: Testing dataset for the third classifier.

	$T\,(°\mathrm{C})$	$G\left(W/m^2\right)$	$I_{mpp}(A)$	$V_{mpp}(V)$
Center (x_0, y_0, z_0, t_0)	400	74.075	400	400
Center (x_1, y_1, z_1, t_1)	77.155	400	$7.72656e+3$	$5.70441e+3$

Table 3.14: Center coordinates found using the GPC algorithm for the third classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 15.

$$Confusion_{matrix3} = \begin{bmatrix} 107 & 0 \\ 63 & 118 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
78.13%	63%	100%	0.040(s)

Table 3.15: Metrics values of the third classifier.

Figure 3.8 represents the evolution of the accuracy and cost function over iterations.

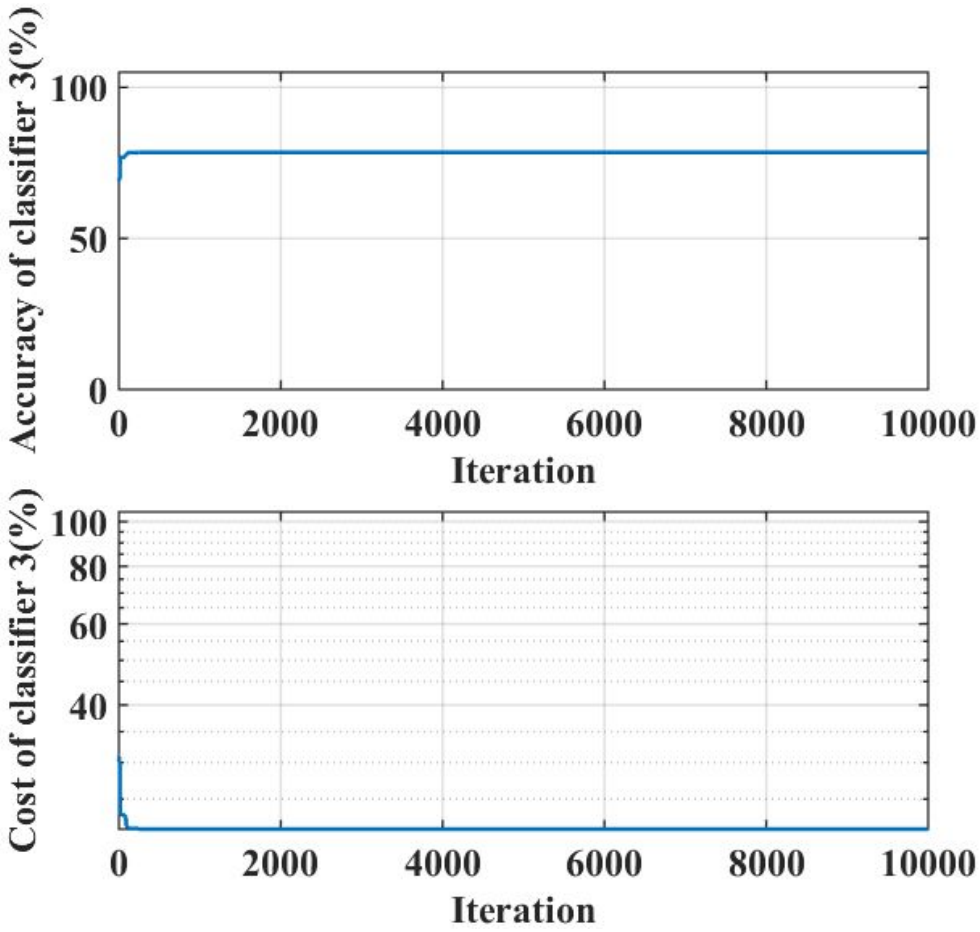

Figure. 3.8: Accuracy and the cost function of the third classifier in terms of iteration.

3.4.3 Obtained results using the classical KNN algorithm

3.4.3.1 Classifier 1 (for K = 6)

From figure 3.9, it can easily seen that the best result is obtained when $k = 6$.

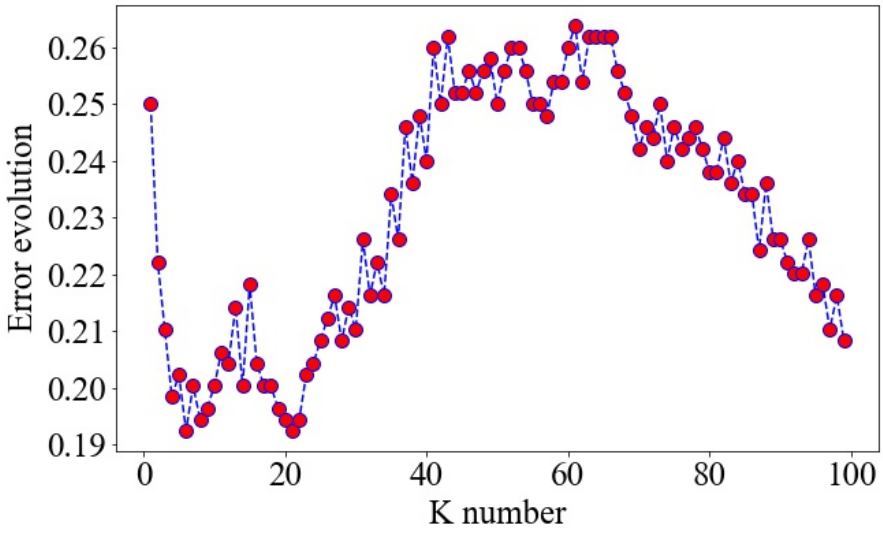

Figure. 3.9: Error in terms of K values for the first KNN classifier.

The number of samples of the testing dataset, given in table 16, is used to evaluate the performance of the first KNN classifier.

Class 0	Class 1	Class 2	Class 3	Total
134	135	118	117	504

Table 3.16: Testing dataset for the first KNN classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 17.

$$Confusion_{matrix1} = \begin{bmatrix} 318 & 69 \\ 28 & 89 \end{bmatrix}$$

K	Accuracy	Precision	Recall	Execution time (100)
6	81%	82%	92%	0.072(s)

Table 3.17: Metrics values of the first KNN classifier.

3.4.3.2 Classifier 2 (for K = 5)

Figure 3.10 shows that the best result is obtained when $K = 5, K = 7$, or $K = 8$.

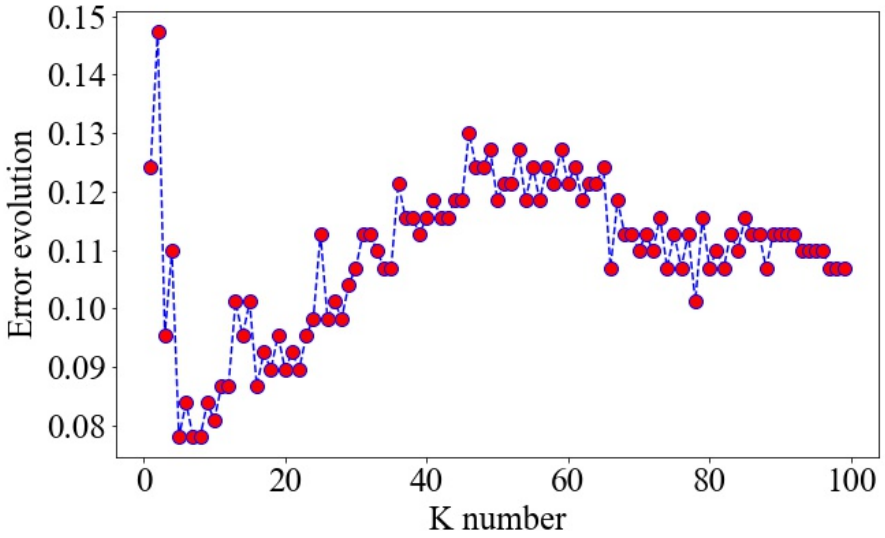

Figure. 3.10: Error in terms of K values for the second KNN classifier.

The number of samples of the testing dataset, given in table 18, is used to evaluate the performance of the second KNN classifier.

Class 0	Class 1	Class 2	Class 3	Total
67	134	117	28	346

Table 3.18: Testing dataset for the second KNN classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and computed and given in table 19.

$$Confusion_{matrix2} = \begin{bmatrix} 265 & 14 \\ 13 & 54 \end{bmatrix}$$

K	Accuracy	Precision	Recall	Execution time (100)
5	92%	95%	95%	0.046(s)

Table 3.19: Metrics values of the second KNN classifier.

3.4.3.3 Classifier 3 (for K = 8)

Figure 3.11 depicts the evolution of the error rate versus the value of K. It can easily seen, from this figure, that the best result is obtained when $k = 8$.

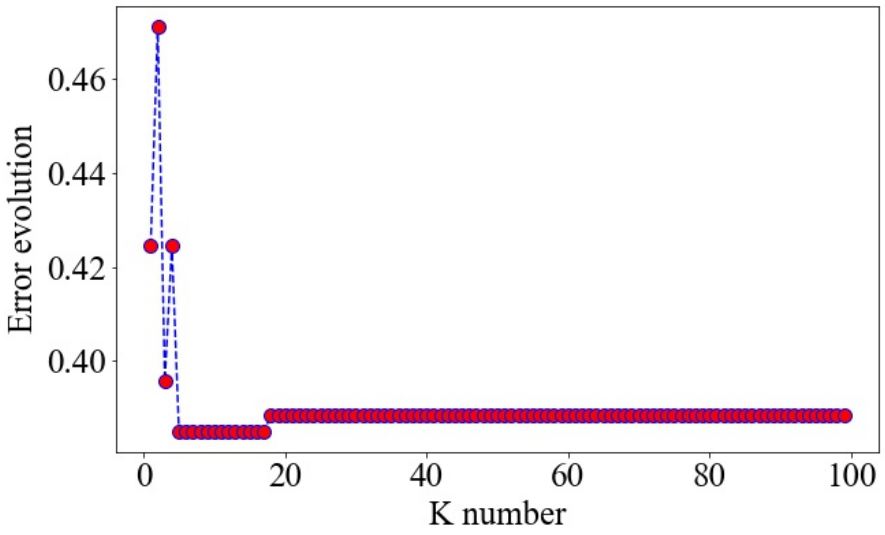

Figure. 3.11: Error in terms of K values for the third KNN classifier.

The number of samples of the testing dataset, given in table 20, is used to evaluate the performance of the third KNN classifier.

Class 0	Class 1	Class 2	Class 3	Total
13	265	0	0	278

Table 3.20: Testing dataset for the third KNN classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and computed and given in table 21.

$$\text{Confusion }_{\text{matrix 3}} = \begin{bmatrix} 171 & 107 \\ 0 & 0 \end{bmatrix}$$

K	Accuracy	Precision	Recall	Execution time (100)
8	62%	62%	100%	0.037(s)

Table 3.21: Metrics values of the third KNN classifier.

3.4.4 Obtained results using support vector machine

The second machine learning algorithm used is the SVM. The SVM hyper-parameters (C and gamma) are tuned using the grid search method.

3.4.4.1 Classifier 1

The tuned values of the first classifier are: $C = 1000$ and $gamma = 0.001$. The number of samples of the testing dataset, given in table 22, is used to evaluate the performance of the first SVM classifier.

Class 0	Class 1	Class 2	Class 3	Total
134	135	118	115	504

Table 3.22: Testing dataset for the first SVM classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 23.

$$Confusion_{matrix1} = \begin{bmatrix} 21 & 28 \\ 96 & 359 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
75%	18%	43%	0.083(s)

Table 3.23: Metrics values of the first SVM classifier.

3.4.4.2 Classifier 2

The tuned values of the second classifier are: $C = 1000$ and $gamma = 0.001$. The number of samples of the testing dataset, given in table 24, is used to evaluate the performance of the second SVM classifier.

Class 0	Class 1	Class 2	Class 3	Total
106	135	118	96	455

Table 3.24: Testing dataset for the second SVM classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 25.

$$Confusion_{matrix2} = \begin{bmatrix} 48 & 38 \\ 58 & 311 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
79%	45%	56%	0.11(s)

Table 3.25: Metrics values of the second SVM classifier.

3.4.4.3 Classifier 3

The tuned values of the third classifier are: $C = 0.1$ and $gamma = 0.0001$. The number of samples of the testing dataset, given in table 26, is used to evaluate the performance of the third SVM classifier.

Class 0	Class 1	Class 2	Class 3	Total
58	135	118	58	369

Table 3.26: Testing dataset for the third SVM classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 27.

$$Confusion_{matrix3} = \begin{bmatrix} 67 & 04 \\ 28 & 247 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
91%	76%	96%	0.048(s)

Table 3.27: Metrics values of the third SVM classifier.

3.4.5 Obtained results using decision tree

In this sub-section, the decision tree algorithm is used to design the three classifiers required to detect and diagnose the considered faults.

3.4.5.1 Classifier 1

The number of samples of the testing dataset, given in table 28, is used to evaluate the performance of the first DT classifier.

Class 0	Class 1	Class 2	Class 3	Total
134	135	118	117	504

Table 3.28: Testing dataset for the first DT classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 29.

$$Confusion_{matrix1} = \begin{bmatrix} 103 & 08 \\ 14 & 379 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
96%	92%	88%	0.0027(s)

Table 3.29: Metrics values of the first DT classifier.

3.4.5.2 Classifier 2

The number of samples of the testing dataset, given in table 30, is used to evaluate the performance of the second DT classifier.

Class 0	Class 1	Class 2	Class 3	Total
129	134	116	14	393

Table 3.30: Testing dataset for the second DT classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 31.

$$Confusion_{matrix2} = \begin{bmatrix} 118 & 20 \\ 11 & 244 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
92%	91%	86%	0.0029(s)

Table 3.31: Metrics values of the second DT classifier.

3.4.5.3 Classifier 3

The number of samples of the testing dataset, given in table 32, is used to evaluate the performance of the third DT classifier.

Class 0	Class 1	Class 2	Class 3	Total
1	126	110	8	255

Table 3.32: Testing dataset for the third DT classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 33.

$$Confusion_{matrix3} = \begin{bmatrix} 0 & 0 \\ 125 & 130 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
51%	0%	0%	0.0028(s)

Table 3.33: Metrics values of the third DT classifier.

3.4.6 Obtained results using random forest

The random forest algorithm is used to design the three classifiers required to detect and diagnose the considered faults.

3.4.6.1 Classifier 1

The number of samples of the testing dataset, given in table 34, is used to evaluate the performance of the first RF classifier.

Class 0	Class 1	Class 2	Class 3	Total
134	135	118	117	504

Table 3.34: Testing dataset for the first RF classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 35.

$$Confusion_{matrix1} = \begin{bmatrix} 103 & 01 \\ 14 & 386 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
97%	88%	99%	0.08(s)

Table 3.35: Metrics values of the first RF classifier.

3.4.6.2 Classifier 2

The number of samples of the testing dataset, given in table 36, is used to evaluate the performance of the second RF classifier.

Class 0	Class 1	Class 2	Class 3	Total
132	135	118	15	400

Table 3.36: Testing dataset for the second RF classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 37.

$$Confusion_{matrix2} = \begin{bmatrix} 118 & 11 \\ 14 & 257 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
94%	89%	91%	0.073(s)

Table 3.37: Metrics values of the second RF classifier.

3.4.6.3 Classifier 3

The number of samples of the testing dataset, given in table 38, is used to evaluate the performance of the third RF classifier.

Class 0	Class 1	Class 2	Class 3	Total
14	131	117	9	271

Table 3.38: Testing dataset for the third RF classifier.

Based on the confusion matrix given below, the accuracy, the precision, and the recall values are computed and given in table 39.

$$Confusion_{matrix3} = \begin{bmatrix} 0 & 117 \\ 0 & 154 \end{bmatrix}$$

Accuracy	Precision	Recall	Execution time (100)
57%	0%	0%	0.059(s)

Table 3.39: Metrics values of the third RF classifier.

3.4.7 Comparison between the hyper-sphere algorithm, the KNN, the SVM, the DT, and the RF algorithms

The results of the proposed strategy for detecting and diagnosing the considered faults using the modified KNN algorithm are compared with those obtained using the same strategy for fault detection and diagnosis, but based on the KNN, SVM, DT, and RF algorithms. This comparison is based on the average values of accuracy, precision, and recall metrics, as well as the average execution time. The results obtained in the preceding sections are summarized in table 40.

	Accuracy	Precision	Recall	Time (s)
Hyper-sphere 1	94.24	96.37	96.12	0.042
Hyper-sphere 2	92	90	99.23	0.041
Hyper-sphere 3	78.13	63	100	0.040
Average	**88.12**	**83.12**	**98.45**	**0.041**
KNN 1	81	82	92	0.072
KNN 2	92	95	95	0.046
KNN 3	62	62	100	0.037
Average	**78.33**	**79.66**	**95.66**	**0.051**
SVM 1	75	18	43	0.083
SVM 2	79	45	56	0.068
SVM 3	91	76	96	0.048
Average	**81.66**	**46.33**	**65**	**0.067**
DT 1	96	92	88	0.0027
DT 2	92	91	86	0.0029
DT 3	51	0	0	0.0028
Average	**76.66**	**61**	**58**	**0.0028**
RF 1	97	88	99	0.083
RF 2	94	89	91	0.073
RF 3	57	0	0	0.059
Average	**82.66**	**59**	**63.33**	**0.071**

Table 3.40: Comparison between the proposed classifier and the KNN, SVM, DT, and RF based classifier.

From table 40, it can be observed that the modified KNN algorithm achieves the highest values for all three metrics, indicating its superior performance compared to the other algorithms. In terms of execution time, the DT-based classifier is the fastest, while the proposed algorithm has a shorter execution time compared to the RF-based classifier, the SVM-based classifier and the KNN-based classifier.

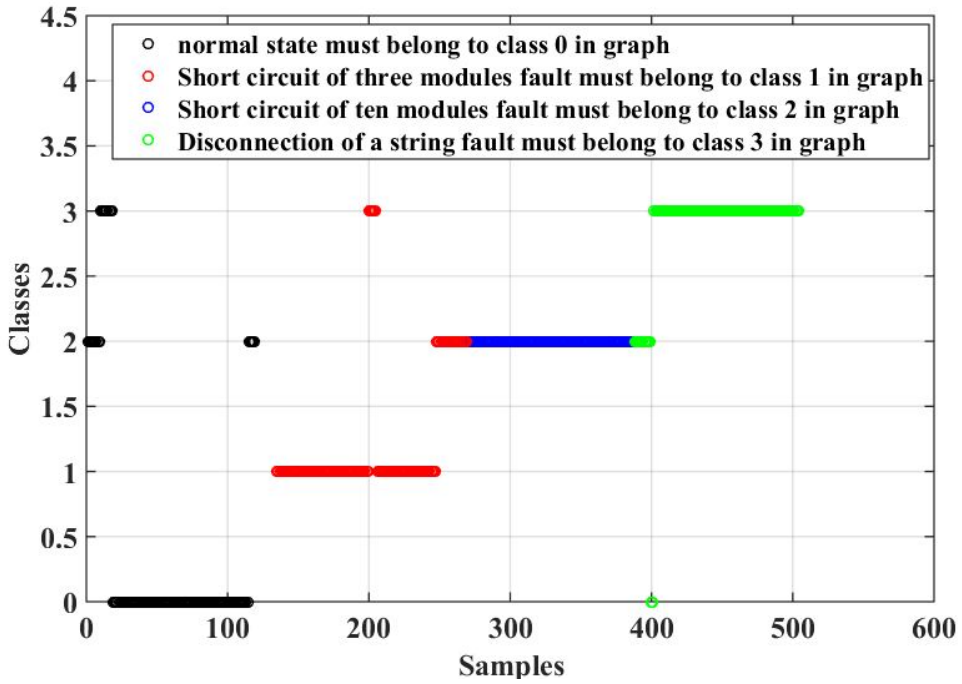

Figure. 3.12: Fault detection diagnosis results using the modified KNN algorithm

Figure 3.12 gives the obtained classification results using the modified KNN-based classifier. The proposed algorithm can classify most of the presented data into their corresponding classes. There are a few cases where it is failed to correctly classify them because of two reasons: The first reason is that this data is located within the bounds of classes, which makes it relatively difficult for the algorithm to classify. As for the second reason, it is that some of this data are considered as outliers.

3.5 Conclusion

This chapter has proposed an algorithm that is inspired from the classical KNN to detect and diagnose faults in PVSs. The proposed algorithm consists of two steps: 1) Generate a dataset for healthy state and three faulty operations using Matlab, 2) Identify and recognize data and its corresponding classes.

For the KNN-based classifier, it mainly depends on calculating all the distances between any new data and all the data in the dataset. In the proposed algorithm, the computing time is reduced, so that the distance between any new data and the center of each hyper-sphere is computed. This development, in addition to the use of the giza pyramids construction algorithm, significantly contribute to improve the accuracy, the precision, the recall, as well as the time of fault detection and diagnosis. In addition to

this, the proposed algorithm was compared with three other algorithms: the SVM, the DT, and the RF based classifiers.

The developed algorithm was tested and evaluated to detect and diagnose three different faults that are: short circuit of three modules, short circuit of ten modules, and a specific string disconnection. The proposed strategy can be easily extended to include other faults by increasing the number of classifiers.

CHAPTER 4

EUCLIDEAN DISTANCE-BASED TREE ALGORITHM FOR FAULT DETECTION AND DIAGNOSIS IN PHOTOVOLTAIC SYSTEMS

4.1 Introduction

This chapter presents a new methodology for fault detection and diagnosis in photovoltaic systems using a novel Euclidean distance-based tree algorithm.

Firstly, the proposed Euclidean-based decision tree classification algorithm is introduced. Then, the utilized database and the considered faults are described. The fault detection and diagnosis procedure, as well as the obtained results, are presented and discussed in this chapter.

4.2 Euclidean-based decision tree classification algorithm

Despite the similarities between the proposed algorithm and the decision trees in their data splitting approach, the key distinction lies in using the Euclidean distance for partitioning data instead of the Gini index. Unlike the decision tree, which requires the use of the Gini index to split the data, this algorithm mainly relies on computing distances between an arbitrary point in the space and the entire data set. Then, the minimum and the maximum distances of each class are extracted and used to split the data into different classes.

Initially, a training dataset, comprising values for N features for each of the two classes (class 0 and class 1), is created. Then, the following steps are performed:

a) Choose an arbitrary point $(x_1, x_2, ..., x_N)$ in an N-dimensional space.

b) Using equations 4.1 and 4.2, compute the Euclidean distances between the chosen point and all samples within the training dataset for each respective class:

$$dist_i^0 = \sqrt{\left(x_1 - x_{i1}^0\right)^2 + \left(x_2 - x_{i2}^0\right)^2 + \ldots + \left(x_N - x_{iN}^0\right)^2} \quad i = 1, 2, \ldots, n \tag{4.1}$$

$$dist_i^1 = \sqrt{\left(x_1 - x_{i1}^1\right)^2 + \left(x_2 - x_{i2}^1\right)^2 + \ldots + \left(x_N - x_{iN}^1\right)^2} \quad i = 1, 2, \ldots, m \tag{4.2}$$

where:
$(x_{i1}^0, x_{i2}^0, ..., x_{in}^0)$ and $(x_{i1}^1, x_{i2}^1, ..., x_{im}^1)$ represent the i^th samples of class 0 and class 1, respectively. n denotes the number of samples in class 0, while m denotes the number of samples in class 1.

c) Determine the minimum and maximum distances for each class:

$$min_0 = \min_{i=1,2,...,n} \left(dist_i^0\right) \tag{4.3}$$

$$max_0 = \max_{i=1,2,...,n} \left(dist_i^0\right) \tag{4.4}$$

$$min_1 = \min_{i=1,2,...,m} \left(dist_i^1\right) \tag{4.5}$$

$$max_1 = \max_{i=1,2,...,m} \left(dist_i^1\right) \tag{4.6}$$

d) Among the following five cases, one may arise:

- case 1: $min_0 < min_1 < max_0 < max_1$
  - Training samples having distances within the interval $[min_0, min_1[$ belong to class 0 (pure data in class 0).
  - Training samples having distances within the interval $]max_0, max_1]$ belong to class 1 (pure data in class 1).

    - Training samples having distances within the interval $[min_1, max_0]$ can not be classified, therefore another random point must be chosen for their classification.

- case 2: $min_1 < min_0 < max_1 < max_0$
    - Training samples having distances within the interval $[min_1, min_0[$ belong to class 1 (pure data in class 1).
    - Training samples having distances within the interval $]max_1, max_0]$ belong to class 0 (pure data in class 0).
    - Training samples having distances within the interval $[min_0, max_1]$ can not be classified, therefore another random point must be chosen for their classification.
- case 3: $min_0 < min_1 < max_1 < max_0$
    - Training samples having distances within the interval $[min_0, min_1[$ or $]max_1, max_0]$ belong to class 0.
    - Training samples having distances within the interval $[min_1, max_1]$ can not be classified, therefore another random point must be chosen for their classification.
- case 4: $min_1 < min_0 < max_0 < max_1$
    - Training samples having distances within the interval $[min_1, min_0[$ or $]max_0, max_1]$ belong to class 1.
    - Training samples having distances within the interval $[min_0, max_0]$ can not be classified, therefore another random point must be chosen for their classification.
- case 5: $min_0 < max_0 < min_1 < max_1$ or $min_1 < max_1 < min_0 < max_0$
    - Training samples having distances within the interval $[min_0, max_0]$ belong to class 0.
    - Training samples having distances within the interval $[min_1, max_1]$ belong to class 1.

e) If the case that occurred in the previous step is case 1, 2, 3, or 4:

- Choose another random point $(x_1, x_2, \ldots, x_N)$.
- Using equations 4.1 and 4.2, compute the Euclidean distances between the chosen point and the unclassified samples within the training dataset for each respective class.

– Go to step c).

f) The algorithm iterates through steps (c) to (e) until all data is classified (case 5) or the stopping criterion is met. It employs early stopping as its stopping criterion to effectively mitigate overfitting without compromising the accuracy of the algorithm [69–71].

Figure 4.1 provides a graphical illustration of the proposed algorithm depicting a given possible situation.

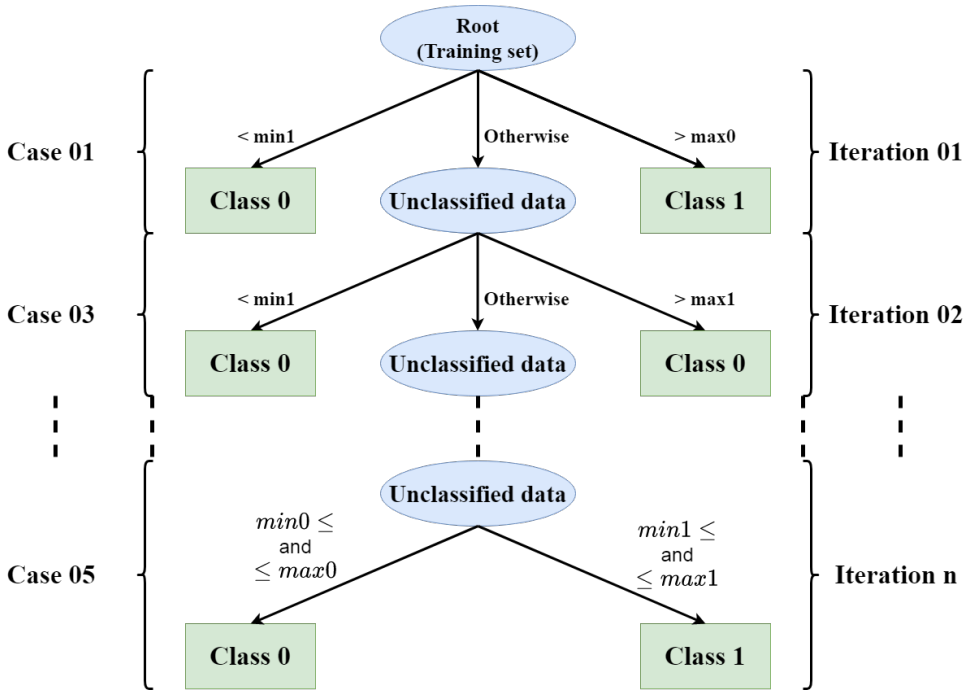

Figure. 4.1: Graphical illustration of the proposed algorithm

The flowchart of the algorithm is given in figure 4.2.

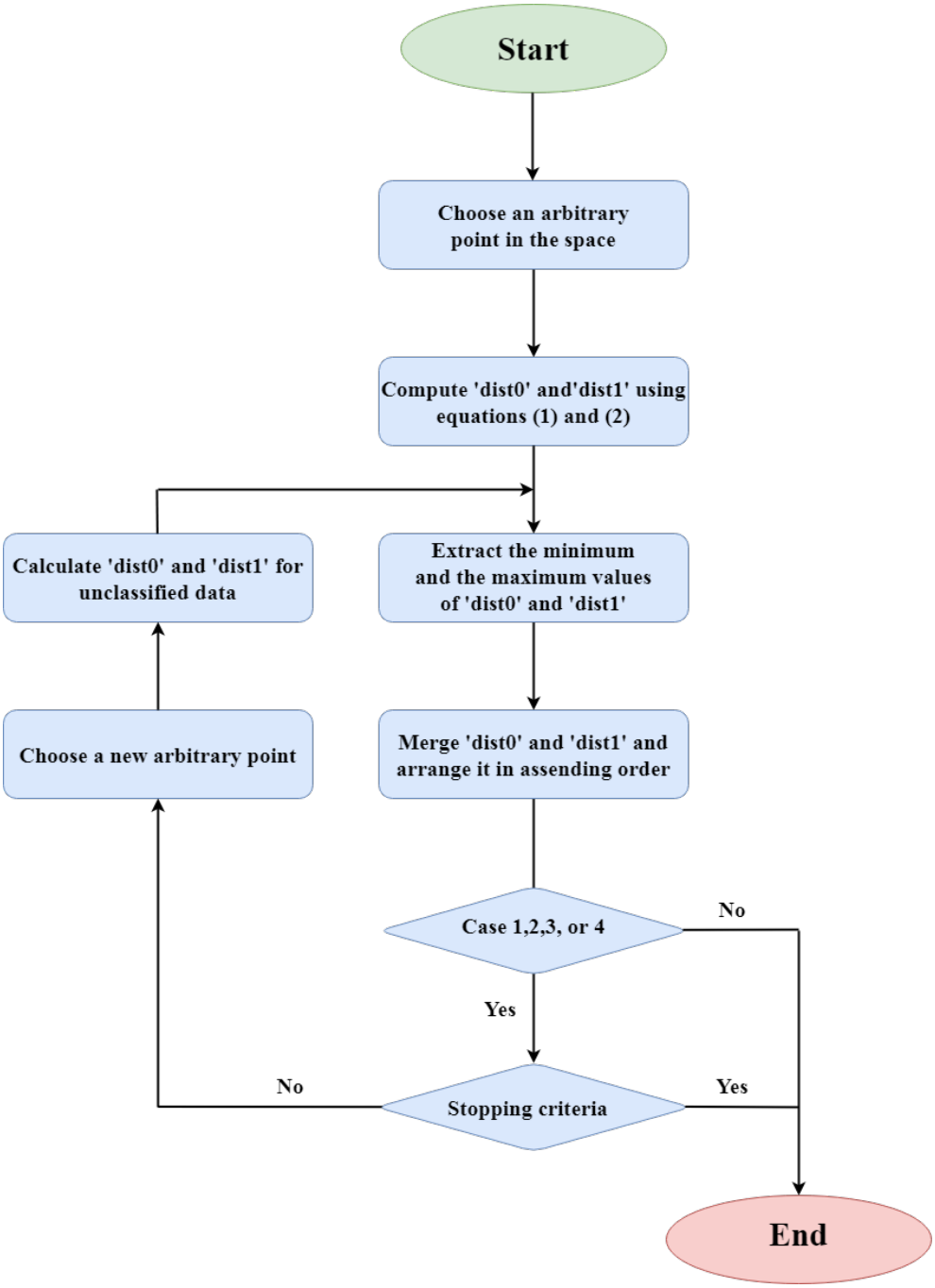

Figure. 4.2: Flowchart of the proposed algorithm

The pseudo code of the proposed algorithm is given below:

Algorithm 1 Pseudo code of the Euclidean Distance-Based Tree Algorithm

STEP (a): Generate a random point.
STEP (b): Using (eq.1) and (eq.2), calculate the distances $dist_i^0$ and $dist_j^1$,$(i = 1, 2, ..., n.j = 1, 2, ..., m)$.
STEP (c): Find $min0, max0, min1, max1$, the minimal and maximal distances of each class.
STEP (d): Store the computed distances into a vector named $dist$ and organize it in ascending order.
STEP (e):
$c = 1$ (the counter for unclassified data).
if $min_0 < min_1 < max_0 < max_1$ (case 1) **then**
  for $i = 1$ to k $(k = n + m)$ **do**
    if $min_0 \leq dist(i) < min_1$ **then**
      The point associated to $dist(i)$ belongs to class 0.
    else if $max_0 < dist(i) \leq max_1$ **then**
      The point associated to $dist(i)$ belongs to class 1.
    else if $(min_1 \leq dist(i) \leq max_0)$ **then**
      $Unclassified(c:) = trainingset(i:)$
      Increment c
    end if
  end for
  $trainingset = Unclassified.$
  if the stopping criteria is not verified **then**
    Choose a new arbitrary point.
    Calculate the distances $dist_i^0$ and $dist_j^1$ for unclassified data.
    Go to step (c).
  else
    Go to step (f).
  end if
end if
if $min_1 < min_0 < max_1 < max_0$ (case 2) **then**
  for $i = 1$ to k **do**
    if $min_1 \leq dist(i) < min_0$ **then**
      The point associated to $dist(i)$ belongs to class 1.
    else if $max_1 < dist(i) \leq max_0$ **then**
      The point associated to $dist(i)$ belongs to class 0.
    else if $(min_0 \leq dist(i) \leq max_1)$ **then**
      $Unclassified(c:) = trainingset(i:)$
      Increment c
    end if
  end for
  $trainingset = Unclassified.$

if the stopping criteria is not verified **then**
  Choose a new arbitrary point.
  Calculate the distances $dist_i^0$ and $dist_j^1$ for unclassified data.
  Go to step (c).
else
  Go to step (f).
end if
end if
if $min_0 < min_1 < max_1 < max_0$ (case 3) **then**
for $i = 1$ to k **do**
if $min_0 \leq dist(i) < min_1$ or $max_1 < dist(i) \leq max_0$ **then**
  The point associated to $dist(i)$ belongs to class 0.
else if $(min_1 \leq dist(i) \leq max_1)$ **then**
  $Unclassified(c:) = trainingset(i:)$
  Increment c
end if
end for
$trainingset = Unclassified.$
if the stopping criteria is not verified **then**
  Choose a new arbitrary point.
  Calculate the distances $dist_i^0$ and $dist_j^1$ for unclassified data.
  Go to step (c).
else
  Go to step (f).
end if
end if
if $min_1 < min_0 < max_0 < max_1$ (case 4) **then**
for do$i = 1$ to k.
if $min_1 \leq dist(i) < min_0$ or $max_0 < dist(i) \leq max_1$ **then**
  The point associated to $dist(i)$ belongs to class 1.
else if $(min_0 \leq dist(i) \leq max_0)$ **then**
  $Unclassified(c:) = trainingset(i:)$
  Increment c
end if
end for
$trainingset = Unclassified.$
if the stopping criteria is not verified **then**
  Choose a new arbitrary point.
  Calculate the distances $dist_i^0$ and $dist_j^1$ for unclassified data.
  Go to step (c).
else
  Go to step (f).
end if
end if

if $min_0 < max_0 < min_1 < max_1$ or $min_1 < max_1 < min_0 < max_0$ (case 5) **then**
  for do$i = 1$ to k.
    if $min_0 \leq dist(i) \leq max_0$ **then**
      The point associated to $dist(i)$ belongs to class 0.
    end if
    if $min_1 \leq dist(i) \leq max_1$ **then**
      The point associated to $dist(i)$ belongs to class 1.
    end if
  end for
  Go to step (f).
end if
Step (f): End (all data is classified or the stopping criterion is met).

4.3 Dataset description

The developed methodology requires four attributes, namely: solar irradiance, temperature, and coordinates of MPP (I_{mpp}, V_{mpp}). The PVA used to generate the dataset, for both healthy and faulty states, consists of two parallel strings. Each string comprises fifteen series-connected $Isofoton$ PVM ($106W/12V$). The Simulink/MATLAB platform is utilized to simulate the current (I_{mpp}) and voltage (V_{mpp}) at the MPP of this PVA under both healthy and faulty states, considering various values of cell temperature (T) and irradiance (G). Through this simulation, 753 samples are generated for each of the considered classes, consisting of the four physical quantities (T, G, I_{mpp}, V_{mpp}). In this study, besides the normal operating state, six faulty states are considered. These states and their corresponding labels are given in table 1.

Table 4.1: Operating states and their labels.

Class name	Label
Normal operation	Class 0
Short circuit of three modules	Class 1
Short circuit of ten modules	Class 2
String disconnection	Class 3
String with 25% of partial shading	Class 4
String with 50% of partial shading	Class 5
String with 75% of partial shading	Class 6

As shown in figure 4.3, utilizing the I_{mpp} as a feature makes it possible to distinguish between three faults: string disconnection, string disconnection with 50% shading, and string disconnection with 75% shading. Meanwhile, in figure 4.4, it appears that the V_{mpp} feature can be used to classify faults such as string disconnection with 25% shading, short circuits of three modules, and short circuits of ten modules. To detect the healthy state class, both I_{mpp} and V_{mpp} features must be used simultaneously.

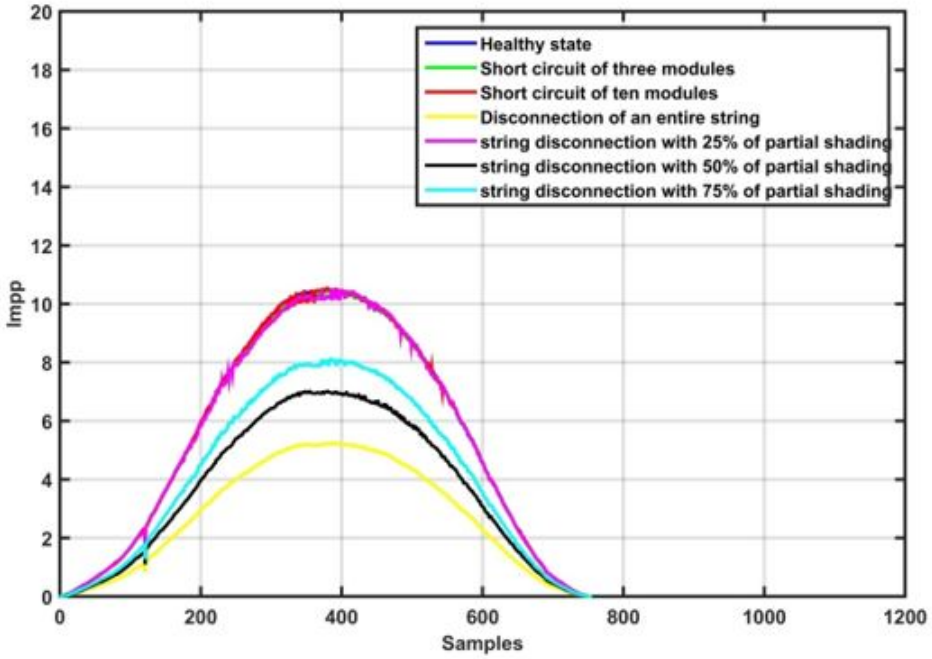

Figure. 4.3: I_{mpp} for various operating states of the PVA

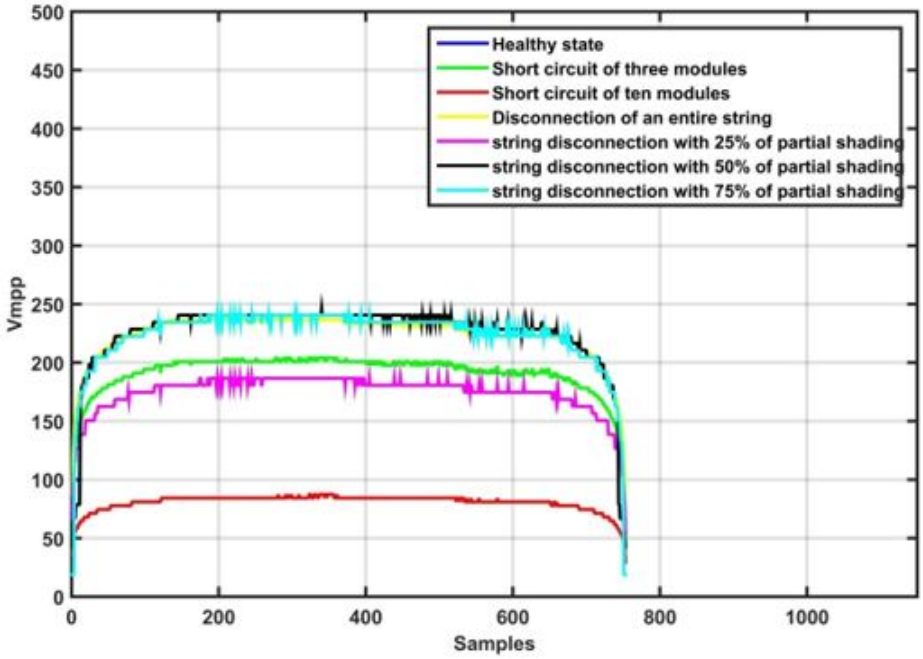

Figure. 4.4: V_{mpp} for various operating states of the PVA

4.4 Fault detection and diagnosis methodology

The developed procedure for fault detection and diagnosis employs the Euclidean distance-based tree algorithm, given above, to classify the considered faults. Figure 4.5 presents the flowchart of the classification strategy employed in this study. As depicted, the algorithm transforms the multi-classification problem into a binary one by isolating a single class at a time, starting from class 0 and ending with class 6. Therefore, six classifiers must be designed for the seven considered classes.

The first classifier separates class 0 from the remaining classes, while the second classifier separates class 1 from the others. The third classifier isolates class 2 from classes 3, 4, 5, and 6. Subsequently, the fourth classifier distinguishes class 4 from classes 3, 5, and 6. The fifth classifier separates class 3 from classes 5 and 6, and the final classifier distinguishes between classes 5 and 6.

Each classifier is designed based on the previously described classification algorithm and utilizes the four specified features (T, G, I_{mpp}, V_{mpp}).

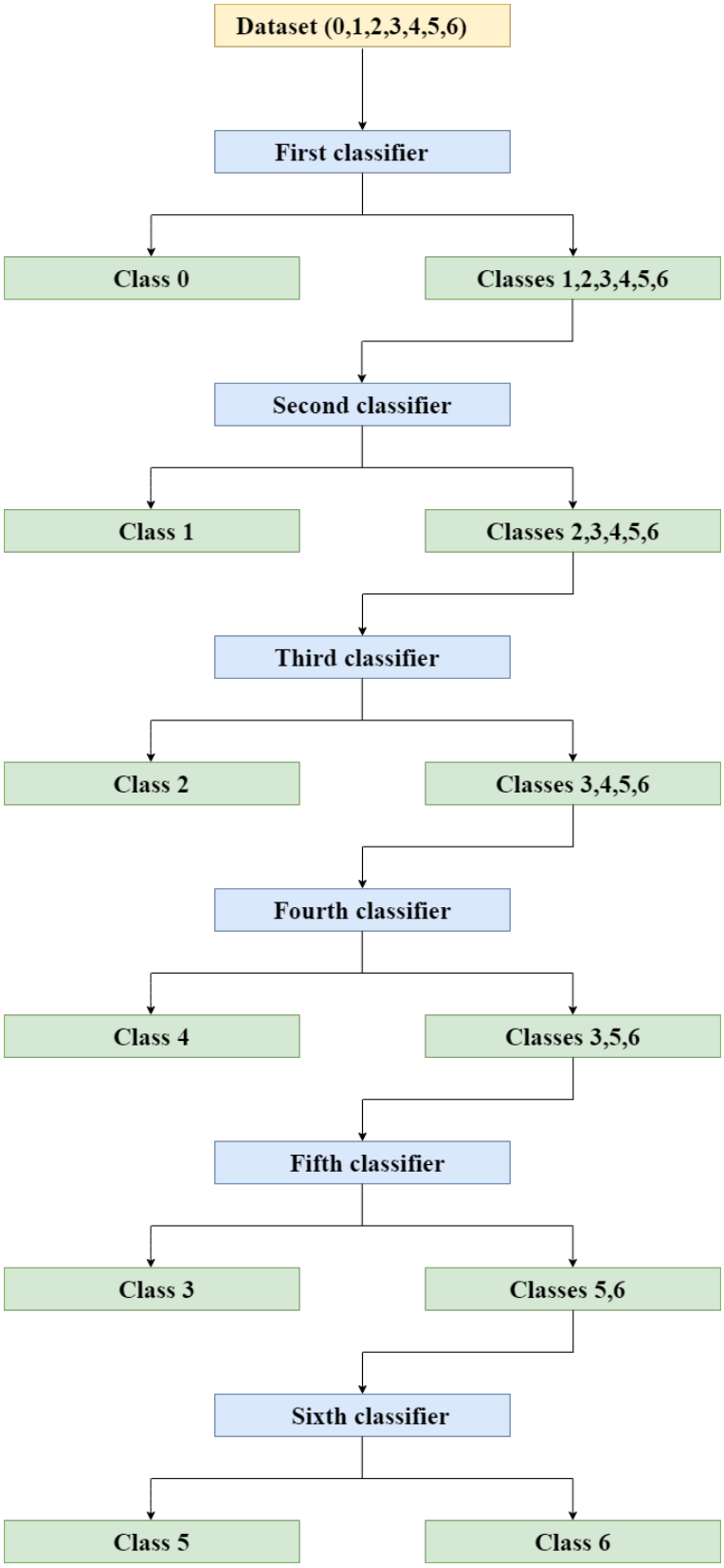

Figure. 4.5: Fault detection and diagnosis flowchart

4.5 Results and discussion

The developed procedure for fault detection and diagnosis is implemented and applied to classify a dataset comprising seven distinct classes (healthy state and six faulty states. A comparison study between the developed procedure and the faults detection and diagnosis methodologies based on Support Vector Machine, Decision Tree, Random Forest, and K-Nearest Neighbors algorithms is conducted.

To evaluate the performance of the different algorithms considered in this study, a mathematical tool called confusion matrix is used. The elements of this matrix are used to compute the accuracy, the precision, and the recall metrics. This matrix is well explained in chapter 03.

4.5.1 Training the fault detection and diagnosis model using the proposed algorithm

Like any other statistical learning algorithm, the proposed algorithm firstly needs to be trained using a training dataset. Following training, its performance is evaluated using a separate testing set. The dataset is partitioned into two subsets: the training set comprises 87% of the global dataset, while the testing set encompasses 13% of the global dataset. As mentioned earlier, six classifiers are necessary to detect and diagnose the specified faults. To mitigate overfitting effectively without compromising the algorithm's accuracy, the early stopping criterion is employed to stop the training process of each classifier.

The accuracy metric for each classifier is calculated at every iteration and illustrated in figures 4.6 through 4.11. As can be seen, for all classifiers the accuracy value increases over iterations. Classifiers 1 to 6 of the trained model require 23, 9, 4, 6, 16, and 17 steps, respectively, to separate a class from the other classes.

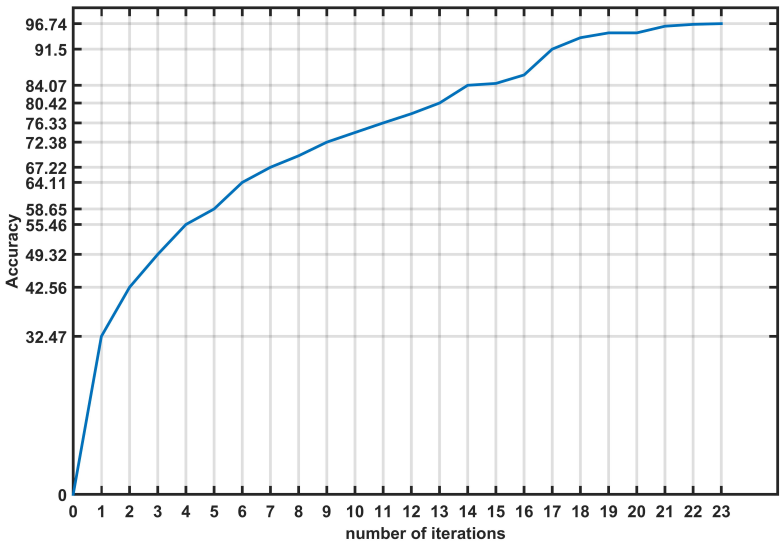

Figure. 4.6: Evolution of accuracy for the first classifier

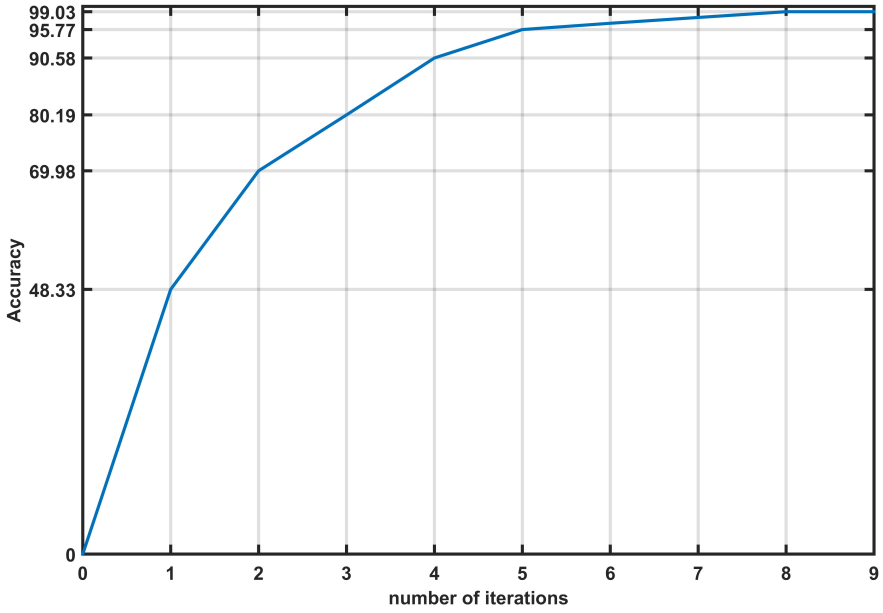

Figure. 4.7: Evolution of accuracy for the second classifier

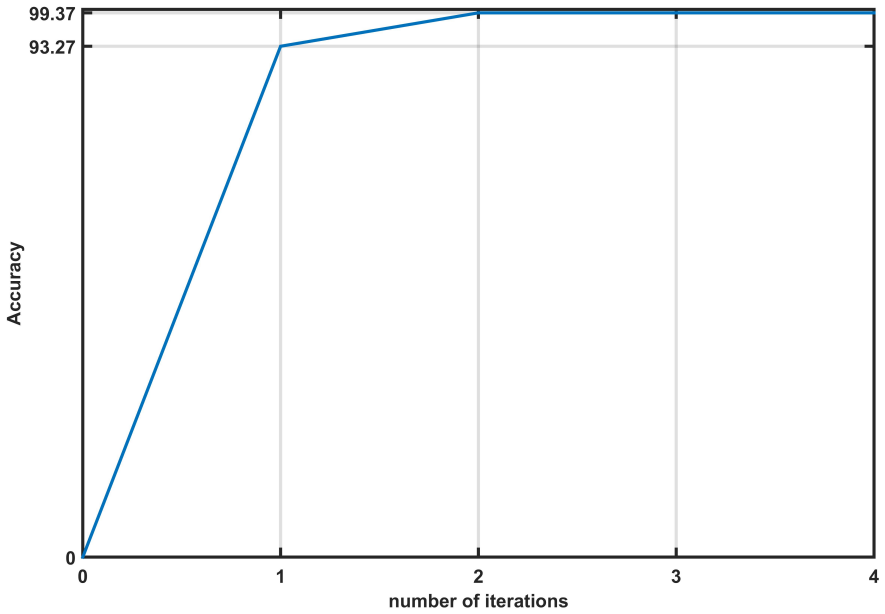

Figure. 4.8: Evolution of accuracy for the third classifier

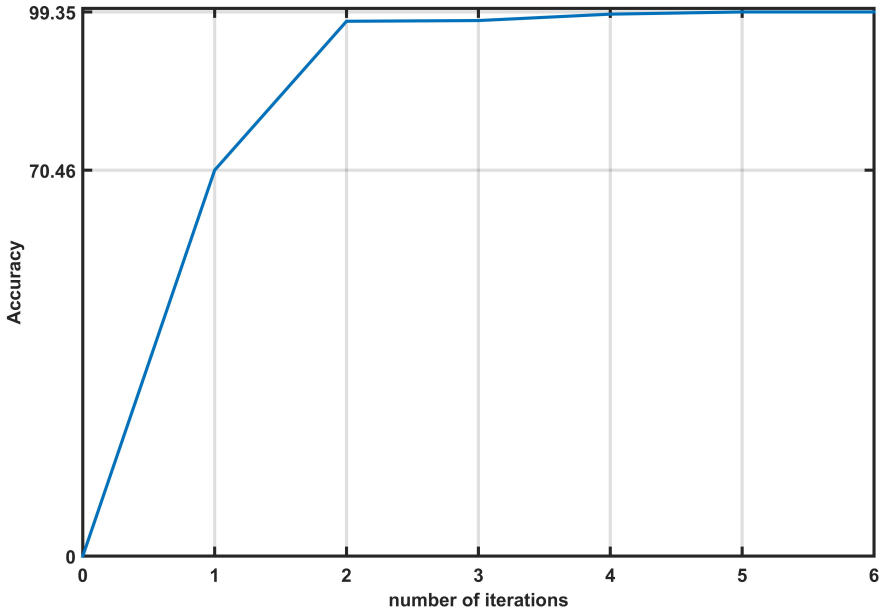

Figure. 4.9: Evolution of accuracy for the forth classifier

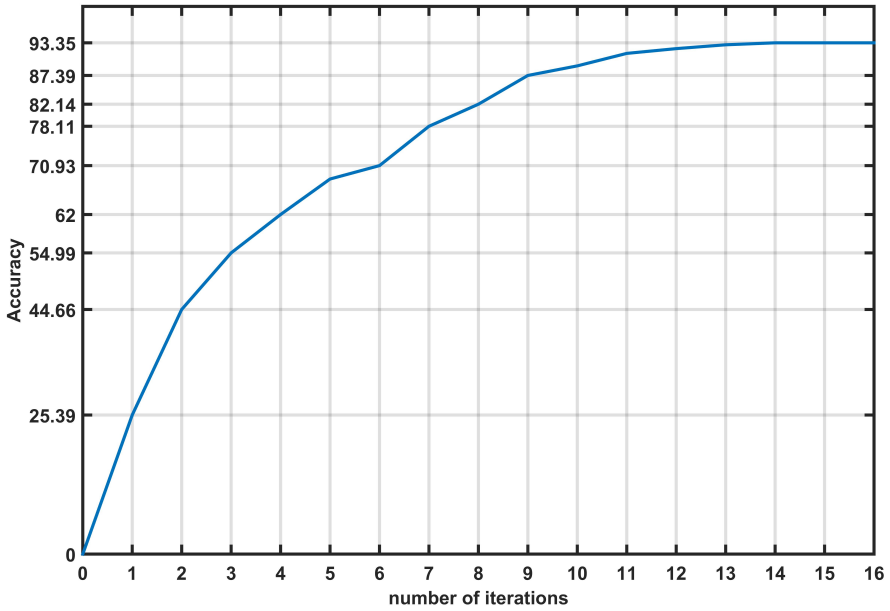

Figure. 4.10: Evolution of accuracy for the fifth classifier

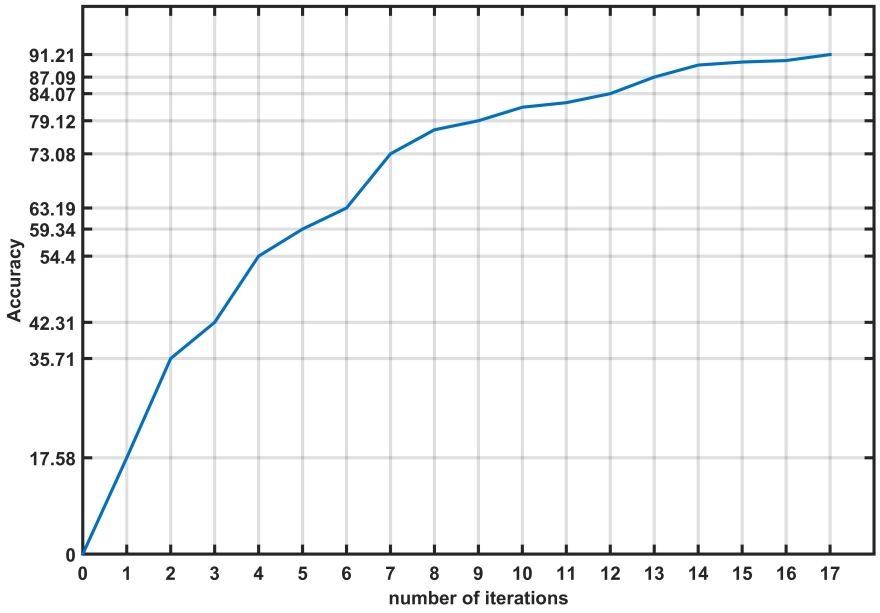

Figure. 4.11: Evolution of accuracy for the sixth classifier

4.5.2 Evaluating the performance of the obtained model using the proposed algorithm

To assess the effectiveness of the proposed approach, the obtained model is evaluated using the average values of the aforementioned metrics. Higher values of accuracy, precision, and recall indicate better performance of the proposed methodology. These metric values are computed from the test dataset and gathered in tables 3 and 4.

Table 3 displays the confusion matrix values corresponding to each classifier within the obtained model. Based on these values, the accuracy, precision, and recall metrics are computed for each classifier and presented in table 4. Additionally, the last row of table 4 gives the average values for accuracy, precision, and recall metrics of the obtained model.

Table 4.2: Confusion matrices for the obtained model.

	TP	FN	FP	TN
Classifier 1	1107	19	29	168
Classifier 2	947	3	4	178
Classifier 3	762	1	3	183
Classifier 4	570	3	1	190
Classifier 5	354	13	10	179
Classifier 6	177	20	5	155

Table 4.3: Metrics values for the obtained model.

	Accuracy (%)	Precision (%)	Recall (%)
Classifier 1	97	97	98
Classifier 2	99	100	100
Classifier 3	100	100	100
Classifier 4	99	100	99
Classifier 5	96	97	99
Classifier 6	93	97	90
Average values	**97.33**	**99**	**97**

4.5.3 Comparative study using various machine learning algorithms

In this comparative study, the fault detection and diagnosis model depicted in the flowchart of figure 4.5 is constructed using various statistical methods, namely the SVM algorithm [23], the DT algorithm [12, 27], the RF algorithm [28, 29, 2], and the KNN algorithm [38–40].

The confusion matrices for the obtained model using the aforementioned algorithms are provided in table 5, while table 6 presents the values for accuracy, precision, and recall, along with the average values of these metrics.

	SVM				DT				RF				KNN			
	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN
Classifier 1	1104	15	165	34	1113	6	23	176	1119	8	15	176	1107	12	155	44
Classifier 2	1086	0	122	61	950	3	3	180	950	1	1	180	1078	1	1	182
Classifier 3	1021	0	103	84	765	2	0	186	764	0	1	186	892	0	0	187
Classifier 4	927	0	107	90	566	3	0	196	568	1	0	196	696	0	0	196
Classifier 5	796	57	131	50	364	10	7	185	371	4	3	185	508	1	167	20
Classifier 6	631	112	84	90	163	29	172	202	157	36	175	202	105	73	201	296

Table 4.4: Confusion matrices for the obtained model using the four algorithms.

Table 4.5: Metrics values for the obtained model using the four algorithms.

	SVM			DT			RF			KNN		
	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)
Classifier 1	86	87	99	98	98	99	99	99	100	87(k = 7)	88	99
Classifier 2	90	90	100	99	100	100	100	100	100	100(k = 1)	100	100
Classifier 3	91	91	100	100	100	100	100	100	100	100(k = 3)	100	100
Classifier 4	90	90	100	100	100	99	100	100	100	100(k = 2)	100	100
Classifier 5	82	86	93	97	98	97	99	99	99	76(k = 35)	75	100
Classifier 6	78	88	84	64	87	54	63	85	53	59(k = 1)	80	60
Average values	85.33	88.65	96	93	97.16	91.50	93.50	97.16	92	87	90.50	93.16

To compare the performance of the proposed algorithm for fault detection and diagnosis with the performance of the models based on the SVM, DT, RF, and KNN algorithms, a summary of the average values of the different metrics is provided in table 7. It can be observed from this table that the proposed method achieves the highest values for the three metrics, indicating its superior performance compared to the other techniques.

Table 4.6: Metrics average values.

	Accuracy (%)	Precision (%)	Recall (%)
The proposed algorithm	**97.33**	**98.66**	**97.5**
SVM	85.33	88.65	96
DT	93	97.16	91.50
RF	93.50	97.16	92
KNN	87	90.50	93.16

Figures 4.12 to 4.16 display the fault detection and diagnosis results using the proposed algorithm-based model and those based on the SVM, DT, RF, and KNN algorithms, respectively. It can be seen from these figures that the smallest number of incorrectly classified data is obtained in the case of both the RF algorithm-based model and the proposed algorithm-based model. The models fail to correctly classify all data due to data overlap and overfitting issues.

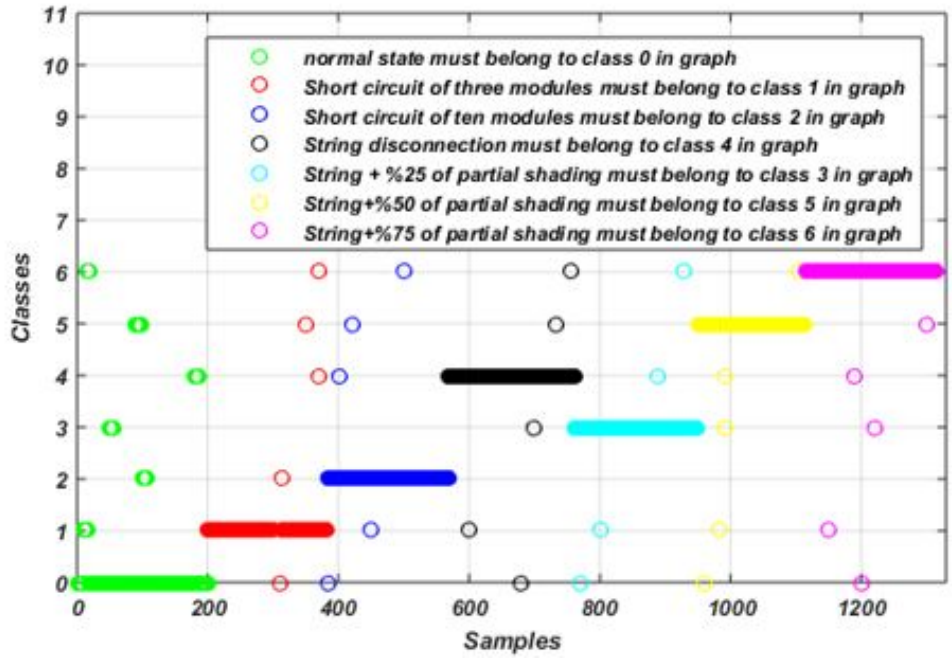

Figure. 4.12: Fault detection and diagnosis results using the proposed algorithm-based model

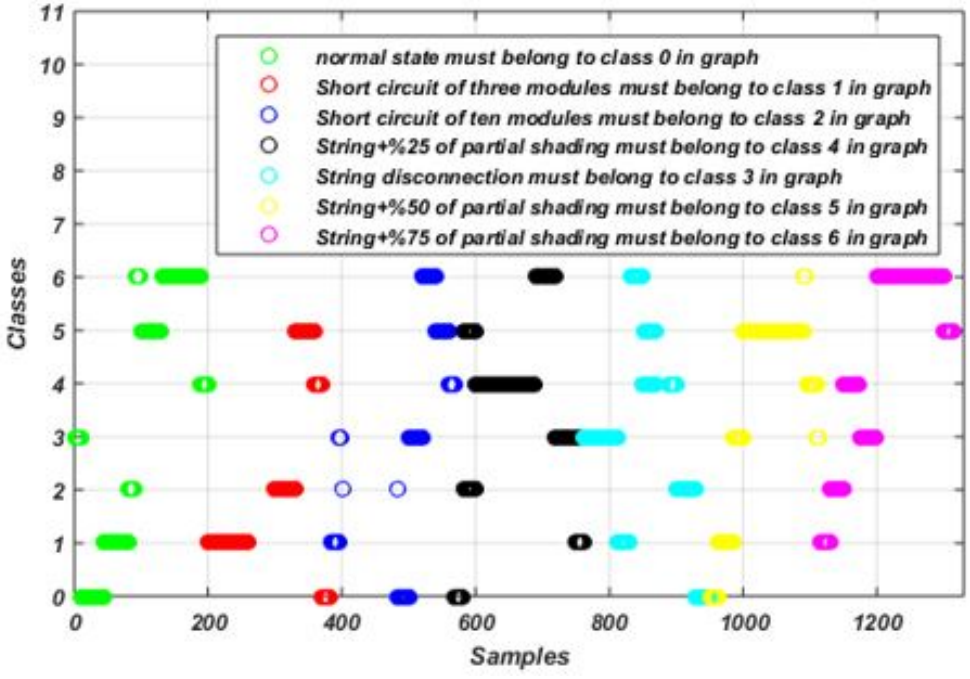

Figure. 4.13: Fault detection and diagnosis results using the SVM algorithm-based model

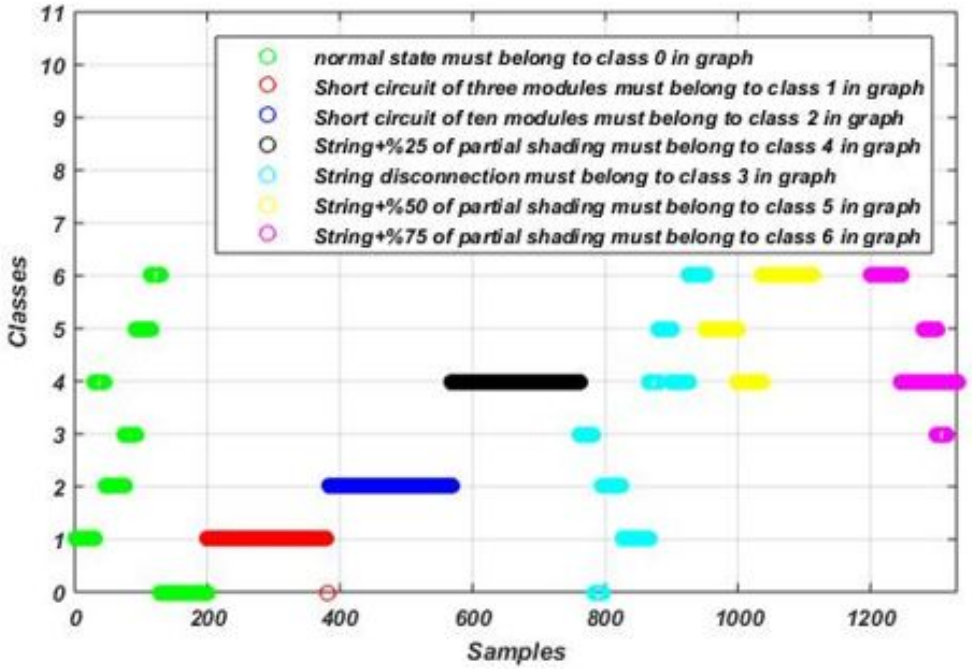

Figure. 4.14: Fault detection and diagnosis results using the DT algorithm-based model

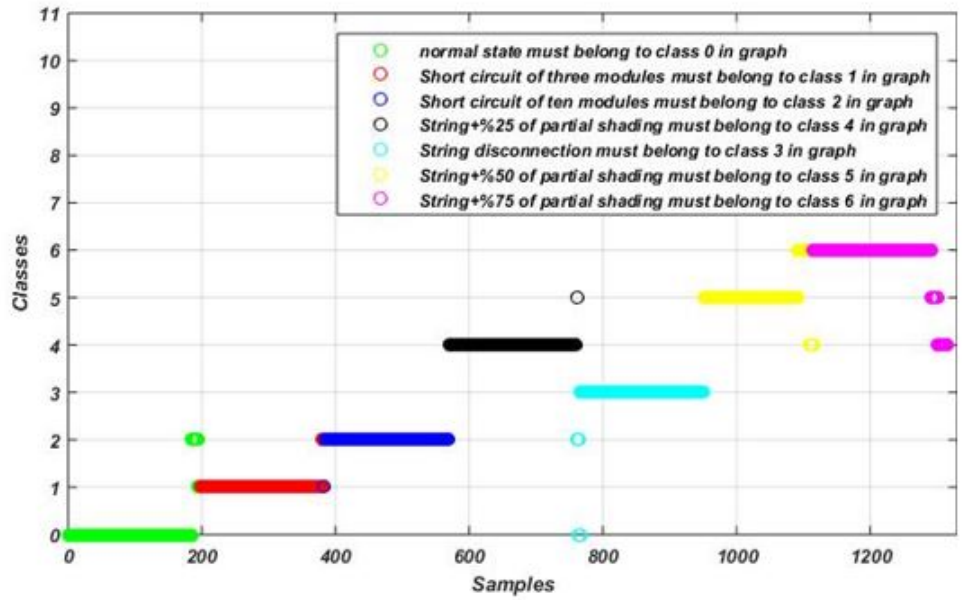

Figure. 4.15: Fault detection and diagnosis results using the RF algorithm-based model

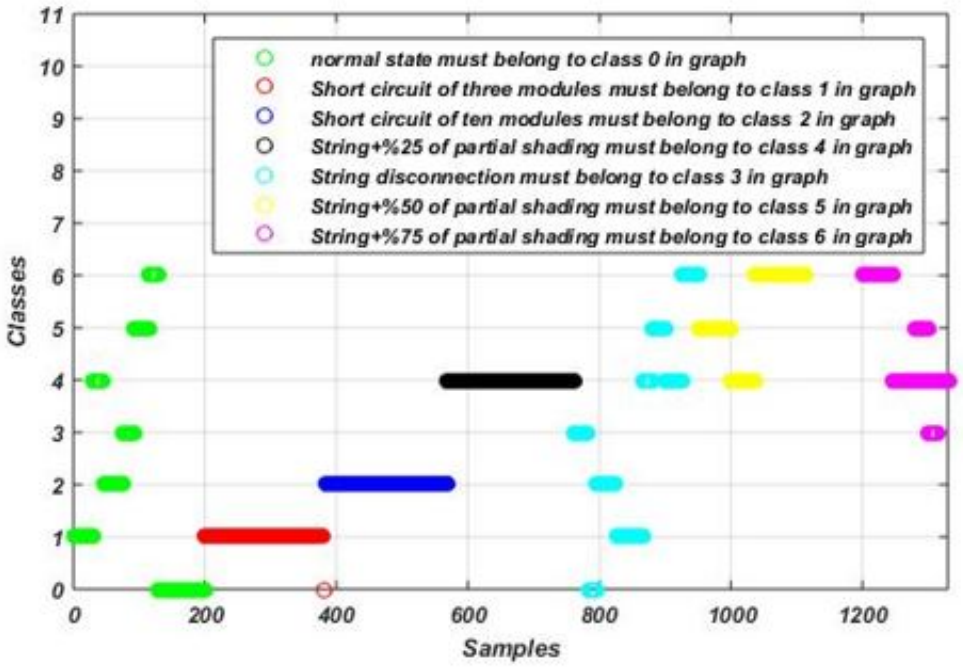

Figure. 4.16: Fault detection and diagnosis results using the KNN algorithm-based model

4.6 Conclusion

In this chapter, an enhanced approach was proposed for identifying and diagnosing PVA faults. A comparative study was conducted between the proposed algorithm-based model and models based on four statistical learning algorithms: SVM, DT, RF, and KNN algorithms. Unlike the decision tree algorithm, which uses the Gini index to split the data onto two classes, the proposed algorithm calculates Euclidean distances between an arbitrary point and the dataset samples. It then utilizes the minimal and maximal distances to separate the samples belonging to each class.

In this study, four features, namely: cell temperature, irradiance, current and voltage of the maximum power point were utilized. The proposed methodology effectively distinguishes the normal operating condition from other abnormal states, achieving a classification accuracy of 97%. The comparative investigation demonstrated that the proposed approach outperformed the other methods considered in this work in terms of accuracy, precision, and recall.

By increasing the number of classifiers, the proposed technique can be easily extended to encompass additional faults.

CONCLUSION AND FUTURE WORK

The main objective of this thesis was to develop effective methods, based on meta-heuristic optimization algorithms and machine learning algorithms, for the detection and diagnosis of faults in a photovoltaic system. Several types of faults on the DC side of PV system, including short circuits of several PV modules, PV string disconnections, and PV string disconnections with varying degrees of partial shading, had to be considered.

The initial task involves deriving the five electrical parameters of the ODM. To accomplish this, an effective procedure based on the EPC and GPC algorithms has been developed for identifying the optimal values of these parameters. These algorithms were chosen for their adeptness in addressing optimization problems, their swift convergence rates, and their ease of implementation in real-time scenarios. The effectiveness of the developed procedure for parameters identification has been confirmed through experimental validation.

Subsequently, the ODM with the obtained values of parameters was employed to formulate a proficient strategy for maximum power point estimation. This strategy underwent experimental validation using experimental measurements. The efficiency evaluation was specifically conducted under clear sky conditions.The comparison between the two meta heuristic algorithms indicates the superiority of the identification procedure based on the GPC algorithm over that based on the EPC algorithm in terms of accuracy. Consequently, the GPC algorithm was employed in conjunction with the first FDD approach developed in this thesis.

In the first developed faults detection and diagnosis strategy, a straightforward modification was introduced to the conventional KNN algorithm. This modification

involves computing the distance between any new data point and the center of each hyper sphere. The modified KNN algorithm with conjunction of the GPC algorithm have been used to build an efficient for faults detection and diagnosis in the DC side of PV systems. To assess the efficiency of the proposed approach, a comparative study was conducted, including the classical version of the KNN, support vector machine, decision tree, and random forest algorithms. The results of the carried-out study have demonstrated the remarkable superiority of the proposed strategy over that based on these algorithms.

A second strategy, based on a novel Euclidean distance-based tree algorithm, for faults detection and diagnosis in the DC side of PV systems has been developed in this thesis. A comparison study between the developed faults detection and diagnosis methodology and the methodology based on support vector machine, decision tree, random forest, and KNN algorithms has been carried out.The obtained results demonstrate the high efficiency and effectiveness of the proposed methodology, with a classification accuracy reaching 97.33%.The comparative analysis has revealed that the faults detection and diagnosis approach that uses the Euclidean distance-based tree algorithm outperforms the approaches that use the other mentioned algorithms in terms of accuracy, precision, and recall.

Although the fault detection and diagnosis strategies implemented in this thesis have yielded promising results, they also prompt several questions and suggest avenues for future research. Specifically, the following aspects warrant thorough consideration:

- Extending the applicability of the two algorithms to identify diverse sets of faults, whether occurring on the DC or AC side.
- Exploring alternative classification strategies to assess the performance of the two algorithms.
- The PCA algorithm can also be integrated with both algorithms to assist in identifying relevant features.

REFERENCES

[1] W. Chine, *Contribution au diagnostic des défauts dans les systèmes photovoltaïques.* University of Jijle.

[2] A. Drews, A. De Keizer, H. G. Beyer, E. Lorenz, J. Betcke, W. Van Sark, W. Heydenreich, E. Wiemken, S. Stettler, P. Toggweiler, *et al.*, "Monitoring and remote failure detection of grid-connected pv systems based on satellite observations," *Solar energy*, vol. 81, no. 4, pp. 548–564, 2007.

[3] Z. Chen, F. Han, L. Wu, J. Yu, S. Cheng, P. Lin, and H. Chen, "Random forest based intelligent fault diagnosis for pv arrays using array voltage and string currents," *Energy Conversion and Management*, vol. 178, pp. 250–264, 2018.

[4] M. H. Rashid, *Power Electronics Devices, Circuits, and Applications.* Pearson Education.

[5] M. H. Qais, H. M. Hasanien, S. Alghuwainem, and A. S. Nouh, "Coyote optimization algorithm for parameters extraction of three-diode photovoltaic models of photovoltaic modules," *Energy*, vol. 187, p. 116001, 2019.

[6] A. Jäger-Waldau, "Snapshot of photovoltaics—february 2020," *Energies*, vol. 13, no. 4, p. 930, 2020.

[7] C. K. M. Khelil, B. Amrouche, A. soufiane Benyoucef, K. Kara, and A. Chouder, "New intelligent fault diagnosis (ifd) approach for grid-connected photovoltaic systems," *Energy*, vol. 211, p. 118591, 2020.

[8] A. Mellit, G. M. Tina, and S. A. Kalogirou, "Fault detection and diagnosis methods for photovoltaic systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 1–17, 2018.

[9] T. Takashima, J. Yamaguchi, K. Otani, K. Kato, and M. Ishida, "Experimental studies of failure detection methods in pv module strings," in *2006 IEEE 4th World Conference on Photovoltaic Energy Conference*, vol. 2, pp. 2227–2230, IEEE, 2006.

[10] L. Schirone, F. Califano, U. Moschella, and U. Rocca, "Fault finding in a 1 mw photovoltaic plant by reflectometry," in *Proceedings of 1994 IEEE 1st World Conference on Photovoltaic Energy Conversion-WCPEC (A Joint Conference of PVSC, PVSEC and PSEC)*, vol. 1, pp. 846–849, IEEE, 1994.

[11] Y. Hu, B. Gao, X. Song, G. Y. Tian, K. Li, and X. He, "Photovoltaic fault detection using a parameter based model," *Solar Energy*, vol. 96, pp. 96–102, 2013.

[12] S. Moulahoum and R. Benkercha, "Fault detection and diagnosis based on c4.5 decision tree algorithm for grid connected pv system," vol. 173, 08 2018.

[13] M. K. Alam, F. Khan, J. Johnson, and J. Flicker, "A comprehensive review of catastrophic faults in pv arrays: types, detection, and mitigation techniques," *IEEE Journal of Photovoltaics*, vol. 5, no. 3, pp. 982–997, 2015.

[14] A. Mohammedi, N. Mezzai, D. Rekioua, and T. Rekioua, "Impact of shadow on the performances of a domestic photovoltaic pumping system incorporating an mppt control: A case study in bejaia, north algeria," *Energy Conversion and Management*, vol. 84, pp. 20–29, 2014.

[15] N. Gokmen, E. Karatepe, B. Celik, and S. Silvestre, "Simple diagnostic approach for determining of faulted pv modules in string based pv arrays," *Solar Energy*, vol. 86, no. 11, pp. 3364–3377, 2012.

[16] A. Mellit, G. M. Tina, and S. A. Kalogirou, "Fault detection and diagnosis methods for photovoltaic systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 1–17, 2018.

[17] Y.-Y. Hong and R. A. Pula, "Methods of photovoltaic fault detection and classification: A review," *Energy Reports*, vol. 8, pp. 5898–5929, 2022.

[18] J. Haney and A. Burstein, "Pv system operations and maintenance fundamentals," *Solar America Board for Codes and Standards Report*, 2013.

[19] S. R. Madeti and S. Singh, "A comprehensive study on different types of faults and detection techniques for solar photovoltaic system," *Solar Energy*, vol. 158, pp. 161–185, 2017.

[20] M. Cubukcu and A. Akanalci, "Real-time inspection and determination methods of faults on photovoltaic power systems by thermal imaging in turkey," *Renewable Energy*, vol. 147, pp. 1231–1238, 2020.

[21] T. Kirchartz, A. Helbig, W. Reetz, M. Reuter, J. H. Werner, and U. Rau, "Reciprocity between electroluminescence and quantum efficiency used for the characterization of silicon solar cells," *Progress in Photovoltaics: Research and Applications*, vol. 17, no. 6, pp. 394–402, 2009.

[22] M. W. Akram, G. Li, Y. Jin, X. Chen, C. Zhu, X. Zhao, A. Khaliq, M. Faheem, and A. Ahmad, "Cnn based automatic detection of photovoltaic cell defects in electroluminescence images," *Energy*, vol. 189, p. 116319, 2019.

[23] G. A. dos Reis Benatto, C. Mantel, S. Spataru, A. A. S. Lancia, N. Riedel, S. Thorsteinsson, P. B. Poulsen, H. Parikh, S. Forchhammer, and D. Sera, "Drone-based daylight electroluminescence imaging of pv modules," *IEEE Journal of Photovoltaics*, vol. 10, no. 3, pp. 872–877, 2020.

[24] J. Bachmann, C. Buerhop-Lutz, R. Steim, P. Schilinsky, J. A. Hauch, E. Zeira, *et al.*, "Highly sensitive non-contact shunt detection of organic photovoltaic modules," *Solar energy materials and solar cells*, vol. 101, pp. 176–179, 2012.

[25] J. Bauer, O. Breitenstein, and J.-M. Wagner, "Lock-in thermography: a versatile tool for failure analysis of solar cells," *Electronic Device Failure Analysis*, vol. 11, no. 3, pp. 6–12, 2009.

[26] Y. Hu, B. Gao, X. Song, G. Y. Tian, K. Li, and X. He, "Photovoltaic fault detection using a parameter based model," *Solar Energy*, vol. 96, pp. 96–102, 2013.

[27] B. Du, Y. He, Y. He, J. Duan, and Y. Zhang, "Intelligent classification of silicon photovoltaic cell defects based on eddy current thermography and convolution neural network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6242–6251, 2019.

[28] J. A. Tsanakas, G. Vannier, A. Plissonnier, D. L. Ha, and F. Barruel, "Fault diagnosis and classification of large-scale photovoltaic plants through aerial orthophoto thermal mapping," in *Proceedings of the 31st European Photovoltaic Solar Energy Conference and Exhibition*, vol. 2015, pp. 1783–1788, 2015.

[29] J. A. Tsanakas, D. Chrysostomou, P. N. Botsaris, and A. Gasteratos, "Fault diagnosis of photovoltaic modules through image processing and canny edge detection on field thermographic measurements," *International journal of sustainable energy*, vol. 34, no. 6, pp. 351–372, 2015.

[30] N. Khera, N. Rana, S. Narendiran, S. K. Sahoo, M. Balamurugan, S. P. Karthikeyan, and I. J. Raglend, "Design of charge controller for solar pv systems," in *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 149–153, IEEE, 2015.

[31] Y. E. A. Eldahab, N. H. Saad, and A. Zekry, "Enhancing the design of battery charging controllers for photovoltaic systems," *Renewable and Sustainable Energy Reviews*, vol. 58, pp. 646–655, 2016.

[32] E. Franklin, "Solar photovoltaic (pv) system components," *no. May*, pp. 1–8, 2018.

[33] A. Shrestha and A. Singh, "Manual for solar technician," 11 2016.

[34] Y. Zhao and R. Lyons Jr, "Line-line fault analysis and protection in pv arrays," *Tech Topics: Photovoltaic protection Note*, vol. 2, 2011.

[35] Y. Zhao and R. Lyons Jr, "Ground-fault analysis and protection in pv arrays," *Proc. Photovoltaic Protection*, pp. 1–4, 2011.

[36] F. Chan and H. Calleja, "Reliability: A new approach in design of inverters for pv systems," in *2006 IEEE International Power Electronics Congress*, pp. 1–6, IEEE, 2006.

[37] S. R. Madeti and S. Singh, "A comprehensive study on different types of faults and detection techniques for solar photovoltaic system," *Solar Energy*, vol. 158, pp. 161–185, 2017.

[38] S. K. Firth, K. J. Lomas, and S. J. Rees, "A simple model of pv system performance and its use in fault detection," *Solar energy*, vol. 84, no. 4, pp. 624–635, 2010.

[39] P. Ducange, M. Fazzolari, B. Lazzerini, and F. Marcelloni, "An intelligent system for detecting faults in photovoltaic fields," in *2011 11th International Conference on Intelligent Systems Design and Applications*, pp. 1341–1346, IEEE, 2011.

[40] A. Chouder and S. Silvestre, "Automatic supervision and fault detection of pv systems based on power losses analysis," *Energy conversion and Management*, vol. 51, no. 10, pp. 1929–1937, 2010.

[41] T. Takashima, J. Yamaguchi, K. Otani, K. Kato, and M. Ishida, "Experimental studies of failure detection methods in pv module strings," in *2006 IEEE 4th World Conference on Photovoltaic Energy Conference*, vol. 2, pp. 2227–2230, IEEE, 2006.

[42] T. Takashima, J. Yamaguchi, and M. Ishida, "Disconnection detection using earth capacitance measurement in photovoltaic module string," *Progress in Photovoltaics: Research and Applications*, vol. 16, pp. 669 – 677, 12 2008.

[43] E. Garoudja, F. Harrou, Y. Sun, K. Kara, A. Chouder, and S. Silvestre, "Statistical fault detection in photovoltaic systems," *Solar Energy*, vol. 150, pp. 485–499, 2017.

[44] M. Miwa, S. Yamanaka, H. Kawamura, and H. Ohno, "Diagnosis of a power output lowering of pv array with a (di/dv)-v characteristic," *Proceeding of IEEE 4th World Conference on Photovoltaic Energy Conversion*, vol. 2, 05 2006.

[45] D. Sera, R. Teodorescu, and P. Rodriguez, "Photovoltaic module diagnostics by series resistance monitoring and temperature and rated power estimation," pp. 2195 – 2199, 12 2008.

[46] H. Zhiqiang and G. Li, "Research and implementation of microcomputer online fault detection of solar array," pp. 1052 – 1055, 08 2009.

[47] E. Garoudja, A. Chouder, K. Kara, and S. Silvestre, "An enhanced machine learning based approach for failures detection and diagnosis of pv systems," *Energy Conversion and Management*, vol. 151, pp. 496–513, 09 2017.

[48] W. Chine, A. Mellit, V. Lughi, A. Malek, G. Sulligoi, and A. Massi Pavan, "A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks," *Renewable Energy*, vol. 90, pp. 501–512, 05 2016.

[49] M. Hajji, M.-F. HARKAT, A. Kouadri, K. Abodayeh, M. Mansouri, H. Nounou, and M. Nounou, "Multivariate feature extraction based supervised machine learning for fault detection and diagnosis in photovoltaic systems," *European Journal of Control*, vol. 59, pp. 313–321, 05 2021.

[50] J. Tan and C. Deng, "Ultra-short-term photovoltaic generation forecasting model based on weather clustering and markov chain," in *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, pp. 1158–1162, 2017.

[51] T. Dong, W. Cheng, and W. Shang, "The research of knn text categorization algorithm based on eager learning," pp. 1120–1123, 08 2012.

[52] G. Guo, X. Ping, and G. Chen, "A fast document classification algorithm based on improved knn.," pp. 186–189, 01 2006.

[53] Y. Zhang, L. Zhu, X. Qiao, and Q. Zhang, "Flexible knn algorithm for text categorization by authorship based on features of lingual conceptual expression," pp. 601–605, 01 2009.

[54] K. Dhibi, M. Mansouri, K. Bouzrara, H. Nounou, and M. Nounou, "An enhanced ensemble learning-based fault detection and diagnosis for grid-connected pv systems," *IEEE Access*, vol. 9, pp. 155622–155633, 2021.

[55] M. M. Badr, M. S. Hamad, A. S. Abdel-Khalik, R. A. Hamdy, S. Ahmed, and E. Hamdan, "Fault identification of photovoltaic array based on machine learning classifiers," *IEEE Access*, vol. 9, pp. 159113–159132, 2021.

[56] H. Rezk, I. Tyukhov, M. Dhaifullah, and A. Tikhonov, "Performance of data acquisition system for monitoring pv system parameters," *Measurement*, vol. 104, 04 2017.

[57] T. L. Floyd., *Electronic Devices.* Pearson Education.

[58] D. L. B. Thomas L. Floyd, *Electronics Fundamentals Circuits, Devices and Applications.* Pearson Education.

[59] D. B. ALBERT MALVINO, *Electronic principles.* McGraw-Hill Education.

[60] J. d. I. Bogdan M. Wilamowski, *Fundamentals oF IndustrIal electronIcs.* Taylor and Francis Group.

[61] L. N. Robert L. Boylestad, *Electronic Devices and Circuit Theory.* Pearson Education.

[62] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, "Emperor penguins colony: a new metaheuristic algorithm for optimization," *Evolutionary Intelligence*, vol. 12, 06 2019.

[63] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, "Optimization in solving inventory control problem using nature inspired emperor penguins colony algorithm," *Journal of Intelligent Manufacturing*, vol. 32, 06 2021.

[64] S. Harifi, J. Mohammadzadeh, M. Khalilian, and S. Ebrahimnejad, "Hybrid-epc: an emperor penguins colony algorithm with crossover and mutation operators and its application in community detection," *Progress in Artificial Intelligence*, vol. 10, 02 2021.

[65] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, "Optimizing a neuro-fuzzy system based on nature-inspired emperor penguins colony optimization algorithm," *IEEE Transactions on Fuzzy Systems*, vol. PP, pp. 1–1, 04 2020.

[66] H. Hammel, "Infrared emissivities of some arctic fauna," *Journal of Mammalogy*, vol. 37, p. 375, 08 1956.

[67] S. Harifi, J. Mohammadzadeh, M. Khalilian, and S. Ebrahimnejad, "Giza pyramids construction: an ancient-inspired metaheuristic algorithm for optimization," *Evolutionary Intelligence*, vol. 14, 12 2021.

[68] A. Wagner, "Peak-power and internal series resistance measurement under natural ambient conditions," 01 2000.

[69] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu, and T. Liu, "Understanding and improving early stopping for learning with noisy labels," 06 2021.

[70] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Networks*, vol. 11, pp. 761–767, 06 1998.

[71] T. Zhang and B. Yu, "Boosting with early stopping: Convergence and consistency," *Ann. Statist.*, vol. 33, 04 2004.