Predictive Maintenance in Renewable Energy Systems using Advanced Artificial Intelligence Techniques

A PhD Thesis
submitted in partial fulfillment of the requirements
of the degree of

Doctor of Philosophy

by

DIGANTA BARUAH
(Roll No. 186151013)

Supervisors:

Dr. Sonali Chouhan
Prof. Senthilmurugan Subbiah

School of Energy Science and Engineering (Earlier Centre for Energy)
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
June 2026

TH-4096_186151013

Abstract

Due to major industrial expansions, population growth, and a steep rise in per capita energy consumption, the world's energy demand has been steadily rising. A large amount of energy needs are met by electricity. The production of electricity using fossil fuels has nearly achieved saturation due to environmental factors and the finite supply of fossil fuels. Therefore, renewable energy sources must fill the gap between demand and supply.

Numerous Photovoltaic (PV) modules connected via cables, junction boxes, converters, battery banks, inverters, and other devices may make up a typical solar PV system. PV modules arranged as arrays, connected to inverters through DC subsystems, with the inverters connected to AC subsystems attached to a grid, make up a grid-connected PV system. Because they are subjected to adverse and unpredictable environmental conditions, including severe weather, rain, heat, storms, dust, snow, etc., PV arrays that are installed outdoors are vulnerable to a variety of defects. Furthermore, compared to other subsystems, inverter errors are more common in PV systems; as a result, they have a major influence on how much energy a PV system can generate. Early detection of inverter faults enables a reduction in system downtime. Even though PV systems are becoming more and more common across all renewable energy sources, faults are serious concerns for their efficiency, safety, and reliability.

Predictive maintenance makes use of predictive tools to decide when to take maintenance actions by monitoring a machine or process continuously, thereby allowing maintenance action only when it is necessary. Predictive maintenance is becoming increasingly prevalent in renewable energy systems due to advancements in sensors, computational power, and the availability of smart and ubiquitous devices. Predictive maintenance is essential for renewable energy systems because a malfunctioning component or faulty equipment can cause power fluctuations, decreased performance, costly downtime, system failure, or even catastrophic failures. PV monitoring systems continuously produce enormous amounts of data. Data-driven prognostics and diagnostics can be accomplished by using Machine learning (ML) and Artificial Intelligence

TH-4096_186151013

(AI) techniques.

This dissertation focuses on the development of advanced AI-based models for early fault detection and diagnosis in renewable energy systems, with a particular emphasis on solar PV systems. The four major contributions are as follows. First, fault detection using regression models is explored where advanced regression techniques, including ensemble methods, deep learning (DL), and contextual DL approaches are utilized to predict AC power and detect anomalies by comparing predicted and actual power values. As part of the second contribution, a novel Context-Sensitive LSTM (CS-LSTM) model is proposed which integrates contextual features like weather conditions and time of day to enhance prediction accuracy for regression-based fault detection. Building classification-based methods to identify and categorize faults into normal and anomalous groups is the third contribution. To address issues with data imbalance, traditional and deep learning classifiers are investigated with a focus on feature selection and data augmentation techniques. To detect and classify PV faults with special emphasis on inverter faults, including earth faults, straight joint faults, IGBT (Insulated Gate Bipolar Transistor) faults, and overtemperature faults, a predictive maintenance framework that integrates regression and classification techniques has been finally developed. This framework uses CS-LSTM and incorporates contextual data to demonstrate the model's diagnostic and predictive abilities.

This research investigates various AI techniques, including traditional ML models, Recurrent Neural Network (RNN) based sequential models, context-integrated ML models, and contextual DL models to predict and classify faults in solar PV systems. The conventional ML models are not effective for early prediction of faults in solar PV systems. However, significant improvements are observed in early fault prediction with our proposed methodologies with RNN-based sequential models, and contextual DL models when contextual features are incorporated with these models. This research demonstrates the effectiveness of integrating contextual information into machine learning models, highlighting its potential for early fault detection and diagnosis in solar PV systems. It is anticipated that the contributions made in this thesis can be applied for early fault prediction of PV systems well in advance. It contributes to the growing field of predictive maintenance, providing practical tools to increase the reliability and efficiency in the operation of renewable energy systems.

TH-4096_186151013

Contents

TH-4096_186151013

TH-4096_186151013

TH-4096_186151013

TH-4096_186151013

List of Tables

TH-4096_186151013

TH-4096_186151013

List of Figures

TH-4096_186151013

TH-4096_186151013

TH-4096_186151013

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी • Indian Institute of Technology Guwahati •

List of Abbreviations

AC	Alternating Current
AE	Autoencoder
AI	Artificial Intelligence
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
BOS	Balance of System
BP	Backpropagation
CCC	Current Carrying Conductor
CF	Collaborative Filtering
CNN	Convolutional Neural Network
CS-LSTM	Context Sensitive Long Short Term Memory
CxDL	Contextual Deep Learning
DBN	Deep Belief Network
DC	Direct Current
DL	Deep Learning
DnCNN	Denoising Convolutional Neural Network
DNN	Deep Neural Network
DT	Decision Tree
EL	Electroluminescence
EM	Expectation Maximization
ESR	Ensemble Stacking Regressor

TH-4096_186151013

EVR	Ensemble Voting Regressor
FDD	Fault Detection and Diagnosis
FF	Fill factor
FL	Fuzzy Logic
GCAD	Global Context Aware Detection
GFDI	Ground Fault Detection Interrupters
GHG	Greenhouse Gas
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
GW	Giga Watt
HDT	Hybrid Detection Technique
HMM	Hidden Markov Model
HS	Hotspot
ICR	Inverter Control Room
IEC	International Electrotechnical Commission
IGBT	Insulated Gate Bipolar Transistor
ITH	Infrared Thermography
kNN	k Nearest Neighbour
LAI	Local Anomaly Index
LCAD	Local Context Aware Detection
LG	Line to Ground
LGR	Logistic Regression
LL	Line to Line
LIT	Lock in Thermography
LSTM	Long Short Term Memory
LR	Linear Regression

TH-4096_186151013

MBDM	Model based Difference Measurement
ML	Machine Learning
MLP	Multilayer Perceptron
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracking
MW	Mega Watt
NEC	National Electric Code
NN	Neural Network
OC	Open Circuit
OCPD	Overcurrent Protection Device
PHM	Prognostics and Health Management
PV	Photovoltaic
PdM	Predictive Maintenance
PvM	Preventive Maintenance
RBF	Radial Basis Function
RDM	Real time Difference Measurement
ReLU	Rectified Linear Unit
R2F	Run to Failure
RF	Random Forest
RFR	Random Forest Regressor
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SCADA	Supervisory Control and Data Acquisition
SHAP	Shapley Additive Explanation
SMOTE	Synthetic Minority Oversampling Technique
SOH	State of Health

TH-4096_186151013

SOM	Self Organizing Map
SVM	Support Vector Machine
SVR	Support Vector Regressor
TSKFRBS	Takagi-Sugeno-Kahn Fuzzy Rule Based System
XAI	Explainable Artificial Intelligence

TH-4096_186151013

Citations to Published Work

Chapter 5 is based on the following conference papers:

- D. Baruah, S. Subbiah, and S. Chouhan, “Exploiting context for fault detection in solar photovoltaic system using context-sensitive recurrent neural network,” in 2023 IEEE International Conference on Power Electronics, Smart Grid, and Renewable Energy(PESGRE), 2023, pp. 1–6. DOI: 10.1109/PESGRE58662.2023.10404256.
- D. Baruah, R. Roy, R. Ahmed, S. Subbiah, S. Chouhan and K. Angappan, "Contextual Approaches to Data-Driven Fault Detection in Solar Photovoltaic System," 2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS), Madrid, Spain, 2024, pp. 1-7, doi: 10.1109/EAIS58494.2024.10570026.

Chapter 6 is based on the following Journal paper:

- D. Baruah, N. Das, S. Chouhan and S. Subbiah, "A Comprehensive Framework for Predictive Maintenance in Solar Photovoltaic Systems Using Contextual Deep Model," in IEEE Access, vol. 13, pp. 191924-191940, 2025, doi: 10.1109/ACCESS.2025.3627866.

TH-4096_186151013

TH-4096_186151013

Acknowledgments

I wish to express a deep sense of gratitude to my supervisors **Dr. Sonali Chouhan** and **Prof. Senthilmurugan Subbiah** for their valuable guidance and constant support at all stages of my PhD study and related research. I would like to thank Azure Power (Haryana) Pvt. Ltd. for providing the necessary data for this research work.

I would like to thank the chairman of the Doctoral Committee Prof. Harshal B. Nemade and other members of the Doctoral Committee, Dr. Moumita Patra and Dr. Sumit Kumar for their valuable comments and suggestions during the PhD Annual Progress seminars (APS), and PhD Synopsis Seminar. I would also like to thank Dr. Reshmi Suresh for being part of the Doctoral Committee till my second APS. I must thank Dr. E. S. N. Raju P. for agreeing to be part of the Doctoral Committee as a substitute member in absence of other member.

I wish to thank the faculty members and staff members of the School of Energy Science and Engineering, Indian Institute of Technology (IIT) Guwahati (Earlier Centre for Energy, IIT Guwahati) for their help and support. I am immensely grateful to Prof. Hiren Kumar Deva Sarmah, former HoD, Department of Information Technology (IT), Sikkim Manipal Institute of Technology (SMIT) and Prof. Ashis Sharma, former Director, SMIT for allowing me to pursue PhD under Quality Improvement Program (QIP) in IIT Guwahati. I would also like to thank Prof. Biswaraj Sen, HoD, Department of IT, SMIT and Dr. Udayan Baruah, former HoD, Department of IT, SMIT for granting me necessary leaves, thereby allowing me to complete my PhD in IIT Guwahati.

I would like to thank my wife Dr. Rashmi Dutta Baruah for her constant support and valuable assistance all the time. I must thank my two sons, Samitinjaye Boruah and Ambarish Avir Baruah for their immense patience. Special thanks go to my parents for their blessings all the time.

June 10, 2026 DIGANTA BARUAH

TH-4096_186151013

TH-4096_186151013

Declaration

I, DIGANTA BARUAH (Roll No. 186151013), hereby declare that the research work presented in this thesis has been carried out under the supervision of Dr. Sonali Chouhan and Prof. Senthilmurugan Subbiah. This work is original and has not been submitted, either in full or in part, to any other institute or university for the award of any degree or diploma.

I further certify that all text, images, diagrams, graphs, ideas, data, concepts, and results taken from external sources have been appropriately cited in the thesis and included in the references.

I also declare that only limited use of AI tools was made during preparation of the thesis, and no part of the thesis may be considered plagiarized.

June 10, 2026 DIGANTA BARUAH

TH-4096_186151013

Chapter 1

Introduction

The energy demand has been gradually increasing for all countries because of large-scale industrial expansions, population growth, and a rapid increase in per capita energy consumption [1]. There is a significant increase in greenhouse gas (GHG) emissions starting from 21st century compared to previous decades [2]. Thus, environmental concerns are also increasing because of increasing trend of global GHG emissions. Renewable energy has drawn the attention of researchers because of increased energy demands and severe environmental problems [2]. Moreover, fossil fuel resources are rapidly depleting[3]. There are many limitations of fossil fuels such as the emission of hazardous gases, continuous depletion, and deterioration of ecology due to the extraction of natural resources [4]. Therefore, there is a need for a rapid transition to renewable energy sources (RES) due to rising energy demand and the urgent need to decarbonise to meet net-zero emissions by the year 2050. Therefore, renewable energy resources have become a viable alternative to energy sources [4]. The sources of renewable energy must therefore fill the gap between supply and demand for energy. Some of the sources of renewable energy are solar, wind, geothermal, biomass, small hydro power etc. [1]. In these systems, the source of energy is supplied by solar radiation, winds, ocean waves, tides etc. These sources are unlikely to run out of supply. Moreover, these sources are clean and pollution free. Among all

TH-4096_186151013

renewable energy sources, solar energy is regarded as the most effective method for harnessing energy from the environment [5]. During the last decade, solar PV systems have attracted lot of attention because of many advantages provided by these systems such as: (i) availability of solar energy world wide, (ii) noiseless operation, (iii) pollution free in nature, (iv) easy installation, (v) modular configuration, (vi) reliable technique for energy conversion, and (vii) capability of integrating solar PV systems in buildings[6]. Therefore, during the last few years, there has been an increased surge in electricity generation using solar PV systems. Moreover, electricity is the main energy demand for consumers [7]. Because of the growing energy demand, the PV sector is likely to expand due to various factors such as decrease in prices of PV modules, advanced manufacturing technologies, government incentives, improvement of power converter technologies etc. [8]. In India, solar energy based power generation is given significant priority in order to increase electricity production from renewable energy sources which is evident from national solar mission of India, where it was envisaged that 175 GW of electricity would be generated from the renewable energy sources by 2022, out of which 100 GW of electricity would be generated from solar energy, and the remaining would be generated from wind, small hydropower, and biomass [9]. In India, the target of 100 GW based on solar energy has already been met, with the present capacity being 132.848 GW as on November 30, 2025 [10].

Table 1.1 shows installed capacity of various energy sources in India as on November 30, 2025 as per Ministry of Power, Government of India (https://powermin.gov.in/en/content/generation-capacity). Installed capacity in India is composed of (i) RES, viz., solar, wind, small hydro, large hydro, biomass/cogeneration, and waste-to-energy, (ii) thermal power, and (iii) nuclear power. Out of the total installed capacity of 509743.09 MW of power, the installed capacities of Renewable Energy Sources (RES), thermal, and nuclear power are 254021.48 MW (49.83%), 246941.62 MW (48.44%), and 8780 MW (1.73%), respectively as on November 30, 2025, thereby providing 51.56% of non-fossil fuel based energy and 48.44% of fossil fuel based energy[10]. In India, the contributions of RES consisting of solar, wind, small hydro power, large hydro power, bio-mass(co-generation), and waste to energy are 26.1%, 10.6%, 1%, 9.9%, 2.1%, and 0.2%, respectively [10] as on November 30, 2025 .

TH-4096_186151013

Table 1.1: Installed capacity of various energy sources in India as on November 30, 2025

Name of Energy Source	Installed Capacity(in MW)
Solar	132848.25
Wind	53986.02
Small hydro power	5158.61
Large hydro power	50414.66
Bio-mass(Cogeneration)	10757.31
Waste to Energy	856.62
Nuclear	8780
Thermal	246941.62
Total	509743.09

1.1 Photovoltaic Fundamentals

A solar cell is a device made from semiconductor materials that generates electricity by harnessing solar energy [11]. It is a P-N junction diode with a very large surface area which can absorb solar radiation. In a P-N junction device, an inherent electric field is present, created by the union of P-type and N-type semiconductor materials. A built-in potential is created because of the recombination of electrons and holes near the junction, due to which a charged region is formed, which is known as the space charge region, also known as the depletion region because of the depletion of mobile carriers. Due to the formation of layers of positive ions and negative ions in the depletion region, there is an electric field that points from N-type to P-type. It is to be noted that the regions outside the depletion region on both P-type and N-type are electrically neutral. When the solar cell is subjected to sunlight, solar radiation is absorbed by the solar cell, thereby creating electron-hole pairs. Because of the presence of built-in potential in the solar cell (i.e., P-N junction diode) due to asymmetry in semiconductor materials used, the electron-hole pairs that are generated by absorbing light are separated from each other. There will be a flow of current if an external load is connected between the P-type and N-type terminals.

Thus, the following two steps are necessary for working of a solar cell [12]:

- Creation of negative and positive mobile carriers called electron-hole pairs in the solar cell

TH-4096_186151013

by absorbing solar radiation. For this to happen, the solar cell must be made of a material which can absorb energy of photons. Increase in the potential energy of charge carriers is performed by semiconductor materials. There is an increase in the potential energy of electrons when photons are absorbed by semiconductor material, thereby keeping the electrons excited in the higher energy levels which increases the probability of separation of electrons.

- Separation of charge carriers: Asymmetry in semiconductor material should be provided such that the generated electron-hole pairs are separated from each other. This can be achieved by combining P-type semiconductor and N-type semiconductor, thus forming a P-N junction which creates a built-in electric field at the junction. A P-type semiconductor can be created by doping intrinsic silicon semiconductor material with trivalent atoms like boron, thus creating covalent bonds with silicon, which are void of electrons (i.e., equivalent to having an excess of holes). Similarly, when an intrinsic silicon semiconductor material is doped with pentavalent atoms like phosphorus, there is an excess of electrons. While an N-type semiconductor contains surplus electrons and a P-type semiconductor possesses surplus holes, both remain electrically neutral. When these materials are joined together to form the P-N junction, electrons from N-type diffuse towards the P-type semiconductor, and holes from P-type diffuse towards the N-type semiconductor to recombine with each other. Electrons move from the N-type region to the P-type region near the junction, resulting in positively charged donor impurity ions being left behind, and holes diffuse from P-type to N-type, thus making the acceptor impurity atoms negatively charged. Thus, a layer of positively charged ions is formed on the N-side, and a layer of negatively charged ions is formed on the P-side, thereby making the region void of mobile carriers. A built-in potential is created because of the recombination of electrons and holes near the junction, due to which a charged region is formed, which is known as the space charge region (also known as the depletion region because of the depletion of mobile carriers). Due to the formation of layers of positive ions and negative ions in the depletion region, there is an electric field that points from N-type to P-type. It is to be noted that the regions outside the depletion region on both P-type and N-type are electrically neutral.

Alexandre-Edmund Becquerel discovered photovoltaic effect in 1839 [13], and Chapin et al

TH-4096_186151013

developed the first silicon solar cell in 1954 [14]. Series connection of solar cells is referred to as PV module [11], [13]. A PV array can be formed by connecting modules in series-parallel configuration [11]. In general, crystalline Si Solar cells are connected in series using external connections to form a module [13]. In case of solar cells made of thin film materials, the series connection is already provided internally [13]. A PV string can be formed by connecting PV modules in series. One simple configuration of PV array is to connect number of PV strings in parallel. A PV system mainly consists of PV arrays and balance of system (BoS). BoS consists of DC to DC converters (optional, depending on requirement), DC to AC converters (inverters), batteries and battery charge controllers (dominantly in off-grid PV systems), supporting structures for mounting the PV panels, protection units, cabling etc.There are mainly two categories of PV systems: grid connected and stand-alone [15],[16]. PV systems can provide both AC and DC supply along with provision of energy storage.

1.2 Types of Faults in Solar PV Systems

Incorrect functioning of a component or equipment is known as a fault, whereas failure denotes the condition of not being able to meet the target. A fault in a PV system refers to an effect that degrades the system's efficiency, for example, in the form of lower power output [17]. Figure 1.1 presents a variety of fault types.

A malfunction in a photovoltaic system can either partially reduce its performance or result in no power output at all, leading to a complete shutdown. On the DC side, faults in the PV array and BOS components might cause failures. Potential reasons for PV array issues include earth faults, arc faults, junction box faults, line-to-line faults, open circuits, short circuits, bypass diode faults, and mismatch faults [18]. Mismatch faults may be permanent or temporary. Permanent mismatch may be due to hotspot, degradation, and soldering. Temporary mismatch may be due to partial shading, soiling (i.e., dust accumulation), leaves/bird dropping etc. Degradation may be due to delamination, cracks, and discolouration. Maximum power point tracking (MPPT) errors, cabling errors, converter faults, and battery bank failures are the main reasons of BOS faults. Islanding and grid irregularities are the two forms of faults that might occur on the AC side of a PV system. Lightning and inverter malfunctions could be the cause of

TH-4096_186151013

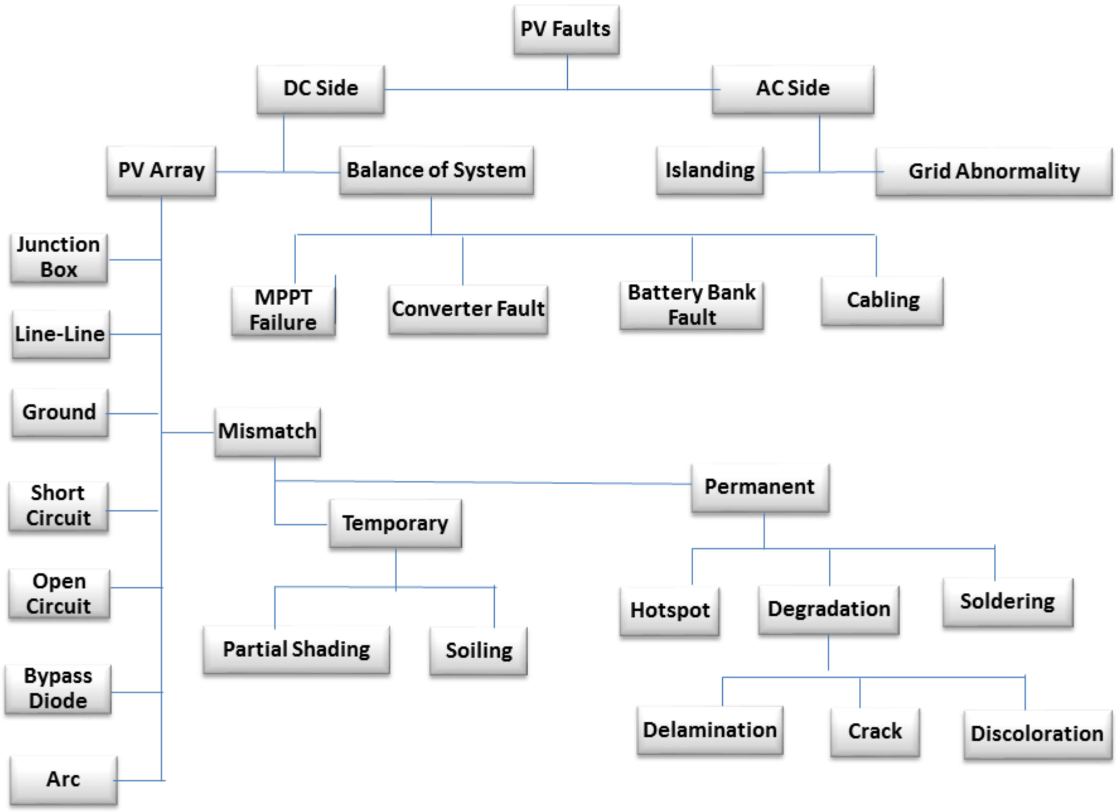


Figure 1.1: Types of faults in PV system

grid irregularities [17]. Mansouri et. el. [19] have classified PV faults into three categories based on time characteristics: abrupt, intermittent, and incipient faults. The faults that occur instantaneously because of PV array malfunction such as ground faults, line to line, open circuit, junction box faults, short circuit, disconnection of connectors, and hot spots are known as abrupt faults. Intermittent or temporary faults are those faults that change over time such as environmental stress (e.g., contamination or dust accumulation), and partial shading. On the DC side, incipient faults may occur because of discoloration of PV modules (e.g., yellowing of solar cells), cracks, delamination, defective anti-reflective coating etc. On the AC side, incipient faults include Insulated Gate Bipolar Transistor (IGBT) faults, overheating, islanding, aging, wiring degradation etc. Incipient faults occur because of inverter faults and grid abnormality, which are the most challenging because of their slow dynamics and small amplitudes. If the incipient faults are not detected well in advance, they will damage the PV cells, resulting in serious faults [19]. Figure 1.2 depicts a pictorial representation of faults in a PV system.

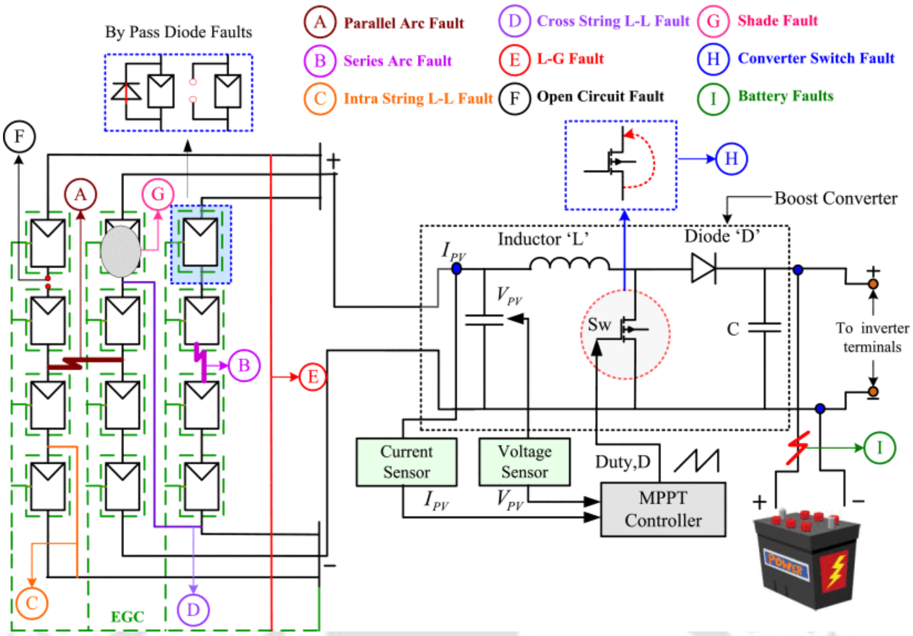


Figure 1.2: All faults in PV system (Adapted from [16])

1.2.1 Photovoltaic(PV) Array Faults

The various PV array faults are described below:

1.2.1.1 Ground fault

In a PV system, distribution panels, mounting structures, module frames, metallic enclosures, converters etc. contain noncurrent carrying metallic parts [20]. Generally, during normal operations, these metallic parts do not carry any current. Because of wrong wiring, wire cutoff, loss of insulation, corrosion etc., there is possibility of formation of electrical connection between the current carrying conductors (CCCs) and non current carrying metallic parts [20]. To prevent electric shock in a PV array, usually the non-current carrying metallic parts are grounded through a conductor known as equipment grounding conductor (EGC) [20]. This is known as equipment grounding. A ground fault is defined as accidental short circuit between ground and current carrying conductors [21]

1.2.1.2 Line-to-Line fault

An LL (line-to-line) fault occurs when there is an accidental short circuit connecting two points on a PV panel that have differing potential levels [21]. These short-circuited points can either be part of the same string or exist between two neighboring strings [22]. Because of LL faults, unintentional pathways for current are created in a PV array [23]. LL fault may remain undetected and can incur severe loss [21]. The voltage drop across PV string will be reduced because of L-L faults, and therefore, current will be back fed from other strings, thereby lowering the system efficiency [23]. PV panels may be damaged, and there is also risk of fire hazards due to LL faults [23], [24]. Some of the causes of LL faults are : cable aging, insulation failures of cables, damaged combiner box, presence of MPPT (e.g., LL faults may be masked) etc. [23].

1.2.1.3 Arc fault

Arc fault can be defined as unintended passage of current through a dielectric material or through air [6]. The maximum electric field which can be tolerated by an insulating material prior to the occurrence of breakdown is determined by the dielectric constant of the material [20]. Air has a dielectric constant of approximately $3V/\mu m$, and it is dependent on presence of impurities, humidity, pressure etc. [20]. Initiation of an arc through an air gap is caused due to ionization of air by high electric fields. Additional charged particles are created because of high-velocity particle collisions. Because of generation of ions, the air medium, which acted as insulator, is converted into conductive medium [20]. According to [25], arc faults are of two types: (i) series arcs, and (ii) parallel arc faults.Various types of arc faults are shown in Figure 1.3.

- Series arc fault: Module interconnects and connectors have almost zero impedance in ideal PV arrays [25]. Because of degradation in junction boxes, cables, solder joints etc., the interconnect impedance increases beyond a nominal value, thereby creating series arc fault [25].

- Parallel arc fault: Because of residual effect of series arc fault or due to insulation breakdown between conductors, arcing between two parallel conductors may occur [24]. There are three variations of parallel arc faults: cross-string, intra-string, and ground [25]. In

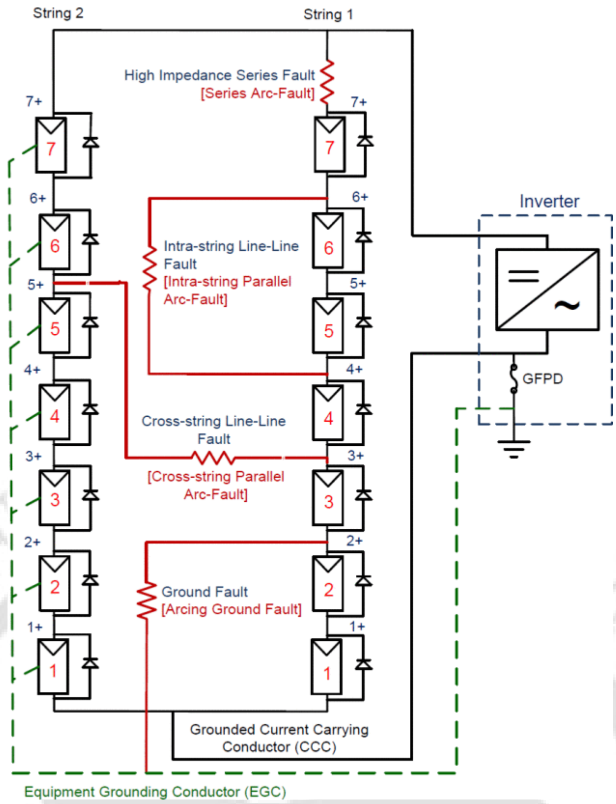


Figure 1.3: Types of arc faults in PV system (Adapted from [25])

case of a parallel arc fault, there may be existence of current paths for fault current while the PV array is open, thereby creating danger of fire and shock hazards [25]. Because of parallel arc fault, the faulted strings appear to be consisting of less number of modules compared to the normal one, thereby creating asymmetry in the array [25]. This in turn will reduce the open circuit voltage which is dependent on the number of modules while short circuit current may not be affected [25]. Identifying the fault location is always an issue with parallel arc fault because I-V curve remains the same for a PV string irrespective of the fault location of the faulty PV module [25].

1.2.1.4 Hotspots

When the photovoltaic cells operate at reverse bias, then they consume power instead of producing it [22]. This phenomenon is known as Hotspot. A PV cell begins to breakdown when it is reverse biased beyond a certain voltage level and as a result more current flows through the cell. Zener and avalanche breakdowns are the two breakdown mechanisms in p-n junctions and both of them are affected by temperature [26]. Because of breakdown, the temperature of a cell becomes very high compared to nearby cells surrounding it. The phenomenon of hotspot is due to mismatch between electrical characteristics of cells which causes a cell to become reverse biased, sink lot of current thereby dissipating power, and heat up [27]. To mitigate this problem (i.e., problem of hotspots), bypass diodes are commonly used in PV strings, but it (i.e., employing bypass diodes) cannot eradicate the problem [27].

Because of Hotspot, photovoltaic cells operate abnormally, thereby causing serious damages in addition to degrading a PV panel and reducing the performance of a PV system [22]. The main contributors of hotspot are shading, failure of bypass diode, mismatch in electrical characteristics etc [22]. Figure 1.4 shows the I-V characteristics of a p-n junction when breakdown occurs:

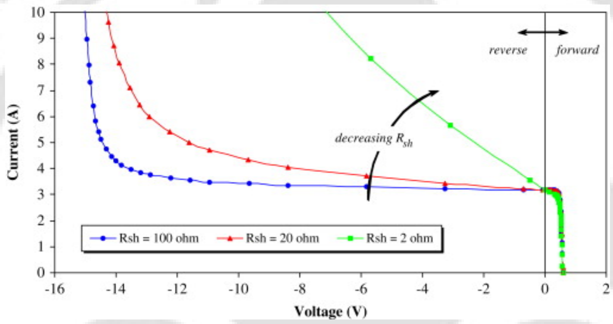


Figure 1.4: I-V characteristics of solar cell when breakdown occurs(Adapted from [28])

The decrease in shunt resistance affects the slope of the I-V curve in the reverse bias condition and causes more power dissipation which in turn results in breakdown of the localized regions of the p-n junction thus leading to formation of hotspot [28]. Using I-V curve, hotspot can be detected , but it can not locate the defective cell dissipating heat [28].

1.2.1.5 Mismatch fault

A mismatch fault occurs if there is lot of variations in the module electrical parameters compared to the initial state [22]. There are two types of mismatch faults: temporary and permanent. Temporary mismatch may occur because of shading of cells leading to hotspot, whereas permanent mismatch occurs due to open circuit fault or presence of defective cell dissipating power [22].

TH-4096_186151013

1.2.1.6 Diode faults

The bypass diodes and blocking diodes are used for reverse voltage protection and reverse current protection, respectively [6]. Open circuit and short circuit in diodes are the electrical faults associated with these diodes. Partial shading of a PV module for a long time may cause these faults [6].

1.2.1.7 Junction box fault

There may be increase in contact resistance due to fretting corrosion in junction box [6]. Formation of electric arc between the contacts will result in wearing out and melting of junction box which will in turn damage the whole string and the module, thereby reducing the amount of energy production [6].

1.2.1.8 PV module fault

A PV module may be shunted or short circuited because of corrosion, delamination of a PV module, current leakage in a module, defects in manufacturing etc., which may isolate the PV array from ground [6]. There may be fire and shock hazards because of PV module fault [6].

1.2.2 Balance of System(BOS) Faults

A PV system consists of several other components other than the PV arrays. These components are referred to as balance of system (BoS) and consists of DC to DC converters, DC to AC converters (inverters), batteries and battery charge controllers (mainly, in off-grid systems), supporting structures for mounting the PV panels, protection units, cabling etc. [12]. The existence of non-functional modules in a PV systems are mainly due to failures in BOS components [29]. The whole PV string can become non-functional even for a failure in single fuse [29]. Based on a survey conducted on 35 PV systems by Sandia national laboratories, the existence of 54% of the non-functional modules were because of BOS component failures like failures in fuses, switches, surge arrestors, dc contactors etc.[30]. Another reason of BOS failure is due to bypass diode faults [29]. The purpose of bypass diode is to allow bypassing

TH-4096_186151013

of current across cracked or shaded cells in order to reduce power losses within a module [29]. Battery bank faults may occur due to abnormal charging and discharging conditions or internal battery cell damage [16].

1.2.3 Faults in AC side of PV System

Grid abnormality and Islanding are the major faults in AC side of PV system [5][18]. Grid abnormalities may occur because of inverter faults and lighting. Islanding operation takes place when power supply from main grid is interrupted [5]. In this situation, the PV array keeps on supplying power to local load or keeps on charging batteries for energy storage [5].

Because of various issues related to failure in photovoltaic(PV) systems including photovoltaic arrays and balance of system(BOS) components, timely maintenance of PV systems is important.

1.3 Grid-connected PV systems and their causes of faults

Figure 1.5 shows a simplified schematic diagram of a typical grid-connected PV system. The DC subsystem is composed of DC switches, DC cables, string combiner boxes, string fuses, surge suppressors, etc. The AC subsystem is composed of switches, cables, fuses, and transformers (in case it is not part of the inverter system).

According to an analysis by Golnas [31], a significant portion of PV system defects (57%) are related to AC subsystems and inverters, whereas just 2% are related to PV modules. Based on an observation of 3500 abnormalities for 27 months starting from January 2010 to March 2012 of 350 PV systems that were maintained and operated by Renewables Operation Center, SunEdison, Belmont, CA, the summary of abnormalities in various subsystems of PV systems is presented in Table 1.2. The percentage of power loss due to faults in various subsystems of PV systems is presented in Table 1.3 .

Based on Golnas' study [31], it is apparent that inverter subsystems experience more faults than other PV system subsystems, which leads to the greatest production loss from inverter

TH-4096_186151013

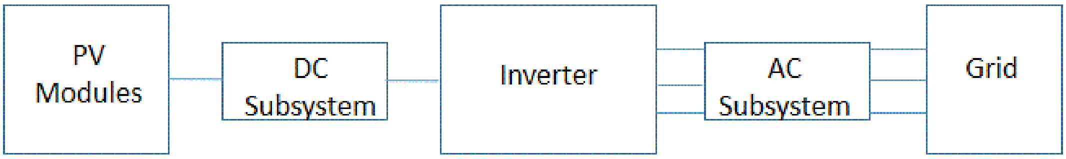


Figure 1.5: Simplified schematic diagram of a grid connected PV system

Table 1.2: Frequency of abnormal events in various subsystems of PV system [31]

Subsystem	**Frequency of abnormal events (%)**
PV modules	2
DC subsystem	6
Inverter	43
AC subsystem	14
Support structure	6
External	12
Outage related	5
Others (weather station, meters, etc.)	12

breakdowns. According to the study, 14% of failures occur in the AC subsystem, which includes everything lying between the inverter and the AC power measurement meter.

Based on observations of 350 PV systems operated and maintained by SunEdison from January 2010 to March 2012, Table 1.4 shows the frequency of abnormal events (i.e., anomalies) found in various inverter subsystems. A more thorough examination of the inverter faults showed that the control software, card/board, IGBT, AC contactor, and fans in the inverter subsystems account for the majority of the faults.[31].

1.4 Types of Maintenance

In industries, maintenance of equipment is a key requirement because it affects efficiency and the operation time of equipment. It is, therefore, necessary to identify equipment faults and accordingly solve the issues related to faults, thereby avoiding production process shutdown[32].

Maintenance can be categorized into three types, viz., run-to-failure (R2F), preventive mainte-

TH-4096_186151013

Table 1.3: Loss contribution of different subsystems in terms of energy (kWh)[31]

Subsystem	**Loss percentage (kWh)**
PV modules	1%
DC subsystem	4%
Inverter	36%
AC subsystem	20%
Support structure	3%
External	20%
Outage related	8%
Others (weather station, meters, etc.)	8%

Table 1.4: Frequency of abnormal events in inverter subsystems

Subsystem	**Frequency of abnormal events (%)**
DC input fuses	2
GFI	3
Internal fuses	3
Internal switches	3
Surge protection	3
Capacitors	3
DC contactor	4
AC fuses	4
Power supply	5
IGBT	6
Fans	6
AC contactor	12
Card/Board	13
Control software	26
Others	5

TH-4096_186151013

nance (PvM), and predictive maintenance (PdM) [33] [32]. They are briefly described below:

1.4.1 Run to Failure (R2F)

In the Run to Failure (R2F) strategy, maintenance actions are initiated only after a component or system has already failed [33]. Although this approach is straightforward to implement, it is generally considered the least efficient maintenance strategy due to the risk of unexpected downtime. Though it is frequently adopted because of its simplicity, the maintenance cost and downtime after failure is usually high compared to planned maintenance taken in advance.

1.4.2 Preventive Maintenance (PvM)

In this approach, maintenance is performed as per planned schedule. This strategy, often termed scheduled or time-based maintenance, relies on predefined maintenance intervals to prevent equipment failures. Despite its ability to enhance reliability, the approach may lead to increased maintenance expenditures since maintenance actions are executed regardless of the actual health state of the system, thereby reducing resource efficiency [34].

1.4.3 Predictive Maintenance

Predictive maintenance (PdM) makes use of predictive tools to decide when to take maintenance actions by monitoring a machine or process continuously, thereby allowing maintenance action only when it is necessary [32]. In PdM, historical data, the current state of monitored equipment, models, and relevant domain knowledge are used to predict behaviour patterns, trends, and correlations by using machine learning or statistical models to anticipate pending failures well in advance to decide maintenance actions to be carried out [35]. This approach predicts failure in advance, thereby enabling maintenance actions to be scheduled proactively. The primary objective of predictive maintenance is to estimate the likelihood and timing of equipment degradation or failure, and in turn decide which maintenance task to perform to maintain a good trade-off between the maintenance frequency and cost [36]. To put it another way, PdM aims to identify impending problems ahead of time so that timely maintenance can be performed [34]. In PdM, maintenance is carried out according to the equipment's or component's state of health [34]. The actual operating conditions of components (or equipment) and systems are used to optimise operation and maintenance costs. Data collected from various sensors and monitoring devices are analysed using predictive algorithms to identify operational patterns and detect potential failures before they occur [36]. By enabling condition-based interventions, predictive maintenance (PdM) improves maintenance efficiency by carrying out maintenance only when required, thereby avoiding both unexpected equipment failures and premature replacement of components that still possess remaining useful life [36]. The process of continuously monitoring the parameters for detecting anomalies is known as condition monitoring [37], and it is a key component of PdM [38]. The predictive maintenance framework typically involves three major phases: data collection, data processing, and maintenance decision-making [39]. Its objective is to reduce unnecessary maintenance by performing interventions only when evidence of abnormal behaviour or degradation is observed in equipment or components [39]. Such an approach can minimise downtime and maintenance costs while improving operational quality and productivity [40]. Diagnostics and prognostics constitute two essential functions of PdM, focusing on fault identification and prediction of future failures, respectively [39].

- Diagnostics: In predictive maintenance, diagnostics encompasses the analysis and categorization of abnormal operating patterns by relating them to established fault signatures [37]. It primarily addresses three key tasks: detecting the existence of faults, isolating the location of the malfunctioning component, and identifying the characteristics of the fault [39]. Specifically, fault detection determines whether anomalous behaviour is present, fault isolation pinpoints the source of the problem, and fault identification determines the fault type [39]. Diagnostics are necessary if a fault occurs or when prognostics for fault prediction fail [39]. It should be mentioned that diagnostics is a type of posterior event analysis [39].

- Prognostics: Prognostics' primary goal is to address fault prediction prior to its occurrence. The objective of fault prediction is to determine whether a fault is likely to occur in the near future and identify its expected time of occurrence [39]. More specifically, prognostics is a predictive analysis technique that estimates the remaining useful life (RUL) of

TH-4096_186151013

equipment by forecasting its future health state before failure occurs, and is a prior event analysis [39]. Prognostics is better than diagnostics for achieving zero-downtime[39]. One can leverage predictive maintenance for prognostics, diagnostics, or both.[39]. The term “prognostics and health management"(PHM) was introduced in 1990s in connection with the Joint Strike Fighter (JSF) project [41], and thus PHM was initially applied in the military aviation domain [42]. Because of improvement in sensor technology and prediction algorithms, PHM is now popular in many domains including renewable energy systems, engineering systems, manufacturing systems, electronic systems, chemical processes, robotics etc. The main objective of PHM is to provide actionable information in order to take timely decisions. According to IEEE standard 1856: 2017, the PHM cycle consists of the following steps [43]:

- Sensors: Sensors are used to sense signals.
- Data acquisition: Data acquisition with reference to relevant parameters is performed in this step.
- Data manipulation: The data acquired by the data acquisition systems is further processed and manipulated in this step.
- State detection: In this step, assessment of state conditions is performed with reference to normal conditions, abnormal condition indicators are generated, if any.
- Health assessment: The aim of health assessment is to obtain the state of health (SOH) of the system.
- Prognostic assessment: This step provides upcoming RUL and SOH information.
- Advisory generation: This step provides alerts or actionable strategies in order to take timely repair and maintenance activities.
- Health management: In this step, the recommended advisory generated in the previous step is used to take maintenance actions in order to bring back the system to a healthy state.

TH-4096_186151013

1.5 Various Approaches of Predictive Maintenance in Photovoltaic (PV) Systems

A substantial body of research has been reported on fault detection and diagnosis (FDD) techniques for photovoltaic (PV) systems. Several review studies, including [6], [5], [1], [17], [24], [8], and [22], provide comprehensive overviews of existing FDD approaches. Mellit et al. [6] focused exclusively on electrical-based FDD techniques and grouped them into five classes: (i)current-voltage characteristic methods, (ii) statistical and signal processing methods, (iii) current and voltage measurement techniques, (iv) analysis based on power loss, and (v) AI-based approaches. Similarly, FDD methods are categorized in [24] into six groups, namely, (i) model-based difference measurement, (ii) ML techniques (iii) real-time difference measurement, (iv) output signal analysis, (v) infrared thermography, and (vi) hybrid detection techniques.

In [5], fault diagnosis methods for the DC side of PV systems were classified into six categories: (i) visual inspection, (ii) electrical characterization, (iii) electroluminescence (EL) imaging, (iv) infrared (IR)/thermal imaging, (v) lock-in thermography (LIT), and (vi) ultrasonic inspection. Chokor et al. [8] further grouped DC-side FDD approaches into two broad categories: physics-based methods and data-driven methods. From the perspective of predictive maintenance, existing techniques can be broadly divided into three major classes: (i) model-based, (ii) data-driven, and (iii) hybrid. These categories are described below.

1.5.1 Model Based Methods

The fundamental premise of these methods is that the behavior of physical systems can be precisely analytically described [42], [44] by employing explicit mathematical equations to incorporate information about the failure behaviors (or degradation) of systems or equipment [45], [44]. The mathematical formulations utilizing explicit mathematical equations in model-based techniques are grounded in a physical understanding of a deteriorating piece of equipment [44]. Developing a model-based approach requires a thorough understanding of the degradation behaviors that result in failure [42].

TH-4096_186151013

Remaining useful life (RUL) can be estimated using knowledge of the mechanisms that lead to failure as well as the electrical, chemical, mechanical, thermal, and radiation factors that affect the equipment's normal operation [44]. A thorough understanding of the physics of the degradation mechanism is necessary for the model-based approach to be accurate [42]. For instance, a rechargeable battery's internal electrochemical activities cannot be seen from the outside [42], making it impossible to accurately explain the battery's deterioration process using model-based techniques [42].

Model-based prognostics techniques are highly dependent on the application. Model-based techniques are typically used at the system or component level. For industrial uses like power systems, especially renewable energy systems, model-based techniques might not be the ideal choice because fault types are likely to spread across components, making them challenging to identify [44]. The lack of access to specific system knowledge is a significant obstacle for model-based prognostics [44]. Furthermore, fault propagation from one component onto the next is likely to be impacted by future operating conditions [44]. In certain circumstances, a model-based approach might not be the best option [44]. The model-based approaches in PV systems can be divided into the following categories.

- Difference calculation-based: This approach's fundamental idea is to estimate PV parameter values and then compare them with actual measurements. PV system faults will impact the output of a PV module or array, such as power output with variations in both current and voltage levels produced by a PV system. A fault is identified if the expected values of outputs computed using a model surpass a predetermined threshold value when compared to real-time measured values. A threshold is commonly set above or below a predefined operating point to identify conditions under which a fault is expected to emerge [8]. To identify errors, the real-time measurements may also be assessed against a threshold value that was previously established [16].

- Adjacent string comparision-based: This approach's fundamental idea is that PV strings connected to a specific combiner box will probably behave similarly if they function normally. Thus, in order to identify problems, the abnormality of a given string can be ascertained by comparing its output (such as current, voltage, or power) with the neighboring strings [46].

- Power (or energy) losses-based: A PV system's power losses can reveal some information regarding potential problems. Power losses can be computed by comparing predicted power obtained from simulation with observed power [5]. This method analyzes the loss of energy in a photovoltaic system and then uses power losses to identify faults [8].

- Heat exchange and temperature-based: If a PV array malfunctions, the temperature of a PV module varies. Heat exchange and module temperature during malfunctioning are taken into account to identify problems, primarily caused by hotspots and cable failures [8].

- Usage of external devices-based: This approach uses external devices for fault identification and diagnosis, such as signal generators and LCR (inductance, capacitance, and resistance) meters [8].

1.5.2 Data Driven Methods

Data-driven methods use condition monitoring data to directly understand equipment behavior. Instead of using in-depth outside information from specialists, these methods rely on data that are readily available. In order to simulate the deterioration trend [47], data-driven techniques extract valuable features through data collected from condition monitoring to detect the current state and predict the RUL of equipment or components based on the data gathered. In this research, every artificial intelligence (AI) approach, encompassing machine learning techniques and knowledge-based methods that derive their knowledge from existing data, is categorised as a data-driven strategy.

1.5.2.1 Machine learning techniques

By analyzing incoming data, ML methods are used for fault identification, diagnostics, and prognostics [8]. The relations between a PV system's inputs and outputs are learned using ML methods, after which the models are trained to identify and categorize faults. In this case, accurate prediction requires enough training data for the model.

TH-4096_186151013

1.5.2.2 Knowledge-based systems

Because of insufficient prior knowledge, it is difficult to build precise mathematical models corresponding to physical systems, and therefore, there are limited applications of model based methods [48]. Knowledge based systems require no physical model, and therefore, they are becoming popular. In knowledge based systems, rules are extracted from domain experts [48]. Examples of knowledge-based systems include fuzzy logic systems and expert systems [48].

- Expert systems: An expert system's approach to problem solving is comparable to that of a human specialist. Expert systems are appropriate for problems that domain experts typically resolve [48]. They use rules to hold the domain knowledge that has been gathered from domain experts. The rules are then used to solve problems through inference and reasoning. An expert system development involves acquiring Knowledge, knowledge representation, and verification and validation of model [48].

  The rules may be either heuristic or domain-specific, and logical operators can be used to combine them [48]. Expert systems rely significantly on the domain experts' expertise to define the system's rules. Nevertheless, obtaining domain expert's knowledge and turning them into rules is difficult. Furthermore, novel circumstances whose rules are not present in the knowledge base cannot be handled by expert systems [48].

- Fuzzy logic: Problems with imprecise, unclear, noisy, or missing inputs can be solved using fuzzy logic [48].In this approach, fuzzy sets are employed to model and represent the behaviour of the system [48]. Fuzzy logic offers a human-like technique of representation and reasoning through the use of linguistic variables. A system's language description is far more effective than its mathematical or numerical description, while being less precise [48].

1.5.3 Hybrid Methods

To enhance fault detection performance in PV systems, hybrid methods typically combine two different methodologies [24]. Additionally, this method is utilized to identify multiple fault occurrences and differentiate between faults with identical signatures [24]. The aforementioned

TH-4096_186151013

methods can be incorporated as needed in hybrid techniques. Unmeasured process parameters required for model-based techniques are often obtained through data-driven methods [44]. To improve accuracy, data-driven and model-based approaches can be combined.

1.6 Requirements of a Data Driven Predictive Maintenance System

Some of the requirements of a typical predictive maintenance system are described below [33]:

R1- Operational assessment: Here, the expenditure for implementing a PdM system and the loss incurred due to failure of equipment should be analysed properly.

R2- Data acquisition: data acquisition should be an indispensable part of the PdM system.

R3- Feature engineering: A major challenge is deciding whether to go for feature engineering or not. Accordingly, one can choose either traditional machine learning (ML) if feature engineering is to be used or go for deep learning (DL) technique if raw signals are to be used.

R4- Modeling: Model building is an important part. Model development, training, and model optimization require lot of attention.

1.7 Importance of Fault Diagnosis and Prognosis in Photovoltaic Systems

Even though PV systems are becoming more and more common across all renewable energy sources, faults are serious concerns for their efficiency, safety, and reliability [16]. Numerous PV modules connected via cables, junction boxes, converters, battery banks, inverters, and other devices may make up a typical solar PV system. These components are susceptible to a variety of problems even though PV systems lack moving parts [49]. PV systems are vulnerable to a variety of faults since they are installed outdoors and are subjected to adverse and unpredictable environmental conditions, including severe weather, rain, heat, storms, dust, wind, snow, etc.

TH-4096_186151013

[50]. Therefore, exposure to adverse external conditions may impair the system's ideal performance [5].

Moreover, because of series–parallel connections of PV modules in a PV array, voltage and current ratings are increased which may cause direct current (DC) arcs or higher risk due to large fault currents [49]. Although large-scale PV systems have diagnostic and fault detection systems, there are still instances of catastrophic failures that result in fire threats due to unidentified flaws. According to an analysis of 80 fire incidents associated with PV systems in the United Kingdom, 58 of these incidents were caused by PV system faults, 16 were unrelated to PV system faults, 6 were caused by unknown reasons, and up to 33 of these incidents happened during installation [51].

According to [51], a number of fire-related events were caused by malfunctions in DC isolators, connectors, cables, inverters, PV modules, and combiner boxes. In addition to creating additional damages and lowering the capacity to generate energy, if the faults are not identified before they happen, they will accelerate system ageing and perhaps increase the danger of fire hazards [8]. As a result, there is a high likelihood of reduced system efficiency [49], quicker system ageing [49], fire hazard risk [49], damage to PV modules and associated cabling [15], as well as heightened risks of DC arc faults [15],[51].

Although a lot of advancements have been made so far in fields like array reconfiguration, maximum power point tracking, grid interconnection, etc., awareness of fault diagnostics of PV systems is still lacking [16]. In order to avoid electric shocks and risk of fire hazards, standards like National Electric Code (NEC) 690, IEEE standard 1374, and European International Electrotechnical Commission Standard (IEC) 62548 are required to be followed in PV systems [16]. Inspite of following these standards also, faults occur in PV arrays which is evident from recent fire related accidents because of undetected faults [51], [20], [24], [16]. Generally, in PV systems, protection devices like ground fault detection interrupters (GFDI), overcurrent protection devices (OCPD), etc. are already available for the protection of PV components and fault detection. But because of high fault impedance, low irradiance, nonlinear output of PV arrays, availability of MPPT controllers etc., faults may remain undetected in PV systems [49], thereby deceiving GFDI and OCPD devices. However, there are some limitations with these protection devices, e.g., they can detect faults if there is large fault currents [16]. But there are situations where there may be slow and gradual leakage of current over a long period of time which may

TH-4096_186151013

cause large accumulation of current after a certain duration, thus aiding arc faults. Because of this difficulty in the protection schemes which is known as blind spots, there is always risk of DC arcing and fire hazards [49]. Therefore, the existence of blind spots require special attention [15].

Moreover, PV systems have unique fault scenarios which are different from conventional power sources [15]. Generally, some major faults, namely, line-ground (LG), line-line (LL), and arc faults are protected by PV protection standards [16]. The PV arrays are still vulnerable to other faults like bypass diode failures, faults due to shading, open circuit (OC) faults, hotspots, connection faults, module degradation etc. [16]

Fault analysis in PV arrays is necessary to improve system efficiency, safety, and reliability [15]. For instance, two incidents of fire hazards in solar PV systems were reported by Zhao et al.[15], one of which was caused by a ground fault in a PV system in Bakersfield, California, and the other of which was connected to a line-to-line fault in PV arrays in California, USA. Faults in PV systems may be the cause of significant energy loss in PV arrays in addition to fire threats [52]. According to a monitoring research by Firth et al., fault-related PV system power loss was predicted to be 18.9% annually [52]. Partial shade results in an annual energy loss of 10–20% or more, according to [53]. This is because certain PV modules have low irradiance, and mismatch loss results from variations in irradiance across the system. Zhao et al. divided faults into three categories based on their location: (i) PV array faults, (ii) power conditioning unit failure, and (iii) grid abnormalities [15].

Thus, to increase the reliability and lifetime of a PV system, there is an urgent need to have the following requirements: monitoring of energy production in real-time, diagnostics of PV systems, and prediction of the RUL of equipment or systems [8]. Therefore, there is need for extensive research for developing better and foolproof diagnostics and prognostics techniques for PV arrays.

A PV system consists of several other components other than the PV arrays. These components are referred to as balance of system (BoS) and consists of DC to DC converters, DC to AC converters (i.e.,inverters), batteries, battery charge controllers, supporting structures for mounting the PV panels, protection units, cabling etc. As reported in literature, there have been a number of fire related hazards caused by arcing in PV systems [51], [20]. Few of these incidents had occurred due to faults in balance of systems (BOS) components. Based on a survey conducted on 35 PV systems by Sandia national laboratories, the existence of 54% of the non-functional modules were because of BOS component failures like failures in fuses, switches, surge arrestors, dc contactors etc. [30]. Inverter faults are related to many PV incidents [54]. Identifying inverter fault well ahead of time may prevent PV incidents and reduce downtime. Thus, in order to ensure adequate safety and better performance, there is a need of diagnostics, prognostics, and health management of BOS components, including inverters in PV systems.

Thus, there is an urgent need to have monitoring, diagnosis, and prognosis for PV arrays, and BOS components along with inverters in photovoltaic systems. The aim of FDD techniques is to detect faults in systems or equipment and diagnose their causes. Prognostics is another mechanism to predict early fault detection and estimate the RUL of the system or equipment. Diagnostics, and prognostics mechanisms are collectively known as predictive maintenance. Many authors also refer to it as condition based maintenance. PHM is another term commonly used by researchers in the PHM community. There are many advantages of predictive maintenance such as failure prediction ahead of their occurrences, minimized unscheduled maintenance, decreased downtime, timely repair actions, extended maintenance cycles, reduced life-cycle costs, decreased inventory etc. [8].

1.8 Research Motivation

The research is motivated by the following factors:

- Because of intermittent nature of photovoltaic (PV) output due to varying weather and environmental conditions, setting up of threshold for fault analysis may not be accurate all the time. Therefore, there is a need for learning algorithm.
- In large scale PV systems, because of continuous monitoring of PV systems, usually through supervisory control and data acquisition (SCADA) systems, massive amount of data are accumulated over a period of time. The traditional classifiers may be inadequate to handle these data in order to identify faults. On the contrary, deep learning algorithms can handle a huge amount of data.
- Deep learning has shown its potential in areas like image and audio processing. They

TH-4096_186151013

are capable of learning complex input to output mappings. It is worth noting that their applicability is limited to dynamic situations where the mappings can be influenced by context. Moreover, the application of contextual deep learning to the problem of predictive maintenance of solar PV systems is novel. Therefore, this is the main motivating factor to investigate and develop learning mechanisms using Contextual Deep Learning (CxDL) for predictive maintenance, where the relationships within the data and operational environment play an important role.

- Application of Neural networks and other AI-based approaches has been widely used in the area of predictive maintenance. Also, AI-based condition monitoring of solar PV systems has been extensively studied and implemented by numerous researchers. However, deep learning techniques are yet to be explored fully in fault diagnosis and prognosis of PV systems.
- Currently, the Indian Government is taking several initiatives towards promoting the use of solar PV systems in smart grids, residential rooftops, and building applications. Maintenance is crucial for such systems as a malfunction of one of the components of such systems can cause hazardous accidents like fire in addition to downtime, and failure.
- The traditional approach of manual and visual inspection is not feasible for large scale photovoltaic (PV) systems. Therefore, there is a need for automated condition monitoring and predictive maintenance of such systems.

1.9 Problem Definition

Condition monitoring and predictive maintenance are essential in renewable energy systems since a malfunctioning component can result in power fluctuations, costly downtime, and system breakdowns. This underscores the need for early fault detection and proactive intervention. While FDD techniques have been explored extensively, solar PV systems have yet to fully leverage AI-based, particularly deep learning approaches for predictive maintenance. Moreover, current approaches often overlook the integration of contextual data such as geographical location, weather, and component-specific details, which can enhance the accuracy and relevance of predictive models. For monitoring applications, context information is important since it allows for more insightful data analysis and interpretation. With reference to predictive maintenance, the notions of "context-aware systems" and "context" are yet to be explored [55]. Additionally, issues in AC subsystems and inverter faults are especially underrepresented in existing research. To summarize, the key research gaps (RGs) with reference to FDD in solar photovoltaic systems are as follows:

- **More emphasis on model-based methods (RG1):** Researchers predominantly give more emphasis on model-based, more specifically electrical methods, with little attention towards data-driven methods for FDD in PV systems.
- **Limited adoption of deep learning (RG2):** Although deep learning techniques have demonstrated significant potential across various domains, their application to fault detection and diagnosis in renewable energy systems, particularly in solar PV systems, remains relatively unexplored.
- **Incorporation of contextual information in deep learning models (RG3):** The integration of contextual features into DL models for predictive tasks has received limited attention in the literature. To the best of our knowledge, no prior study has employed context-aware DL methods in solar PV systems for the purpose of FDD.
- **Limited research on early fault prediction (RG4):** Data-driven methods for FDD primarily focus on detecting faults after their occurrence, with limited emphasis on predicting potential failures in advance.
- **Lack of adequate research on inverter subsystems (RG5):** Among the different subsystems of a PV system, inverter subsystems account for the largest share of abnormal events relative to PV arrays. Despite this, inverter faults have not been adequately investigated in the existing research.

The following are the primary research questions (RQs):

- **RQ1:** Generally, in large scale grid connected photovoltaic systems, weather data such as wind speed, ambient temperature, module temperature, horizontal irradiation, tilt irradiance etc., and inverter data such as energy produced at current time, cumulative energy upto current time, DC current, DC voltage, AC power, DC power, power factor etc. are

TH-4096_186151013

continuously monitored and stored using supervisory and data acquisition (SCADA) systems (e.g., every 15 minutes on a daily basis for several years). Thus, massive amount of SCADA data are available from such systems. Inverter faults also occur frequently in PV systems. How can SCADA data, which continuously monitors and stores weather parameters (e.g., wind speed, ambient temperature, irradiation) and inverter metrics (e.g., DC current, AC power, power factor), along with contextual information be leveraged to develop AI-based models for early fault detection and diagnosis of various faults in large-scale grid-connected photovoltaic systems? Specifically, how can AI-based models be developed and applied to identify and prognose inverter faults, and other faults in PV systems such as partial shading, degradation, short circuit faults, open circuit faults, ground faults etc., thereby reducing system downtime and improving operational efficiency?

- **RQ2:** Various contextual attributes, such as seasonal variations, time of day, geographical location, weather, and operating environments, have a significant impact on PV system performance. Can incorporating contexts into predictive models improve early fault prediction in PV systems?

1.10 Research Objectives

The primary objective of this research work was to investigate and develop both traditional machine learning (ML) models and deep learning (DL) based models, particularly leveraging context information for predictive maintenance of renewable energy systems with special emphasis on diagnostics and prognostics of PV faults. Further, the potential benefits of deep neural networks and context information were leveraged for early detection and diagnosis of faults such as short circuit faults, open circuit faults, partial shading, and inverter faults for grid-connected PV systems while considering the research gaps (RG1-RG5) and research questions (RQ1-RQ2) discussed under problem definition.

The primary objective was achieved through the following sub-goals:

- SCADA data from a 10 MW operational grid-connected PV plant located in the state of Gujarat, India, were analyzed to develop predictive models addressing the research

TH-4096_186151013

questions (RQ1-RQ2).

- Fault detection approaches based on both regression and classification techniques were explored.
- Contextual information was utilized to develop deep learning-based models aimed at improving performance in fault detection and diagnosis.
- In addition to the real-world dataset, an open dataset was used to evaluate the effectiveness of the developed models.
- The best-performing model was selected to establish a comprehensive framework for early fault prediction and categorization of various faults as part of predictive maintenance.

1.11 Research Contributions

The key contributions of this research are as follows:

- A comprehensive analysis of SCADA data from a 10 MW operational grid-connected PV plant is conducted to examine parameter correlations and the influence of contextual factors such as seasonal variations.
- Fault detection methods based on both regression and classification are explored using a range of traditional ML and DL models.
- Various methods for integrating context into the existing ML models are explored, and a comparative analysis is performed.
- A novel deep sequential model, Context-Sensitive Long Short-Term Memory (CS-LSTM), is proposed by incorporating contextual information for improved fault detection.
- An end-to-end framework for early fault detection and diagnosis is developed, integrating CS-LSTM for early fault prediction and a Random Forest classifier for fault diagnosis.

TH-4096_186151013

1.12 Organization of the report

The subject matter of the report is presented in the following seven chapters:

- ✓ Chapter 1 gives an overview of the research background, and then discusses various types of faults in solar photovoltaic (PV) systems. Types of maintenance are discussed next, emphasising various components of predictive maintenance. Various approaches of predictive maintenance in solar PV systems are discussed next. Finally, this chapter provides the problem definition, research issues, research objectives, and contributions of the research.

- ✓ Chapter 2 explores the prior works of FDD in solar PV systems. The chapter focuses on data-driven approaches and provides a systematic review of various AI/ML-based methods, viz. Artificial Neural Networks, Support Vector Machine, Decision Tree, and Fuzzy Logic etc. This chapter also describes some of the widely used DL architectures for predictive maintenance and provides a review of DL methods for solar PV FDD.

- ✓ Chapter 3 presents solar PV system description and data analysis. Throughout this research work both real-world and open datasets from Laboratory installations are used. The real dataset is obtained from a 10 MW PV plant connected to a grid that is operational in Gujarat. The open dataset is generated from a PV system which is installed in the Laboratory of Innovation and Technology in Embedded Systems and Energy (LIT), Universidade Tecnológica Federal do Paraná-UTFPR, Brazil. This chapter details the two PV systems, describes the datasets used for model development and evaluation, and presents a comprehensive data analysis.

- ✓ Chapter 4 includes fault detection and diagnosis using traditional ML and DL models. Different regression models used for fault detection are described in this chapter. This includes conventional models and deep learning-based sequential models. It also covers fault classification models framed as a multi-class classification task to distinguish between different types of faults.

- ✓ Chapter 5 presents the various contextual approaches for early fault prediction. This chapter also describes the proposed contextual model, Context Sensitive Long Short-

TH-4096_186151013

Term Memory (CS-LSTM) which integrates contextual information such as season and time-of-day to improve the performance of predictive model and thereby improving fault detection.

✓ Chapter 6 presents an end-to-end framework for early fault detection and diagnosis in solar PV systems, integrating CS-LSTM for early fault prediction and a Random Forest classifier for fault diagnosis.

✓ Chapter 7 contains summary, conclusion, limitations, and future work.

TH-4096_186151013

This page was intentionally left blank.

TH-4096_186151013

Chapter 2

Literature Review

2.1 Introduction

In this chapter, the emphasis of the discussion is on reviewing the existing literature on data-driven predictive maintenance (PdM) methods in solar photovoltaic (PV) systems with special emphasis on PV arrays and Balance of System (BOS) failures. The traditional artificial intelligence (AI) based techniques for PdM of solar PV systems are discussed first, followed by a review of existing literature on deep learning (DL) methods applied to solar PV systems. Then, a review of related patents is done.

With the objective of identifying the problem of research, the preliminary survey was performed using the following databases: Web of Science, Google Scholar, IEEE Xplore digital library, and ScienceDirect with key words like "predictive maintenance", " fault classification", "fault diagnosis", "deep learning", and "solar photovoltaics" from 2010 to 2020. Similarly, searches were also conducted using keywords like "remaining useful life prediction of photovoltaic systems". As the research progressed further, relevant papers were also reviewed during the period 2021-2025. The summary of some of the results obtained is listed in table 2.1.

TH-4096_186151013

Name of database	Period	Number of publications retrieved	Number of relevant publications	Remarks
Web of Science	2010-2020	99	7	Keyword used is" fault diagnosis of solar photovoltaic systems" to search in all fields
Google Scholar	2010-2020	969	12	Keywords used are "deep learning for predictive maintenance in solar photovoltaic systems " containing the exact phrase " fault diagnosis" plus " fault classification" OR "fault detection"
IEEEXplore	2010-2020	5	4	Containing the phrases "predictive maintenance" OR "fault classification" OR "fault diagnosis", and refined by the keywords "deep learning", and then by"solar photovoltaics"
ScienceDirect	2010-2020	705	9	Containing the phrase "deep learning for predictive maintenance of solar photovoltaic systems"

Table 2.1: Number of research publications during the period 2010-2020

Research methodology: The main purpose of this review is to investigate the existing literature on AI methods in predictive maintenance of photovoltaic arrays and battery banks in photovoltaic systems. Therefore, it has been attempted to compile the relevant existing literature with the scope of this investigation limiting to the period from 2010 to 2025. As part of this research, various journal articles and conference papers pertaining to deep learning in predictive maintenance published in the above-mentioned period have been reviewed. Even though the scope of the investigation spans over the last 15 years, more emphasis is given to research work published in the last 10 years, considering their relevance from a research point of view. The articles were initially searched using Google Scholar and Web of Science and found to be scattered in

TH-4096_186151013

various sources. Finally, a literature search was conducted using two main electronic databases with scientific scope, namely, IEEE Xplore Digital Library and ScienceDirect. A total of 32 research papers were selected, which are of great relevance to the above-mentioned area. The inclusion and exclusion criteria are as follows:

Inclusion criteria:

- papers published in 2010 and later.
- papers containing applications such as fault diagnosis, condition monitoring, prognosis, and remaining useful life.
- Papers containing traditional AI techniques such as artificial neural network (ANN), support vector machine (SVM), decision tree (DT), fuzzy logic, etc.
- Papers containing deep learning architectures such as recurrent neural networks (RNNs), deep belief networks (DBNs), autoencoders (AEs), and convolutional neural networks (CNNs).
- Containing review or some form of experimental results.

Exclusion criteria:

- Published before 2010.
- Papers not related to predictive maintenance.
- Published as only abstract, editorial review, poster, book chapter, report.
- Papers that do not present any type of comparison or experimental results.
- Papers that make only propositions.
- Non-English articles

Latest developments in ML, DL, and advanced real-time monitoring systems have been instrumental in driving PdM applications in solar PV systems. PdM is generally categorized into three main approaches: (i) data-driven, (ii) model-based, and (iii) knowledge-based. Of

TH-4096_186151013

these, data-driven methods have attracted significant interest among researchers[33]. Data-driven techniques, particularly ensemble learning approaches and deep neural architectures such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), have shown considerable potential for accurately detecting faults in photovoltaic systems, including ground faults, line-to-line faults, and potential-induced degradation [56], [57]. Hybrid approaches integrating physics-based models with artificial intelligence (AI) techniques have demonstrated improved predictive performance, particularly when contextual variables such as irradiance and temperature are incorporated to enhance robustness and minimize false alarms [58]. Additionally, explainable AI (XAI) is becoming more popular to guarantee operator trust and transparency in automated diagnosis. To enhance fault localisation, electrical data is currently being combined with drone-based inspection systems and infrared thermography (IRT) [59]. These advancements show a distinct move towards autonomous, scalable, and smart FDD frameworks designed for solar PV system predictive maintenance, grid dependability, and efficient operation.

The literature provides a number of surveys and review studies [60], [61] on FDD for solar PV systems that cover several data-driven methodologies. There are also a number of review articles that emphasise on visual defects. For example, identifying and then categorising PV defects using DL methods like CNNs has been reviewed in [62]. It covers electroluminescence, thermal, and other imaging methods for the visible spectrum. The review identified important problems that must be addressed under various circumstances; these problems might be mitigated by contextual feature integration, but they are largely absent from the methods studied. For a thorough explanation of FDD in relation to PV systems, readers may consult these works.

In this chapter, we will specifically review only data-driven approaches that emphasise recent studies that are confined to ML and DL techniques.

2.2 Review of Data Driven Approaches for Diagnostics of Faults in Solar Photovoltaic (PV) Systems Using Traditional Artificial Intelligence (AI) Based Techniques

2.2.1 Based on Artificial Neural network (ANN)

Riley and Johnsohn [63] presented an ANN-based method for the prognostics and health management of photovoltaic systems. They observed two systems, each with 1.1 kWp capacity. The ANN model used ambient temperature, wind speed, and solar irradiance as inputs to forecast the AC power. For each system, they developed a separate ANN model. Each ANN comprised a 4×20×1 feed-forward network. Particle swarm optimization was used for training. The training data was obtained from a monitoring system for a duration of two months starting from system installation. The estimated AC power by the ANN models were compared with the measured power obtained from the monitoring system, a threshold of 10% was set for generating alarms. The authors mainly observed the effect of training period length on fault detection capability and evaluated performance based on 10% energy loss criteria. Initially, the ANN models were trained on two months of data, and best performance was obtained only for temperatures under which the models were trained, and for other temperature conditions (i.e., other seasons over which it was not trained) the ANN models produced lot of seasonal errors. When the ANN models were trained considering data for six months, their prediction performance increased significantly. Each of the ANN model could detect decrease in power when more than 10% of energy losses occured for a window period of 2 days. The author noticed that reducing the time windows increases the time sensitivity, but causes more false alarms, specifically when the irradiance is poor. The authors also observed that increasing window period decreased time sensitivity. For example, an alarm threshold of less than 10% energy loss can be used if the window period is increased to several days. A major drawback of the proposed system is that it can only detect faulty condition, but can not specify why it happens. Another disadvantage of the ANN model is that it is prone to more false positives for situations like temporary shading or when the inverter is off, and as a result high power loss will indicate fault. The authors did not mention the accuracy of their models. Also, there is no description of the types of faults detected other than comparing the measured and estimated powers. Moreover, this approach can not classify(i.e., isolate) the faults.

Chine et al. [64] presented a fault isolation framework for PV arrays that integrates ANN with threshold-based techniques. In their approach, fault detection is achieved by evaluating the deviation between the simulated and measured power outputs of the PV array. Following fault detection, several PV string parameters, including short-circuit current (I_{SC}), open-circuit voltage (V_{OC}), output current, output voltage, and the number of maximum power point tracking (MPPT) controllers, are supplied to an ANN for further analysis. The authors introduced the concept of *fault signatures*, which represent distinct combinations of these parameters under different fault conditions. Subsequently, a threshold-based mechanism is employed to isolate and classify faults according to the collective behaviour of the attributes (e.g., no change, increase, or decrease). For example, for a fault like an open-circuited module, the attribute values of V_{OC}, I_{SC}, maximum power point (MPP) current (I_{MPP}), and MPP voltage (V_{MPP}) are decreased. Chine et al. [64] utilized ANNs, specifically focusing on radial basis function (RBF) and multilayer perceptron (MLP) models, to categorize fault signatures that threshold-based methods could not distinguish. The MLP model featured two hidden layers, each containing 13×13 neurons, and was trained with the Levenberg–Marquardt algorithm, employing logarithmic sigmoid activation functions. Conversely, the RBF network featured a single hidden layer comprising 49 neurons and utilized Gaussian kernels, with their parameters determined through k-means clustering. A total of 775 simulated patterns were created, with 80% designated for training and 20% for testing. The ANN models were employed to identify faults like bypass diode, short circuits, reverse-biased bypass diodes, shunted bypass diodes, and module connection faults, which could not be detected using threshold-based methods. The MLP and RBF models achieved classification accuracies of 90.3% and 68.4%, respectively. The threshold-based approach could isolate open-circuit fault, partial shadow fault, shadow effect with faulty bypass diode, connection fault, and shadow effect with connection fault in a PV array. The authors used a PV string comprising four PV modules in series, with each module containing 72 cells connected in series. However, the ANN models were developed using simulated data for different attribute values rather than measurements collected from real PV systems. Moreover, fault detection relied on a model-based approach in which simulated outputs were compared with measured values using predefined thresholds. In addition, the classification performance

TH-4096_186151013

achieved by the ANN models was relatively poor.

An ANN-based approach was introduced by Mekki et al. in[65] for detecting partial shading in PV modules. The developed model used cell temperature (T) and irradiance (G) as inputs to estimate current and voltage at the PV module's output. The ANN architecture utilised an MLP that included two neurons in the input layer. The two hidden layers contained 7, and 12 neurons, respectively. The output layer comprised two neurons corresponding to current and voltage. Electrical measurements, including current and voltage, together with meteorological parameters such as temperature and irradiance, were collected from a rooftop PV module over a period of 30 days. Several backpropagation algorithms were evaluated for training, among which resilient backpropagation yielded the best performance. 70% of the dataset was reserved for training, and the remaining was used for testing. To assess the impact of partial shading, different shading scenarios were considered by varying the shaded cells between 5 and 50. The voltage and current estimated by the ANN model were compared with measured values to characterize the operating condition of the PV module. For the case of 5 shaded cells, the mean absolute errors (MAEs) in voltage and current estimation were 1.71 V and 0.0667 A, respectively, whereas for 50 shaded cells, the corresponding MAEs increased to 11.47 V and 0.52 A. Despite these promising results, the proposed approach was not compared against other existing methods. Furthermore, the experimental study was conducted using data collected from a single PV module, which limits the generalizability of the models to larger PV systems operating under diverse weather and environmental conditions.

An ANN based method was presented by Syafaruddin et al. in [66] to locate short-circuit fault in a PV array. The key principle is based on the fact that when a module is short-circuited, its output voltage becomes zero, which in turn affects all the modules of the same PV string as well as the adjacent strings. The architecture of the ANN included an input layer, a single hidden layer, and an output layer, utilizing a tangent sigmoid function for activation. Solar irradiance (G), cell temperature (T), MPP current (I_{MPP}), and MPP voltage (V_{MPP}) were provided to the ANN model as inputs. The output layer was composed of six neurons with each neuron indicating the estimated output voltage of a PV module corresponding to a PV array where the modules were configured in 3×2 array. The quantity of neurons was set during training in the hidden layer. The authors used the gradient descent algorithm for weight adjustment. Current and voltage were recorded when there were variation in cell temperature and irradiance, and 30

TH-4096_186151013

samples were collected by the authors for training the network. Multiple identical ANN models were used to locate short circuit faults at various modules (e.g., the authors used 12 ANN models to analyse 12 fault cases). However, this approach could precisely identify the location of short-circuit fault in a single string only. Since multiple ANN models need to be used for various fault cases even for a smaller configuration (e.g., the authors used only 3×2 array of modules), this approach is likely to be infeasible in case of larger PV systems. Moreover, for a large system, enormous amount of data will need to be trained considering huge number of ANN models, which in turn will require huge amount of memory.

Akram and Lotfifard [67] introduced a framework for classifying faults in photovoltaic (PV) systems, specifically targeting open-circuit and line-to-line faults. The framework utilized a probabilistic neural network (PNN) that relied on radial basis functions and was structured with an input layer, a hidden layer, a summation layer, and an output layer. The network's input features comprised the MPP current (I_{MPP}), voltage at MPP (V_{MPP}), power at MPP (P_{MPP}), cell temperature (T), and solar irradiance (G). The hidden layer's size was determined by the number of training samples, while the summation layer included four neurons, each corresponding to one of the four operating conditions: normal operation, line-to-line fault, 75% open-circuit fault, and 50% open-circuit fault. The final classification decision was provided by a single output neuron. Simulation-based evaluation of the proposed method yielded 98.53% classification accuracy. Despite the high reported accuracy, the study relied on data derived from manufacturers' datasheets rather than measurements obtained from actual PV installations. Furthermore, the fault scenarios considered in the study were artificially simulated and did not represent real fault conditions encountered in operating PV systems.

A fault prediction method for inverter was developed by Betti et al. in [68]. The fault prediction method consists of two modules: a generic fault prediction module implemented using a clustering algorithm called self organizing map (SOM), and specific prediction module for fault class implemented using artificial neural network (ANN). The authors considered SCADA data such as DC voltage, DC current, DC power, AC voltage, AC current, AC power, module temperature, ambient temperature, global horizontal irradiance, and global tilt irradiance as inputs to both the modules. As claimed by the authors, generic inverter faults can be predicted by 7 days ahead of time of occurrence with 95% sensitivity and 80% specificity, and specific fault class can be predicted ahead of time occurence from few hours to 7 days in advance.

Charkhgard and Farrokhi [69] introduced a hybrid method that integrates an extended Kalman filter (EKF) with ANN for estimating state of charge (SOC) in Lithium-ion (Li-ion) batteries. The ANN structure comprised an input layer, one hidden layer, and an output layer. The network received inputs such as the battery voltage from the previous sampling moment, the terminal current at the present sampling moment, and the estimated SOC at the current time step. A Gaussian activation function was utilized in the hidden layer, and the network's output represented the battery terminal voltage at the current sampling moment. The SOC, used as an input for the ANN, was derived through the ampere-hour counting method. This model was combined with EKF by the authors to estimate the battery's SOC. For data acquisition, an interfacing circuit was employed to sample the current, voltage, and temperature of a Li-ion battery. The proposed method's performance was evaluated using both offline and online data, revealing a root mean square error of 2% and 3% between the actual and estimated SOC, respectively. However, the authors performed their investigations on a single battery only, and the data for training the ANN were obtained from a new battery that was healthy. Therefore, the trained ANN is unlikely to give acceptable output as the battery ages. Moreover, the authors performed their experiment only at room temperature and did not explore the proposed method in different environmental conditions and ambient temperatures.

2.2.2 Based on Support Vector Machines (SVM)

In [23], Yi and Etemadi introduced an SVM-based method to identify line-to-line faults in PV arrays. The authors extracted four features from the PV array itself. The first feature was the fill factor (FF), the second feature was the ratio between the multiplication of normalised current output by cell temperature and normalised voltage output by solar irradiance, where normalised current output is the ratio between current output and short-circuit current, and normalised voltage output is the ratio between voltage output and open-circuit voltage.

$$Feature_2 = \frac{I_{NORM} \times T_{cell}}{V_{NORM} \times G} \tag{2.1}$$

TH-4096_186151013

where,

$$I_{NORM} = \frac{I_{PV}}{I_{SC}} \tag{2.2}$$

and

$$V_{NORM} = \frac{V_{PV}}{V_{OC}} \tag{2.3}$$

The changing rate of normalized output current and voltage of the PV array were the third and fourth features, respectively. By varying irradiance, cell temperature, fault impedance, and mismatch percentage, various fault conditions were simulated by the authors, and considered 6 faults as labels for training purpose and 426 faults were used for testing. Though it becomes difficult to detect faults with decreasing mismatch percentage or increasing fault impedance, the results showed that 50% or more L-L faults can be detected with 20% mismatch using the proposed approach. The main demerit of the proposed method was that the sample size for training the SVM was very less (e.g., only 6 in this case). Moreover, the faults were created using simulation instead of real fault data.

A technique for identifying short-circuit faults in PV systems, employing SVM in conjunction with the K-NN algorithm, was introduced by Rezgui et al. in [70]. Data were obtained from a database containing PV generator data. As claimed by the authors, the classification accuracy was 68 to 75.8%. However, the authors did not specifically mention the features used as inputs in the SVM. Also, detailed description of the datasets used were not explained properly.

Patil et al. [71] developed a method with two stages of support vector machines (SVM) to estimate RUL of Li-ion batteries. Data from discharge cycles of Li-ion batteries, which were subjected to different operating conditions, were examined by the authors. The datasets used by the authors contained information about current, voltage, capacity, voltage load, current load, and temperature of batteries corresponding to every discharge cycle. All parameters, aside from cell capacity, were acquired over time with non-uniform sampling rate during discharge. The authors used the operating parameters of 19 batteries in their work. Given that battery ageing affects measured voltage, current, and temperature, the authors identified eight parameters from the voltage and temperature curves. These parameters include the minimum and maximum values for each curve, along with the times at which they occur. Additionally, 13 more parameters were derived from voltage, temperature, and current like capacity of battery, energy of voltage and temperature signals, fluctuation index of voltage and temperature signals, curvature index of voltage and temperature signals, concave convex index of voltage and temperature signals, skewness index of voltage and temperature signals, kurtosis index of temperature signals. The features of discharge cycles were used to build the classification model. Counting of discharge cycles were continued till end of life (EOL) of a battery, and each discharge cycle was categorized into four classes, out of which one class represented the samples the near the end of life of battery. For extracting critical parameters, the authors used principal component analysis (PCA). The regression model was used for estimating the RUL of the already classified battery data near the end of life. The authors conducted experiments pertaining to multiple battery data, single battery data, two batteries at multiple temperatures, multiple battery data at low and high temperatures etc. For multiple battery, the authors considered three batteries that have same operating temperature, EOL condition, number of cycles, and varying discharge current and end voltage. For classification, SVM was used with radial basis function as the kernel. Voltage of energy signal and fluctuation index of voltage signal were the features selected for training the SVM. The authors allocated 70% of the data for training purposes, while the remaining 30% was reserved for testing, achieving 94.15% classification accuracy with an RMSE of 0.2126. They examined how well their proposed model could predict the RUL of a battery by focusing on the class close to the end of its life, using 70% of the data for training and 30% for testing. The inputs to the SVM were the voltage of the energy signal and the fluctuation index of the voltage signal, with RUL as the target output. As claimed by the authors, the root mean squared error for regression was 0.2420, and gave a prediction accuracy greater than 99%. Though the authors did an extensive studies based on various battery combinations, manual feature extraction is always a tedious task.

Nuhic et al. [72] developed a method for the prediction of RUL of Li-ion batteries using SVM. They used input features like the capacity of the cell at the beginning, cell throughput measured in Ah, the minimum and maximum values of SOC (state of charge), the most recent capacity degradation, cycling temperatures, and the time span between the initial and target capacity assessments. The target training vector was exclusively determined by the cell capacity measured at the end of the driving session. The authors investigated six Li-ion battery cells, three of which had undergone cycle ageing, while the remaining three retained higher capacities. A total of 18 capacity tests were conducted, accounting for variations in temperatures, C-rates, and SoCs. Of these, 11 datasets were employed for training, and 7 were used for testing. The authors used an innovative idea of processing capacity tests with load collector profiles to increase the number of datasets and the features in the datasets. They evaluated the training results using conventional datasets, a single load collective, and all load collectives, finding that the best training outcomes were achieved with all load collectives. Though the authors did not specifically mention the RMSE for SoH estimation in their paper, it can be inferred that their experimentation produced RMSE of less than 0.0008 for conventional datasets, 0.0001 with one load collective, and 0.000025 with all load collectives, thus experimentally demonstrating best performance with all collectives. It is to be noted that the authors did not use any feature reduction technique in order to determine the critical features which could have given more insight regarding dependence on important critical features. Also, the authors conducted only 18 capacity tests in their investigation. Though the authors showed simple RUL prognosis, prediction likelihood was not included in their result.

2.2.3 Based on Gaussian Mixture Model (GMM)

In [46], a data-driven framework was proposed to identify and categorize anomalies at the PV string level by utilizing PV SCADA data. This framework includes two distinct methods: one is an unsupervised learning technique known as hierarchical context-aware anomaly detection, and the other is a supervised learning method aimed at classifying anomalies. The process of detecting anomalies was carried out in two consecutive phases to identify abnormal patterns. Initially, Local Context Aware Detection (LCAD) was employed, followed by Global Context Aware Detection (GCAD). LCAD is responsible for detecting unusual PV strings from individual combiner boxes. The proposed framework assumes that PV strings connected to the same combiner box exhibit similar current profiles under normal operating conditions, and therefore only anomalous strings deviate from this common behaviour. The LCAD stage employs a Gaussian Mixture Model (GMM), referred to as AutoGMM, whose parameters are determined using the Expectation Maximisation (EM) algorithm. Among the identified clusters, the one with the highest current centroid is treated as representing normal operation, while the others are treated as abnormal. To quantify the severity of anomalies, the authors introduced an index,

TH-4096_186151013

known as local anomaly index (LAI), which signifies the proportion of time for which a PV string's current is classified as abnormal. A higher LAI suggests a greater chance of abnormal operation. At the system level, the GCAD stage uses LAI values to minimise false alarms. GCAD specifically integrates K-means clustering with second-order differencing to assess if a PV string is anomalous. For classifying anomalies, both aggregate and spectral features are extracted. Aggregate features consist of the mean, standard deviation, median, and maximum of the deviation sequence calculated as the difference between the normal cluster centroid's current and that of a PV string over time. Fast Fourier Transform (FFT) was used to derive the spectral features. The classifiers evaluated include SVM, Bagging, and XGBoost. The proposed method achieved detection accuracies ranging from 78.8% to 90.2%, while XGBoost yielded the best performance using both aggregate and spectral features, attaining a classification accuracy of 93.0% and a recall of 92.8%. Furthermore, the approach demonstrated robustness to variations in irradiance and weather conditions. However, its capability to detect and isolate shunt faults was not investigated for a negligible maximum power point current.

2.2.4 Based on Collaborative Filtering (CF)

A collaborative FDD framework for PV systems was introduced in [73] that uses collaborative filtering techniques commonly employed in recommender systems to predict the behaviour of a target user based on information from similar users. Extending this concept to PV systems, the proposed method estimates the power generated by a target PV component using the power outputs of similar components. The method utilises current measurements acquired from the SCADA system and performs anomaly detection in three phases. The first phase is responsible for current prediction, where the similarity between a PV string and its neighbouring strings is quantified using the Euclidean distance metric. These similarity measures, together with the neighbouring strings' current values, are used to estimate the current of each PV string. During the second phase, the seriousness of faults is evaluated by determining the daily total of residuals that arise from the difference between the forecasted and actual current values. Under the assumption of Gaussian-distributed noise, noise-induced residuals are expected to remain close to zero, whereas residuals associated with actual faults exhibit a monotonically increasing trend. Finally, PV strings that exhibit larger fault severity indices are recognised as defective.

The proposed method is capable of detecting faults even when fault-induced power losses are difficult to distinguish from variations caused by noise. Experimental results demonstrated that the approach outperformed existing data-driven techniques in terms of computational efficiency, effectiveness, and robustness, and achieved detection accuracy between 96.3% and 97.1%.

2.2.5 Based on Fuzzy Logic (FL)

Ducange et al. [74] introduced a method for detecting faults by employing a Takagi-Sugeno-Kahn Fuzzy Rule-Based System (TSK-FRBS). This system was utilised to predict the output power of a PV array. The model for fault detection took temperature (T) and solar irradiance (G) as inputs, and DC power (estimated) was the output. The estimator was designed using TSK-FRBS. For fuzzy input partitions and generating rule antecedents, subtractive clustering was used, and the rule consequent was estimated by the linear least squares algorithm. The estimator was trained using simulated data. The estimated power obtained from the model was compared with the actual power of a PV array, and a fault was detected if the difference in power, expressed as the mean absolute percentage error, exceeded a threshold of 5%. Fault conditions due to short circuit in cells, shading, and broken cells on a module were considered for experimentation. The authors used simulation framework for generating both faulty and normal conditions, and the results showed that 90% or more faults could be detected by the system. However, predefined fuzzy rules are needed by the proposed method, which are not easy to determine precisely since they require a lot of human expertise and experience in addition to having a proper system configuration. Moreover, the authors considered only short circuit, partial shading, and open circuit faults.

Bonsignore et al. [75] introduced a neuro-fuzzy method to identify faults in PV systems. They employed adaptive neuro fuzzy inference system (ANFIS) and used a simulation, referred to as the "hybrid simulator PV model," to produce the data required for training. Through simulation, they generated I_{MPP}, V_{MPP}, I_{SC}, and V_{OC} for various levels of irradiance and module temperatures. Two parameters D_1 and D_2 were derived based on I_{MPP}, V_{MPP}, I_{SC}, V_{OC} which are defined as :

$$D_1 = \frac{I_{MPP} - I_{SC}}{V_{MPP} - V_{OC}} \tag{2.4}$$

$$D_2 = \frac{0 - I_{MPP}}{V_{OC} - V_{MPP}} \tag{2.5}$$

The ANFIS model receives module temperature and solar irradiance as inputs, and the target outputs are I_{MPP},V_{MPP}, I_{SC}, V_{OC}, D_1, and D_2. Six characteristics were determined based on six parameters which differed from current-voltage curve in normal operating condition because of presence of faults. The authors defined the six characteristics based on variation of I_{MPP}, V_{MPP}, I_{SC}, V_{OC}, D_1, and D_2 which were different from normal I-V (current-voltage) curves, and these characteristics were used to detect faults. The authors also demonstrated fault diagnosis capability by using norm test in order to categorize the faults into various types like short circuited diode, partial shading, lower earth fault, upper earth fault, and normal condition. The results demonstrated that the method could categorize normal and faulty operations for earth faults, diode short circuit, and partial shading. However, the training data were generated using simulation only.

2.2.6 Based on Decision Tree (DT) and other advanced Machine Learning (ML) techniques

In their study, Zhao et al. [21] presented a technique that employed decision trees for identifying and categorising faults in solar PV arrays. Feature selection was performed using information gain, and the most informative inputs included normalised current, fill factor, PV array power, PV array voltage, and open-circuit voltage. This approach was tested for detecting partial shading, open-circuit, and line-to-line faults in PV modules. Based on experimental findings, the accuracy of fault detection and classification was found to be between 93.56% - 99.98%, and 85.43% - 99.8%, respectively.

TH-4096_186151013

Abdelsattar et al. [76] investigated various ML techniques for predicting power and detecting various PV faults. They used a dataset consisting of 97,333 samples and evaluated various ML models such as Linear Regression, XGBoost, Random Forest, CatBoost, Random Trees, and Gradient Boosting (GB). GB and CatBoost stood out in power prediction, achieving a peak R^2 score of 0.994. With reference to FDD of PV faults, CatBoost surpassed the other models, achieving 99.97% accuracy, 99.72% precision, 99.93% recall, and an F1-score of 0.9982, highlighting its effectiveness for FDD applications.

Noura et al. [77] introduced a framework for FDD in PV systems using the Extra Trees classifier. This framework functions in two consecutive phases: the first phase involves binary classification to detect faults, and the second phase employs multi-class classification for diagnosing the faults, resulting in high accuracy. To improve the framework's interpretability and transparency, the authors utilized explainable artificial intelligence (XAI) with SHapley Additive exPlanations (SHAP). Additionally, the authors highlighted the significance of using oversampling techniques to address class imbalance challenges in PV fault datasets.

2.3 Review of Data Driven Approaches on Diagnostics of Faults in Solar Photovoltaic (PV) Systems Using Deep Learning (DL) Techniques

ANNs with more than one layer of hidden units can be considered as Deep Neural Networks (DNNs). Such deep models with multiple levels of nonlinearity provided through multiple hidden layers having nonlinear activation functions in the hidden units are capable of modelling highly complex and nonlinear relations between input and output. However, the conventional training algorithms, such as backpropagation, are not suitable for DNN due to various limitations. The class of learning approaches, called Deep learning, aims at efficient training of deep architectures, mainly the DNNs. The earliest reported deep architecture was a multilayer perceptron comprising numerous hidden layers, introduced by Yann LeCun in 1989 [78].This network employed the backpropagation algorithm for handwritten ZIP code recognition. However, the increased network depth resulted in high computational cost and introduced optimiza-

TH-4096_186151013

tion difficulties, particularly the vanishing gradient phenomenon. As a consequence, efficient parameter estimation in deep networks became extremely difficult, causing research interest in deep neural architectures to decline for several years. Interest in deep neural architectures was revived in 2006 following the influential contribution of Hinton et al. [79], which demonstrated effective strategies for training deep models and significantly accelerated research in this area. Further progress was achieved through the work of Ranzato et al. [80] and Bengio et al. [81], who proposed stacked autoencoder architectures that substantially advanced the development of deep representation learning. Since then, DL methods have evolved into the most prominent paradigm in AI and have been successfully applied across a broad spectrum of domains, including speech and signal processing, computer vision, and natural language processing. Compared to conventional ML approaches, DL offers several important advantages [82]. By employing multiple layers of representation learning, these models can automatically derive meaningful features from raw input data, analogous to the hierarchical information processing observed in the human brain [83]. Moreover, deep learning systems are capable of learning task-specific representations directly from unprocessed data, thereby reducing the dependence on manual feature engineering for detection and classification tasks [84].

The following are some of the definitions of deep learning that are available in the literature [85],[86]:

- **Definition 1**: According to [82], “Deep learning is about learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features.”
- **Definition 2**: "Deep learning (also known as deep structured learning or hierarchical learning) is learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised" [87] [84].
- **Definition 3**: According to [87], [86] “deep learning is a class of machine learning algorithms that: (1) uses a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input, (2) learns multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.”

DL techniques can broadly be classified into discriminative (supervised) and generative (unsu-

TH-4096_186151013

pervised). Examples of discriminative models are CNNs and RNNs. Deep Boltzmann Machines (DBMs), DBNs, and AEs are examples of generative models. In addition to these, Deng et al. [86] mentioned another class i.e. hybrid models. The aim of hybrid models is discrimination but such models are assisted with generative or unsupervised deep models. For example, a DBN (generative model) can be trained in an unsupervised way to get the initial value of parameters. This is often referred to as pre-training. After pre-training, the same network can be used for discriminative task using the labels.

The body of research on deep learning-based methods for predictive maintenance in solar photovoltaics is rather small. Although there are many papers in solar forecasting, publications on predictive maintenance utilising certain DL techniques have only recently increased. The next section discusses a few deep learning techniques for solar photovoltaic system predictive maintenance.

2.3.1 Based on Convolutional Neural Networks (CNN)

The CNNs, introduced by Lecun et. al. [78], have been successfully applied to several practical applications. These networks employ the convolution operation in place of mathematical operation in at least one of their layers [88].

CNNs generally process data structured as grids or multiple arrays. For instance, a color image is made up of three two-dimensional arrays that hold pixel intensity values for the RGB channels, while sequences and signals are represented as one-dimensional arrays. A CNN is composed of multiple convolutional and pooling layers, typically ending with one or more fully connected layers that resemble traditional feedforward networks, as illustrated in Figure 2.1.

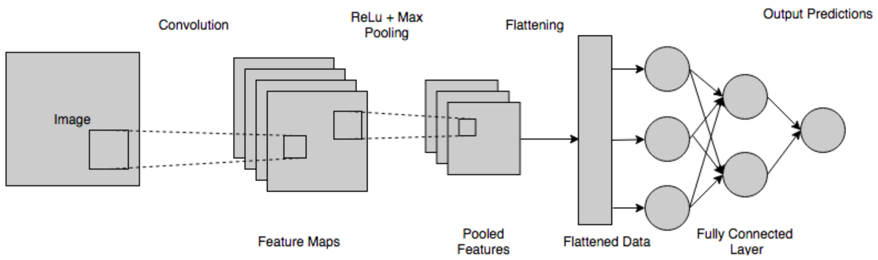


Figure 2.1: Basic architecture of CNN (source:https://www.i2tutorials.com/)

The initial layer of hidden units, which carries out the convolutional process, is linked to the input layer. However, not all neurons of the input layer are connected to the hidden layer; only local patches from the input neuron space (pixels of image) are connected to the hidden layer neurons as depicted in Figure 2.2. Each such local patch can be considered as a window and this window slides over the entire input neuron space. Each position of the window corresponds to a specific neuron within the hidden layer. The window slides through the entire input neuron space, moving horizontally from left to right and vertically from top to bottom. Let us consider an array of size, say, w x h x d, where w, h, and d indicate width, height, and depth, respectively. For example, a grayscale image of size 7 x 7 x 1 or an RGB image of 7 x 7 x 3. For the given grayscale image, a window (or patch) size of 3 x 3 x 1 will return a hidden layer with 5 x 5 x 1 neurons when the window slides over the entire input space as shown in Figure 2.2. Every neuron within the hidden layer is linked to a specific bias value b and a weight matrix W (called a filter or kernel) through which it is connected to the previous layer. The depth of this filter has to be the same as the depth of the input. The activation value at a neuron is computed by taking the scalar product between the filter and the patch connected to that hidden neuron and summing them up and adding the bias. Finally, a nonlinear function, such as a ReLU, processes the locally weighted sum. For each hidden neuron linked to a particular neuron in the input layer, this procedure is carried out repeatedly. The resulting array is called a feature map or activation map. All units within a feature map utilise the same filter and bias, while distinct filters are applied to various feature maps within a layer. A network configuration that includes only one feature map is capable of identifying one type of localised feature. In image recognition tasks, it is necessary to use multiple feature maps. Therefore, a full convolutional layer is made up of various distinct feature maps, as illustrated in Figure 2.1.

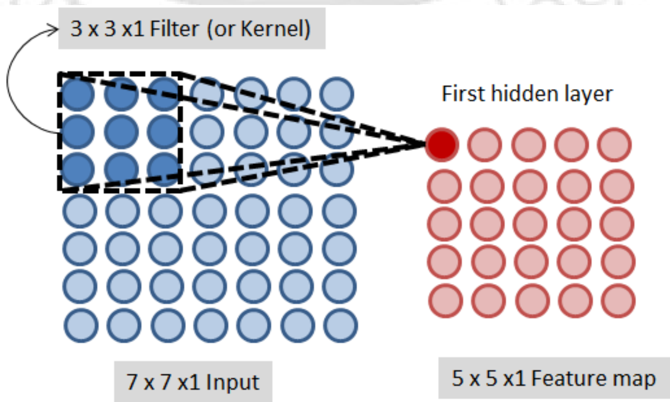


Figure 2.2: Input and hidden layer of CNN

The pooling layer follows the convolutional layer. Its primary function is to present the feature map from the preceding layer in a more compact form. In the pooling layer, a typical unit identifies the highest value from a specific section (i.e., patch) of the feature map. Adjacent pooling units receive input by shifting patches by more than one row or column. Typically, a CNN consists of two or more stacked layers of convolution, ReLU, and pooling, followed by fully connected layers. The usual stochastic gradient and backpropagation algorithms are used for training the network to learn the parameters.

Huuhtanen and Jung [89] investigated the use of CNNs for detecting abnormalities in PV panels. Their approach utilised historical power measurements, represented as time-series data, to identify malfunctioning panels. The underlying principle was to compare the observed power curve of a target panel with predicted power curve derived from neighbouring panels. Significant deviations between the measured and predicted power outputs were considered indicators of abnormal operation. The authors evaluated two CNN architectures: a fully convolutional model with two convolutional layers (CC) and a hybrid model consisting of a fully convolutional first layer followed by an unshared convolutional second layer (CUC). The proposed models were tested using both synthetic power data spanning 1000 days and real-world measurements collected over 157 days. For the synthetic dataset, the CC and CUC architectures achieved mean squared errors (MSEs) of 0.000037 and 0.000010, respectively, whereas for the real dataset the corresponding MSE values were 0.002589 and 0.002346. The overall MSEs reported for the synthetic and real datasets were 0.000421 and 0.003561, respectively. Although the proposed approach demonstrated promising performance on synthetic data, its effectiveness on real-world datasets was not very satisfactory.

Aziz et al. [90] proposed a CNN-based architecture for fault detection in PV arrays using simulated data. The method employed two-dimensional scalogra ms generated from PV system measurements to extract discriminative features for fault classification. Five fault scenarios were considered in the study: partial shading (PS), open-circuit (OC) faults, line-to-line (LL) faults, high-impedance/series arc faults, and faults occurring under partial shading conditions. The dataset comprised several electrical and environmental parameters, including solar irradiance (G), temperature (T), short-circuit current (I_{SC}), open-circuit voltage (V_{OC}), PV current (I_{PV}), MPP current (I_{MPP}), MPP voltage (V_{MPP}), and power at MPP (P_{MPP}). Additionally, the maximum power, current, and voltage at the boost converter were also utilized. Experimental results indicated that the fine-tuned pre-trained CNN achieved 73.53% accuracy for fault

TH-4096_186151013

detection . The authors also emphasized the significance of extracting discriminative and representative features instead of directly using raw data in noisy environments, obtaining a detection accuracy of 70.45%. Despite these contributions, the study relied entirely on simulated PV data rather than measurements from real PV installations. Furthermore, the achieved fault detection accuracy suggests that there remains scope for improving the proposed approach.

Tan et al. [91] introduced a method for assessing dust accumulation on PV panels using denoising convolutional neural networks (DnCNN). The proposed framework integrated the feed-forward DnCNN architecture introduced by Zhang et al. [92] with three variants of Residual Networks (ResNet), namely ResNet-101, ResNet-50, and ResNet-34. For comparative evaluation, the authors also combined DnCNN with VGG-16 and AlexNet architectures. The experimental dataset consisted of PV panel images collected over a period of four months. Out of the total dataset, 440 and 110 images were used for training and testing, respectively. The performance of the different DnCNN-based architectures was assessed after 1500 training iterations. The obtained accuracies for AlexNet, VGG-16, ResNet-101, ResNet-50, and ResNet-34 were 31.59%, 30.45%, 38.6%, 87.3%, and 85.9%, respectively. Among the evaluated models, DnCNN combined with ResNet-50 achieved the highest accuracy for classification for dust accumulation assessment.

Li et al. [93] introduced a DL-based approach for the analysis of defects in PV modules. The method employed CNNs to identify defect patterns from images captured using unmanned aerial vehicles (UAVs). The proposed framework was evaluated for several types of module defects like dust-induced shading, snail trails, gridline corrosion, encapsulant delamination, discolouration, as well as defect-free modules. The reported classification accuracies for these categories were 97.8%, 97.8%, 98.0%, 99.5%, 98.1%, and 98.5%, respectively, demonstrating the effectiveness of the CNN-based approach for automated PV module inspection.

Priyadarshini et al. [94] investigated the detection of visual defects in PV modules, including cell damage, glass fractures, soiling, and shading, which are often difficult to identify under challenging environmental conditions. For fault classification, the authors employed the EfficientNet-B0 deep learning architecture. Based on testing, the authors showed that the model effectively distinguished among various visual defects in RGB images of PV modules, yielding a classification accuracy of 97.24%.

Sridharan et al. [95] proposed a CNN-based framework for the automated classification of

TH-4096_186151013

defects in PV modules using aerial images acquired from UAVs. The proposed method demonstrated lower training time while achieving superior performance compared to conventional methods and pre-trained architectures such as VGG16 and ResNet50. Specifically, classification accuracies of 95.07% and 94.145% were obtained for uniform and non-uniform datasets, respectively.

2.3.2 Based on Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) represent a specialised class of deep learning architectures that are well suited for modelling sequential and temporal data [96],[97]. An important feature of an RNN is the presence of at least one feedback connection, which enables activations to circulate in a loop while having hidden states. This configuration allows for temporal processing and sequence learning. There are various kinds of RNN architectures. The basic RNN includes three layers: the input layer, the recurrent hidden layer, and the output layer, as illustrated in Figure 2.3 [97].

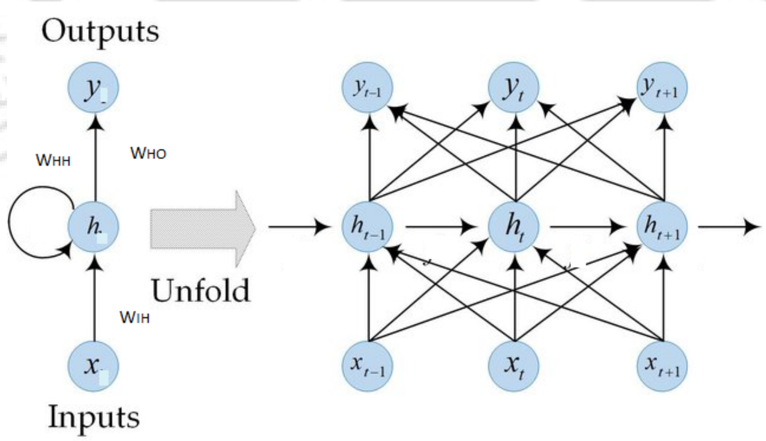


Figure 2.3: Basic structure of RNN

The input to the RNN is a time series vector such as $\{\ldots,\mathbf{x}_{t-1},\mathbf{x}_t,\mathbf{x}_{t+1},\ldots\}$, where where each input at time step t is given by $\mathbf{x}_t = (x_1, x_2, \ldots, x_N)$. In a fully connected RNN,the matrix $\mathbf{W}_{IH}$ defines the connection weights between the input layer and the hidden layers. The hidden units are interconnected through recurrent links and are also connected to the output layer as shown in the unfolded network of Figure 2.3. The recurrent connections enable the hidden layer to retain the network's internal state, which acts as its memory, and is given by (2.7).

$$\mathbf{h}_t = f_H(\mathbf{o}_t) \tag{2.6}$$

$$\mathbf{o}_t = \mathbf{W}_{IH}\mathbf{x}_t + W_{HH}\mathbf{h}_{t-1} + \mathbf{b}_h \tag{2.7}$$

Here, f_H denotes the activation function for the hidden layer, and the bias vector for the hidden units is denoted by $\mathbf{b}_h$.

The output of the network is given by (2.8)

$$\mathbf{y}_t = f_O(\mathbf{W}_{HO}\mathbf{h}_t + \mathbf{b}_o) \tag{2.8}$$

Here, f_O represents the activation function for the output layer, while $\mathbf{b}_o$ denotes the bias vector associated with the output units.

The network's loss function is determined by evaluating the difference between the target output $\mathbf{y}_t$ and the network's output $\mathbf{y'}_t$ at each time step, and it can be expressed as

$$\mathscr{L}(\mathbf{y} - \mathbf{y}') = \sum_{t=1}^{T} E(\mathbf{y}_t, \mathbf{y'}_t) \tag{2.9}$$

The training of simple RNN architectures can be achieved using gradient descent approaches such as back-propagation algorithm.

Appiah et al. [98] utilised long short-term memory (LSTM) networks for fault diagnosis in PV arrays. The authors generated simulated datasets comprising 2240 line-to-line (LL) fault instances, 1961 hotspot (HS) fault instances, and 1866 healthy operating conditions. A total of 6067 samples were created by varying operating parameters such as partial shading, line-to-line faults, solar irradiance, fault impedance, and temperature. Current, voltage, and power measurements obtained from the simulations were organized into separate datasets, namely I_dataset, V_dataset, and P_dataset. Each sample was subsequently transformed into a 400-dimensional feature vector. The proposed LSTM model automatically extracted features from the raw data, and the learned representations were supplied to a softmax regression classifier for fault classification. The model was trained and evaluated using both noiseless and noisy datasets containing normal (N), line-to-line (LL), and hotspot (HS) fault conditions. For the noiseless dataset, the reported classification accuracies for N, LL, and HS classes were 100%, 99.03%, and 100%, respectively. When noise was introduced, the average classification accuracies decreased slightly to 99.23%, 97.66%, and 98.78% for the corresponding classes. Despite the promising results, the study was based entirely on simulated fault scenarios and did not incorporate measurements from real PV systems. Furthermore, the rationale behind representing each sample as a 400-dimensional vector was not clearly explained, and the attributes constituting each sample were not explicitly specified. Furthermore, this approach is not suitable for comparison with experiments conducted on real datasets.

Liu and Yu [99] used an Elman Neural Network (ENN) for identifying faults in PV systems. The authors designed a laboratory setup for a small PV power station consisting of solar panels, a switching controller, a controller for MPPT charging and discharging, an inverter, battery storage, AC loads, DC loads, remote data interaction, and a signal detection unit. There is a power detection module for detecting irradiance, temperature, input current, output current, input voltage, and output voltage in each main power device. A halogen lamp was used to simulate sunlight irradiation in the range 300-1000 w/m^2, and irradiance and temperature sensors were used to collect environmental data. Two modules were connected to form one string, and four strings were connected in parallel, thereby producing a 2×4 PV array. The faults considered were No fault (i.e., normal state), partial shading, overall shading, and open circuit fault. The collected parameters were: output voltage and output current of PV array, output voltage of inverter, temperature, irradiance, DC load voltage. Total 1600 samples were used, out of which 70% were used for training, and the remaining 30% were used for evaluating the performance of the models. Neural Network toolbox in MATLAB was used to train the ENN structure by keeping 2000 as maximum training epoch with error precision equal to 0.0001 and 0.01 as initial learning step size. The authors compared performance of their model with Back Propagation (BP) Neural Network, and claimed that diagnosis error of their model was less than BP method. The main delimitation of the proposed method was that it used only simulated data, and that too for a very small PV system (e.g., 2×4 PV array). Moreover, the authors did not extend their work to large PV systems.

To accurately detect and classify defects in PV modules, particularly those caused by complex and dust-related issues, a hybrid DL model was introduced in [100]. This model integrates Bidirectional GRU and ResNet along with other components. By extracting information from I-V curves, it increases diagnostic accuracy and efficiently differentiates between similar fault types. Achieving an accuracy of 98.21% on noisy data and 99.94% on clean data, the model surpassed traditional methods such as ANN, RF, ResNet, and other models derived from ResNet, making it appropriate for FDD.

Zhang et al. [101] used LSTM-RNN for RUL prediction of Li-ion batteries. The framework that was developed included four primary elements: an LSTM-RNN, a method for optimising network parameters, a dropout technique to avoid overfitting in the neural network, and a Monte Carlo simulation for generating prediction uncertainties. The authors employed the RMSprop method for training the LSTM-RNN network and applied a dropout strategy to mitigate overfitting in the LSTM-RNN. The authors also used Monte Carlo (MC) simulation for generating uncertainties associated with RUL prediction. According to the authors, LSTM-RNN can be employed to estimate the RUL of a battery without the need for offline training data. Tests conducted on four cells at varying temperatures, such as 25 °C and 40 °C, and under different current rates showed that the proposed technique could predict RUL more accurately as compared to SVM and simple RNN. But the author did not mention percentage of accuracy of their proposed method.

2.3.3 Based on Deep Belief Network (DBN)

DBNs were one of the non-convolutional models that demonstrated successful training of deep architectures [88]. Prior to DBNs, deep models were considered to be very difficult to train and models like SVMs were popular compared to the deep models.

A DBN is a type of generative graphical model created by layering several levels of latent or hidden units on top of each other. Typically, the hidden units are binary, while the visible units can be either binary or real-valued. A DBN represents the joint distribution of the observed input layer along with the hidden layers, and the joint distribution can be given as

TH-4096_186151013

$$p(\mathbf{x}, \mathbf{h}^1, \ldots, \mathbf{h}^l) = p(\mathbf{h}^{l-1}, \mathbf{h}^l) \left(\prod_{k=1}^{l-2} p(\mathbf{h}^k \mid \mathbf{h}^{k+1}) \right) p(\mathbf{x} \mid \mathbf{h}^1) \tag{2.10}$$

where $p(\mathbf{h}^{l-1}, \mathbf{h}^l)$ is the prior and the joint distribution is a Restricted Boltzmann Machine (RBM) and the conditional distribution is given as

$$p(\mathbf{h}^{k-1} \mid \mathbf{h}^k) = \prod_j p(h_j^{k-1} \mid \mathbf{h}^k) \tag{2.11}$$

Hinton et. al [102] showed that DBN can be trained by stacking RBMs as depicted in Figure 2.4. The last layer of the DBN is added when discriminative task is required to be performed. This will be discussed in the later paragraphs.

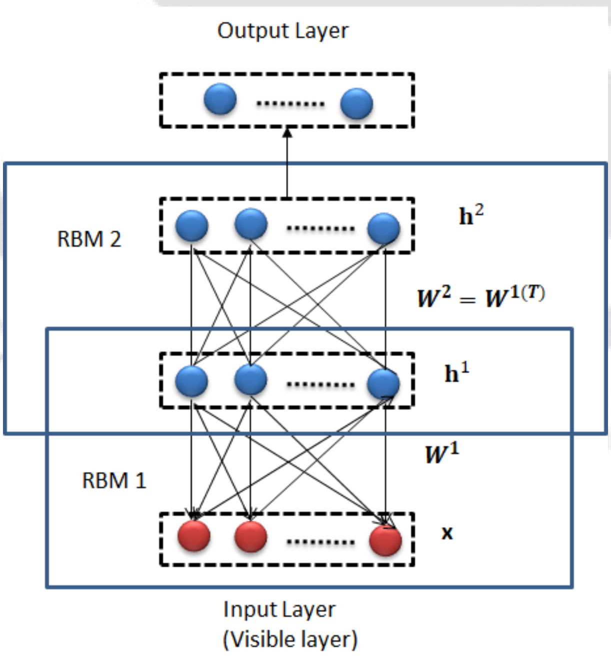


Figure 2.4: A basic architecture of deep belief network

Considering that DBN is formed by stacking RBMs, the joint distribution $p(\mathbf{h}^{l-1}, \mathbf{h}^l)$ can be given in terms of the energy function (or Boltzmann distribution).

$$p(\mathbf{h}^{l-1},\mathbf{h}^{l}) = \frac{1}{Z}\exp\left\{(-E(\mathbf{h}^{l-1},\mathbf{h}^{l}))\right\} \\ = \frac{1}{Z}\exp\left\{(\mathbf{h}^{l\mathbf{T}}\mathbf{W}^{3}\mathbf{h}^{l-1} + \mathbf{b}^{l-1\mathbf{T}}\mathbf{h}^{l-1} + \mathbf{b}^{l\mathbf{T}}\mathbf{h}^{l})\right\} \tag{2.12}$$

Here, the energy function E can be given by (2.13)

$$E(\mathbf{h}^{l-1},\mathbf{h}^{l}) = -\mathbf{h}^{l-1T}\mathbf{w}\mathbf{h}^{l} - \mathbf{b}^{T}\mathbf{h}^{l-1} - \mathbf{c}^{T}\mathbf{h}^{l} \tag{2.13}$$

The weight matrix $\mathbf{w}$, which has real values, is linked to the units $\mathbf{h}^{l-1}$ and $\mathbf{h}^{l}$. Additionally, the bias terms $\mathbf{b}$ and $\mathbf{c}$, also real valued, are connected to $\mathbf{h}^{l-1}$ and the hidden layer $\mathbf{h}^{l}$, respectively.

A DBN is trained through a greedy layer-wise unsupervised method [79], [81], where each RBM layer is individually trained. The input for each subsequent layer is the output from the preceding one. This unsupervised learning stage is known as pre-training. This phase turns the hidden layers as feature detectors. It can represent hidden patterns in the input observations. After pre-training, DBN can further be trained in a supervised manner to be used for discriminative task. This phase is referred to as fine tuning. For supervised learning the output layer is used and backpropagation approach can be applied to update the weights and biases.

To improve forecasting accuracy of short term PV power forecasting, an Enhanced Deep Belief Network (EDBN) framework has been proposed in [103] which employs a stacking ensemble strategy consisting of multiple base learners such as Extreme Learning Machine (ELM), Mondrian Forest (MF), Extremely Randomized Trees (ERT), and K-Nearest Neighbours (KNN) operate at the first level, while a DBN acts as the meta-learner. As a key contribution, the authors have integrated the Tree-Structured of Parzen Estimators (TPE) algorithm for hyperparameter optimization, which enables efficient exploration of the hyperparameter space, thereby improving generalization performance and reducing overfitting. The proposed EDBN is validated using datasets obtained from Desert Knowledge Alice Springs Center (DKASC) located in Australia with a sampling interval of five-minute resolution. For assessing the performance of the proposed model, various types of weather fluctuations were considered such as rainy, foggy, partially cloudy, and sunny. Comparative performance evaluation of the proposed model demonstrates that it outperforms all standalone baseline models such as ELM, MF, ERT, KNN, DBN and other DL models like MLP, GRU, LSTM, and BiLSTM by achieving an overall RMSE of 3.88 kW. However, the proposed method incurs higher computational complexity and increased training time due to the ensemble structure and hyperparameter optimization process. Furthermore, the study focuses on a single-site dataset, and its generalization across various climatic regions remains an open challenge.

Tao et al.[104] presented an intelligent FDD framework for PV arrays by integrating a DBN with a Genetic Algorithm (GA) to address a well-known limitation of conventional DBNs, namely their sensitivity to randomly initialised weights and biases, which often leads to local optima and unstable diagnostic results. In the proposed GA-DBN model, the GA simultaneously optimises the initial weights and biases of the DBN during the unsupervised pre-training stage. The reconstruction error of the Restricted Boltzmann Machine (RBM) is used as the fitness function, enabling effective evaluation of network performance. This optimisation improves convergence speed and enhances feature extraction capability before supervised fine-tuning using backpropagation. The proposed method is validated using a simulated PV array consisting of twelve modules under five operating conditions: open circuit, partial shading, grounded short circuit, abnormal ageing, and normal operation. Four electrical parameters—open-circuit voltage, short-circuit current, current at MPP, and voltage at MPP—are used as input features. Comparative experiments are conducted against traditional DBN, SVM, and GA-optimized backpropagation neural networks (GA-BP). Results demonstrate that the GA-DBN achieves superior fault diagnosis performance, with an overall accuracy of 95.73%, outperforming DBN (93.02%), SVM (86.77%), and GA-BP (85.42%). However, the GA-DBN requires higher training time due to the computational overhead of genetic optimization, limiting its suitability for real-time applications. Moreover, the authors used simulated data for validation, and did not evaluate performance under real-world operating conditions.

2.3.4 Based on Autoencoder (AE)

An autoencoder (AE), a type of neural network, is employed in unsupervised learning to primarily capture the encoding of input data within its hidden layers. It can be considered as an unsupervised nonlinear feature extraction approach. An AE generally includes an input layer,

TH-4096_186151013

several hidden layers designed to learn representations of this input, and an output layer that mirrors the input layer, as illustrated in Figure 2.5. An AE is classified as deep when it contains more than one hidden layer. In dimensionality reduction, the hidden layer contains fewer units than the input layer. Conversely, when mapping features to a higher-dimensional space, the hidden layer has more units than the input layer [86]. An AE can be trained using one of the backpropagation methods such as stochastic gradient descent. However, as the number of layers increases in the network, the problem of vanishing gradient appears wherein the error propagated to the first few layers becomes very small leading to inefficient training. There are several variants of the basic AE model, e.g., Sparse, Denoising, Contractive, and Variational AE [105].

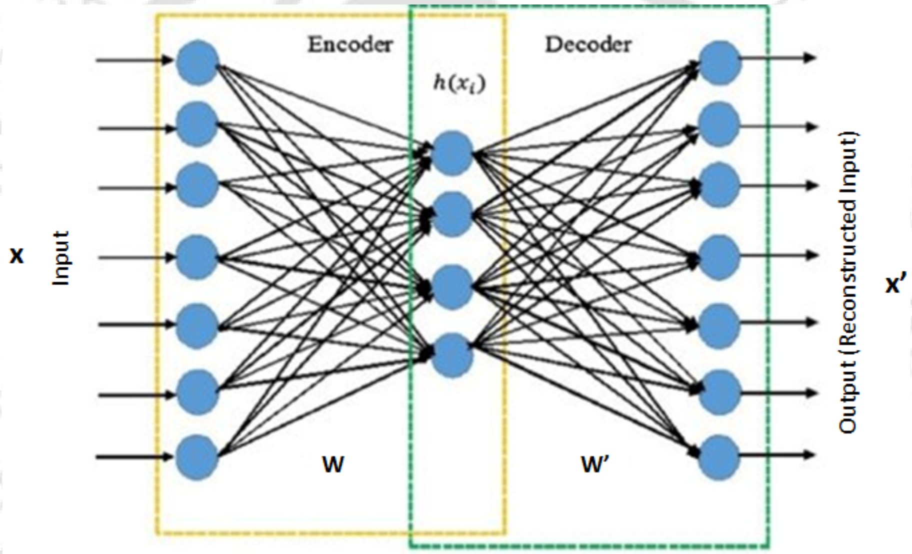


Figure 2.5: Basic architecture of autoencoder

The basic AE typically comprises two parts: the encoder and the decoder. In the simplest case, with one hidden layer, the encoder takes the input **x** and maps it to **h** as given in (2.14) , where the dimension of **h** can be higher or lower than **x** depending on the task's requirement.

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{2.14}$$

Here, **h** is commonly known as code, latent representation, or latent variables. The activation function, denoted as f, can be either a sigmoid function or a ReLU. The weight matrix is represented by **W**, while **b** stands for the bias vector.

The decoder component translates $\mathbf{h}$ into the reconstructed vector $\mathbf{x}'$ as given by (2.15).

$$\mathbf{x}' = g(\mathbf{W}'\mathbf{h} + \mathbf{b}') \tag{2.15}$$

The process of training an AE focuses on reducing the reconstruction error, which is described by the loss function as follows:

$$\mathscr{L}(\mathbf{x} - \mathbf{x}') = \| \mathbf{x} - \mathbf{x}' \|^2 \tag{2.16}$$

The weights and biases are updated using backpropagation.

Hadrawi et. al.[106] proposed an unsupervised deep autoencoder-based model to identify mismatch faults in PV modules using data obtained from a 20 MW grid-connected PV plant consisting of 520 PV modules. The authors derived eight electrical features from I–V and P–V curves, viz., short-circuit current, current at maximum power, voltage at maximum power, open-circuit voltage, maximum power, module efficiency, fill factor, and thermal voltage. The distribution of faults considered was: Normal (103 modules), Hotspot (157 modules), Short-circuit faults (49 modules), and bypass diode failures (211 modules). The authors compared the proposed method's performance with Deep Feedforward Neural Networks (DFNN) and Deep Belief Networks (DBN), and obtained an accuracy of 97.12% for the proposed method, 92.84% for DBN, and 92.41% for DFNN. However, the authors did not do any experiment with various variations of autoencoders. Moreover, no experiment was done to combine the autoencoder with other methods to check performance improvements.

Dairi et al. proposed an autoencoder model, known as a Variational AutoEncoder (VAE) for forecasting PV power [107]. The authors conducted an extensive evaluation of autoencoders, including Variational Autoencoder (VAE), Restricted Boltzmann Machine (RBM), and Stacked Autoencoder (SAE), as well as RNN-based models such as RNN, GRU, LSTM, ConvLSTM, and BiLSTM, for short-term and multi-step-ahead PV power forecasting. The forecasting framework is based on univariate time-series modelling, in which the input to all models is a sequence of historical PV power measurements, and the output is the predicted PV power for single- and multi-step-ahead horizons. The authors evaluated the performance of the models using datasets obtained from two PV plants connected to grid: a 9 MW PV Plant situated in

TH-4096_186151013

Algeria and a 243 kW PV array located in a parking lot in the US. The results demonstrated that VAE was the best performing model as compared to all other models for both datasets, achieving R^2 values of 0.995. However, external meteorological variables (i.e., weather data) are not used for PV power forecasting. Moreover, the authors did not consider multivariate forecasting.

Seghiour et al.[108] proposed a DL based FDD framework for grid-connected PV systems, aiming to improve robustness and accuracy under noisy operating conditions. The proposed methodology integrates physical modelling with data-driven technique. Initially, the PV system was modelled using the One Diode Model (ODM), whose unknown parameters were extracted using the Coyote Optimization Algorithm (COA). This optimization step enables realistic simulation of the PV plant behaviour under varying environmental conditions. Based on the validated model, a comprehensive dataset is generated for both healthy and faulty operating states, including voltage, current, solar irradiance, module temperature, and MPP power. Five AEs were utilized to derive features from parameters like irradiance, voltage, current, power at the maximum power point, and module temperature. These features served as inputs for an ANN, creating a DNN composed of five AEs and an ANN with a single hidden layer containing eight neurons. The authors experimentally validated their model by using monitored data from a 9.54 kW PV system located in Algeria. As claimed by the authors, the accuracy of the proposed method is 99.9% under noisy data, and 100% under noiseless data. However, the authors focused only on module short circuit faults and string disconnections, whereas other PV faults such as partial shading, inverter faults, degradation, etc. were not considered.

2.4 Review of Related Patents

A few relevant patents pertaining to the current research are described below.

- US20190384257A1, "Automatic health indicator learning using reinforcement learning for predictive maintenance", by Chi Zhang, Chetan Gupta, Ahmed Khairy Farahat, Kosta Ristovski, Dipanjan Ghosh, Dec.19, 2019: Zhang et al. invented a method that employed reinforcement learning to predict maintenance needs using health indicators. The method integrates operational condition data, sensor data, and failure event data to construct a model that assesses health indicators. These indicators, derived from state values, learned

TH-4096_186151013

policies, and rewards, serve as measures of equipment performance. External sensor readings and operational data for a piece of equipment are used to validate the model, which then provides a recommendation based on its analysis.

- US9939485B1, “Prognostics and health management of photovoltaic systems”, by Jay Johnson, Daniel Riley, April 10, 2018: Johnsohn and Riley used ANN to create a PV system’s prognostics and health management system. By utilising data related to at least one parameter of the PV system’s present operation and generating an expected operating metric as an output related to at least one parameter, the ANN model makes it easier to analyse an output of a photovoltaic (PV) system. Training data collected before the PV system’s present operation was used to configure the ANN. Additionally, the system has an operational component that is set up to receive data about an output of the PV system while it is currently working and to produce an actual operating metric related to the output of the PV system. Additionally, a comparison component is set up to ascertain whether the value of the expected operating metric and the actual operating metric differ. If there is a difference, an evaluation component is set up to ascertain whether the difference is greater than a threshold, which might be a specified amount, percentage, or mathematical formula.

- US9541598B2, “Smart junction box for photovoltaic systems”, by Mei Zhang and Jianhong Kang, Jan 10, 2017: A smart junction box for photovoltaic systems was created by Mei Zhang and Jianhong Kang. It gives electrical measurements of strings of PV cells to identify early cell degradation, bypass diode failure, and arcing, reports the results to a central location, and/or allows for the automatic disconnection of a specific string of PV cells.

- US10389300B2, “Photovoltaic system having fault diagnosis apparatus, and fault diagnosis method for photovoltaic system” by Sukwhan Ko, Youngseok Jung, Gi-Hwan Kang, Youngchul Ju, Hyemi Hwang, Junghun So, Hee-Eun Song : Ko et al. presented invention related to a fault diagnosis for a photovoltaic system. The fault diagnosis apparatus consists of a temperature sensor unit to measure both internal temperature of the junction box and a surface temperature of the solar module; a voltage sensor to measure voltage in the junction box, an operation unit to determine whether fault occurs inside the solar modules or inside the junction box, and the inverter by comparing measured values of the temperature sensor unit with a measured value of the voltage sensor. In addition, it is possible to detect solar module faults that are difficult to be checked by the naked eye.

- US9761913B2, "High efficiency high voltage battery pack for onsite power generation systems", by Sandeep Narla, Sept. 12, 2017 : For an on-site power generation system, Sandeep Narla created a high-voltage battery pack with battery modules set up to deliver a minimum voltage of 170V. Between the battery modules and the positive and negative output terminals of the battery pack, high-speed switches and a rapid current detection circuit are connected in series. When a fault situation is identified, a control circuit linked to the current detection circuit activates to open one or more switches, thereby disconnecting the battery modules from the battery pack's output terminals.

2.5 Summary

In this chapter, the existing literature on data-driven predictive maintenance methods in solar PV systems with special emphasis on diagnostics of PV array faults, and prognostics of Balance of System failures, including inverter faults, has been reviewed. The AI-based techniques for PdM of solar PV systems were discussed first, and then a literature review of existing methods using DL techniques is presented. It should be noted that the applications of DL methods in PdM of solar PV systems are in their nascent stages. Based on the survey of existing literature, it has been observed that DL methods are yet to be explored for PdM, especially in solar PV systems for diagnostics and prognostics of PV array faults, health estimation, RUL of BOS components, and prediction of inverter faults. Table 2.2 lists some of the significant research done on fault diagnosis of PV arrays during the period 2010-2014 using traditional AI-based techniques. Table 2.3 lists the reviewed papers on fault diagnosis of PV arrays using traditional AI-based techniques during the period from 2015 to 2020. Table 2.4 summarises the research done on fault diagnosis of PV arrays using deep learning during the period 2010-2020. Table 2.5 lists some of the reviewed papers on prognostics of BOS components using data driven methods during the period 2010-2020.

According to the literature survey, the application of DL techniques in predictive maintenance for renewable energy systems, especially solar PV systems, remains largely untapped. While

TH-4096_186151013

numerous studies have concentrated on fault detection and diagnosis, there is a scarcity of published work on predicting faults early by using contextual information.

To summarize, the following research gaps (RGs) have been identified:

- RG1: Most of the publications are related to model-based techniques, especially electrical methods. Data-driven methods have not been extensively explored by many researchers in PV applications.

- RG2: Very few researchers have used DL approaches in the domain of renewable energy systems for PdM. Based on the survey of existing literature, it has been observed that DL methods are yet to be explored for PdM, especially in solar PV systems.

- RG3: There is a lack of approaches that take into account the context information. Though there are few researchers who have used DL for PdM in solar PV systems recently, there is very little work on contextual approaches for PdM in solar PV systems.

- RG4: Though artifial intelligence based condition monitoring and fault diagnosis of solar PV systems has been extensively studied and implemented by numerous researchers for quite a long time, the main idea of predictive maintenance is lacking in many of the research works. There are many researchers who identified the faults after occurrences of the faults. In predictive maintenance, the algorithm should be able to identify the potential fault(s) that is/are likely to happen well ahead of its/their occurence(s). This particular philosophy of predictive maintenance is missing in many of the publications.

- RG5: In the literature, most of the publications are confined to PV array faults. There is very little research on inverter faults in PV systems.

- RG6: There is a lack of data-driven approaches that consider a system-level perspective rather than a component-level view, where interactions among components are ignored.

Table 2.2: Review of papers on fault diagnosis of PV arrays using traditional artificial intelligence based techniques during the period 2010-2014

.

Author,Ref., Year	Type of fault	Approach	Inputs and outputs	Performance
Riley and Johnsohn, [63], 2012	PV faults in general	ANN	Inputs: ambient temperature(T_a), wind speed(w), solar irradiance(G). Output: predicted AC power	Prone to false positives
Syafaruddin et al., [66], 2011	Short circuit fault	ANN	Inputs: cell temperature(T), solar irradiance(G), maximum power point current(I_{MPP}), and maximum power point voltage(V_{MPP}). Output: estimated voltages of PV modules.	Can identify the location of short-circuit fault in a single string.
Rezgui et al., [70], 2014	Short circuit fault	SVM	Inputs: simulated PV data. Outputs: fault class	Classification accuracy obtained was 68 to 75.8%.
Ducange et al., [74], 2011	Short circuit in cells, shading, broken cells on a module	FL	Inputs : temperature(T) and solar irradiance(G). Output: DC power(estimated).	90% or more faults could be detected by the system
Bonsignore et al., [75], 2014	Partial shading, earth faults, and short circuited diode	TSK-FRBS	Inputs: module temperature(T_m) and solar irradiance(G). Outputs: I_{MPP},V_{MPP}, I_{SC}, V_{OC}, D_1, and D_2, where D_1, and D_2 are derived parameters as explained in [75]	Normal and faulty operations for partial shading, diode short circuit, and earth faults can be categorized.
Zhao et al., [21], 2012	Line-to-line faults, open circuit faults, and partial shading	DT	Inputs: normalized current(I_{NORM}), fillfactor(FF), voltage of PV array(V_{PV}) , open circuit voltage(V_{OC}), power of PV array(P_{PV}). Output: fault class.	Fault detection accuracy:93.56% to 99.98%, fault classification accuracy: 85.43% to 99.8%.

Table 2.3: Review of papers on fault diagnosis of PV arrays using traditional artificial intelligence based techniques during the period 2015-2020

Author,Ref., Year	Type of fault	Approach	Input and Output	Performance
Akram and Lotfifard, [67], 2015	Line-to-line and open circuit faults	ANN	Inputs: Maximum power point current(I_{MPP}), maximum power point voltage(V_{MPP}), power at maximum power point(P_{MPP}), temperature(T), solar irradiance(G). Output: decision of classification.	Classification accuracy of 98.53%.
Chine et al., [64], 2016	Short circuit, diode faults, connection faults	ANN	Inputs:open circuit voltage ratio, maximum power point voltage ratio, and maximum power point current ratio between real and simulation based data. Ouput: fault class.	90.3% for MLP based ANN, 68.4% for RBF based ANN.
Mekki et al., [65], 2016	Partial shading	ANN	Inputs: cell temperature(T) and irradiance(G). Outputs: current and voltage of PV module	Mean absolute errors are 0.52 A and 11.47 V for current and voltage, repectively.
Yi and Etemadi, [23], 2016	Line-to-line	SVM	Input features: Feature 1 is fill factor, $feature2 = \frac{I_{NORM} \times T_{cell}}{V_{NORM} \times G}$, features 3 and 4 are changing rates of normalized output current and voltage of the PV array. Ouput: fault class.	50% or more L-L faults can be detected with 20% mismatch
Zhao et al., [46], 2019	PV string fault	GMM	Inputs: Aggregate features and spectrum features derived from SCADA data, where the aggregate features are mean, median, standard deviation, and maximum of the deviation sequence. Output: fault class	Best classification accuracy is 93.0%
Zhao et al., [73], 2020	PV string faults	CF	Inputs:current values from SCADA system. Output: decision of fault detection.	The overall detection accuracy varies from 96.3% to 97.1%.

TH-4096_186151013

Table 2.4: Review of papers on fault diagnosis of PV arrays using deep learning techniques during the period 2010-2020

Author,Ref., Year	Type of fault	Approach	Input data	Performance
Huuhtanen and Jung, [89], 2018	Anomaly in PV panel	CNN	Inputs: Synthetic and real power measurement data.	Overall mean squared error obtained was 0.000421 and 0.003561 for synthetic and real datasets, respectively.
Aziz et al., [90], 2020	Line to line fault (LL), partial shading (PS), faults in PS, arc fault, open-circuit fault (OC).	CNN	Solar irradiance(G), temperature(T), I_{PV}, V_{OC}, I_{SC}, I_{MPP}, V_{MPP}, P_{MPP} from the PV array, and maximum current, voltage, power from boost converter.	Fine-tuned pre-trained CNN achieved a detection accuracy of 73.53%. .
Tan et al., [91], 2019	Dust accumulation (i.e., soiling)	DnCNN	Images of PV panels	Accuracy of ResNet-34, ResNet-50, ResNet-101, VGG-16, and AlexNet models were 85.9%, 87.3%, 38.6%, 30.45%, and 31.59%, respectively.
Appiah et al., [98], 2019	Hotspot(HS) and line-to-line(LL)	LSTM	Simulated PV data.	Classication accuracies were 99.23%, 98.78%, and 97.66% for No fault, HS, and LL, respectively..
Li et al., [93], 2019	PV module defect	CNN	Defect patterns of images.	Classification accuracies of 97.8%, 99.5%, 98.5%, 98.0%, 97.8%, and 98.1% for shading, delamination of encapsulant, no defects, corrosion of gridline, snail trails, and discoloration, respectively.

Table 2.5: Review of papers on prognostics of battery banks and inverters using data driven methods during the period 2010-2020

Author,Ref., Year	Type of failure prediction	Approach	Input and Output	Performance
Charkhgard and Farrokhi, [69], 2010	Estimation of the state of charge (SOC)	ANN	Inputs: Battery voltage at previous sampling time, the estimated SOC at current sampling time, and the battery terminal current at current sampling time. Output: battery terminal voltage at the current sampling time.	The root mean square errors between actual SOC and estimated SOC are 2% and 3% for offline data and online data, respectively.
Patil et al., [71], 2015	Remaining useful life	SVM	Inputs: Voltage of energy signal and fluctuation index of voltage signal. Output: Remaining useful life (RUL) of battery.	Mean squared error for regression is 0.2420 with prediction accuracy greater than 99%.
Nuhic et al., [72], 2013	State of health (SOH) estimation, remaining useful life prediction.	SVM	Inputs: capacity of cell at the beginning, throughput of the cell in Ah, minimum and maximum values of SoC (state of charge), last known capacity degradation, temperatures during cycling, time between initial and target capacity measurements. Output: Cell capacity.	Root mean square error of less than 0.0008 for conventional datasets.
Zhang et al., [101], 2018	Remaining useful life Prediction	LSTM	Inputs: State of charge(SOC), ambient temperature, charging profiles of cells, battery capacity. Ouput: Remaining useful life.	Accurate prediction of remaining useful life(RUL).
Betti et al., [68], 2019	Prediction of inverter faults	ANN and SOM	Inputs: DC current, DC voltage, DC power, AC current, AC voltage, AC power, module temperature, ambient temperature, and global horizontal irradiance, global tilt irradiance. Ouput: Anomaly prediction and fault class prediction.	Generic fault prediction upto 7 days ahead of occurrence, specific fault class prediction from few hours to 7 days ahead of occurrence .

TH-4096_186151013

Chapter 3

System Description and Data Analysis

This chapter presents an overview of the photovoltaic (PV) systems and the datasets used. During this study, two datasets are used, the first dataset is from an operational PV plant, and the second is generated from a PV system installed in the Laboratory of Innovation and Technology in Embedded Systems and Energy (LIT), Universidade Tecnológica Federal do Paraná-UTFPR, Brazil. The data obtained from this installation is made available by Lazzaretti et. al.[109]. The latter part of the chapter focuses on exploratory data analysis (EDA), highlighting key patterns, trends, and faults relevant to predictive maintenance.

We begin with a description of the operational PV plant in Section 3.1. We then describe the lab-installed PV system in Section 3.2. Section 3.3 describes the various datasets. Analysis of datasets are presented in sections 3.4 and 3.5.

3.1 Operational PV System Description

The PV system being analysed is an operational solar PV plant with a capacity of 10 MW located in the state of Gujarat, India. The Plant is grid-connected and consists of 36480 PV panels

TH-4096_186151013

Figure 3.1: Simplified schematic diagram of a grid-connected PV system

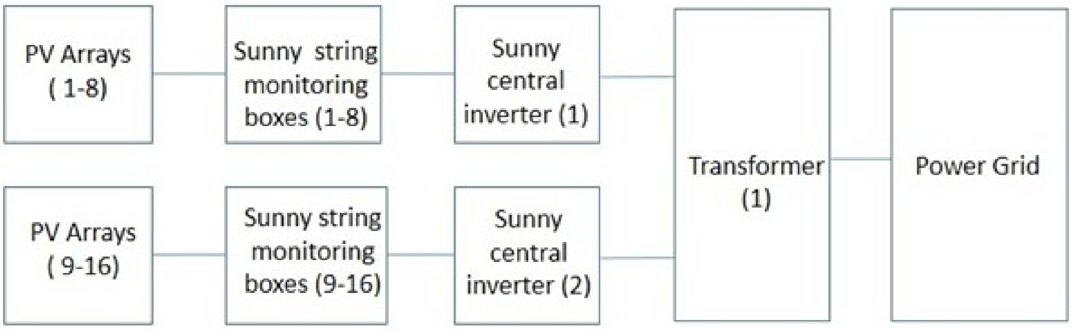


Figure 3.2: A detailed schematic diagram of a section of the PV plant

and 16 inverters. The number of PV modules in a panel is 57 (19 modules, each consisting of 5 strings).

3.1.1 PV System details

A simplified configuration of the solar PV plant is illustrated in Figure 3.1, and a detailed schematic of a section of the layout is depicted in Figure 3.2. The PV plant comprises 16 inverters, each interfaced with eight PV arrays through Sunny String Monitoring (SSM) units. Each SSM unit is associated with a single PV array. A PV array consists of 5 PV strings connected in parallel, where each PV string contains 19 PV modules connected in series. Consequently, one SSM monitors an array consisting of five strings, with each string comprising 19 series-connected modules. Thus, 19 PV modules are connected in series, which indicates one string, and 5 such strings are connected to one SSM, which forms one PV Array. Eight such SSMs are connected to one inverter of rating 630kW. Two such Inverters (630kW) are connected to one transformer rated at 1300 kVA through 1Cx300 sq mm copper cable.

3.1.2 Module Specification

The solar cells used in the modules are polycrystalline 156 mm×156 mm solar cells of rating 280 Watt manufactured by Suntech with module efficiency of 14.4%. There are 72 solar cells in a module arranged in 6×12 configuration. Each module has a dimension of 1956 mm×992mm×50mm or 77.0 inches×39.1 inches×2.0 inches. Under standard test conditions, which include an irradiance of 1000W/m^2, a module temperature of 25°C, and an air mass ratio (AM) of 1.5, each module operates with a maximum voltage (V_{MP}) of 32.2 V and a maximum current (I_{MP}) of 7.95A. Additionally, the open circuit voltage (V_{OC}) is 44.8V, while the short circuit current (I_{SC}) reaches 8.33A. Table 3.1 shows the technical specifications of Suntech 280 Watt polycrystalline solar modules under standard operating conditions.

Table 3.1: Specifications of the PV Module

Name of Parameter	**Range / Value**
Maximum power at STC	280 W
Short circuit current (I_{SC})	8.33 A
Open circuit voltage (V_{OC})	44.8 V
Maximum operating voltage (V_{MP})	35.2 V
Maximum operating current (I_{MP})	7.95 A
Efficiency of module	14.4%
Maximum system voltage	1000 V DC (IEC) / 600 V DC (UL)
Operating temperature range of modules	-40°C to $+85$°C
Maximum series fuse rating	20 A
Power tolerance	0 / +5%

3.1.3 Inverter Specification

The solar PV plant uses SMA-630HE sunny central inverters for DC (direct current) to AC (alternating current). There are 16 inverters in the system with each inverter having a rating of 630kW. The SMA-630HE inverter does not have a transformer in it. The rating of each

TH-4096_186151013

inverter is 630kW with 315 V AC output voltage and 1155A AC output current operating with a frequency of 50Hz. Two SMA-630HE inverters can be connected to a transformer of rating 1300 KVA, 315V AC voltage, and 11000A AC current. 5 strings (each string consisting of 19 PV modules connected in series) are connected to one Sunny String Monitoring box (SSM) which forms one Array. 8 nos. of such SSMs are connected to 1no. of Inverter of rating 630kW. 2 nos. of such Inverters of rating 630 kW are connected to 1no. of transformer of rating 1300kVA through 1Cx300sqmm Copper cable. Table 3.2 shows the technical specification of SMA-630HE sunny central inverter.

Table 3.2: Technical Specifications of the Power Conversion System

Parameter	**Range / Value**
Maximum DC voltage	1000 V DC
Maximum DC current	1350 A DC
Maximum power point (MPP) voltage range	500–820 V DC
Number of DC inputs	9
Nominal AC power	630 kVA
Nominal AC voltage	315 V AC
Nominal AC current	1155 A AC
Maximum AC current	1283 A AC
Grid frequency	50 Hz / 60 Hz
Maximum total harmonic factor	$< 3\%$
Maximum efficiency	98.7%
Operating temperature range	$-20°$C to $50°$C

3.2 Lab-Installed PV System Description

Figure 3.3 shows the PV system set-up in LIT. The system comprises 16 PV modules arranged into two strings, with each string containing eight modules connected in series. The DC power generated by these strings is supplied to two grid-connected inverters, which convert the DC energy into single-phase AC power for grid integration. The overall installed peak capacity of the system is approximately 5 kW. The installation also houses an electrical panel consisting of parts utilised in a photovoltaic installation, viz., Main Circuit Breaker, String Circuit Breakers, Fuses, and Transient Voltage Suppressors. In addition, certain additional parts are installed to artificially induce faults. Specifically, twelve sockets are available for connection to create various short-circuit scenarios or, alternatively, to be used with auxiliary circuit breakers to introduce specific resistances. These circuit breakers can be employed independently to produce open-circuit faults. Furthermore, opaque objects can be used to block solar radiation to generate shadowing.

In addition to the PV array and inverters, the installation includes an electrical distribution panel equipped with several essential PV system components, such as fuses, transient voltage suppressors forming the circuit protection block, string circuit breakers, and a main circuit breaker. To facilitate fault analysis and experimental investigations, the setup is also provided with twelve configurable sockets. These sockets can be interconnected to emulate various short-circuit fault scenarios or combined with auxiliary circuit breakers to introduce external resistive elements of desired values into the circuit.

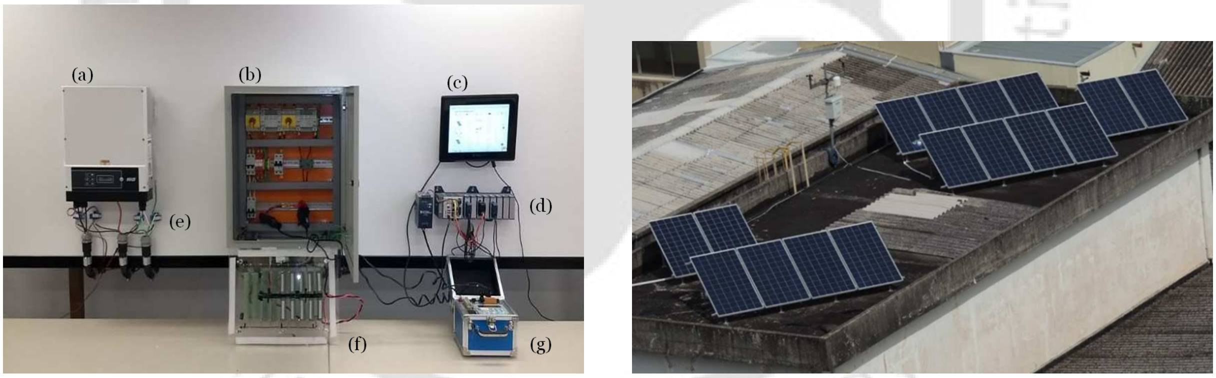


Figure 3.3: (adopted from Lazzaretti et. al.[25]) PV system installed at LIT (a) Power Inverter, (b) Electrical Panel with Protection Devices and Signal Conditioning, (c) Local Display, (d) Compact RIO and Signal Acquisition Modules, (e) alternating current (AC) and direct current (DC) Current Transducers, (f) Degradation Resistors, and (g) Power Quality Analyzer.

3.3 Data Description

The first dataset (Dataset 1) from the operational solar plant comprises eleven months of SCADA records spanning the period from April 2019 to February 2020. The measurements were acquired from the Inverter Control Rooms (ICRs) at a sampling interval of 15 minutes. The dataset contains around 30 inverter-related parameters, including daily energy generation (kWh), AC power, DC power, AC voltage, DC voltage, AC current, and DC current, among other operational variables. In addition to the electrical measurements, the dataset includes meteorological information recorded during plant operation. The available weather-related attributes comprise horizontal irradiance, plane-of-array (tilt) irradiance, module temperature, wind speed, and ambient temperature. Furthermore, a separate fault log is provided, documenting the inverter fault events that occurred during the data collection period. Table 3.3 shows all the parameters of a particular inverter (Inv 9) that are monitored. Table 3.4 presents SCADA Parameters and description of ICR1-ICR4.

Based on SCADA data of inverter control rooms (ICRs) received and the fault logs, 25 abnormal events associated with inverters and other equipment were observed. Table 3.5 shows the causes of faults and their frequency of occurrences.

The second dataset (Dataset 2) from the PV installation in LIT was collected for 16 days between 07:30 to 17:00, and three different faults were introduced at various times: degradation, open circuit and short circuit. Partial shadowing occurred naturally as the sunlight was obstructed by adjacent buildings at various times in the morning and afternoon. The distribution of faults and normal data samples is shown in Table 3.6 A data acquisition system was used to collect the following variables: Idc, Vdc, Iac, Vac and module temperature, along with the weather variables such as ambient temperature, irradiance, relative humidity, wind speed, and wind direction, with a sample frequency of 1 Hz.

3.4 Analysis of Dataset 1

In this section, we conduct a comprehensive analysis to enhance our understanding of the data for all 16 inverters at the power plant. Initially, we have plotted the average AC power across all months for each of the 16 inverters. It is noted that the plant's production was elevated in January, February, April, and May, while it significantly decreased in July, August, and September. Figures 3.4 and 3.5 illustrate the AC power for four inverters as examples. This trend is consistent across all 16 inverters, with minor deviations in a few cases.

TH-4096_186151013

Table 3.3: Description of Inverter Parameters(Inv 9)

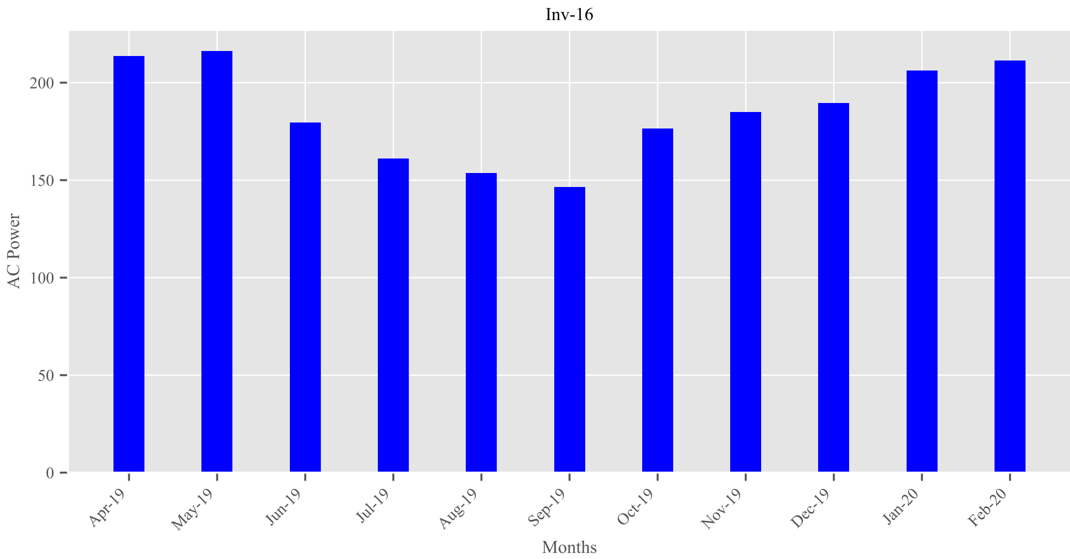
Sl. No.	Parameter	Description
1	Date	Day of the month
2	Time	Time of day
3	Energy_today_1	Generation of the inverter in kWh for a particular day
4	AC_Frequency_1	AC frequency
5	DC_Current_1	DC current
6	AC_Real_Power	AC power
7	Insulation_Resistance_1	Insulation resistance
8	AC_Voltage_1	AC voltage
9	External_Temp_1	External temperature
10	Energy_Total_1	Total energy since commissioning (kWh)
11	AC_Current_1	AC current
12	Mode_1	Mode of operation
13	Power_Factor_1	Power factor
14	Reactive_Power_1	Reactive power
15	Active_Power_1	Active power
16	DC_Voltage	DC voltage
17	Internal_Temp_1	Internal temperature
18	DC_Power_1	DC power
19	Vry_1	Voltage between R and Y phases
20	Vyb_1	Voltage between Y and B phases
21	Vbr_1	Voltage between B and R phases
22	Current_R_1	Current in R phase
23	Current_Y_1	Current in Y phase
24	Current_B_1	Current in B phase

TH-4096_186151013

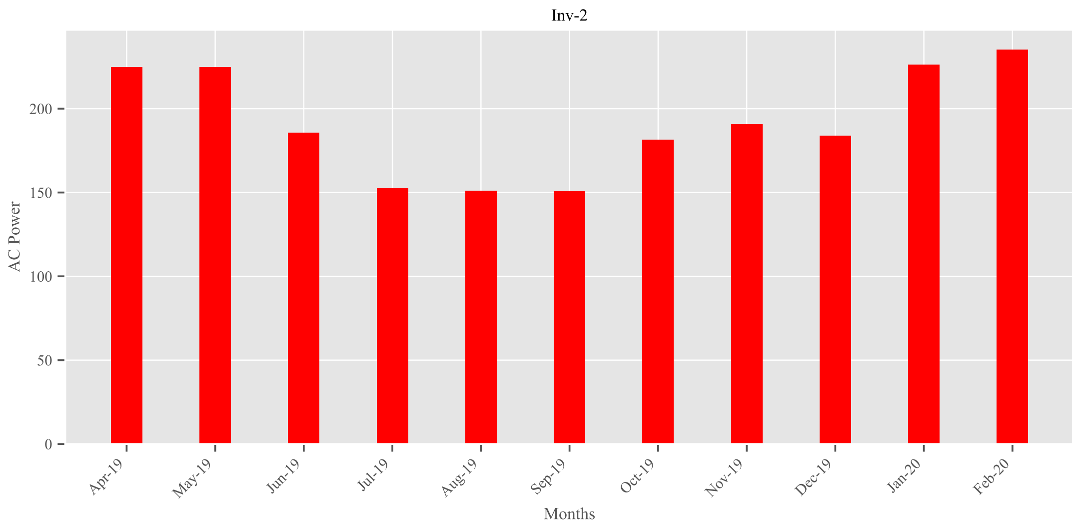
Table 3.4: Description of ICR1-ICR4 (Sunny Central inverter) Parameters

Sl. No.	Parameter (ICR)	Description
1	TimeStamp	Time of day
2	E-heute	Energy fed into the grid by the Sunny Central on the current day
3	Fac	Power line frequency
4	Ipv	DC input current
5	Iac	Output AC current
6	Ppv	DC input power of the Sunny Central
7	Pac	Output AC power
8	Upv	DC input voltage of the Sunny Central
9	R-Insul	Insulation resistance
10	E-Total	Total energy fed into the grid during the entire operating period
11	Op. Mode	Mode of operation
12	PF	Power factor
13	Qac	Reactive power
14	Sac	Apparent power in kVA
15	h-on	Total operating hours
16	h-total	Total operating hours in feed-in operation
17	Error	Sunny Central error status
18	VacL1-L2	Grid output voltage between L1 and L2
19	VacL2-L3	Grid output voltage between L2 and L3
20	VacL3-L1	Grid output voltage between L3 and L1
21	Reg.SMUs	Number of registered Sunny String Monitors (SSMs)
22	Komm.FahlerSMU	Disturbed communication with Sunny String Monitor number
23	PB-DigiInput1	Output current of SSM 1 at digital input 1 of the inverter
24	PB-DigiInput2	Output current of SSM 2 at digital input 2 of the inverter
25	PB-DigiInput3	Output current of SSM 3 at digital input 3 of the inverter
26	PB-DigiInput4	Output current of SSM 4 at digital input 4 of the inverter
27	PB-DigiInput5	Output current of SSM 5 at digital input 5 of the inverter
28	PB-DigiInput6	Output current of SSM 6 at digital input 6 of the inverter
29	PB-DigiInput7	Output current of SSM 7 at digital input 7 of the inverter
30	PB-DigiInput8	Output current of SSM 8 at digital input 8 of the inverter

TH-4096_186151013

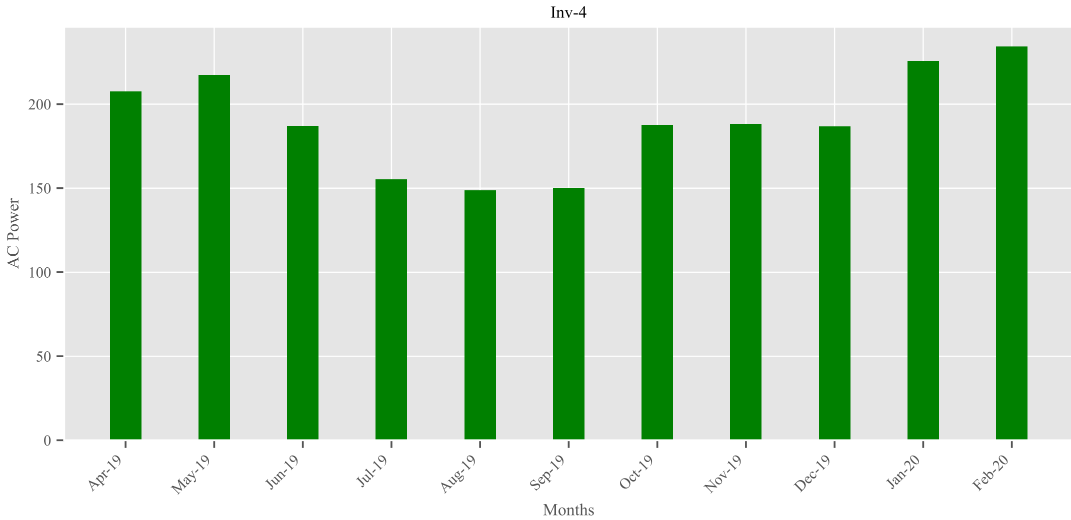


(a) Inverter 16 average AC Power

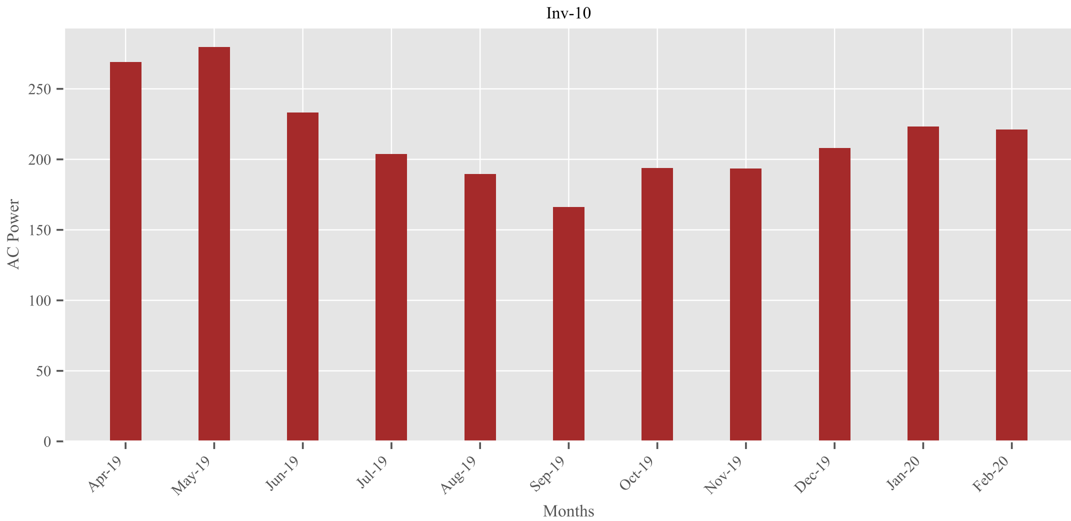


(b) Inverter 2 average AC Power

Figure 3.4: Average AC Power for inv 16 and inv 2 for all months with no fault record



(a) Inverter 4 average AC Power



(b) Inverter 10 average AC Power

Figure 3.5: Average AC Power for inv 4 and inv 10 for all months having fault records

Table 3.5: Frequency Distribution of Inverter Faults

Reason of Fault	**Frequency**
Inverter fault due to earth fault (GFI failure)	32%
Inverter fault due to IGBT failure	24%
Inverter not started	12%
Inverter fault due to inverter control supply failure	8%
Inverter fault due to straight joint failure	4%
Tripping in master relay in inverter control room due to motor failure	4%
Inverter fault due to overtemperature	4%
Fault in AC subsystem due to failure in AC protector	4%
Fault in DC cable	4%
Inverter failure due to C-tech failure	4%

Table 3.6: Distribution of Data Samples Across Operating Conditions

Type	**No. of Samples**
Normal data samples	309,253
Degradation	10,371
Short-circuit	5,999
Open-circuit	6,024
Partial shadowing	184,311
Total	**515,958**

As can be seen from figure 3.4, both inverters 16 and 2 produce almost identical patterns for all the months because these two inverters do not have any fault records. Slight variation for some months may be due to cloud cover, or due to partial shading because of the presence of trees. As shown in Figure 3.5 (b), the pattern of inverter 10 is slightly different as compared to the other inverters towards the end. This may be due to the presence of faulty conditions in the inverter 10.

Apart from irradiance and ambient temperature, other reasons for low output in July may be the faults in the plant. During July, the highest number of faults was reported (7 faults for

TH-4096_186151013

varying duration), whereas in August and September, 3 and 4 faults were reported, respectively (as shown in Figure 3.6).

Figure 3.7 shows inverter-wise average the AC power. Although faults occurred in Inv-10, it has the highest average AC real power, slightly above 210 kW. Inv-9 and Inv-11 also perform well, close to Inv-10. Inv-12, Inv-13, and Inv-14 exhibit the lowest average AC power values, slightly below 180 kW. To examine seasonal trends, the average AC power output was calculated using two-hourly intervals and plotted for all months, as shown in Figure 3.8. The month-wise profiles indicate that months belonging to the same season exhibit similar power-generation patterns. For instance, April and May display comparable average AC power characteristics, while July, August, and September form another group with closely matching profiles. The figure also reveals that AC power output exhibits consistent behaviour not only within the same season but also during corresponding periods of the day. Such observations suggest that both seasonal and temporal factors have a significant influence on PV power generation, highlighting the importance of incorporating contextual information into predictive modelling frameworks.

Figure 3.9 shows the month-wise average AC power. It shows the monthly variation in average AC real power output (in kW) from a photovoltaic (PV) system over the period. April and May record the highest average AC power, exceeding 250 kW. February and April to June also show strong performance, indicating clear-sky conditions and optimal solar irradiance during the pre-monsoon and summer period. July through September exhibit a sharp decline, with the lowest outputs in August and September (around 130 kW). This drop aligns with the monsoon, suggesting reduced irradiance due to cloud cover and rainfall. Also, as mentioned earlier, the higher frequency of faults during these months has contributed to the reduced AC power output. There is a noticeable recovery in October, likely due to clear skies and moderate irradiance. December and January show moderate output (around 190–215 kW), possibly due to shorter daylight hours and lower solar angles, although the skies may remain relatively clear. The system's output is highly seasonal, peaking in pre-monsoon/summer and dipping during the monsoon. Thus, seasonal modelling or forecasting of power generation would benefit from explicitly incorporating weather and climatic factors.

The heatmap in Figure 3.10 shows how the average AC power output from a PV system varies across different months and hours of the day. Based on the heatmap, the contextual profile of the PV system can be described by analyzing the variation of average AC power output with respect

TH-4096_186151013

to different hours of the day and months of the year. The figure shows that PV power generation follows a strong diurnal and seasonal pattern. The AC power generation begins around 7 AM, rises quickly, and peaks between 11 AM and 2 PM across most months. Power generation drops after 3 PM and tapers off by around 6 PM, which aligns with solar irradiance patterns. The monthly trends are the same as discussed before, with February to May (yellow) around mid-day indicating highest power outputs. The intensity of mid-day power output is a strong seasonal indicator. It is highest in April/May and lowest during the monsoon (August/September). Morning and evening contributions are consistently low throughout the year, reinforcing that most power is generated during mid-day hours. Therefore, Operational scheduling and maintenance should avoid mid-day hours, especially in high-output months (April–May), to prevent revenue loss. The contextual profile highlights the importance of considering both temporal (e.g., time of day) and seasonal variations for accurate PV performance assessment, fault detection, and predictive modelling, as normal operating behaviour varies significantly across different time periods and seasons. The power generation pattern reflects the combined effects of solar irradiance, ambient temperature, seasonal variations, varying weather conditions, and operational factors. During peak sunlight hours, typically around midday, the PV system produces higher AC power due to increased solar energy availability, while output decreases during early morning and evening hours because of reduced irradiance. Similarly, monthly variations occur due to seasonal differences in sun position, daylight duration, cloud cover, and environmental conditions. Understanding this contextual profile is essential for analyzing PV system performance, identifying abnormal operating conditions, improving fault detection models, and developing accurate prediction strategies by considering natural variations in power generation patterns.

Finally, the fault data was analysed to identify underlying patterns. We plotted the faults that occurred in all the inverters to see the pattern. Here, Inv-7 is shown as a representative. As a first step, AC power output during normal operating conditions was plotted to serve as a baseline for comparison with faulty states (Figure 3.11a). As shown in Figure 3.11a, normal operation is characterised by a smooth and gradual rise and fall in power throughout the day. Figure 3.11b shows the power pattern before and after the day the fault occurred. On 10-08-19, fault occurred during 07:45 – 11:30 in Inv-7, and the reason is “Inverter not started”.

As shown in Figure 3.12a, the fault that occurred on 04-04-201 during 13:45-16:00 is due to “Failure in Over-temperature”. Figure 3.12b shows the plot of the fault that occurred due to

TH-4096_186151013

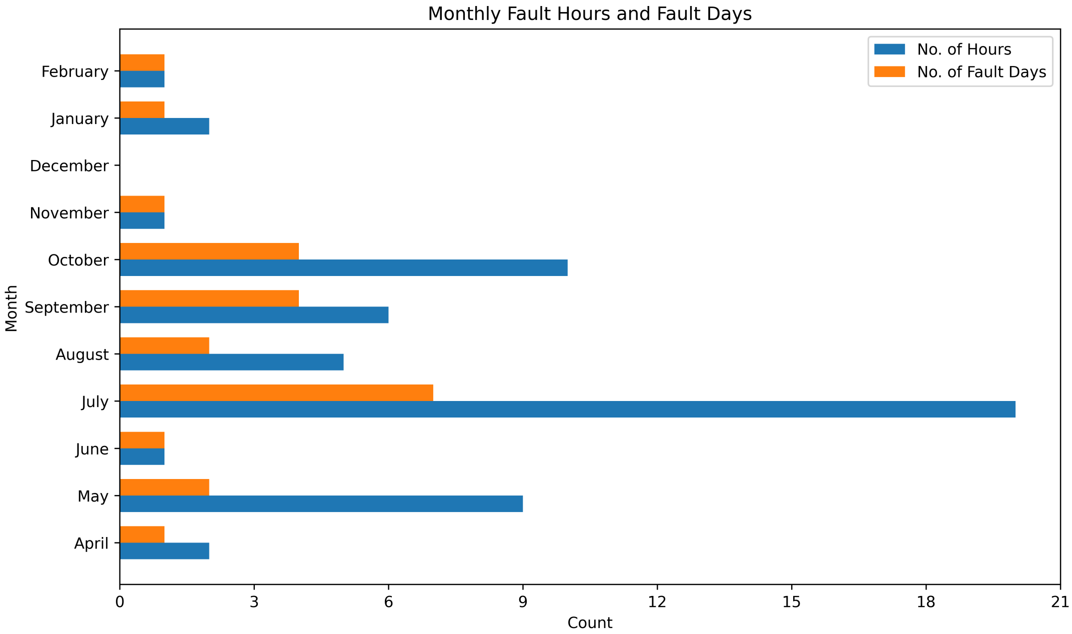


Figure 3.6: Frequency of faults

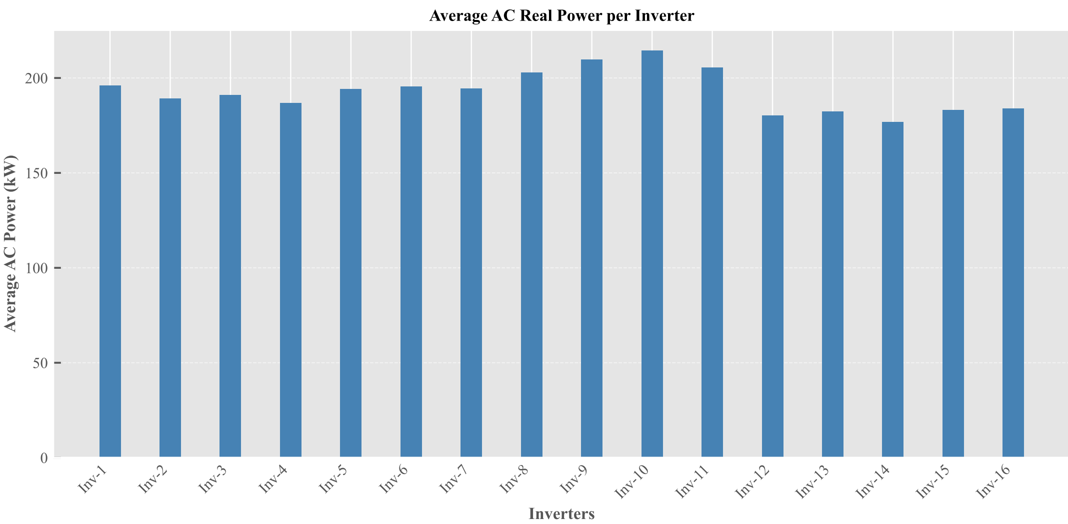
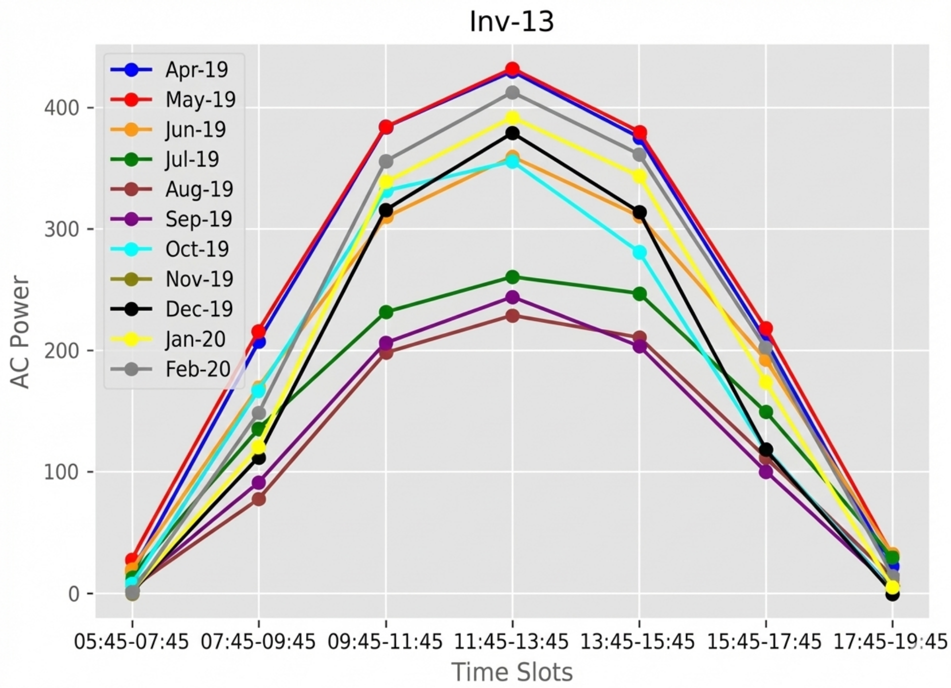
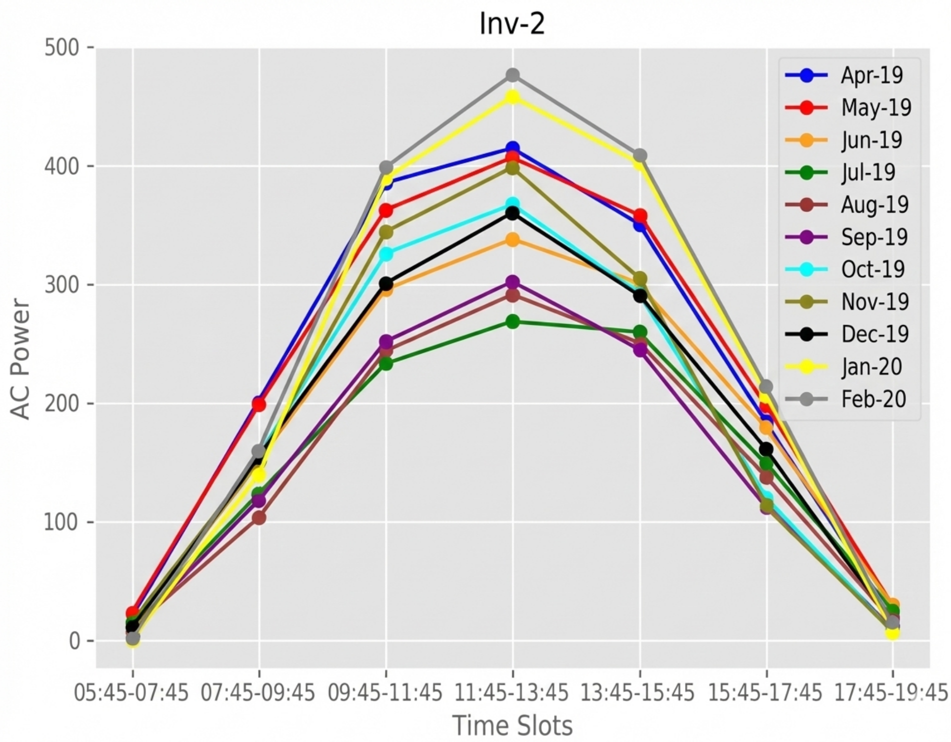


Figure 3.7: Average AC power per inverter



(a) Inverter 2 average AC Power



(b) Inverter 13 average AC Power

Figure 3.8: Average AC Power for inv 2 and inv 13 for all months

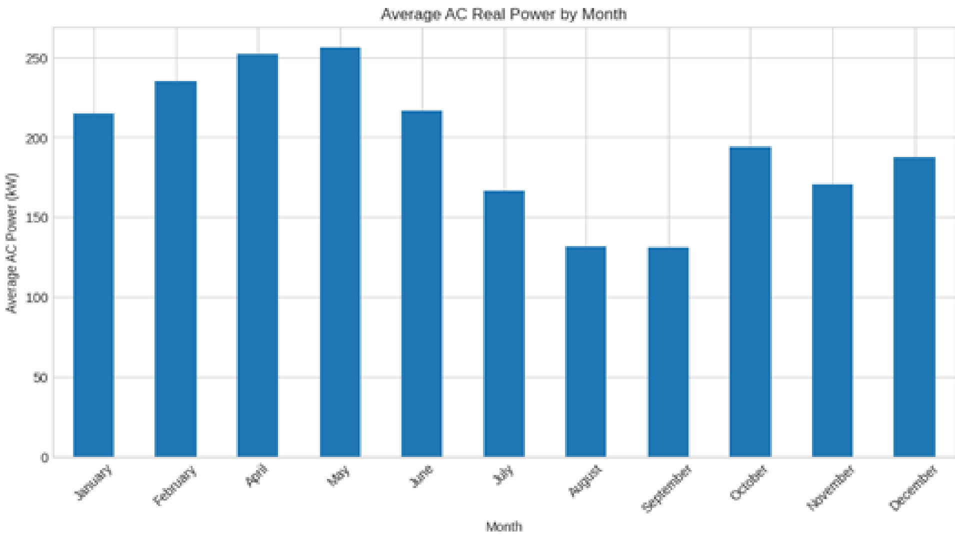


Figure 3.9: Month-wise average AC power

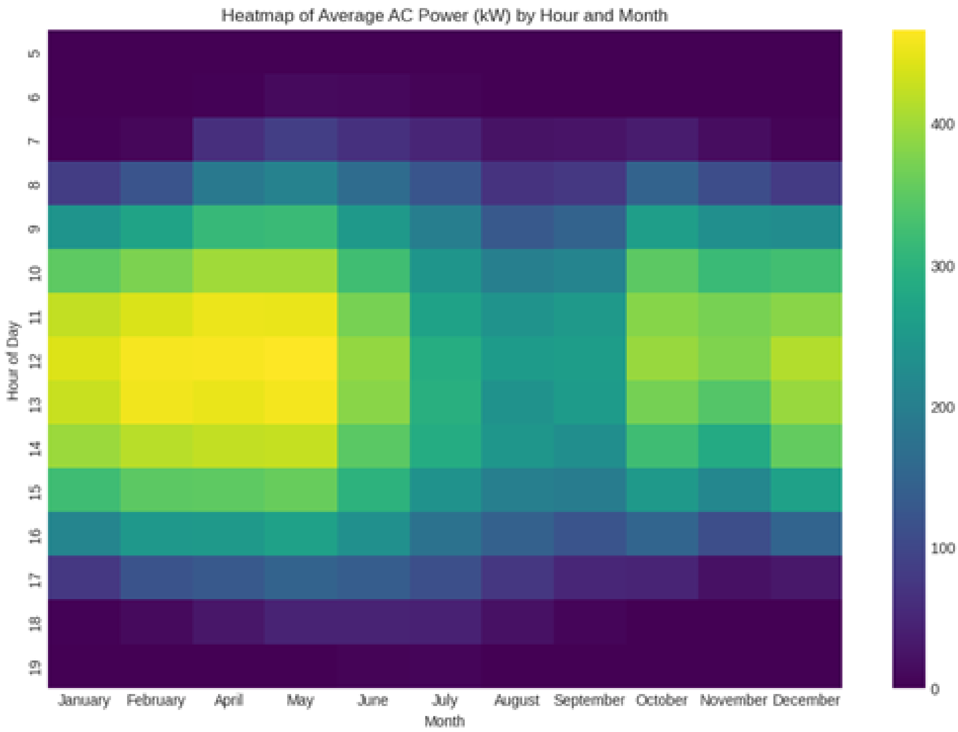
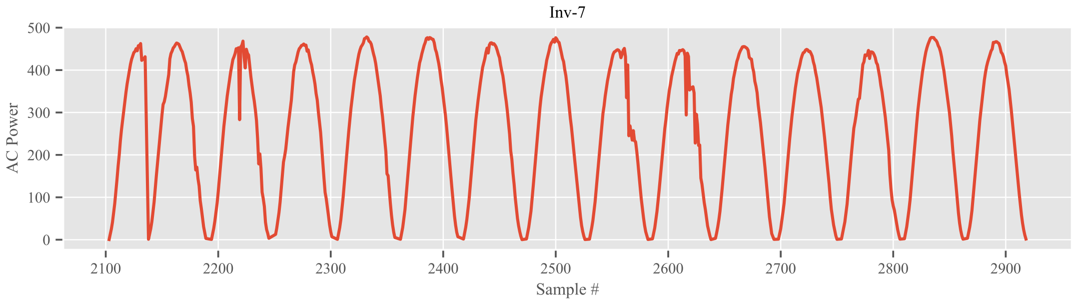
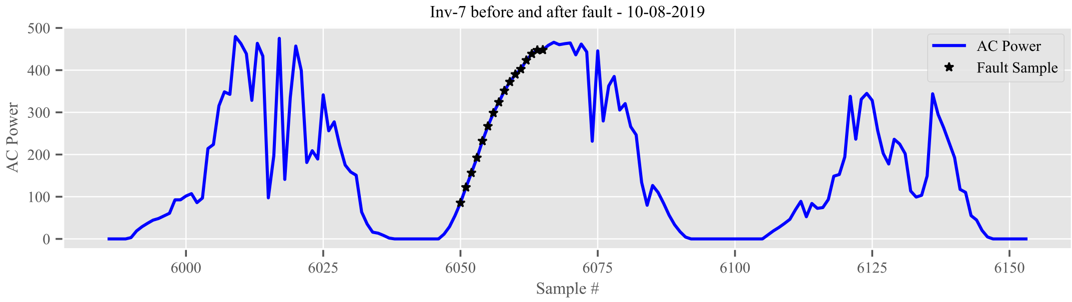


Figure 3.10: Heatmap of average AC power by hour and month depicting the contextual profile



(a) Normal operation of inverter 7



(b) Faulty operation of inverter 7

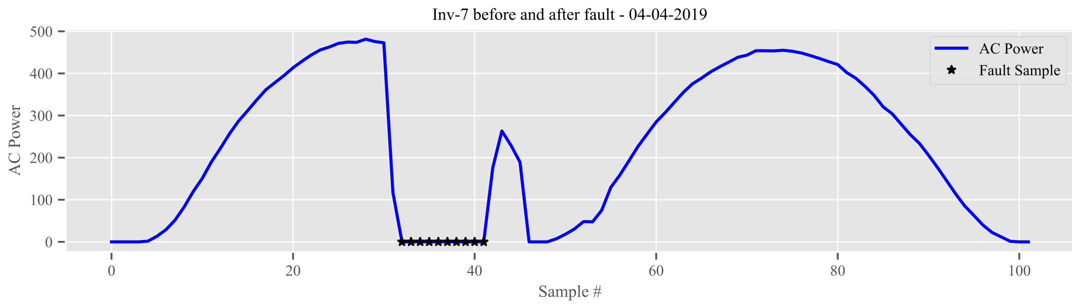
Figure 3.11: Normal operation and faulty operation of inverter 7

"Inverter control failure". Finally, the faults are compared across the inverters to see if they have the same pattern. It could not be done for all the faults, as many have been reported once in the available fault data.

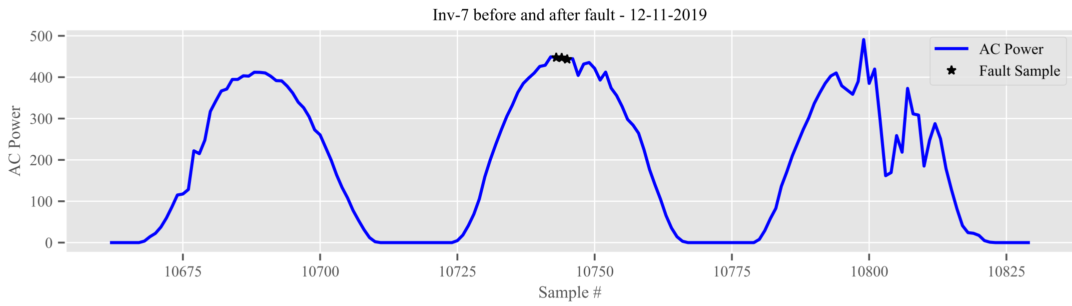
Figures 3.12c and 3.12d depict the Earth fault of inverter 15 on two different days, viz., 23 Oct, 2019 and 7 Feb, 2020. On Oct 23, 2019, the Earth fault occurred during 15:30-16:00 hrs (for a duration of 30 minutes), and on Feb 7, 2020 it occurred during 12:00 -12:45 hrs (for a duration of 45 minutes) As can be seen, on both the days there are dip in AC power with moderate fluctuations.

The Earth faults of inverters 3 and 6 are depicted in Figure 3.13 On 7 July, 2019 and 8 July, 2019, inverter 3 was affected due to Earth fault during the period 6:15-19:15 hrs (for a duration of 13 hours on both days). As shown in Figures 3.13a and 3.13b , on both days, almost the same patterns are exhibited with lot of AC power fluctuations during the day. On June 26, 2019, inverter 6 was having Earth fault from 14:30 hrs to 15:30 hrs (1 hour), and the fault pattern is depicted in Figure 3.13. It can be concluded that Earth fault causes moderate to high AC power fluctuations, which depends on the total duration of Earth fault.

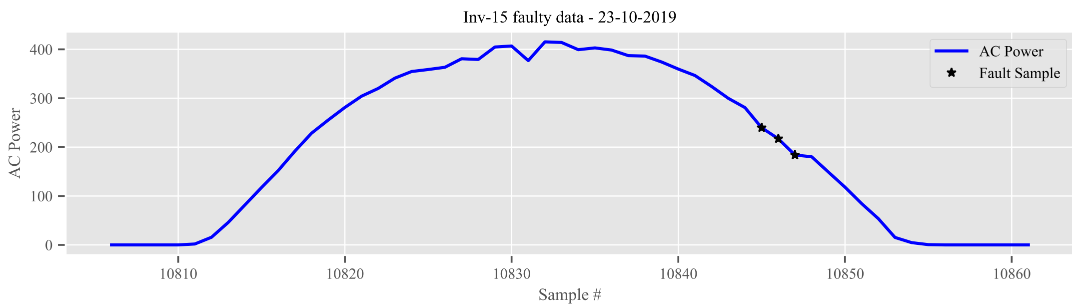
TH-4096_186151013



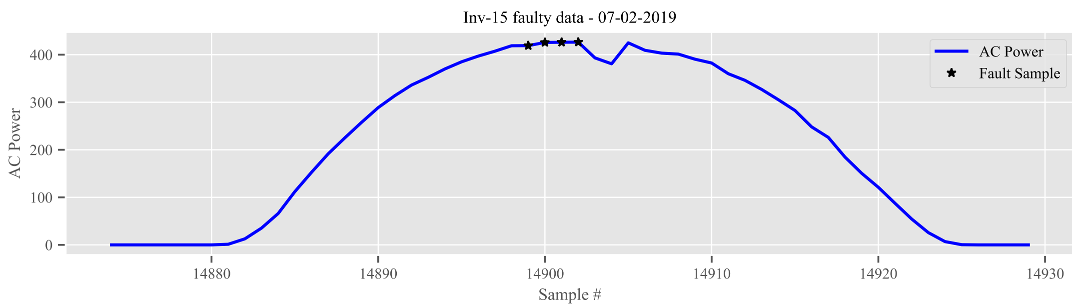
(a) Over-temperature in inverter 7



(b) Control failure in inverter 7



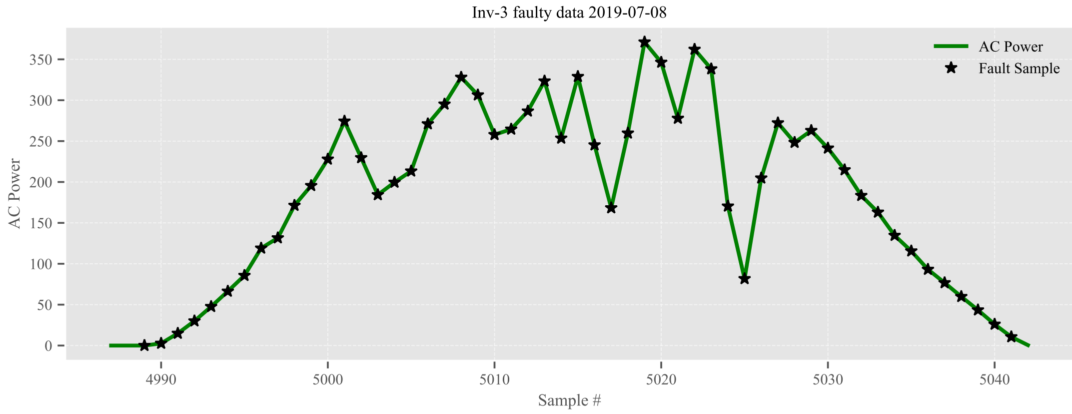
(c) Earth fault in Inverter 15-Day1



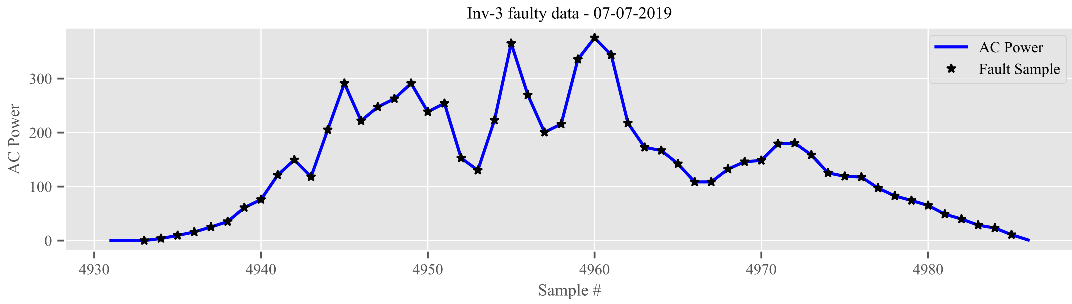
(d) Earth fault in Inverter 15-Day 2

Figure 3.12: Various faults observed in Inverter 7 and 15

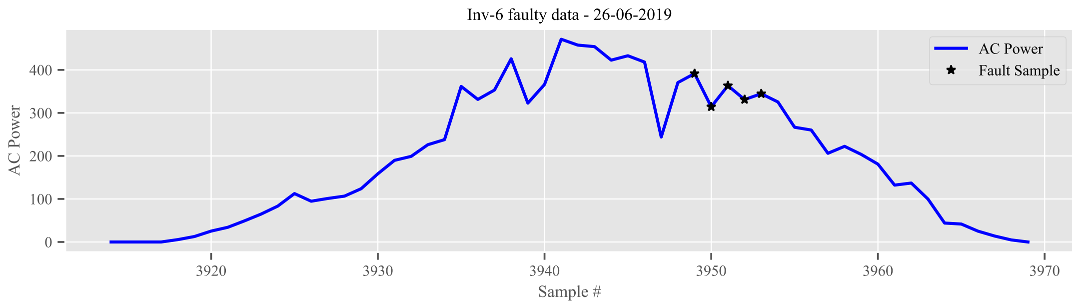
TH-4096_186151013



(a) Earth Fault in Inverter 3- Day 1



(b) Earth Fault in Inverter 3- Day 2

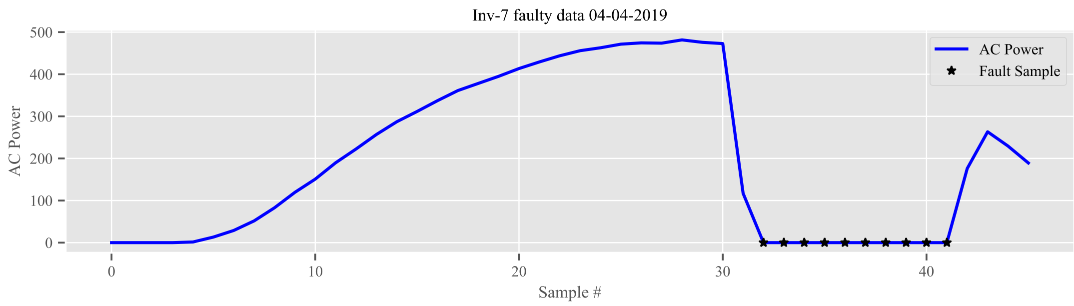


(c) Earth Fault in Inverter 6

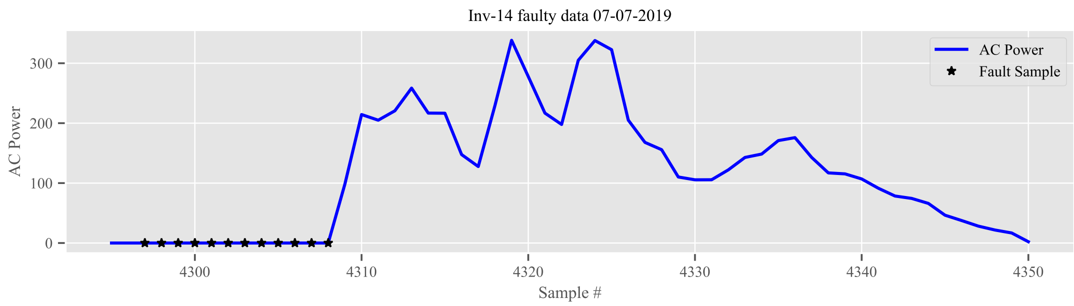
Figure 3.13: Failure due to Earth fault in Inverter 3 and Inverter 6

Figures 3.14 - 3.16, show the patterns of “Over temperature”, “Inverter not started”, and “Inverter control supply fail” faults, respectively. During “Over temperature” fault, Initial normal or partial operation followed by flat zero-output periods and unstable output fluctuations. "Inverter Not Started" faults may not always result in zero output from the beginning of the day. Instead, they often show a normal ramp-up in power generation, followed by a disruption or failure to maintain steady-state power, and early or abrupt drops in power output compared to expected patterns. “Inverter Control Supply Fail” fault, a condition where the inverter’s internal control circuitry loses power, potentially causing momentary or partial disruptions in energy conversion. These faults are brief and localised, not leading to complete power loss. They occurred at or near peak production periods, where inverter circuits are under maximum load. Such faults resulted in minor power dips or irregularities rather than prolonged outages.

Figure 3.17 shows the fault “IGBT driver board failure” for four different inverters. In inverters 7 and 8 the power drops abruptly to zero, whereas the other two inverters do not show such behaviour. It can be observed that there is no uniform pattern present across the inverters.

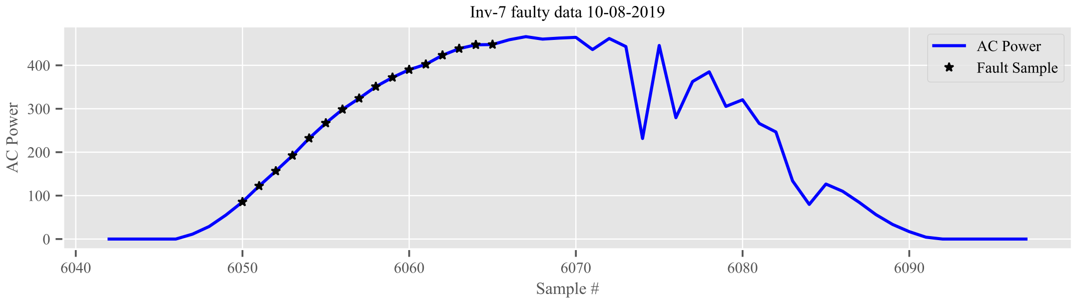


(a) Failure in overtemperature for inverter 7

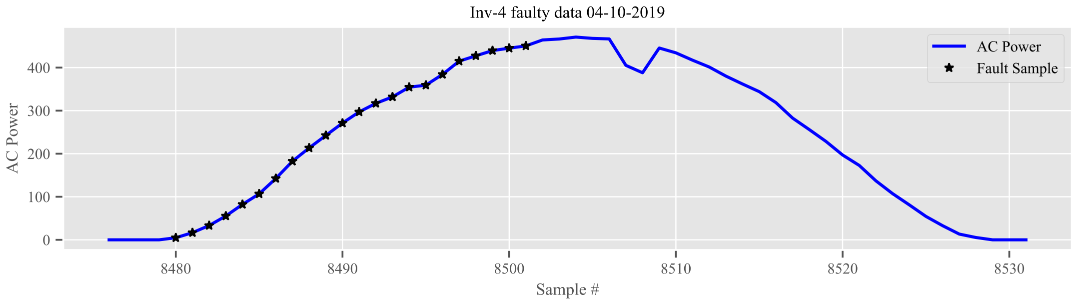


(b) Failure in overtemperature for inverter 14

Figure 3.14: Failure in overtemperature for inverter 7 and inverter 14

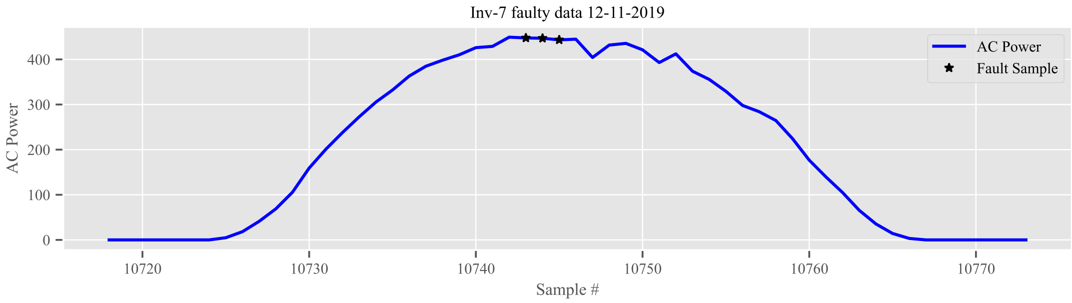


(a) Inverter not started fault for inverter 7

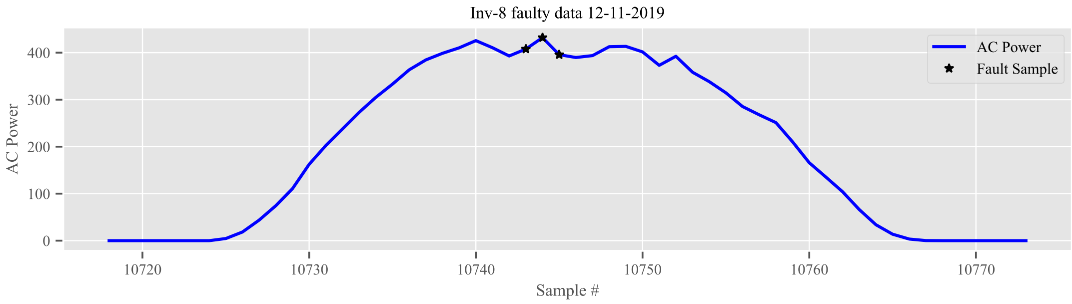


(b) Inverter not started fault for inverter 4

Figure 3.15: “Inverter not started” fault for inverters 7 and 4

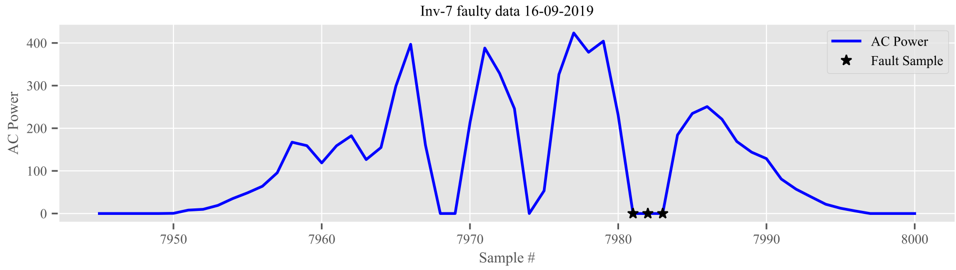


(a) Inverter control supply failure for inverter 7

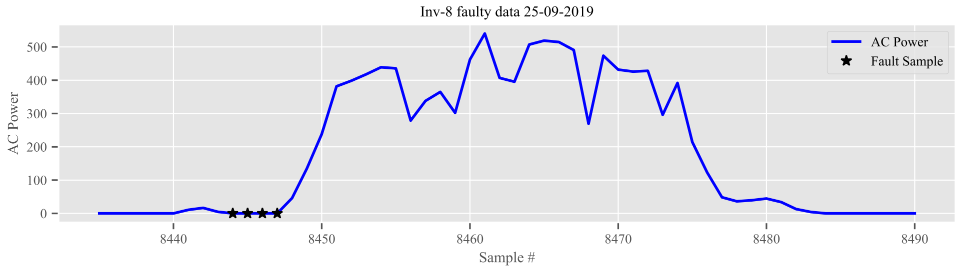


(b) Inverter control supply failure for inverter 8

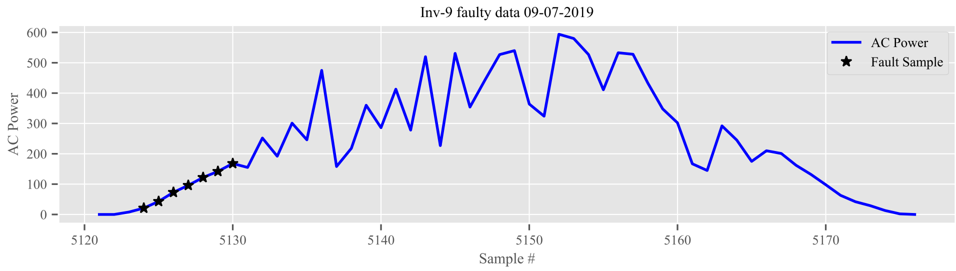
Figure 3.16: Fault due to “inverter control supply fail" for inverters 7 and 8



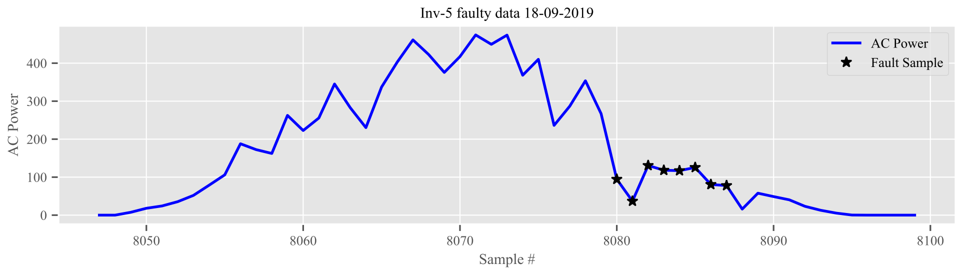
(a) IGBT fault of inverter 7



(b) IGBT fault of inverter 8



(c) IGBT fault of inverter 9



(d) IGBT fault of inverter 5

Figure 3.17: Failure of IGBT driver board.

3.5 Analysis of Dataset 2

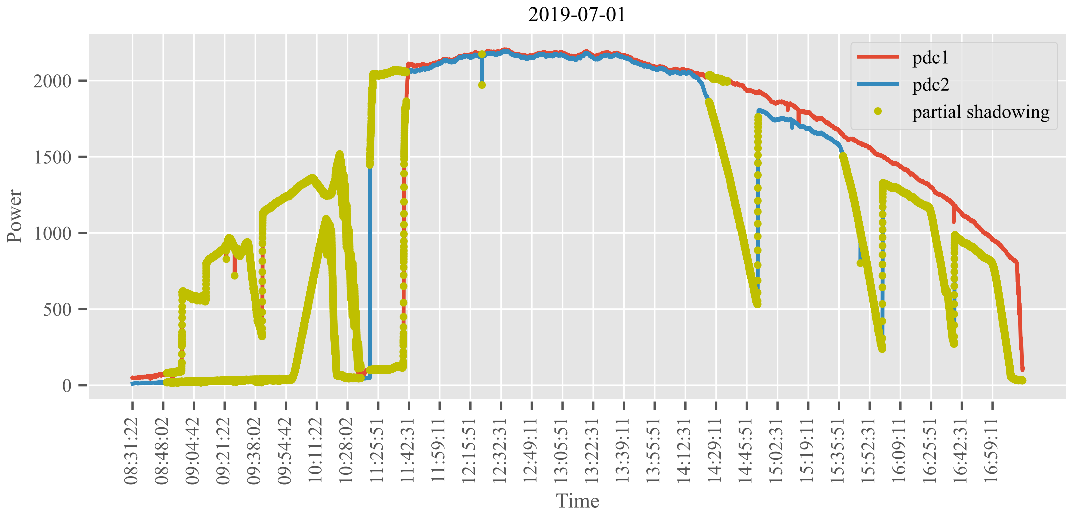
We plotted the faults to observe their patterns. Figures 3.18 (a)-(c) show the patterns of various faults. It is to be noted that String 1 has mainly partial shadowing faults, whereas String 2 has all four kinds of faults, which are partial shadowing, degradation, short-circuit, and open circuit. Also, for clarity each of the figures 3.18 (a) and (b) show only one type fault even though other faults are present. Similarly, Figure 3.18 (c) highlights only short circuit and open circuit faults.

From the figures it can be observed that during partial shadowing the power fluctuates and in case of open-circuit, the power drops to zero. The degradation is reflected by the power dropping to zero and then remaining constant for a specific time. In short circuit fault also, the power is seen to be constant for a specific time. These patterns are the same across all the days when faults occurred primarily due to the reason that the faults are introduced artificially to the installation. On the contrary, for the real dataset from Azure, we did not observe uniform patterns across the same type of fault.

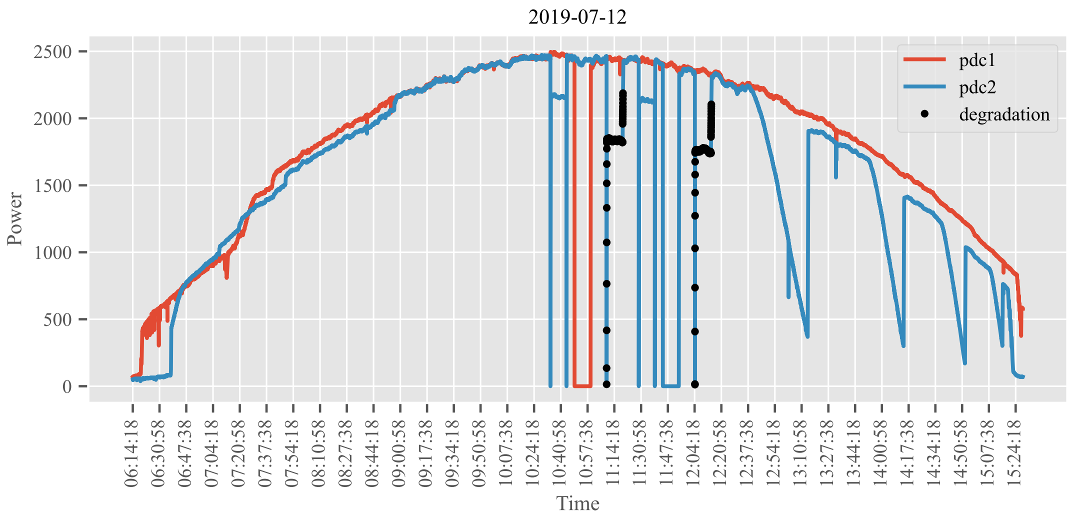
3.6 Summary

This chapter presented a detailed analysis of PV SCADA data obtained from a real grid-connected PV Plant with a capacity of 10 MW, along with special emphasis on the fault patterns of various inverter faults. First, it presented the analysis of the data, focusing on the AC Power of all 16 inverters. The average AC power is analysed across months and also across seasons to see its variation. It is observed that for the months of July, August, and September, the average AC power generated by the Plant is low as compared to other months due to environmental factors as well as due to faults. It was also observed that some inverters produced less power compared to other inverters in the same month due to occurrences of faults. Finally, to see the pattern of the faults, the possible patterns of all dominant faults were analysed. Some of the faults did give some specific patterns, as illustrated through various diagrams in the chapter. In Dataset 2, fault patterns were analysed across two strings, with String 1 showing partial shadowing and String 2 exhibiting all four fault types. Artificially introduced faults showed consistent signatures, e.g., power drop to zero in open-circuit and fluctuations in shadowing, whereas real-world data

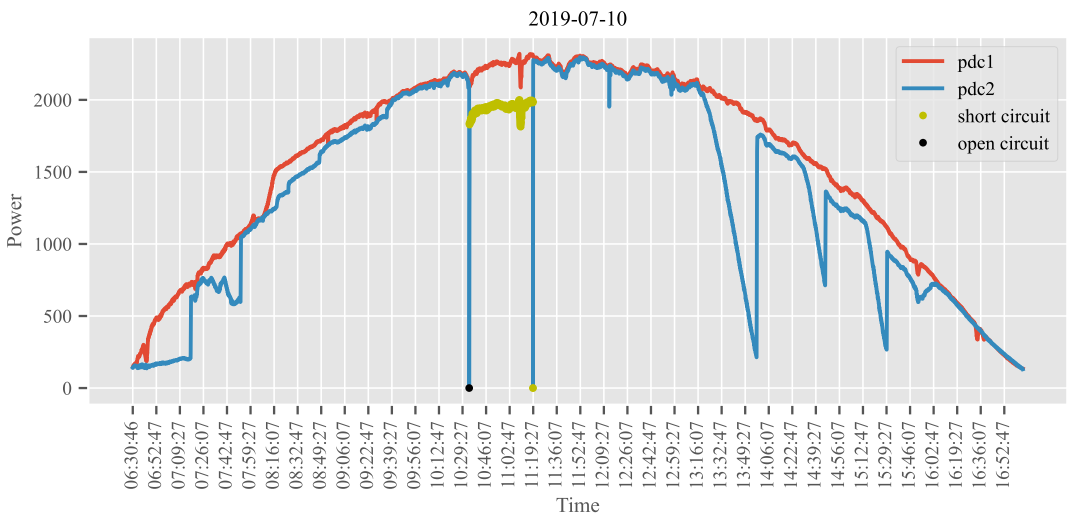
TH-4096_186151013



(a) Partial Shadowing



(b) Degradation



(c) Short circuit and Open circuit

Figure 3.18: Various fault patterns observed in Dataset 2

lacked such uniformity.

This chapter also includes profiles of context-sensitive features. For example, Figures 3.8–3.10 present the variation of average AC power with respect to time of day and month/season. Figure 3.8 shows the average AC power across different time slots for different months, Figure 3.9 presents the month-wise average AC power, and Figure 3.10 provides a heatmap of average AC power by hour and month for depicting the contextual profile. These figures demonstrate that PV power generation exhibits systematic temporal and seasonal variation, thereby supporting the inclusion of time of day and season as contextual features in our study. This chapter highlights that average AC power is strongly correlated with time of day and within a season, which supports the need to integrate seasonality and temporality into predictive models.

TH-4096_186151013

This page was intentionally left blank.

TH-4096_186151013

Chapter 4

Fault Detection Using Traditional Machine Learning (ML) and Deep Learning (DL) Models

In this chapter, various models that were developed to detect faults (or anomalies) in the 10 MW grid-connected Photovoltaic (PV) system developed by Azure Power (Haryana) are discussed, which rely on regression and classification models. This chapter also presents the development of predictive models based on recurrent neural networks (RNNs), such as simple RNN (Elman), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU) to predict power by considering the temporal features, and illustrates an anomaly detection mechanism.

4.1 Fault Detection using Traditional Regression Models

When regression models are used, a two-step process is used for fault detection. The first stage involves estimating the AC power output using a predictive model. In this work, the estima-

TH-4096_186151013

tion is performed through a data-driven machine learning (ML) approach. In the next step, the predicted AC power is compared with the corresponding measured AC power obtained from the PV system. A significant deviation between the predicted and observed values, exceeding a predefined threshold, is considered an indication of abnormal system behaviour. When such deviations persist continuously over a specified period, an alarm is generated to signal the presence of a potential fault. Figure 4.1 shows the fault detection approach using regression. The following sections describe the model development and the fault detection approach.

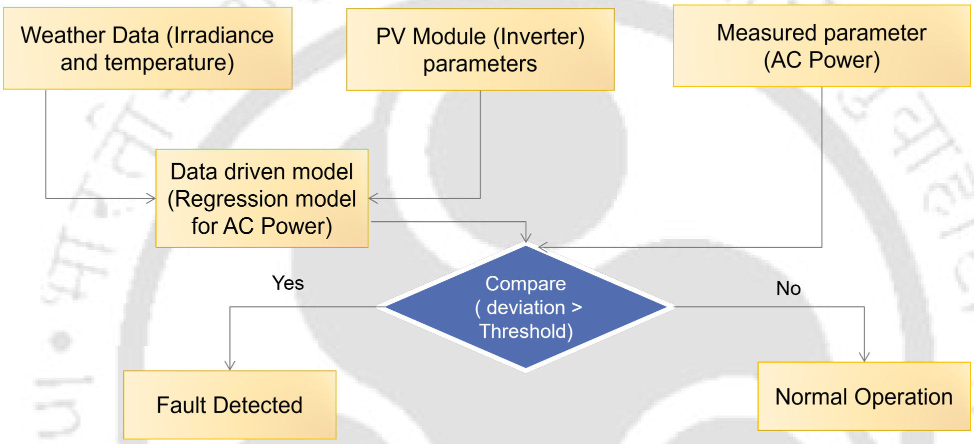


Figure 4.1: Fault detection using regression model

4.1.1 Data preprocessing and Feature selection

For the development of the ML-based regression model, data corresponding to periods when the PV plant is not generating power are excluded during the preprocessing stage. Only samples recorded between 05:45 and 19:30 hours are retained for further analysis. Since the regression model is intended to learn the behaviour of the system under healthy operating conditions, data points associated with anomalies and fault events are subsequently filtered out. The resulting number of valid samples available for each inverter in the plant is summarized in Table 4.1. For modeling, three input features, viz., Ambient Temperature, Horizontal Irradiation, and Tilt Irradiation are considered, and AC power is the output to be predicted. The correlation matrix is shown in Figure 4.2. Min-max normalization is used for scaling the data samples.

TH-4096_186151013

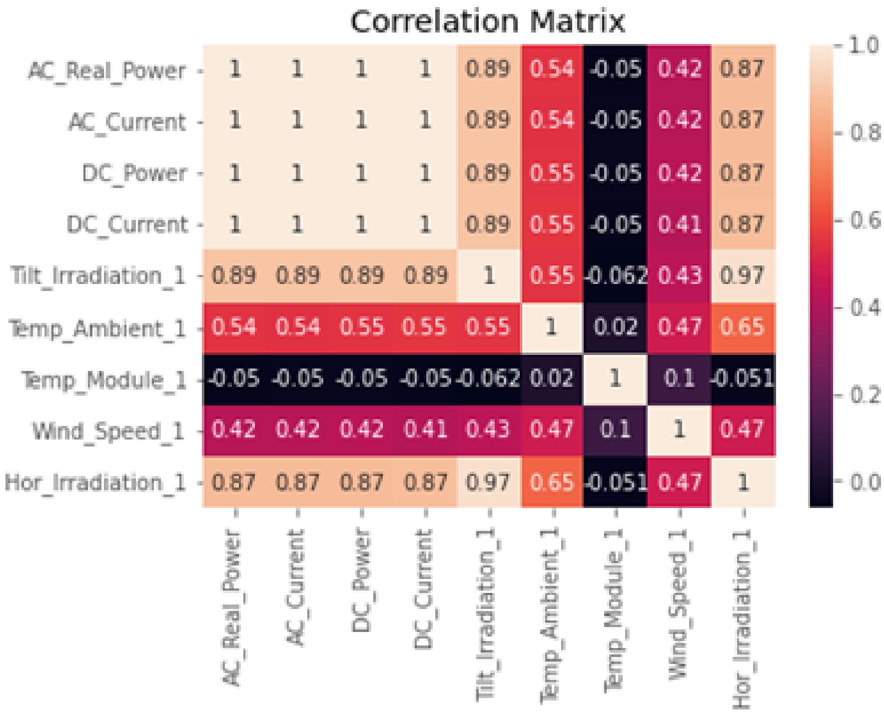


Figure 4.2: Correlation matrix

Table 4.1: Summary of data samples

Category	**Number of samples**
Normal operation data samples	2,39,359
Fault data samples	327
Total data samples	2,39,686

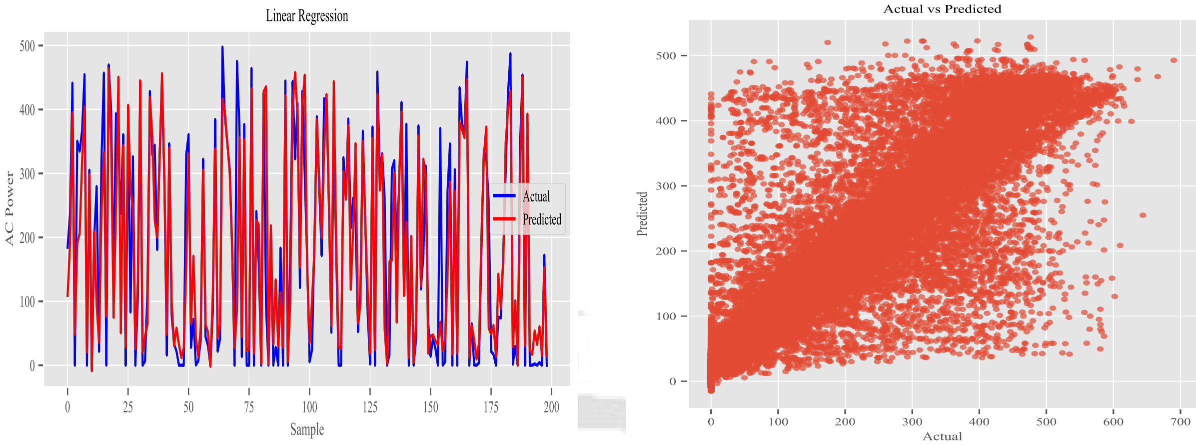
Table 4.2: Performance comparison of regression models

Model	Hyper-parameters	RMSE	R^2 Score
Linear Regressor (LR)	—	0.1026	0.7941
Random Forest Regressor (RFR)	Depth = 25, Number of estimators = 200	0.0658	0.9154
Support Vector Regressor (SVR)	Kernel = RBF (other parameters default)	0.0984	0.8106
ANN-based Regressor (ANN)	Hidden layers = (50, 100), Iterations = 500, Activation = ReLU	0.0942	0.8265
Ensemble Voting Regressor (EVR)	Estimators: MLP, RFR, K-NN (40 neighbours)	0.0708	0.9019
Ensemble Stacking Regressor (ESR)	Base estimators same as EVR, Final estimator: RFR	0.0655	0.9160

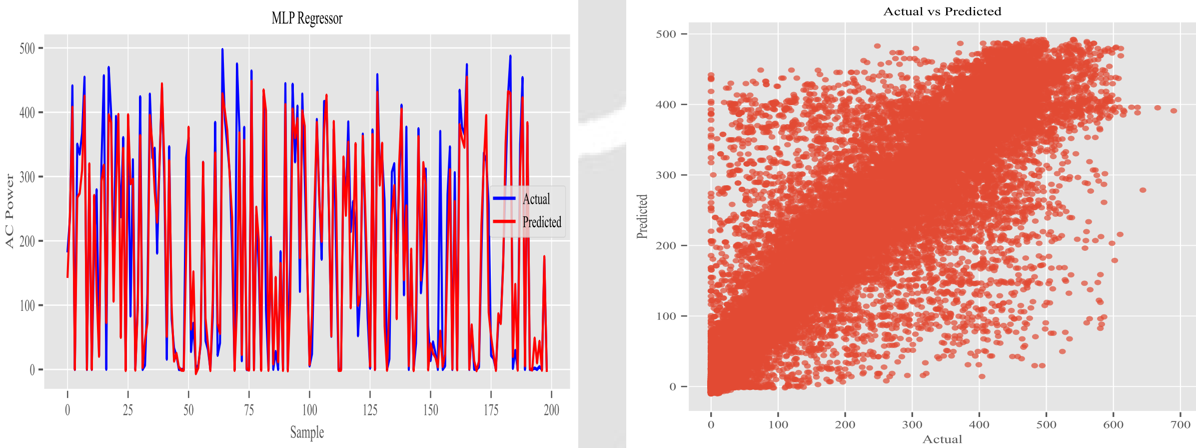
4.1.2 Model Development

For modelling, in addition to Linear Regression (LR), Support Vector Regressor (SVR), Random Forest Regressor (RFR), and Artificial Neural Network (ANN), two more ensemble techniques, namely, Ensemble Voting Regressors (EVR) and Ensemble Stacking Regressors (ESR), are considered. The data is divided into a training set (70%), a validation set (15%) and a test set(15%). The hyperparameters of each of the models are tuned using the validation set. Table 4.2 shows the results of various models. Selected plots are shown in Figure 4.3. For RFR, the best result is achieved with Depth 25 and number of estimators set to 200. ESR also gave equally comparable results with Estimators: MLP, RFR, K-NN (40 neighbours). Out of the six models, RFR and ESR performed well for the given data. Therefore, these models are the right candidates for the next step of fault detection process. Finally, RFR model is retained for the next step of fault detection process (because of higher computation time compared to RFR, ESR model is not selected).

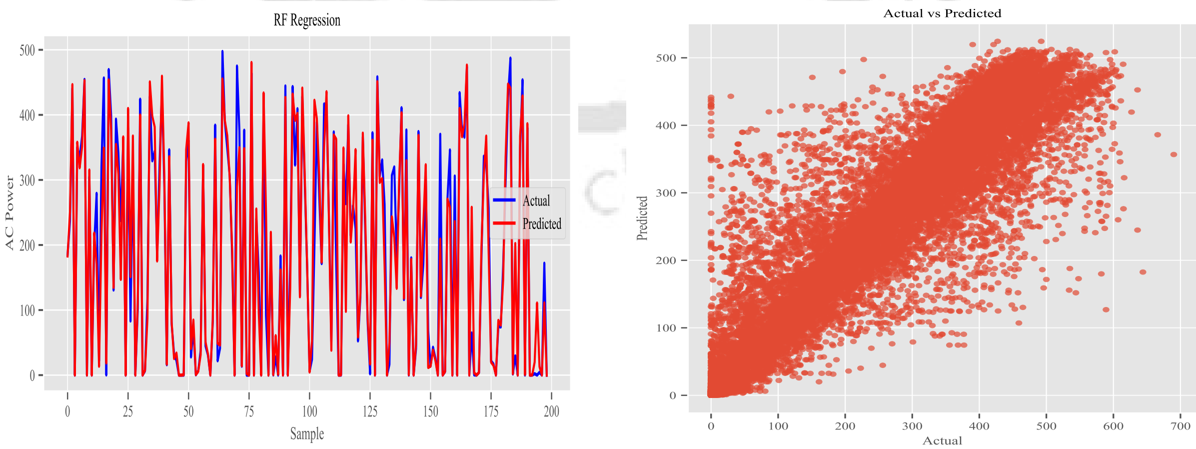
TH-4096_186151013



(a) Linear Regression



(b) MLP Regressor



(c) RF Regressor

Figure 4.3: Performance of Various Regressors

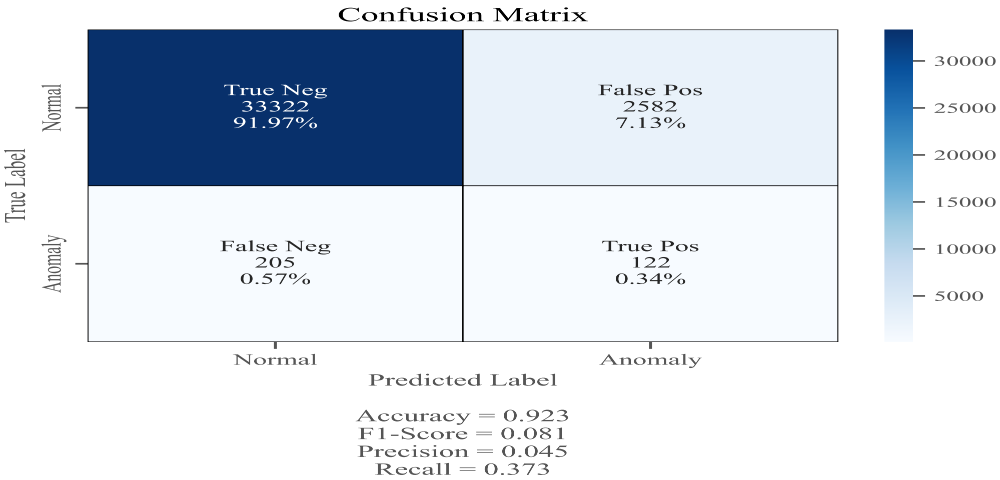
4.1.3 Fault detection

During the second phase of fault detection, it is necessary to establish a threshold value, which serves as the criterion for determining the presence of a fault. As previously discussed in the chapter, the difference between the actual AC Power and the predicted AC Power is evaluated against this threshold to identify any anomalies. To determine the threshold value, first the difference between the actual power available in the dataset and the estimated power is calculated for all the models and then the standard deviation (sigma) is determined. The threshold value is set to 2 * sigma, 3 * sigma , and 4 * sigma respectively for the RFR model.

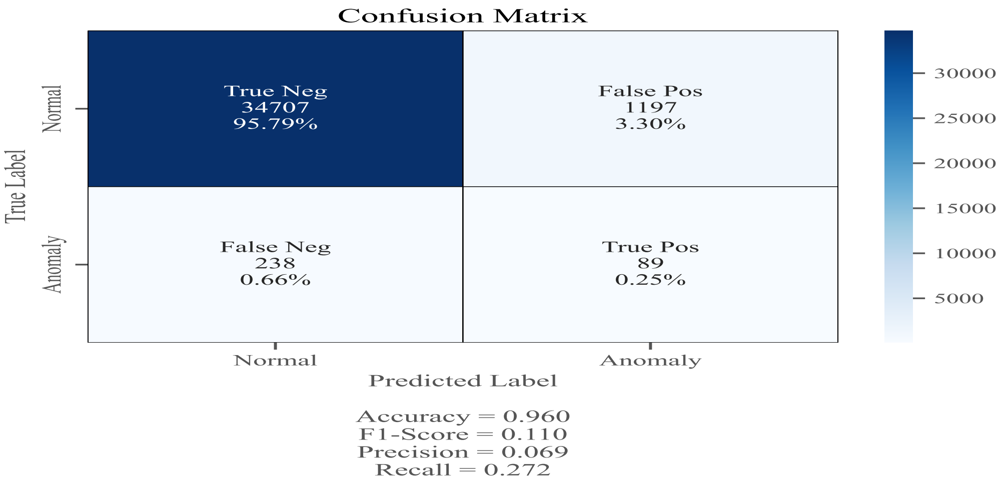
4.1.3.1 Fault detection results

The test set includes data from normal operations as well as anomalous instances. Figure 4.4 presents the related confusion matrices. To evaluate the performance of the models, three primary metrics are considered: Precision, Recall, and F1 Score as given in equations (4.1)–(4.4). Precision indicates how precise or accurate the model is, i.e., how many actual positive cases the model identifies out of the predicted positive cases. Precision is high when there are fewer false positives, meaning instances that are actually normal but are mistakenly identified as anomalies by the ML model. Recall, on the other hand, indicates the proportion of true positive cases correctly identified among all positive cases. This metric is particularly valuable when false negatives carry a significant cost. In the context of fault detection, our goal is to enhance recall by minimizing false negatives, which are anomalies incorrectly labeled as normal. The F1 score, which is the harmonic mean of precision and recall, is beneficial for us as it helps achieve a balance between these two metrics, especially given our data's uneven class distribution. Accuracy, however, is not a suitable performance metric for our task because the number of positive cases (anomalies) is much smaller compared to negative cases (normal operations). The accuracy measure is given by equation (4.4) . As apparent from the confusion matrices of RFR model, the number of true negatives is very high compared to the true positives due to which the resulting accuracy would be high.

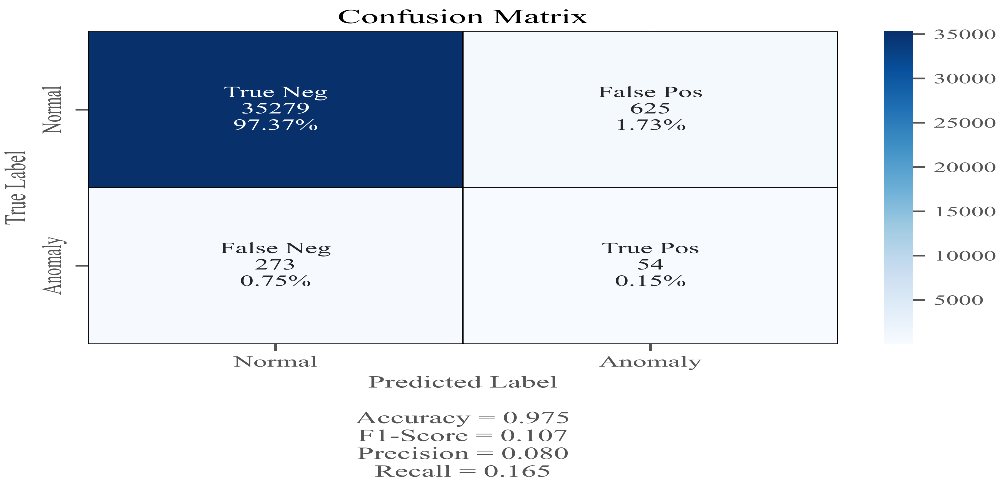
TH-4096_186151013



(a) Confusion matrix of RFR with threshold of 2*sigma



(b) Confusion matrix of RFR with threshold of 3*sigma



(c) Confusion matrix of RFR with threshold of 4*sigma

Figure 4.4: Confusion Matrices

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4.1}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{4.2}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

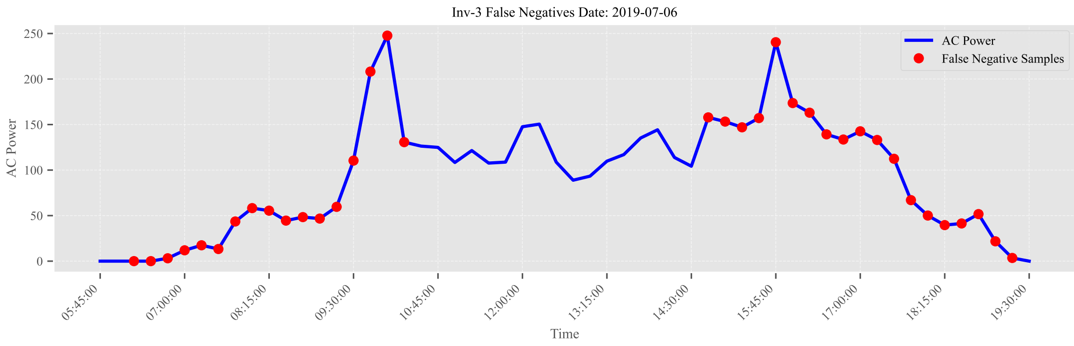
$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative}} \tag{4.4}$$

4.1.3.2 Analysis of false negatives

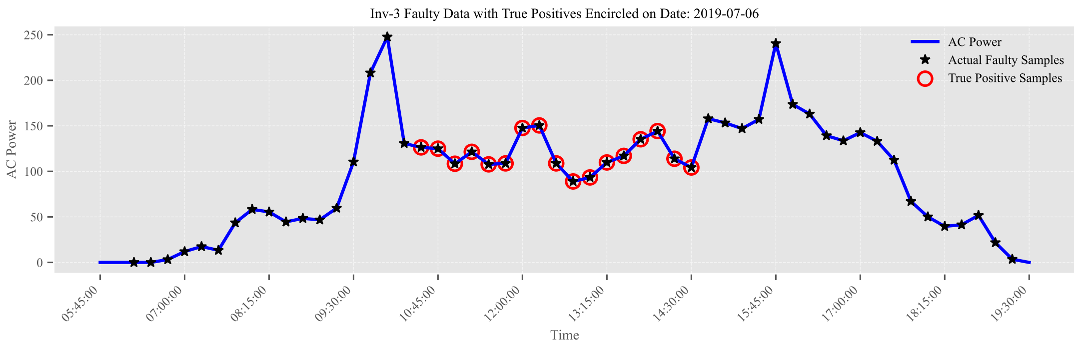
There are 205 false negative data points. We are presenting here only few selected ones from various inverters. We investigated which of the anomalous samples (positive cases) are falsely classified as normal by these models. The figure 4.5a shows the anomalous data samples that could not be detected by the model i.e. the false negatives. As given in the fault report, on 06/07/2019 fault occurred in Inv 3 from 6:15 to 19:15 and the error occurred due to problem in -ve side of DC cable. Figure 4.5b shows all the faulty data. The encircled points show the faulty data that are correctly identified as faults (True positives). Figure 4.6a shows false negatives for Inv 4 where the reason of fault is “Inverter not started”. Figure 4.6b shows false negatives for Inv 5. The fault was caused due to earth fault and Figure 4.6c shows false negative for Inv 15 for “Earth fault”.

4.2 Fault Detection using Traditional Classification Models

This approach aims at developing classifiers using labeled or tagged data. The developed classifiers can detect or classify a given data sample as normal or anomaly. If such anomalous values are detected continuously for a specific duration of time then an alarm is triggered indicating fault. Figure 4.7 shows the schematic representation of classifier based anomaly detection. For anomaly detection two types of classifies are considered here, one-class classifiers and the binary classifiers. The key difference between these two types of classifiers is in their training



(a) False negative of inverter 3



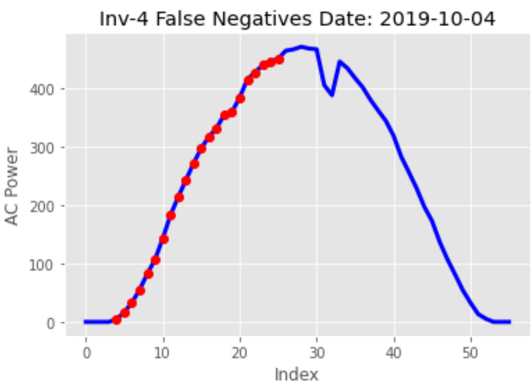
(b) Faulty data of inverter 3

Figure 4.5: False negatives and faulty data of inverter 3

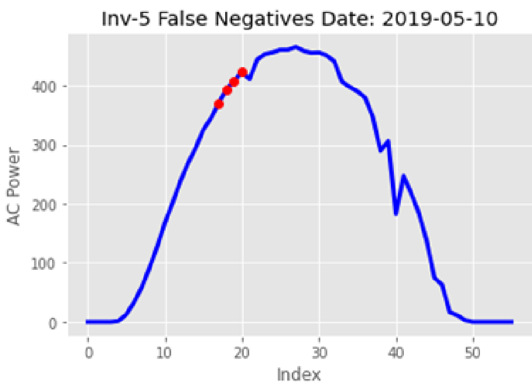
data. Binary classifiers necessitate a thoroughly labelled training dataset where the data from both classes are expected to be balanced. In contrast, one-class classifiers are trained exclusively with samples from the class of interest. Specifically, for the task of anomaly detection, one-class classifiers are trained using samples from the normal class. The subsequent subsection discusses the process of developing the classification models for detecting anomaly.

4.2.1 Data Preprocessing and Feature extraction

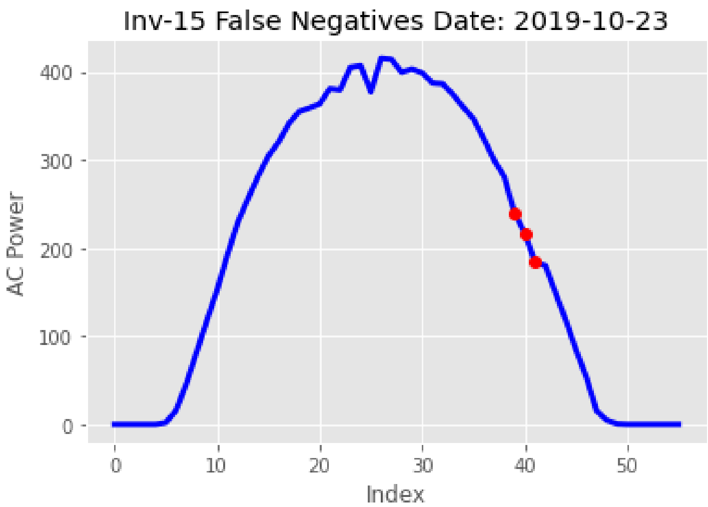
To develop the classification models, first the data is labeled using the available inverter fault report considering all the inverters. The labeled data consists of two classes: Normal (labeled as 0) and Anomaly (labeled as 1). For our task, Anomaly is the positive class. The data is cleaned by removing the data samples where AC Power is 0 as these values indicate that the plant is not in operation, particularly during early morning and night time. After cleaning, the data is scaled by applying standardization such that that the mean of observed values is 0 and the



(a) False negative of Inverter 4



(b) False negative of Inverter 5



(c) False negative of Inverter 15

Figure 4.6: False Negatives in various Inverters

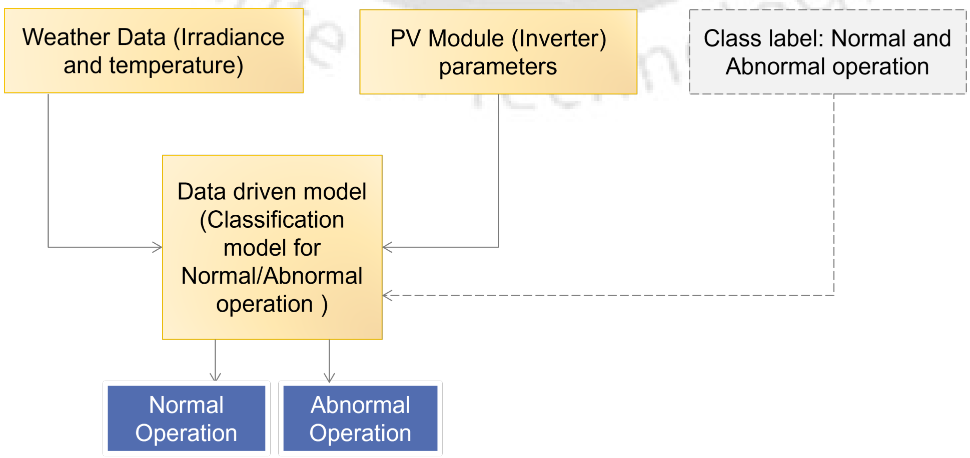


Figure 4.7: Schematic of classifier-based fault detection

TH-4096_186151013

standard deviation is 1. As the data is imbalanced i.e. the number of data samples that belong to normal class is very high compared to the data from anomalous class, and therefore, data augmentation technique is used. One of the data augmentation approaches is SMOTE (Synthetic Minority Oversampling Technique) where new samples for minority class are synthesized from existing samples. Here, SMOTE is used for augmenting the data of anomaly class. After data augmentation, the resulting data consist of 239,919 data samples each of normal and anomalous class. The data is divided for training and testing in the ratio of 70:30. It is to be noted that for training one-class classifier, data augmentation is not required.

To extract the required features, XGBoost classifier is used to get the feature importance. Based on the feature importance top 3 features are selected: DC_Current , Active_Power, Energy_Total, Hor_Irradiation

4.2.2 Models and Results

Five approaches were used to train the classifiers, Logistic Regression (LGR), Support Vector Machine (SVM), Multi-layer Perceptron (MLP), Decision Trees (DT), and Random Forest (RF).

The results based on precision, recall, F1-score, and accuracy are depicted in the Table 4.3. SVM gave best performance in terms of recall as compared to all other models.

Table 4.3: Performance comparison of classification models

Model	**Hyperparameters**	**Precision**	**Recall**	**F1-Score**	**Accuracy**
LGR	C = 0.05	0.62	0.55	0.58	0.61
SVM	Kernel = RBF	0.81	0.89	0.85	0.84
MLP	Layer = 1, Hidden units = 100, Iterations = 500	0.79	0.70	0.74	0.76
DT	Max depth = 5, Min samples leaf = 5	0.95	0.37	0.53	0.67
RF	Number of estimators = 100, Max depth = 5	0.98	0.21	0.35	0.60

TH-4096_186151013

4.3 Recurrent Neural Network-based Deep Learning Models for Power Prediction and Fault Detection

The RNN, and more specifically the LSTM networks, are dominantly used in sequence modelling, including language modelling, speech processing, voice recognition systems, etc. Considering the fact that solar PV data are time series data, various techniques such as RNN, LSTM, and GRU can be applied to analyse solar PV time series signals and weather data in order to predict possible faults. In this section, first an overview of sequential models is provided, and then the results obtained from these models are presented and discussed.

4.3.1 Brief Overview of Sequential Models

Many real-world phenomena generate data in the form of sequences, where observations are collected over time or follow a specific order. In such cases, the assumption of independence between data samples does not hold, as the current observation often depends on past states or events. Sequential models are a class of machine learning and statistical models specifically designed to capture and exploit these temporal or ordered dependencies in data.

Sequential models process input data as an ordered sequence and learn relationships across successive time steps. Unlike conventional machine learning models that treat each data instance independently, sequential models incorporate historical information through internal states, memory mechanisms, or probabilistic transitions. This makes them particularly suitable for applications involving time-series data, signals, text, event logs, and system monitoring.

Formally, a sequential modeling problem involves an input sequence $\{x_1, x_2, ...x_T\}$, where the objective is to predict an output that may be associated with a single time step, multiple time steps, or the entire sequence. The output can take various forms, such as a future value forecast, a sequence of labels, or a system state classification.

Early sequential models were mainly probabilistic, with approaches such as Hidden Markov Models (HMMs) [110] and state-space models representing systems using latent states and probabilistic transitions. While these models are effective for systems with clearly defined state

TH-4096_186151013

dynamics, they are limited in handling complex nonlinear behavior and high-dimensional data. With advances in computational power, neural network-based sequential models became prominent. Recurrent Neural Networks (RNNs) introduced feedback connections to capture temporal dependencies, but their ability to learn long-term relationships is hindered by vanishing and exploding gradient issues. To overcome these issues, LSTM networks and GRUs were created.

4.3.1.1 Recurrent Neural Network (RNN)

An RNN [111] belongs to a category of neural networks that is suitable for modelling sequential data. A key characteristic of RNNs is the presence of at least one feedback connection, which enables activations to circulate in a loop while having hidden states. This configuration allows for temporal processing and sequence learning. There are various kinds of RNN architectures. The basic RNN is composed of three layers, namely, input, recurrent hidden, and output layers as shown in Figure 4.8.

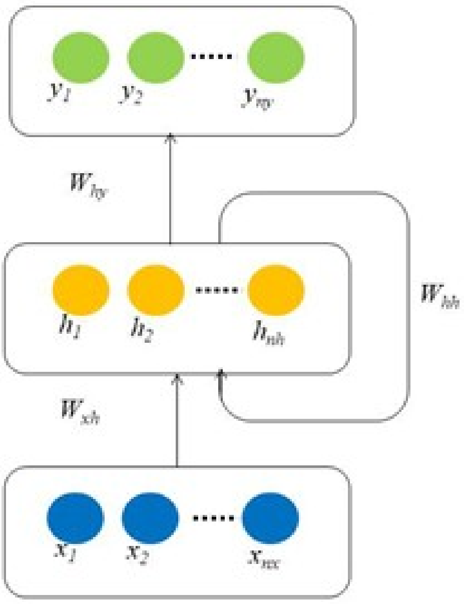


Figure 4.8: Basic RNN structure

The computation of the output $\hat{y}_t$ and the hidden state h_t at time step t in an RNN is given by:

$$\hat{y}_t = f\left(W_{hy}^T h_t + b_y\right) \tag{4.5}$$

$$h_t = \tanh\left(W_{xh}^T x_t + W_{hh}^T h_{t-1} + b_h\right) \tag{4.6}$$

where $f(\cdot)$ denotes the output activation function, W_{hy} is the hidden-to-output weight matrix, and b_y is the output bias vector. The vector x_t represents the input at time step t, W_{xh} and W_{hh} are the input-to-hidden and hidden-to-hidden weight matrices, respectively, and b_h is the hidden bias vector.

4.3.1.2 Long Short Term Memory (LSTM) Network

The LSTM architecture was introduced by Hochreiter and Schmidhuber [112] in 1997 as an enhancement to traditional RNNs to address the vanishing and exploding gradient issues that arise during training. By incorporating specialized memory mechanisms, LSTM networks are capable of capturing long-range temporal dependencies more effectively than standard RNNs. Owing to this capability, LSTMs are widely used DL architectures in sequential data analysis and time-series forecasting [96]. In an LSTM network, the architecture generally includes an input layer, followed by one or more hidden layers, and ends with an output layer. The distinguishing feature of the architecture is the hidden layer, which contains memory cells responsible for retaining and updating information over time. Each memory cell comprises several components, including a forget gate, an input gate, a candidate input node, an output gate, a recurrent unit, and a cell state acting as long-term memory [97],[18]. As depicted in Figure 4.9, the variable h_t represents the hidden state, which captures short-term information, whereas the cell state, denoted by C_t, retains the long-term information. Gating mechanisms are responsible for controlling how information moves through the network by determining which information should be retained, updated, discarded, or propagated to the output at every time step [97],[113]. The long-term state from previous LSTM cell denoted as C_{t-1} traverses through the current LSTM network, and drops some memory under the control of forget gate, and then some new memory is added via the addition operation. So, at every timestep, C_{t-1} drops some memory, and some new memory is added to it in order to get the long term state C_t at current time t. Furthermore,

TH-4096_186151013

after allowing the long term state to pass through *tanh* function, it is filtered by the output gate resulting in h_t (i.e., the short-term state) which is also produced as output at y_t.

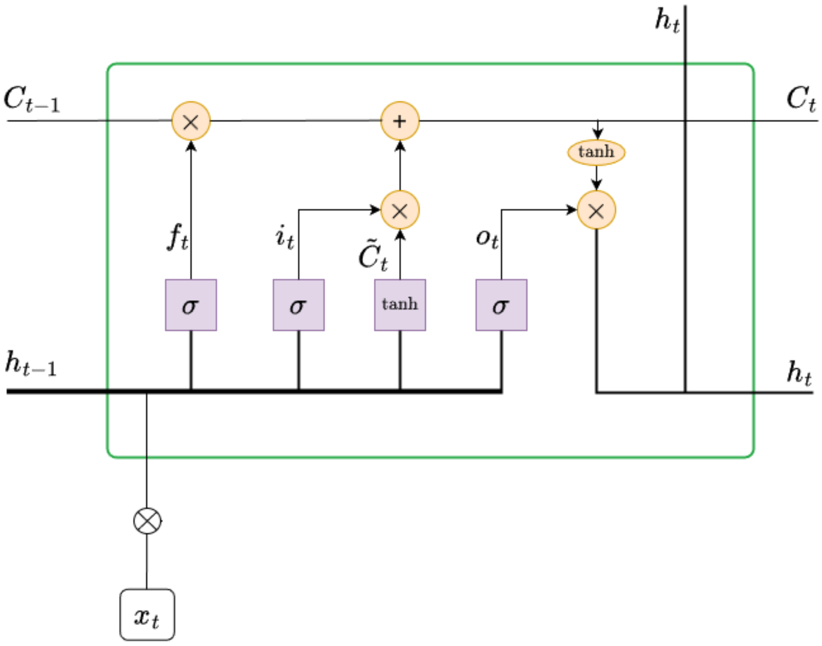


Figure 4.9: LSTM cell

In the LSTM cell, the previous short-term state h_{t-1} and the current input vector x_t are provided as inputs to four different fully connected layers. As shown in the Figure 4.9, the layer that outputs $\tilde{C}_t$ is the main layer. It is also known as input node. It analyses x_t and h_{t-1}. In LSTM, the long term state contains partial output from this layer (i.e. input node). The other three layers are basically gate controllers, and these gate controllers use sigmoid activation functions, so their outputs lie between 0 and 1. It is to be noted that the outputs of these gates are subjected to element wise multiplication (also, referred to as Hadamard product). Therefore, an output of 0 will close the gate, and an output of 1 will open the gate. Other values lying between 0 and 1 will decide the proportion of information to be passed through a particular gate. More specifically, the task of the forget gate f_t is to decide the parts of long term state to be erased [113]. The decision regarding which parts of $\tilde{C}_t$ should be incorporated into the long term state is controlled by the input gate i_t. Finally, the proportion of the long term state that is retrieved and presented as output at a particular time step is controlled by the output gate o_t. The forget gate f_t, the input gate i_t, the output of input node denoted as $\tilde{C}_t$, the long term state C_t, and the short term state (i.e., hidden state) h_t of an LSTM cell are updated by the following equations:

TH-4096_186151013

$$f_t = \sigma\left(W_{fh}h_{t-1} + W_{fx}x_t + b_f\right) \tag{4.7}$$

$$i_t = \sigma\left(W_{ih}h_{t-1} + W_{ix}x_t + b_i\right) \tag{4.8}$$

$$\tilde{C}_t = \tanh\left(W_{gh}h_{t-1} + W_{gx}x_t + b_g\right) \tag{4.9}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4.10}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{4.11}$$

4.3.1.3 Gated Recurrent Unit (GRU)

The GRU, introduced by Cho et al. [114] in 2014, serves as a simplified alternative to the LSTM. In contrast to LSTM, the GRU combines the short-term state, h_t, with the long-term state, C_t, into one unified vector h_t. One gate controller is responsible for managing both the forget and input gates. An output value of 1 from the gate controller results in the activation of the input gate, while the forget gate is kept in a closed state. On the contrary, when the gate controller is 0, the exact reverse happens. Another important point to note is that there is no output gate, resulting in the entire state vector being generated at each time step at the output. Therefore, at each time step, the full state vector is produced at the output. A new gate controller determines the portion of the previous state that is presented to the main layer.

4.3.2 Estimation of AC power with sequential models

The previous set of models do not take into account the temporal relation in the data such that AC Power can be predicted ahead (for example, $t+k$ time step ahead where t is the current time) Three sequential models are considered for next set of experiments: Recurrent Neural Networks (Elman), LSTM , and GRU for prediction of AC Power. The input parameters for these models are: Tilt Irradiation, Ambient Temperature, Horizontal Irradiation, and AC Power (at time t) and Output parameter: AC Power at time step $t+k$. The models use past 5 samples to predict the next sample, i.e. 15 mins ahead as shown in Figure 4.10. The distribution of

TH-4096_186151013

data used for training, testing and validation is: 192323, 24041, 24041 respectively. Table 4.4 shows the performance of all the three models. Figure 4.11 shows the plot for predicted versus actual values of AC power for a small number of samples and also the correlation plot. It can be observed that GRU preformed slightly better compared to LSTM and RNN. However, if R2 score is considered the results are not significantly different.

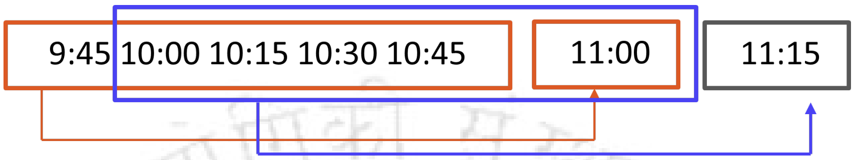


Figure 4.10: Length of time sequence

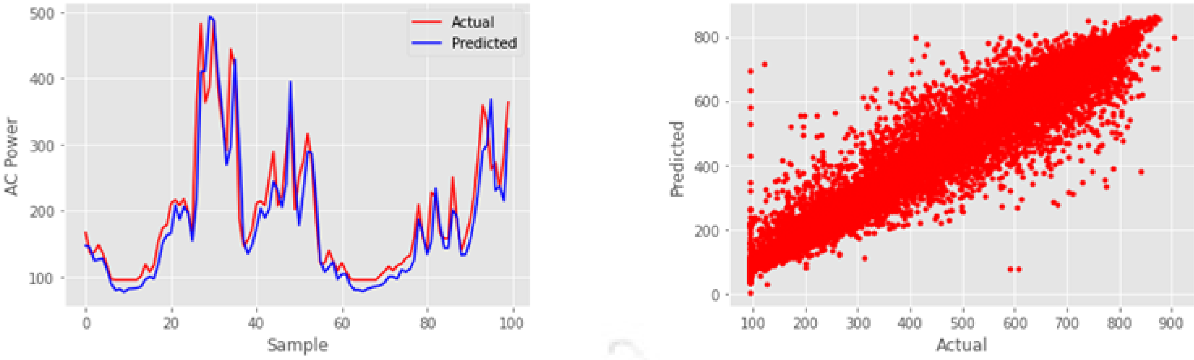
Table 4.4: Performance comparison of sequential models

Model	**Hyper-parameters**	**RMSE (not normalized)**	**R^2 Score**
RNN	Layers = 1, Hidden units = 110, Learning rate = 0.00165	46.86	0.963
LSTM	Layers = 1, Hidden units = 138, Learning rate = 0.00314	48.37	0.960
GRU	Layers = 1, Hidden units = 170, Learning rate = 0.00672	46.32	0.964

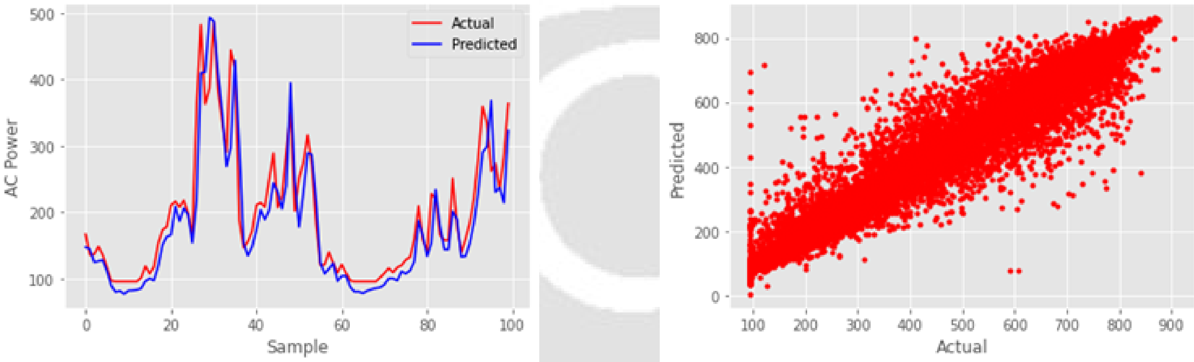
4.3.3 Fault Detection

For testing the fault detection part, we created the test series by combining the data of one day ahead of fault, the fault day and one day after the fault. The training error is modeled using a Gaussian distribution to ascertain the mean and standard deviation, which are then employed to establish the threshold for fault detection as shown in Figure 4.12. Table 4.5 shows the results with a threshold probability of 0.005. Ideally, threshold probability of 0.001 should have been chosen, but within 0.001 threshold none of the anomalies were detected as fault. So, after experiments 0.005 was chosen as threshold.

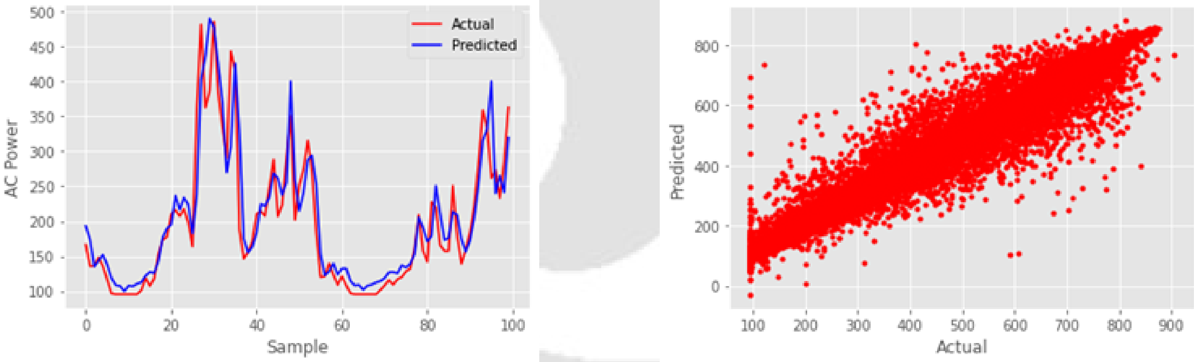
It is apparent from Table 4.5 that if we consider the F1-score, LSTM performed better. However, there is not significant difference in comparison to other two models. If we consider the recall,



(a) Prediction of RNN



(b) Prediction for LSTM



(c) Prediction of GRU

Figure 4.11: Predicted versus actual values and correlation plots

Table 4.5: Performance comparison of sequential classification models

Model	Precision	Recall	F1-Score	Accuracy
RNN	0.208	0.332	0.256	0.681
LSTM	**0.222**	0.329	**0.265**	**0.699**
GRU	0.179	**0.424**	0.252	0.585

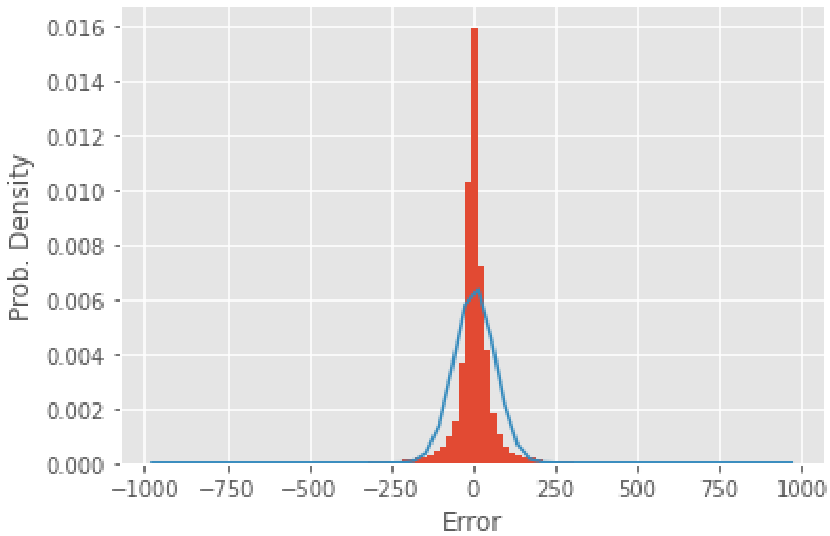
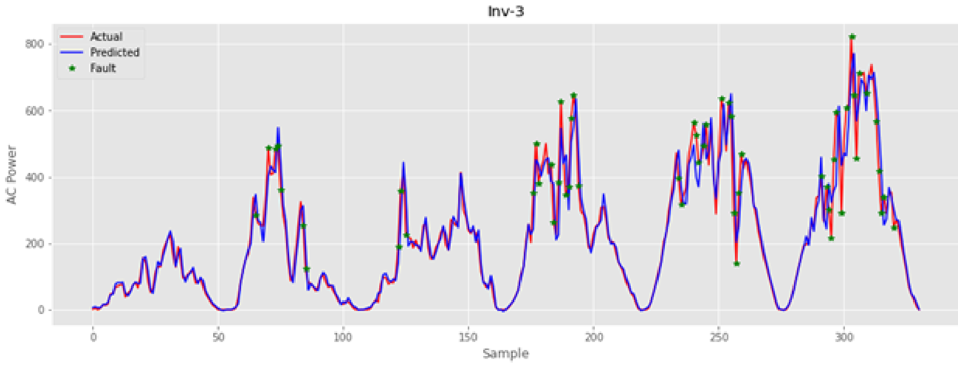


Figure 4.12: Probability density of Error

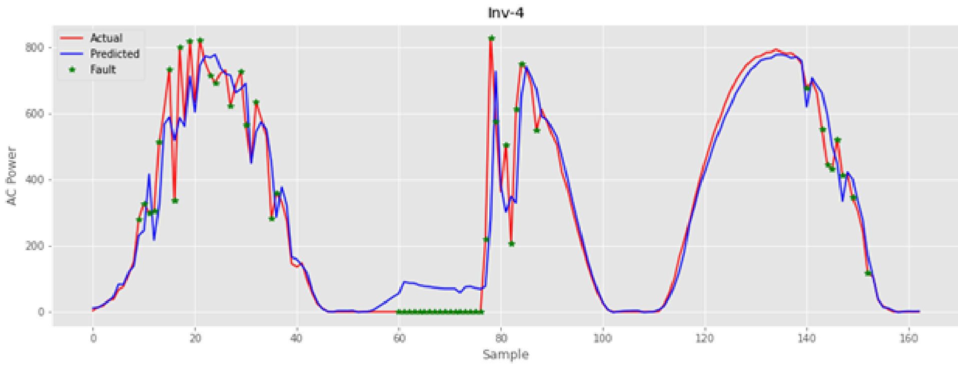
GRU has the best recall factor which means it has least false negatives. Next, we analysed the False Negatives (faults that the models could not identify). As GRU has least number of false negatives, we will consider the results of GRU for analysis.

Figure 4.13 – Figure 4.16 show the plots of all the faults identified by GRU for each inverter. Both false positives and true positives are shown(highlighted with green star markers). It can be observed that whenever there is a deviation of the actual AC power from the estimated value more than the threshold value, the sample is labeled as fault. Figure 4.17 – Figure 4.21 show the False negatives (faults that could not be identified by the model) and True negatives (faults that could be identified by the model) for each of the inverters where fault occurred. Table 4.6 shows the type of Inverter fault for each of the inverters.

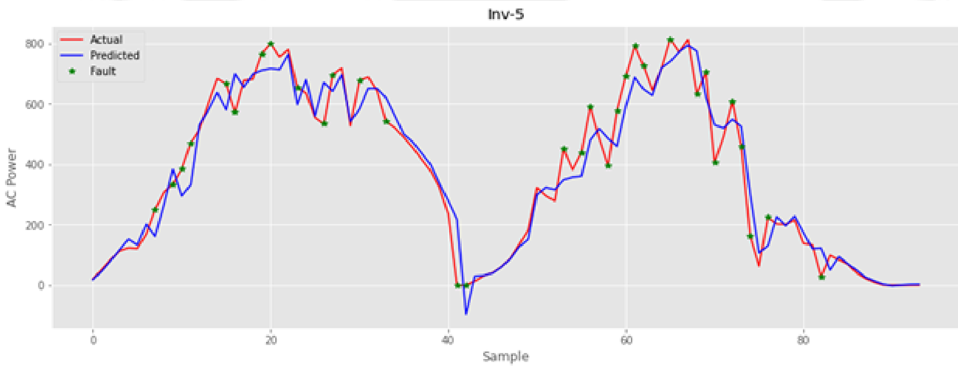
It can be observed from Figure 4.17 and the Table 4.6 that false negatives are higher for "Earth fault" in Inverter 3. In comparison to Inverter 3, False negatives are less in case of Inverter 4, 5, and 6. Highest number of true positives can be observed in Inverter 4, where the fault is "inverter no start". However, for Inverter 7 "Inverter no start" faults on 12-11-2019 could not be identified and the model gave more false negatives. The other type of faults in inverter 7 could be identified by the model. In Inverter 8, the error that occurred on 10-08-2019 could not be tagged as fault for all the samples whereas the other two types of fault viz. "IGBT failure" and "Inverter control supply fail" could be tagged. For inverter 9,10, 11 all the "straight joint failure" faults could be identified whereas for inverter 9 the fault on day 09-07-2019 could not be identified as "IGBT failure". For inverter 14 and 15 the type of fault is "heat sink overtemperature" and



(a) Inverter 3 faults

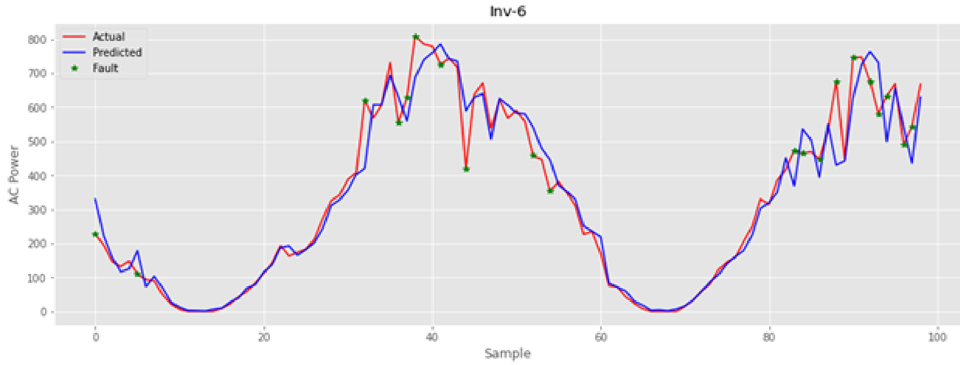


(b) Inverter 4 faults

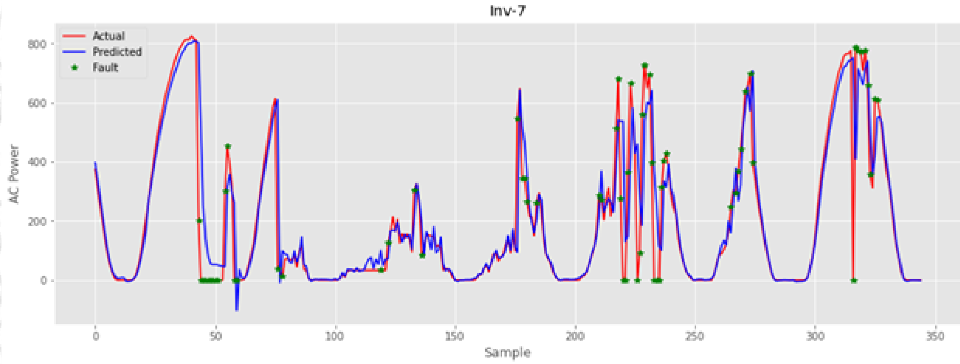


(c) Inverter 5 faults

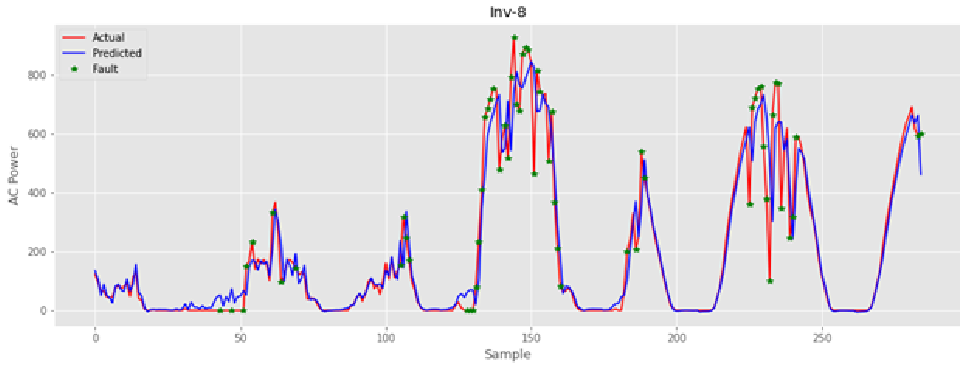
Figure 4.13: Faults identified by GRU (False positives and True positives)- Inverter 3, 4, and 5



(a) Inverter 6 faults

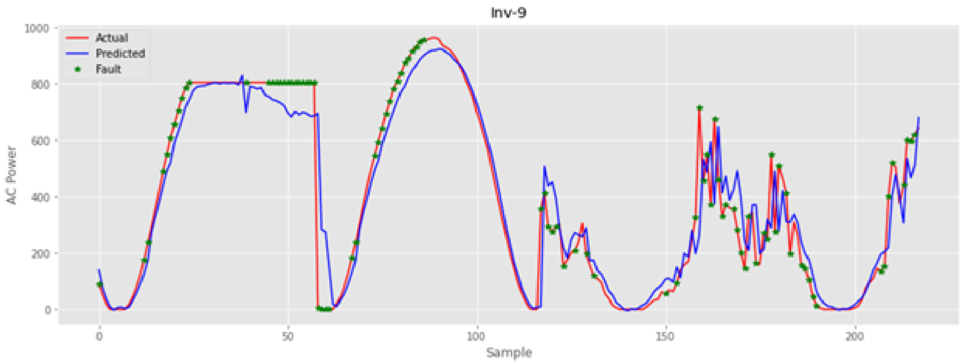


(b) Inverter 7 faults

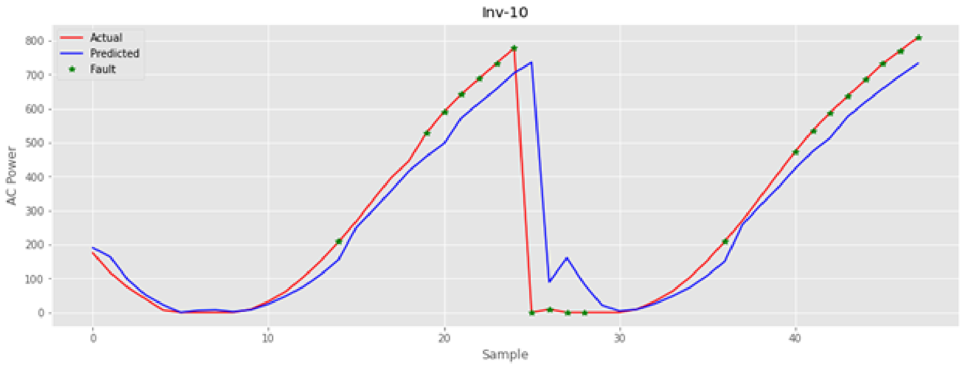


(c) Inverter 8 faults

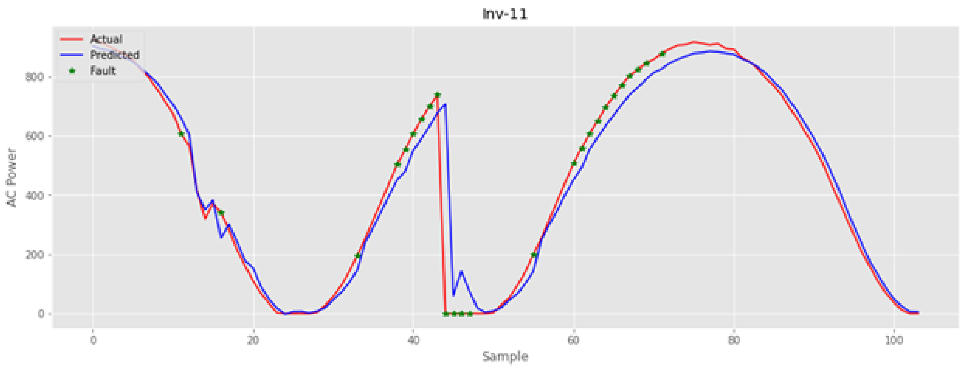
Figure 4.14: Faults identified by GRU (False positives and True positives)- Inverter 6, 7, and 8



(a) Inverter 9 faults

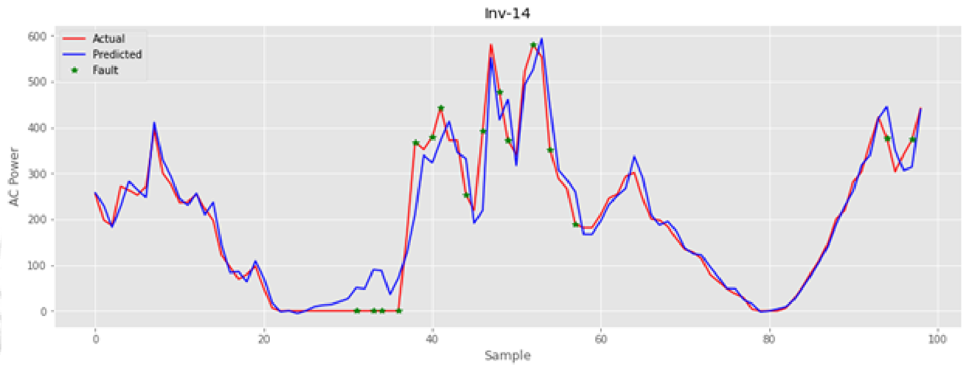


(b) Inverter 10 faults

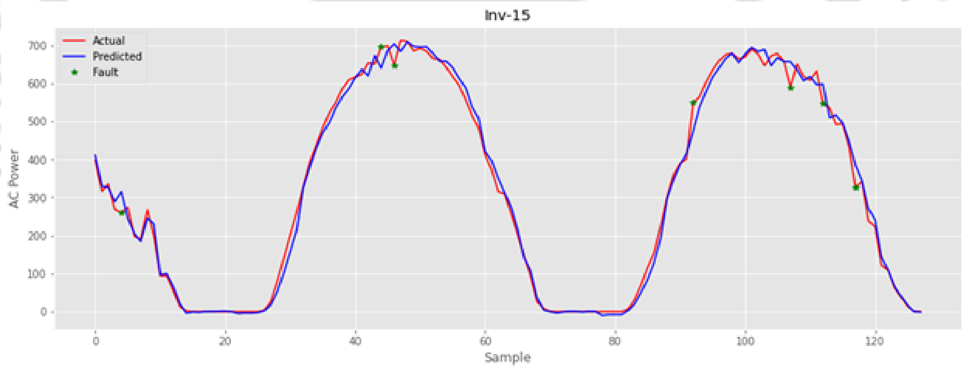


(c) Inverter 11 faults

Figure 4.15: Faults identified by GRU (False positives and True positives)- Inverter 9, 10, and 11



(a) Inverter 14 faults



(b) Inverter 15 faults

Figure 4.16: Faults identified by GRU (False positives and True positives)- Inverter 14 and 15

TH-4096_186151013

“earth fault” respectively. Both the faults could not be identified properly.

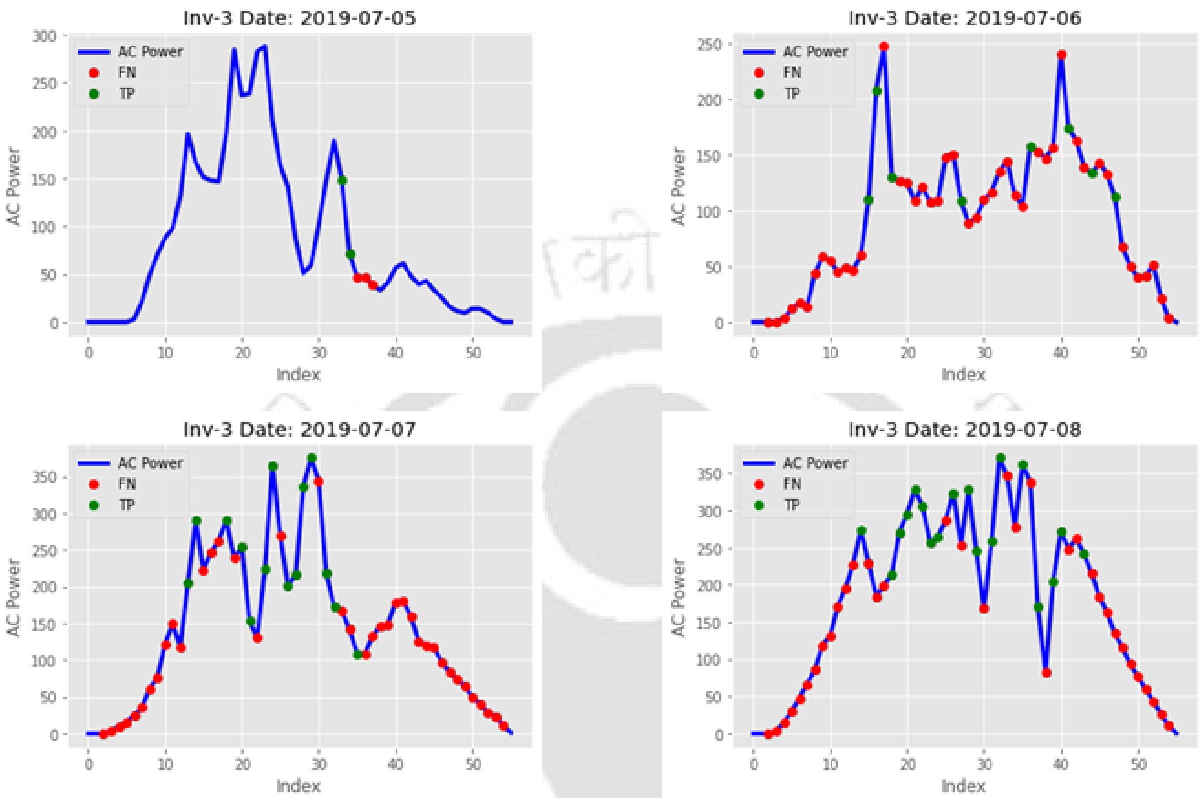


Figure 4.17: Inverter 3 faults (FP and TP)

The problem with RNN-based models is that the prediction error accumulates continuously and the error becomes larger with time [115]. This can be observed in Figure 4.13 – Figure 4.16 where in the beginning there is a little deviation of actual AC Power from the predicted and as time elapses, this deviation increases. This is one of the causes of more false positives. It can be voided by considering limited number of faulty samples in the test data, which means as the fault occurs say 2 or 3 samples can be considered rest can be discarded till the issue is resolved. This is observed in Inverter 10 and 11 (Figures 4.21 (b)) where only one sample is recorded when the fault occurred (first green dot) and the second green dot is when the fault is addressed. During the time when the fault was still existing there were no samples recorded. There are no false negatives for these two inverters. On the other hand, in such a situation the model will get only few anomalous samples and hence the error will not accumulate leading to lesser false positives.

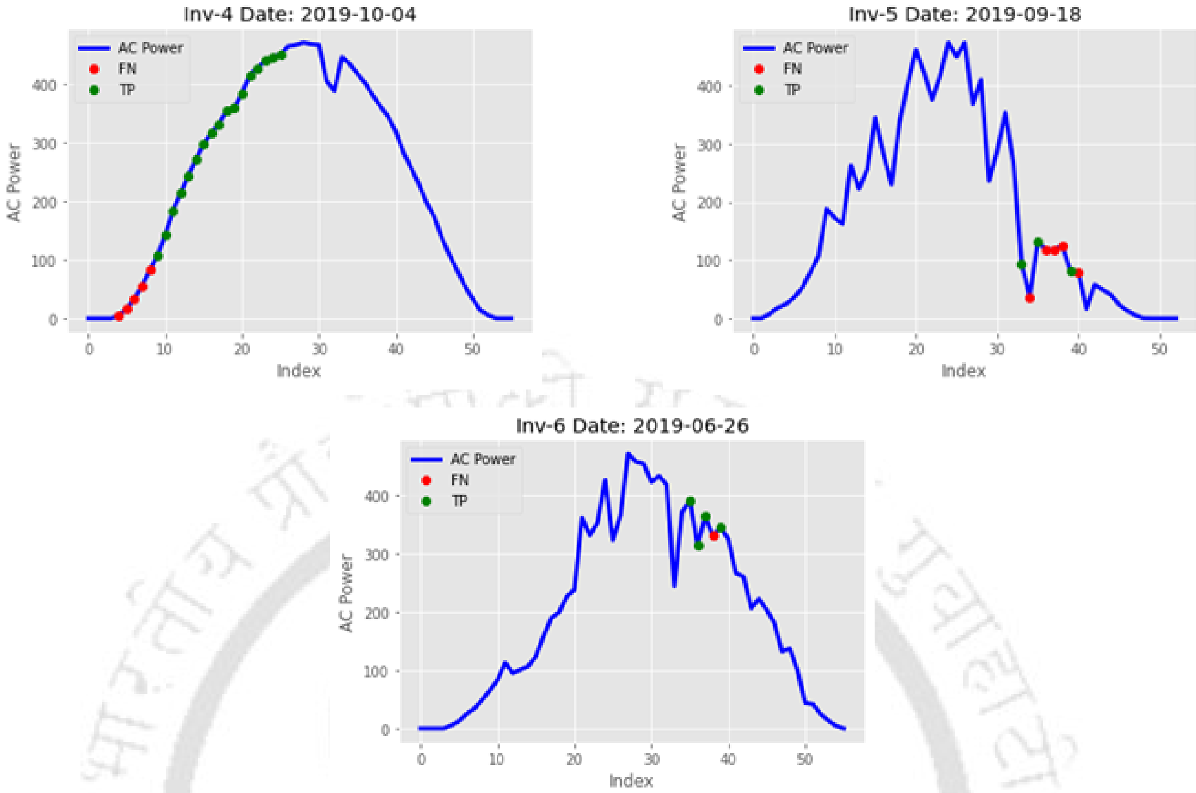


Figure 4.18: Inverter 4,5,6 faults (FP and TP)

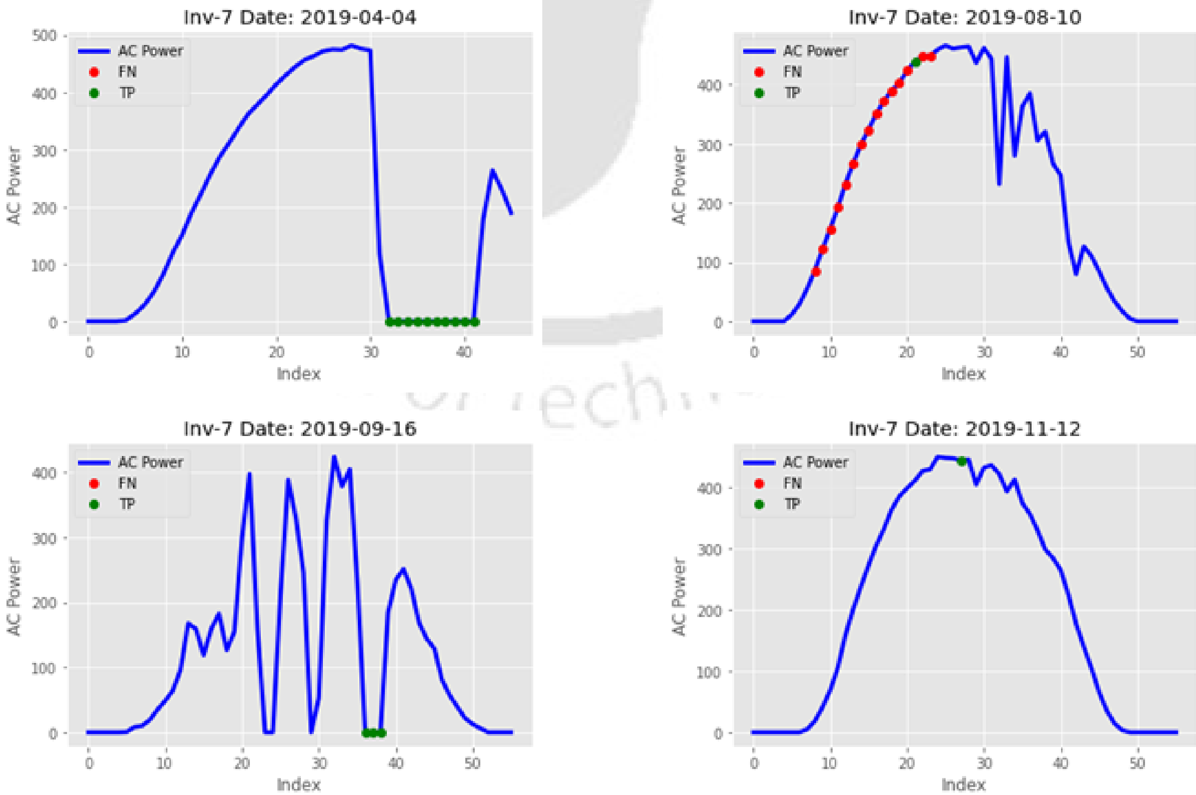


Figure 4.19: Inverter 7 faults (FP and TP)

Table 4.6: Inverter-wise fault occurrence details

Inverter	Date	Time	Fault
Inv-3	05-07-2019	14:00–15:00	Earth Fault
	06-07-2019	06:15–19:15	0 IR value in the -ve cable
	07-07-2019	06:15–19:15	Earth Fault
	08-07-2019	06:15–19:15	Earth Fault
Inv-4	04-10-2019	06:45–12:00	Inverter not start
Inv-5	18-09-2019	14:45–16:30	L2 IGBT driver board failure
	10-05-2019	10:00–10:45	Earth Fault
Inv-6	26-06-2019	14:30–15:30	Earth Fault
Inv-7	04-04-2019	13:45–16:00	Overtemperature failure
	10-08-2019	07:45–11:30	Inverter not start
	16-09-2019	14:45–15:15	IGBT driver board L3 failure
	12-11-2019	12:00–12:30	Earth Fault
Inv-8	10-08-2019	07:00–12:00	Tripped with error code 105
	25-09-2019	08:00–08:45	L2 IGBT driver board failure
	12-11-2019	12:00–12:30	Inverter control supply failure
Inv-9	19-05-2019	10:00–18:45	Straight joint failure
	09-07-2019	06:30–08:00	IGBT error failure
Inv-10	19-05-2019	10:00–18:45	Straight joint failure
Inv-11	19-05-2019	10:00–18:45	Straight joint failure
Inv-12	19-05-2019	10:00–18:45	Straight joint failure
	03-07-2019	06:15–07:45	C-tech failure
	15-08-2019	10:30–11:45	IGBT driver board failure
Inv-14	07-07-2019	06:15–09:00	Heat sink failure (high temperature)
Inv-15	23-10-2019	15:30–16:00	Earth Fault
	07-02-2020	12:00–12:45	Earth Fault

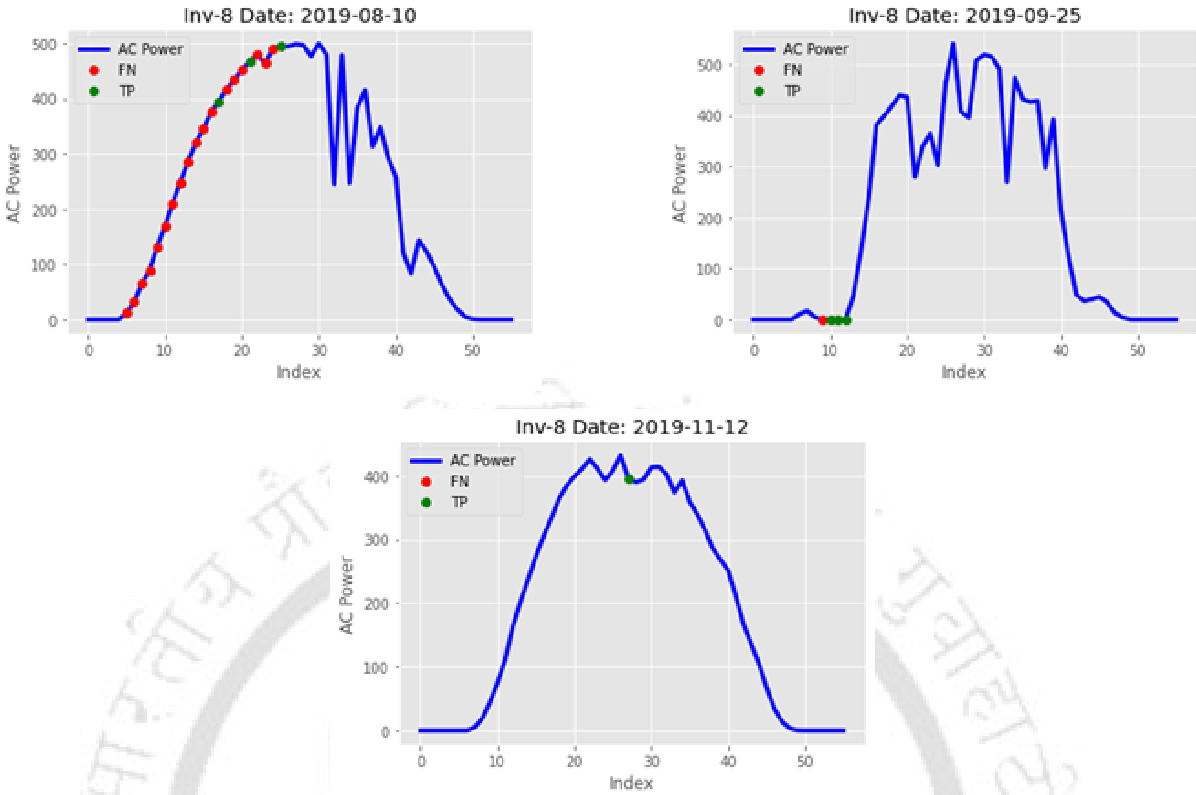
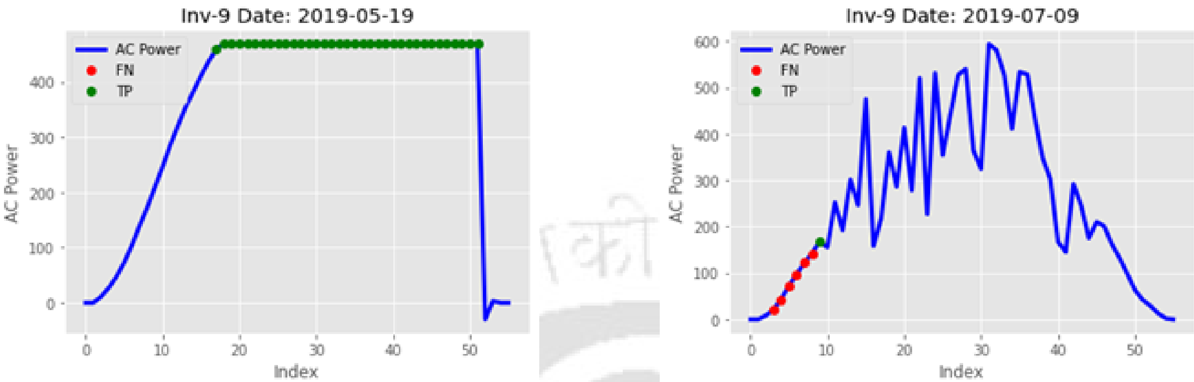


Figure 4.20: Inverter 8 faults (FP and TP)

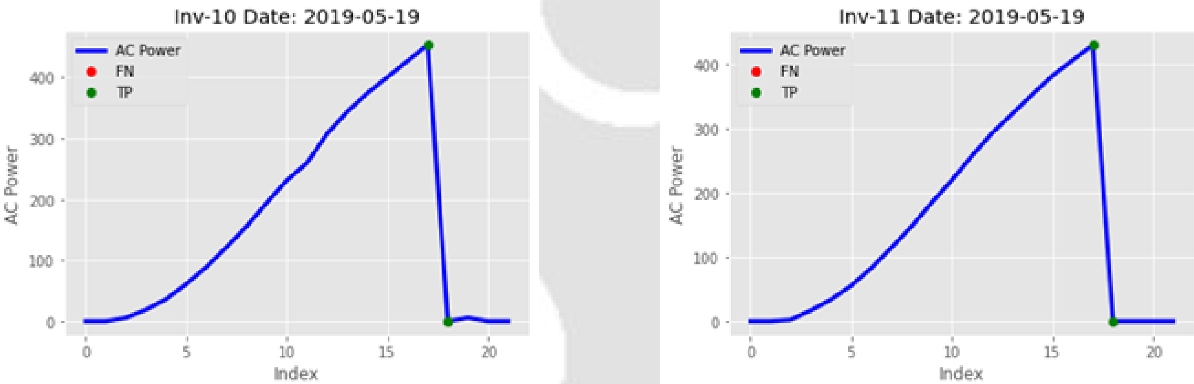
4.4 Sensitivity Analysis

Threshold selection is very important for regression-based fault detection, since the threshold directly affects false negatives, false positives, F1-score, precision, and recall. Sensitivity analysis of various threshold values such as 2*sigma, 3*sigma, and 4*sigma in terms of accuracy, F1-score, precision, and recall are shown in Figure 4.4 for the Random Forest (RF) regressor. Sensitivity analysis of the performance of the RF regressor with thresholds of 2*sigma, 3*sigma, and 4*sigma is also depicted in Table 4.7 , giving accuracies of 0.923, 0.960, and 0.975, respectively. However, the highest recall (0.373) was obtained with a threshold of 2*sigma. Since false negatives are critical in fault detection, the lower threshold was considered more useful for detecting more faults, even at the cost of more false positives.

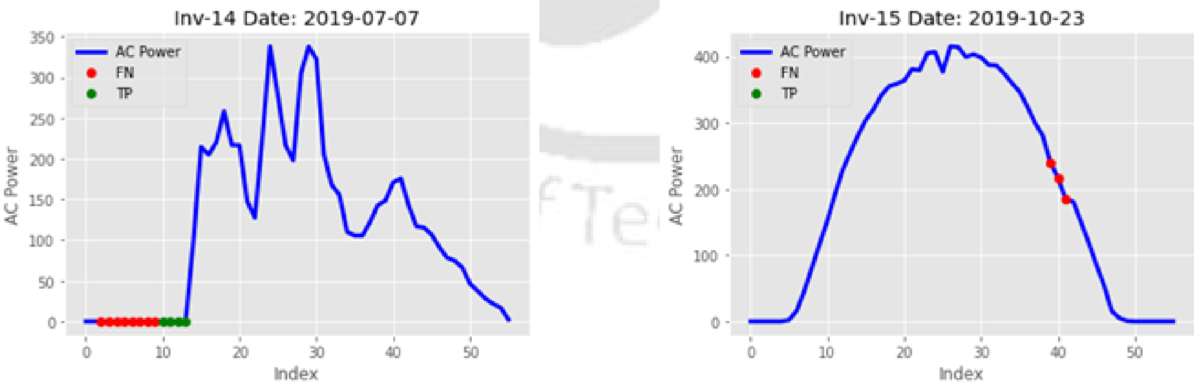
Furthermore, in this chapter, the fault detection threshold for sequential models such as RNN, LSTM, and GRU has been selected by modelling the training prediction error using a Gaussian distribution. Based on our experiment, a probability threshold of 0.001 did not detect any anomalies, and therefore, 0.005 was selected after experimentation. The resulting performance



(a) Inverter 9 faults



(b) Inverter 10 and 11 faults patterns



(c) Inverter 14 and 15 faults (FN and TP)

Figure 4.21: Inverter faults (True Positives and False negatives)

Table 4.7: Sensitivity analysis of Random Forest (RF) regressor for different detection thresholds.

Threshold	TN	FN	FP	TP	Accuracy	F1-score	Precision	Recall
2σ	33322	205	2582	122	0.923	0.081	0.045	0.373
3σ	34707	238	1197	89	0.960	0.110	0.069	0.272
4σ	35279	273	625	54	0.975	0.107	0.080	0.165

was reported in terms of accuracy, F1-score, precision, and recall for RNN, LSTM, and GRU models in Table 4.5 with a threshold probability of 0.005, and GRU provided the highest recall of 0.424. The threshold choice affected anomaly detection performance.

Missing a true fault is more serious than only reducing false alarms. Therefore, emphasis has been given to metrics such as precision, recall (i.e., true positive rate), F1-score, and accuracy that are appropriate for fault detection models. Figure 4.4 contains confusion matrices of the Random Forest Regressor (RFR) with various threshold values in terms of accuracy, F1-score, precision, and recall. Performance comparison of various classification models such as LGR, SVM, MLP, DT, and RF in terms of precision, recall, F1-score, and accuracy has been depicted in Table 4.3. Table 4.5 contains performance comparison of sequential classification models such as RNN, LSTM, and GRU in terms of accuracy, recall, precision, and F1-score.

4.5 Summary

The work presented in this chapter can be summarised into three broad sub-tasks:

- Developing the models for anomaly detection using 6 regression-based models.
- Developing the models for anomaly detection using 5 classification-based models.
- Developing the temporal (sequential) models for prediction of AC Power

The first set of experiments relies on regression models for estimation of AC Power, and then the deviation of measured power from the estimated power is used to detect if the measured power

TH-4096_186151013

is an indication of an anomaly. For estimation of AC power, six ML models were developed, viz., LR, RFR, SVR, MLP, EVR, and ESR. It is observed that the performances of RFR and ESR are better than other models in terms of RMSE and R^2 Score. Based on the performance, RFR model is selected for further detecting faults as they are computationally faster than ESR models. When a threshold of 2*sigma is used, the Random Forest-based regression model resulted in the maximum number of True Positives i.e., the highest recall. Considering the F1-score, the RFR model with 3*sigma provided the highest F1-score. It is observed that faults that are caused by "Earth fault" or "Inverter not start" could not be detected by the model, as these faults cause a dip in the AC Power.

The second approach is to detect anomaly by using the conventional classification models. For classification-based anomaly detection, five binary classifiers were trained, viz., LGR, SVM, MLP-based classifier, DT, and RF. It has been found that, overall, SVM performed better than all other classifiers, and after SVM, MLP provided better scores. Among all the models SVM model provided high performance in terms of F1- score and Accuracy. However, overall the classifier models did not give satisfactory results.

Next, three RNN-based sequential models were considered, Elman RNN, LSTM, and GRUs. In this chapter, AC power forecasting and fault detection using sequential models, RNN, LSTM, and GRU are also presented that can consider the temporal correlations. It has been observed that GRU produced the best results as compared to RNN and LSTM in terms of RMSE and R^2 Score for prediction of AC Power. When the models are used for threshold-based fault detection, GRU provided the highest recall while LSTM provided the highest precision. We further analysed the false positives and false negatives of GRU model. Higher number of false positives can be attributed to the error accumulation. The error accumulation can be reduced my reducing the noisy input during the period when fault exists. This will also help in reducing false negatives. Another solution is to apply a different approach such as bidirectional RNNs. As far as power prediction is concerned, all the models gave comparable results, while GRUs performed slightly better. For fault detection, GRU provided higher recall and LSTM gave higher F1-score.

TH-4096_186151013

Chapter 5

Contextual Approaches for Early Fault Detection

Contextual information can substantially influence the performance of predictive models. Although context-aware learning has been widely investigated in fields like computer vision [116], natural language processing [117], and recommender systems [118], its utilization in solar photovoltaic applications has received comparatively less attention. This chapter examines the role of contextual variables, including time of day, seasonal variations, and weather conditions, in improving fault detection for PV systems using machine learning (ML). Given that solar irradiance, which directly influences PV panel output, is affected by these contextual factors, integrating them can significantly improve detection accuracy [119]. In the exploratory analysis presented in Chapter 3, we also observed that seasons and time of day influence the power production. Motivated by these observations, we explore strategies for embedding such contextual features into ML-based fault detection frameworks to enhance performance and the reliability of solar PV systems.

This chapter begins by discussing various approaches for handling contextual features. These approaches are then used to train five different models, whose performances are subsequently

TH-4096_186151013

compared. Following this, we introduce a novel approach for incorporating contextual information into an LSTM network, aiming to enhance model performance. In the proposed method, the input weights are tuned based on the context during learning. We refer to this model as Context Sensitive-LSTM (CS-LSTM). The CS-LSTM is evaluated on two datasets, and the results are analysed and compared with those from existing approaches.

5.1 Strategies for Integrating Contextual Features

The concept of *context* is used extensively across many disciplines, often with different meanings depending on the application. In the domain of machine learning, we follow the definition of context and contextual features as described by Turney [120]. According to this view, the input variables used for training a predictive model can be broadly classified into three groups: (i) *primary features*, (ii) *contextual features*, and (iii) *irrelevant features*[120].

- **Primary features :** Primary features are those features that possess a direct and significant relationship with the target variable. These features contain the core information required for prediction and are capable of producing satisfactory predictive performance even when used independently. They contain strong, intrinsic relationships with the target variable and can yield good predictive performance even when used alone. In the context of solar PV power prediction, variables such as solar irradiance, module temperature, and historical power output can often be considered primary features because of their strong influence on the generated power.

- **Contextual features:** Contextual features provide supplementary information describing the circumstances under which the primary features are observed. Although they generally exhibit limited predictive power when used in isolation, they can substantially improve model performance when combined with primary features. In other words, contextual features do not have strong predictive power by themselves but enhance prediction when used alongside primary features. Contextual features help explain variations in the relationship between the target output and the input variables. For solar PV systems, examples of contextual features include season, time of day, weather conditions, and other environmental factors that influence the operating conditions of the plant.

TH-4096_186151013

- **Irrelevant features:** The features that do not contribute meaningful information to the prediction task are known as irrelevant features. These features neither exhibit a significant relationship with the target variable nor improve predictive performance when combined with other features. Inclusion of irrelevant features may increase model complexity, computational cost, and the risk of overfitting, thereby degrading the overall effectiveness of the learning algorithm. Irrelevant features provide no useful information for prediction even if they are combined with other features or used independently[120], [121].

The distinction between these feature categories is particularly important when developing context-aware predictive models. While primary features capture the fundamental characteristics of the system, contextual features provide additional information that can help the model adapt to varying operating conditions. Consequently, the effective integration of contextual information can lead to more robust and accurate predictions, especially in solar PV applications such as fault detection and power forecasting where environmental and temporal variations significantly influence system behaviour.

For example, in solar energy forecasting, where the goal is to predict AC power output of a solar PV system, primary features could include solar irradiance (W/m^2) and ambient temperature (°C). Solar irradiance is directly proportional to power output and thus serves as a strong, standalone predictor. Contextual features might include panel orientation, tilt angle, timestamp (time of day/day of year), or cloud cover index. For instance, solar irradiance of 800 W/m^2 may result in different power outputs depending on the panel orientation or the time of day (e.g., 8 A.M. vs. noon). These contextual features help the model interpret how effective a given irradiance level will be under current conditions. Irrelevant features could be unrelated metadata, such as panel serial number, which do not affect energy generation. In this framework, the context is defined by the set of contextual features. Even subsets of these, say just timestamp and cloud cover, define a particular context under which the model may interpret primary features differently. Understanding the role of context is critical in such applications where environmental and system-specific conditions can significantly affect the output even when primary variables appear similar. For instance, two PV systems might receive the same solar irradiance, but due to differences in tilt or weather conditions, their actual power output could vary widely. Properly integrating contextual features helps capture such nuanced dependencies, leading to more accurate and robust forecasting models. In [115], five different strategies are presented

TH-4096_186151013

for incorporating explicit contextual information into classification tasks. These strategies are: contextual normalisation, contextual expansion, contextual model selection (i.e., selection of a contextual classifier), adjustment of contextual classification, and contextual weighting [115], [121].

- Contextual normalisation adjusts primary features that vary with context by normalising them based on contextual information, thereby reducing their dependence on contextual fluctuations.
- Contextual expansion involves combining the additional context-related features with the primary features to create an extended feature space. This enriched feature set, which captures both the main and contextual information, is then used to train the learning model.
- The contextual classifier selection strategy follows a two-step approach: first, contextual features are used to select the most appropriate classifier from a pool of available models; and then the selected classifier is used to perform the final prediction based on the primary features.
- In contextual classification adjustment, the model first performs classification using only the primary features, and then applies contextual information to refine or adjust the initial prediction.
- In contextual weighting, the influence of each primary feature is adjusted based on contextual information, so that features most relevant in a given context receive higher importance during learning.

5.2 Model Development using Contextual Approaches

This section outlines the experimental setup used to evaluate the contextual feature-handling methods. It describes how different contextual approaches are integrated into model architectures, trained on the real dataset, and assessed using common metrics. Details of hyperparameters and performance criteria are also discussed. As outlined in Chapter 4, fault detection can be approached either as a regression problem or as a classification problem. In this work, a

TH-4096_186151013

regression-based model is adopted to detect faults. To evaluate the effectiveness of different context-handling strategies, five regression models were developed and compared: Linear Regression (LR), Multilayer Perceptron (MLP), Support Vector Regressor (SVR), Random Forest Regressor (RFR), and Long Short-Term Memory (LSTM) networks. Each model was trained under three distinct settings that reflect different levels of contextual integration. In the first setting, the models were trained using only the primary features, serving as the baseline for comparison. In the second setting, contextual normalisation was applied, where context-sensitive primary features were normalised with respect to contextual variables, to minimise their dependency on context. The third setting implemented contextual expansion, where contextual features were appended to the primary features, allowing the models to learn directly from the combined feature space. Additionally, a context-specific modelling strategy was explored by training four separate models, each corresponding to a distinct seasonal context. During inference, the appropriate model was selected based on the contextual condition (i.e., the current season) of the input data. The data preprocessing steps followed the procedure outlined in Chapter 3. Two contextual features, viz.,season and time of day were incorporated. The seasonal context was encoded numerically (with the first season labelled as 1 and so on), while the time of day was represented using cyclical encoding, where sine and cosine transformations were applied to capture its periodic nature. For all strategies except contextual normalisation, min–max normalisation is applied. In contextual normalization, the data normalisation is done using the following formulation [122], [121]:

$$v_i = \frac{x_i - \mu_i(\mathbf{c})}{\sigma_i(\mathbf{c})} \tag{5.1}$$

where the expected mean value of x_i is denoted by $\mu_i(\mathbf{c})$, and the expected variation of x_i is denoted by $\sigma_i(\mathbf{c})$ as a function of the context $\mathbf{c}$.

During training, model performance was optimized by tuning hyperparameters on a validation set. Table 5.1 presents the values of a selected subset of these hyperparameters[121], while the corresponding results are discussed in the subsequent subsection.

TH-4096_186151013

Table 5.1: Selected hyperparameters for various models

Model	Hyperparameter	Baseline Configuration	Context-Integrated Configuration
Linear Regressor (LR)	Fit intercept	True	True
Multilayer Perceptron (MLP)	Hidden layer sizes Epochs Activation Optimizer Learning rate	(50,100) 500 ReLU Adam Adaptive	(200) 200 ReLU Adam Adaptive
Support Vector Regressor (SVR)	Kernel	RBF	RBF
Random Forest Regressor (RFR)	Depth Number of estimators	25 200	20 200
Long Short-Term Memory (LSTM)	Sequence length Number of layers Hidden units Learning rate Epochs Optimizer	5 1 50 0.0003 100 RMSprop	10 2 200 0.001 100 Adam

TH-4096_186151013

Table 5.2: Performance comparison of different models based on RMSE

Name of Model	RMSE			
	Base Model	***Contextual Expansion***	***Contextual Normalisation***	***Contextual Model Selection***
Multilayer Perceptron (MLP)	0.0942	0.0950	**0.08155**	0.0949
Linear Regressor (LR)	0.1026	**0.0860**	0.0932	0.1010
Random Forest Regressor (RFR)	0.0658	**0.0589**	**0.0590**	0.0665
Support Vector Regressor (SVR)	0.0984	0.0940	**0.0803**	0.0935
Long Short-Term Memory (LSTM)	0.0811	**0.0801**	0.09678	0.1012

5.2.1 Results and Discussion

As regression-based fault detection is adopted, for each input sample, a regression model is used to estimate the expected AC power, which is then compared against the measured AC power to detect faults. We begin by discussing the results obtained from various regression models for predicting AC power. The results obtained from various regression models based on normalised data are presented in Table 5.2 [121].

The corresponding plot is shown in Figure 5.1. Here, the LSTM-based regression model was employed to predict AC power one time step (15 minutes) ahead.

Across all models, incorporating contextual information generally led to slightly improved prediction accuracy compared to the base models. The Random Forest Regressor (RFR) achieved the lowest RMSE overall, indicating its superior ability to capture nonlinear patterns in the data. Specifically, RFR performed best with Contextual Expansion (RMSE = 0.0589), closely followed by Contextual Normalisation (RMSE = 0.0590). Linear Regression (LR) and Sup-

TH-4096_186151013

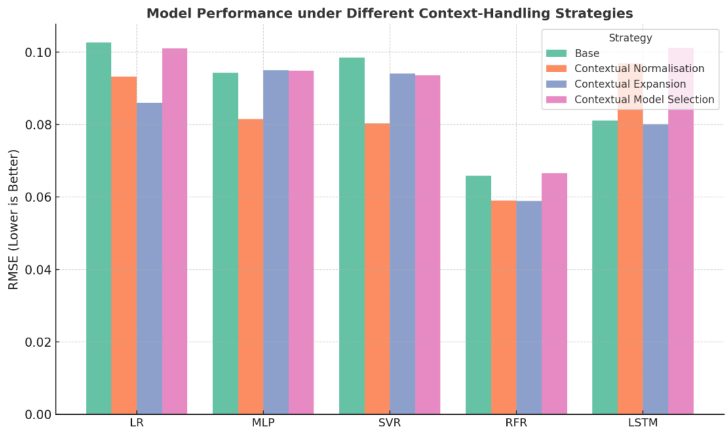


Figure 5.1: Performance of different regression models under different context-handling strategies

port Vector Regression (SVR) also benefited from context-based preprocessing, which reduced RMSE from 0.1026 to 0.0860 for LR under Contextual Expansion, and from 0.0984 to 0.0803 for SVR under Contextual Normalisation. The MLP model showed modest improvement with Contextual Normalisation (RMSE = 0.08155), while the LSTM model displayed inconsistent behaviour, its performance declined with contextual normalisation but improved slightly with contextual expansion. Overall, the analysis confirms that context-aware strategies enhance model reliability for most approaches, though the degree of improvement varies. The RFR stands out as the most robust and accurate model for AC power prediction, making it well-suited for downstream fault detection tasks.

The fault detection process relies on defining a threshold value to evaluate the deviation between the predicted and the measured AC power. In this work, the threshold is established based on the training error distribution. The training error values are modelled using a Gaussian distribution to estimate their mean (μ) and standard deviation (σ). Thresholds corresponding to 2σ and 3σ were tested and their performance compared. It was found that using 2σ as the threshold provided better results, as it led to fewer false negatives and false positives. Additionally, to further minimise false alarms, a fault condition is confirmed only when four consecutive samples exceed the specified deviation threshold. Over an 11-month period, faults were observed in 9 of the 16 inverters. For testing, we considered the full-day data corresponding to a fault in a specific inverter, along with the two preceding and two subsequent days. This ensured that the test dataset included both normal and faulty operating conditions. The times of actual fault occurrences were compared with the times at which the models detected these faults, with the goal of evaluating the system's capability for early fault detection—a critical requirement in PV

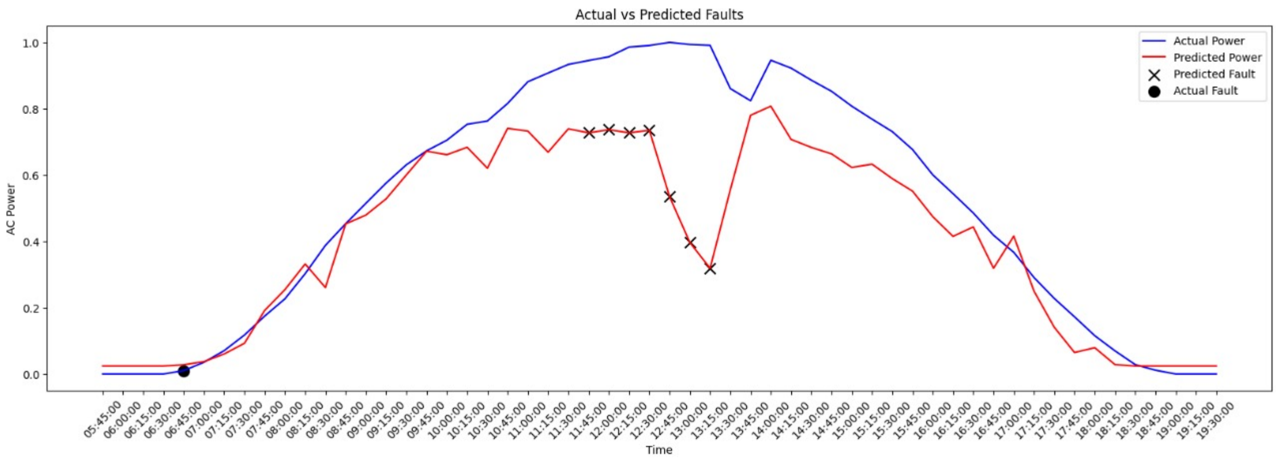


Figure 5.2: Actual versus predicted faults for Inv-4 using Contextual Model selection.

systems. Table 5.3 presents a detailed comparison between the actual fault occurrence times and the detected fault times, along with the models responsible for the detections. In the case of Inv-3, the initial fault was identified at 14:00. However, both the contextual expansion strategy and the contextual model selection strategy recognised the fault earlier, at 11:00. Instances of early fault detection by the system are highlighted in bold.

Out of a total of 13 faults, the system successfully detected 6 faults in advance, while the remaining 7 faults were identified later than their actual occurrence. Late detections were predominantly associated with faults occurring during early morning hours. Figure 5.2 illustrates a representative example of delayed fault prediction, where predicted power closely follows the actual power with minimal deviation, and power gradually increases in the early morning, reflecting normal conditions. Consequently, the model was unable to detect faults during these hours [121]. Addressing this limitation for early-morning fault detection will be a focus of future work.

5.3 Context Sensitive Long Short-Term Memory (CS-LSTM)

In the previous section, we discussed three approaches for handling context in fault detection. Building on these ideas, we will now introduce a novel model that extends the standard LSTM framework to explicitly incorporate contextual information. Before discussing the proposed

Table 5.3: Comparison between predicted and actual fault occurrence times

Date of Fault	**Inverter Number**	**Predicted Fault Time**	**Actual Fault Time**	**Methodology**
26-06-19	Inv-6	**11:00**	14:30	Contextual Expansion
05-07-19	Inv-3	**11:00**	14:00	Model Selection and Contextual Expansion
07-07-19	Inv-14	08:00	06:15	Contextual Expansion and Contextual Normalisation
09-07-19	Inv-9	12:00	06:30	Model Selection
10-08-19	Inv-7	08:30	07:45	Model Selection
10-08-19	Inv-8	08:30	07:00	Model Selection
16-09-19	Inv-7	**11:45**	14:45	Model Selection
18-09-19	Inv-5	**11:45**	14:45	Model Selection, Contextual Expansion and Contextual Normalisation
25-09-19	Inv-8	08:30	08:00	Model Selection, Contextual Expansion and Contextual Normalisation
01-10-19	Inv-3	10:45	07:00	Contextual Expansion
04-10-19	Inv-4	08:30	06:45	Contextual Normalisation
23-10-19	Inv-15	**09:00**	15:30	Contextual Expansion
07-02-20	Inv-15	**10:15**	12:00	Contextual Expansion and Contextual Normalisation

TH-4096_186151013

approach, we first provide an overview of the standard LSTM network, followed by a detailed discussion of the CS-LSTM architecture.

In 1997, Hochreiter and Schmidhuber introduced the LSTM architecture [123] as an enhancement to traditional RNNs. The main reason for its creation was to address the vanishing and exploding gradients, which impede the efficient training of traditional RNNs [97],[18]. The distinguishing feature of an LSTM network is its memory cell, which enables the model to selectively retain, discard, and propagate information over long temporal sequences. This selective information processing is accomplished through three gating units, namely the forget gate, input gate, and output gate[97],[113]. Together, the three gates decide which information needs to be retained, modified, removed, or passed on to future time steps. This controlled information flow enables the LSTM architecture to effectively model long-term temporal dependencies in sequential data [96]. Thus, the collective action of these gating mechanisms regulates the memory cell by selectively updating, retaining, discarding, and propagating information across successive time steps.

The long-term memory state from the previous cell, denoted as $\mathbf{c}_{t-1}$, is propagated through the current LSTM cell, where a portion of the stored information is selectively forgotten under the control of the forget gate, while new information is incorporated through an addition operation. Thus, at every time step, the LSTM cell updates its memory by partially retaining the past information and integrating new relevant information, resulting in the updated long-term state $\mathbf{c}_t$ at time (t). Subsequently, the long-term state $\mathbf{c}_t$ is passed through a hyperbolic tangent (tanh) activation function, and then the resulting signal is modulated by the output gate to produce $\mathbf{h}_t$, the short-term (hidden) state. This hidden state $\mathbf{h}_t$ also forms the output $\mathbf{y}_t$ of the LSTM unit. To compute the gating signals and state updates, the LSTM cell processes both the previous hidden state $\mathbf{h}_{t-1}$ and the current input vector $\mathbf{x}_t$ through four separate fully connected layers corresponding to the different components of the architecture[97].

As illustrated in Figure 5.3, the layer producing $\tilde{\mathbf{c}}_t$, often referred to as the input node, serves as the main transformation layer, analysing the combined effect of $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$. The long-term state incorporates a portion of the output from this input node. The remaining three layers function as gating mechanisms: the input gate, forget gate, and output gate, all employing sigmoid activation functions, which constrain their outputs to the range [0, 1]. These gate outputs are applied via element-wise multiplication (Hadamard product) to control the flow of information.

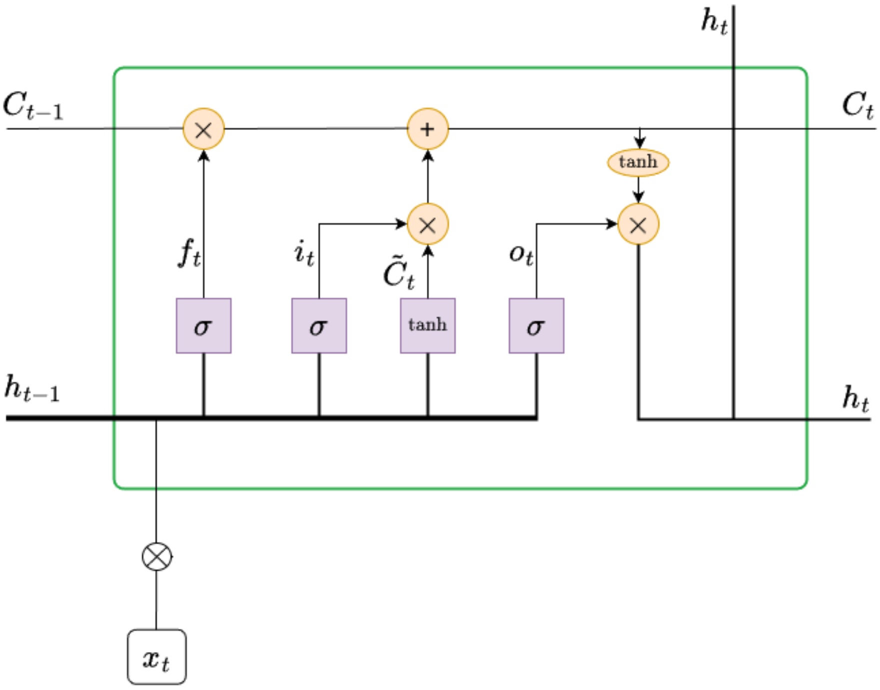


Figure 5.3: LSTM Cell

A gate output of 0 completely blocks the passage of information, while an output of 1 allows full passage[113]. Intermediate values between 0 and 1 determine the fraction of information transmitted through each gate. Specifically, the forget gate $\mathbf{f}_t$ determines which parts of the previous cell state $\mathbf{c}_{t-1}$ should be discarded, the input gate $\mathbf{i}_t$ regulates the addition of new candidate memory $\tilde{\mathbf{c}}_t$ to the cell state, and the output gate $\mathbf{o}_t$ decides how much of the updated cell state should influence the hidden state and output[18],[113].

The updates to the cell state $\mathbf{c}_t$ and hidden state $\mathbf{h}_t$ are given by the following equations [18]:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \\
\tilde{\mathbf{c}}_t &= tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \\
\mathbf{h}_t &= \mathbf{o}_t \odot tanh(\mathbf{c}_t)
\end{aligned} \tag{5.2}$$

In the above formulation, $\mathbf{x}_t \in \mathrm{Re}^{n\times 1}$ denotes the input vector at time t; The vectors $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t \in \mathrm{Re}^{m\times 1}$ denote the activations of the input, forget, and output gates, respectively; $\mathbf{c}_t \in \mathrm{Re}^{m\times 1}, \tilde{\mathbf{c}}_t \in$

$\mathrm{Re}^{m\times m}$ denote the cell state and cell input activation,respectively; The hidden state of the network is denoted by $\mathbf{h}_t \in \mathrm{Re}^{m\times 1}$; The matrices $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_c, \mathbf{W}_o \in \mathrm{Re}^{m\times n}$ define the weights associated with the current input vector $\mathbf{x}_t$; $\mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_c, \mathbf{U}_o \in \mathrm{Re}^{m\times m}$ are the weight matrices linked to the previous hidden state $\mathbf{h}_{t-1}$; The bias terms are represented by $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_g, \mathbf{b}_o$. Here, n and m specify the dimensionality of the input vector and the number of hidden units in the LSTM layer, respectively.

Traditional LSTM networks exploit temporal dependencies in data, making them naturally suitable for time-series modelling. The proposed model, termed Context-Sensitive LSTM (CS-LSTM), goes a step further by adapting the LSTM weights according to the specific context. This is achieved through the contextual weighting strategy, which allows the network to dynamically emphasise relevant patterns depending on the prevailing context. The proposed CS-LSTM architecture is fundamentally a Recurrent Neural Network (RNN) that uses LSTM units as its core building blocks, with an added capability to explicitly incorporate contextual information. The model processes two categories of input features: primary features and contextual features. By integrating contextual features, CS-LSTM allows the network's weights for primary features to be modulated according to the context.

The proposed CS-LSTM formulation extends the standard LSTM architecture by introducing a contextual input $\mathbf{z}_t$ in addition to the primary input $\mathbf{x}_t$. The contextual variables are used to modulate the weights connected to the input units. The structure of a CS-LSTM cell is illustrated in Figure 5.4.

While the computation of the cell state $\mathbf{c}_t$ and the hidden state $\mathbf{h}_t$ remains identical to that of a conventional LSTM, the gate activations in CS-LSTM are computed using weights that are functions of the contextual features, as expressed below [18], [124], [125]:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i(\mathbf{z}_t)\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f(\mathbf{z}_t)\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \\
\tilde{\mathbf{c}}_t &= tanh(\mathbf{W}_c(\mathbf{z}_t)\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o(\mathbf{z}_t)\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \\
\mathbf{h}_t &= \mathbf{o}_t \odot tanh(\mathbf{c}_t)
\end{aligned}
\tag{5.3}
$$

TH-4096_186151013

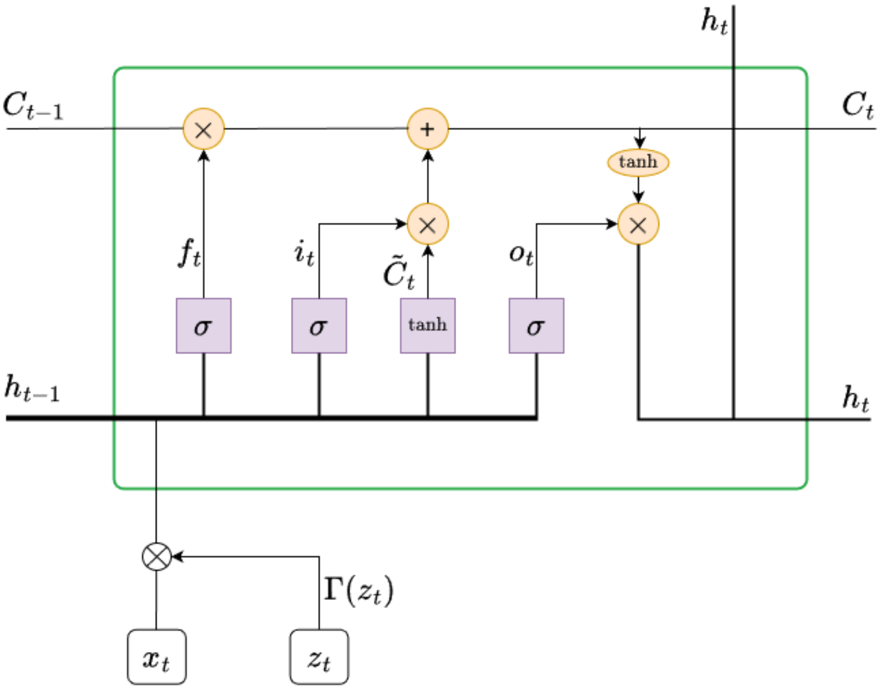


Figure 5.4: CS-LSTM Cell

Let us denote the weights of the input connections represented as a parameter matrix $\mathbf{W}(\mathbf{z}_t)$ having a dimension of $m \times n$ (subscripts are omitted for simplicity of notation). Each element of the parameter matrix denoted by $\mathbf{W}(z_t)$ is defined as shown below [113]:

$$w_{ki} = \mathbf{B}_{ki}\boldsymbol{\Gamma}(z_t), \quad k = 1,\ldots,m, \quad i = 1,\ldots,n \tag{5.4}$$

In the above formulation, the context-dependent weights are parameterized through a coefficient vector represented by $\mathbf{B}_{ki} = [\beta_{ki1}, \beta_{ki2}, \ldots, \beta_{kil}]$. The vector of basis functions, denoted by $\boldsymbol{\Gamma}(z_t) = [\gamma_1(z_t), \gamma_2(z_t), \ldots, \gamma_l(z_t)]^T$ provides a functional representation of the contextual input and serves as a design-specific hyperparameter of the model. The choice of these basis functions is a design consideration and can be treated as a model hyperparameter. The coefficient vector $\mathbf{B}_{ki}$ characterizes how the corresponding network weights vary as a function of the contextual variables. To simplify the representation, a coefficient matrix $\mathbf{B}$ can be constructed by concatenating the vectors $\mathbf{B}_{ki}$ associated with each input feature. Consequently, the k-th row of $\mathbf{B}$ is given by [124]

$$\mathbf{B}_k = [\mathbf{B}_{k1}, \mathbf{B}_{k2}, \cdots, \mathbf{B}_{kn}], \quad k = 1, 2, \ldots, m, \tag{5.5}$$

Here, $\mathbf{B}$ has dimensions $(m \times nl)$, where m denotes the number of hidden units, n represents the number of input variables, and l corresponds to the number of basis functions used to model the contextual information. This matrix serves as a structured representation of the relationship between the network parameters and the contextual variables. Specifically, the elements of $\mathbf{B}$ capture how the effective weights of the model vary under different contextual conditions, thereby enabling the network to adapt its behaviour according to changes in the operating environment. This matrix provides a compact representation of the contextual dependencies embedded within the model parameters.

The introduction of $\mathbf{B}$ provides a compact and mathematically convenient framework for incorporating contextual dependencies into the model. Instead of treating the influence of context separately for each parameter, the contextual effects are aggregated within a single matrix representation, leading to a more efficient formulation. As a result, the original expressions given in (5.3) can be reformulated in a concise matrix form by employing the matrix $\mathbf{B}$ as described in [18], [124], [125], simplifying both the notation and the implementation of the model. Furthermore, this representation highlights the role of contextual information in modulating the network parameters and facilitates the derivation of learning algorithms for estimating the contextual coefficients. Following the formulation presented in [18], [124], [125], the equations in (5.3) can therefore be expressed in the following compact form:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{B}_i(\mathbf{x}_t \otimes \boldsymbol{\Gamma}(\mathbf{z}_t)) + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{B}_f(\mathbf{x}_t \otimes \boldsymbol{\Gamma}(\mathbf{z}_t)) + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \\
\tilde{\mathbf{c}}_t &= tanh(\mathbf{B}_c(\mathbf{x}_t \otimes \boldsymbol{\Gamma}(\mathbf{z}_t)) + \mathbf{W}_{hc} \mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{o}_t &= \sigma(\mathbf{B}_f(\mathbf{x}_t \otimes \boldsymbol{\Gamma}(\mathbf{z}_t)) + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o)
\end{aligned} \tag{5.6}$$

In the above equations, the operator $\otimes$ represents the Kronecker product. The coefficient vectors $\mathbf{B}_{ki}$, each containing m elements, are learned during the training phase in a manner analogous to the parameter estimation procedure employed in conventional LSTM networks. This involves defining an appropriate objective (loss) function and updating the model parameters using the Backpropagation Through Time (BPTT) algorithm, which enables the network to capture temporal dependencies present in sequential data. The gradients obtained through BPTT are subsequently utilized to adjust the contextual coefficient vectors so that the prediction error is minimized. To efficiently optimize the model parameters, gradient-based optimization techniques such as Stochastic Gradient Descent (SGD), RMSProp, or Adam may be employed, depending on the convergence and computational requirements of the application [125].

5.3.1 Case Study I: Real Data from 10 MW Grid-Connected PV System

The initial set of experiments was conducted using the real dataset from the operational PV plant as described in Chapter 3. The PV system investigated in this study is a grid-connected PV plant with a capacity of 10 MW, designed and developed by Azure Power (Haryana) Pvt. Ltd., which is operational in the state of Gujarat in India. This was the largest solar PV plant in the country at the time of commissioning and became operational in the year 2011. We considered LSTM as the base model for comparing the results of CS-LSTM.

5.3.1.1 Model Development

For model development, the available data was partitioned into training and validation subsets. Specifically, from the normal operating data of an inverter, the last s samples were set aside for validation, while the remaining samples were used for model training. It is important to note that s was chosen such that it is an integer multiple of the time window (i.e., sequence length) to ensure compatibility with the sequential learning framework, thereby preserving complete input sequences during validation. Also, observations from each inverter were included for validation. The hyperparameters and their corresponding ranges considered for model tuning are presented in Table 5.4. The network architecture comprised a single recurrent layer with a fixed batch size of 128. Second-order polynomial basis functions were employed to represent the contextual inputs. The RMSProp optimizer was used for optimization. Table 5.5 shows the final models that have been selected based on the validation error.

TH-4096_186151013

Table 5.4: **Hyperparameters and range**

Hyperparameters	**Range**
Number of hidden units	10-30 step size 5
Sequence length	5-15 step size 5
Learning rate	loguniform($1e-5$, $1e-3$)

5.3.1.2 Results and Discussion

Once the model is trained for the estimation of AC power, we use it to detect the faults. We used the two developed models, LSTM and CS-LSTM, for AC power prediction, where it is predicted one time step (15 minutes) ahead. The validation results achieved from both the models are close in terms of Mean Squared Error (MSE) with values of 0.0011 for the LSTM and 0.0015 for the CS-LSTM. Figure 5.5 presents the first 250 samples of the AC power predicted by the CS-LSTM model alongside the actual AC power measurements for an inverter (Inv-12) which was chosen randomly.

As discussed earlier, the second stage of fault detection involves comparing the predicted AC power with the measured values. For this comparison, a threshold value is necessary. This threshold is derived from the training error, which is modelled using a Gaussian distribution. From this distribution, the mean (μ) and standard deviation (σ) are obtained, and a value of $3*\sigma$ is selected as the threshold experimentally. The test dataset includes both normal and faulty conditions. According to the fault report, during the 11-month observation period, faults were recorded in 11 inverters (e.g., Inverters 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, and 15). For each inverter, the data corresponding to the full month in which the fault occurred was used for testing. The performance of the model was evaluated using three metrics: Accuracy, True Positive Rate (TPR), and False Positive Rate (FPR). The results for both LSTM and CS-LSTM models are

Table 5.5: Performance comparison of LSTM and CS-LSTM

Performance Metric	**LSTM**	**CS-LSTM**
Accuracy (%) $\left(\frac{TP}{\text{Total samples}}\right)$	93.2	96.9
TPR (%) $\left(\frac{TP}{TP+FN}\right)$	44.0	45.8
FPR (%) $\left(\frac{FP}{TN+FP}\right)$	6.2	2.6

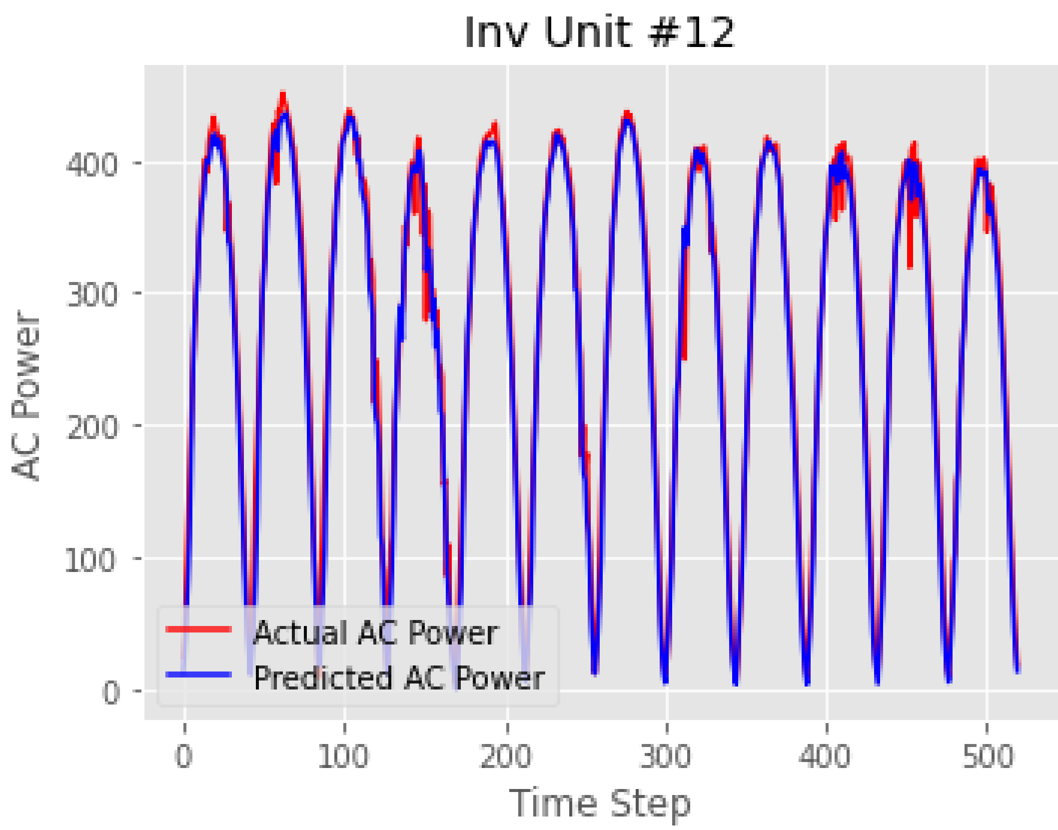
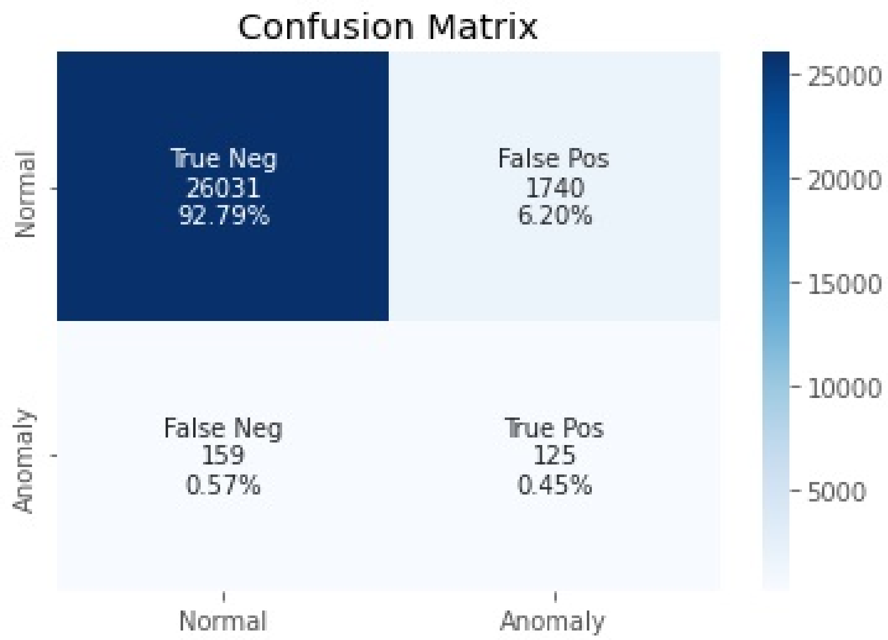


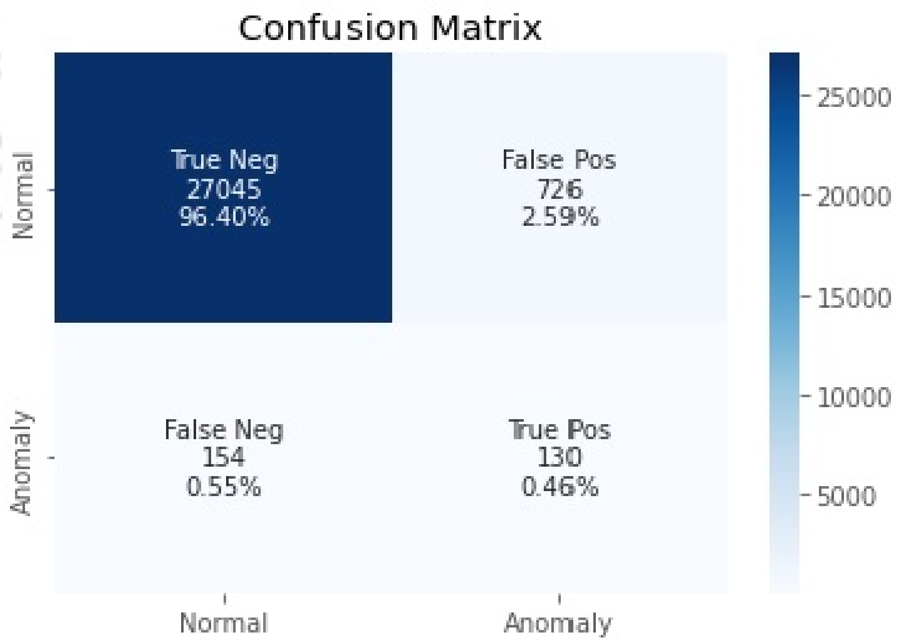
Figure 5.5: Comparison of measured and Estimated AC Power of Inv 12 with CS-LSTM

presented in Table 5.5, with their corresponding confusion matrices illustrated in Figure 5.6. As shown in Table 5.5, the CS-LSTM model outperformed the baseline LSTM model across all performance metrics. Figure 5.7(a) shows month-wise distribution of false negative (FN) and false positive (FP) samples for the CS-LSTM model. There are no FN samples during the months of April, June, November, and February. In May, August, September, and October, the number of false negatives is very small (fewer than 10). The month of July shows the highest contribution to FN samples. The majority of FP cases occur during the period from June to October [18]. Figure 5.7 (b) presents the distribution of FP and FN samples inverter-wise. For Inv-3, both FP and FN counts are high, and the log report confirms that this inverter was faulty in July. Similarly, Inv-7 exhibits the highest number of FPs, corresponding to faults recorded in April, August, September, October, and November.

Figure 5.8 (a) presents the AC power profile of Inv-3 for the first 15 days of July, with a similar pattern observed throughout the month. The figure reveals continuous fluctuations in AC power, lacking the usual smooth pattern of gradual rise and fall. One possible cause for this irregularity could be partial shading from cloud cover. However, according to the error log, the inverter fault

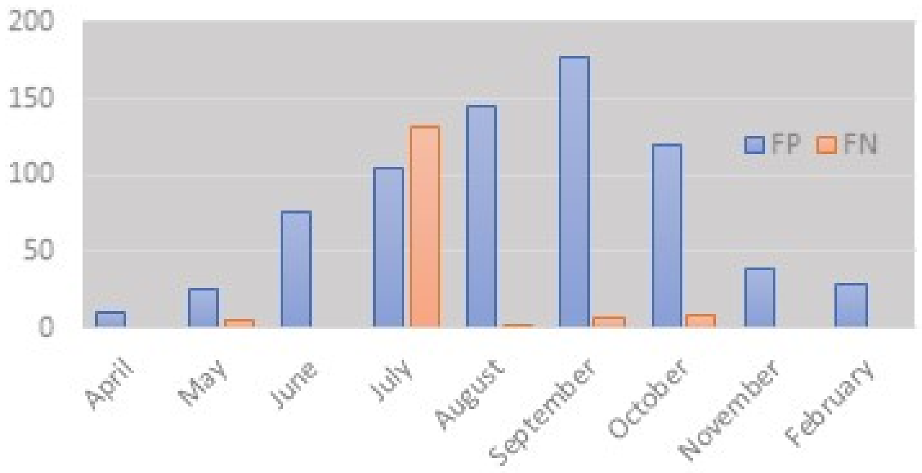


(a) LSTM

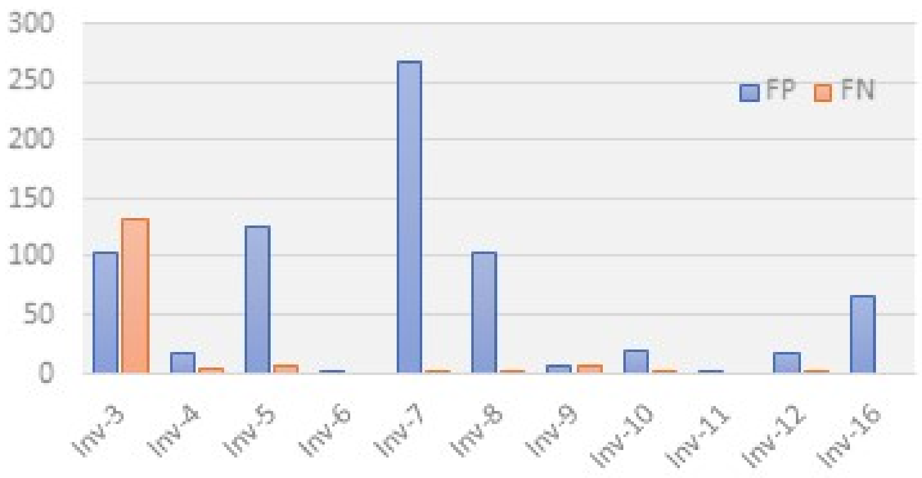


(b) CS-LSTM

Figure 5.6: Confusion Matrices



(a) Month-Wise



(b) Inverter-Wise

Figure 5.7: Distribution of FP and FN data samples from CS-LSTM by month and inverter

was reported only for three days in July. Because of these persistent fluctuations, the model classified a larger number of samples as positive during this month. A comparable behaviour is noted for August, September, and October as well. Figure 5.8(b) illustrates a single day of Inv-3 data from July. The green samples in the plot represent instances detected by the model as faulty, though they are absent in the error log. While these points show clear deviations from the expected AC power, the deviation could result from partial shading rather than an inverter malfunction. As only the log-reported samples are treated as anomalies in the experiments, instances like those shown in Figure 5.8(b) contribute to the false positive rate (FPR).

Figures 5.9(a) and (b) depict the false negative samples for Inv-3 recorded over two days in July, representing instances where the model failed to identify faults. As seen in the figures, the difference between the actual and predicted AC power is minimal, especially during the morning (before 10:00) and evening (after 15:00) periods. Because of this small deviation, the model was unable to classify these instances as faulty. The power output during these hours shows a smooth and gradual rise and fall, leading the model to interpret the samples as normal. These types of samples contribute to the lower True Positive Rate (TPR) observed in the model's performance.

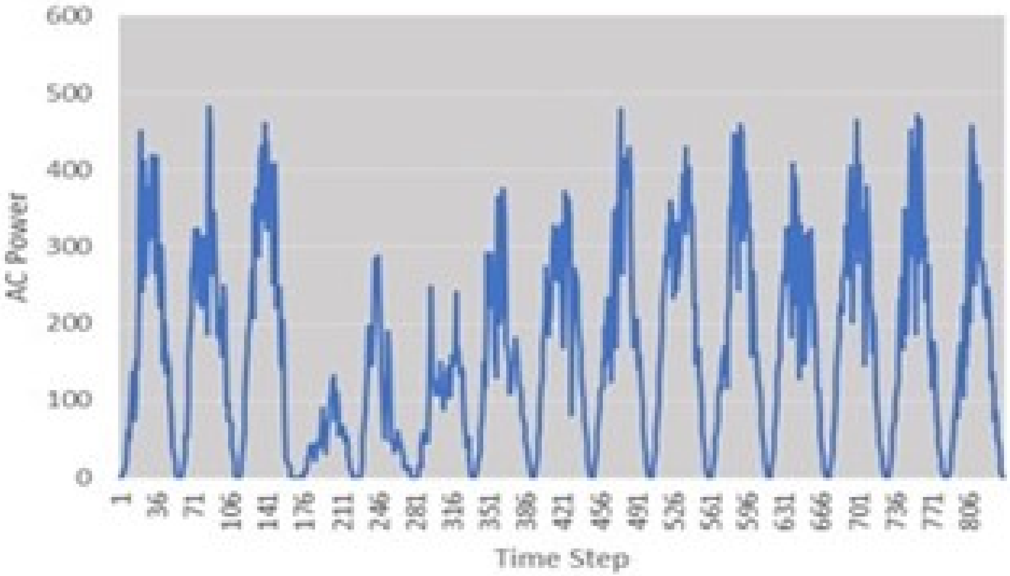
5.3.2 Case Study II: Benchmark Data from PV System Installed in Lab

For the next set of experiments, we use the data generated from the PV system installed in the Laboratory of Innovation and Technology in Embedded Systems and Energy (LIT), Universidade Tecnológica Federal do Paraná-UTFPR, Brazil [109]. The dataset description and analysis were discussed in Chapter 3.

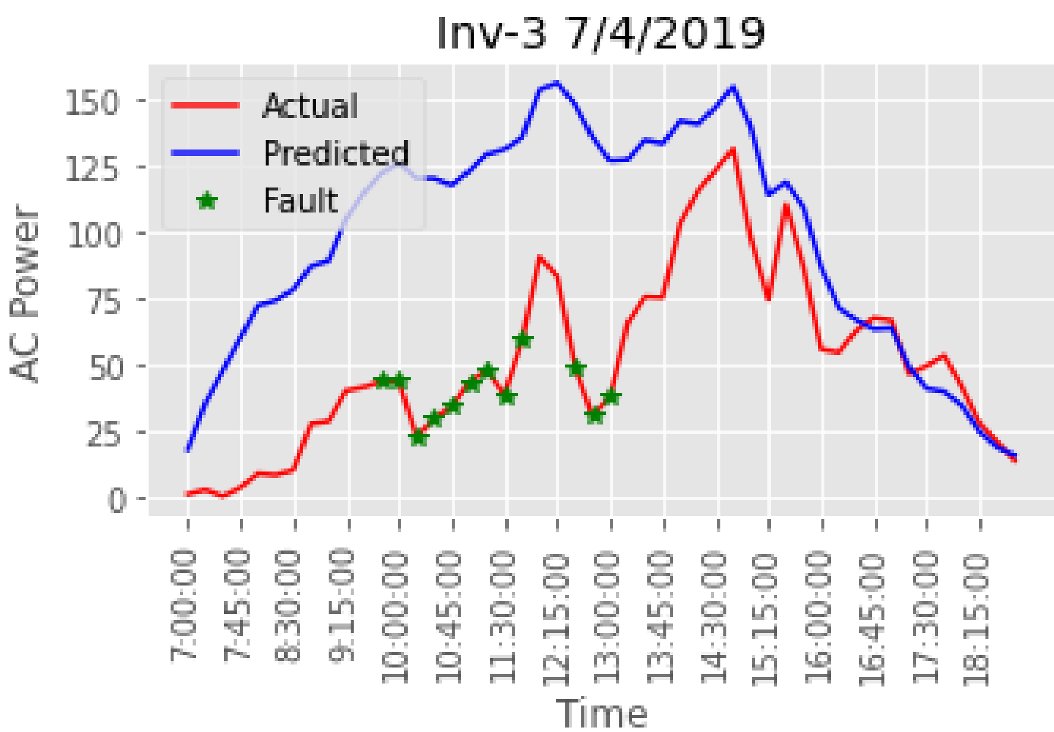
5.3.2.1 Model Development

For model development, two ambient features were chosen based on the correlation index between the variables and the power output of each string [109], as listed in Table 5.6. The contextual feature, Time of the Day, was transformed into the number of seconds since midnight and represented in a cyclical format. Prior to training, the data was preprocessed using Min–Max normalization to ensure uniform scaling across features.

TH-4096_186151013



(a) AC power first 15 days in July



(b) AC power on a selected day

Figure 5.8: AC power of Inverter 3 from July

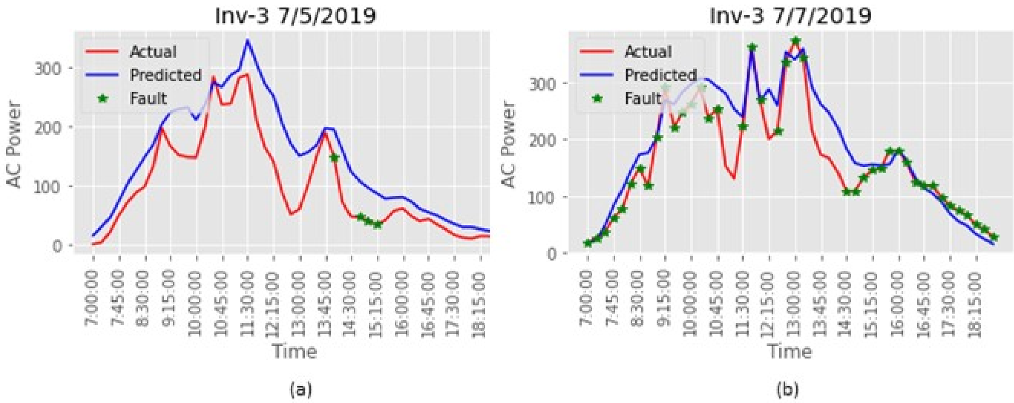


Figure 5.9: False Negatives for selected 2 days

Table 5.6: Selected Features

Input features	Irradiance, PV temperature, and past DC power $(t-k:t-1)$
Context feature	Time of the day
Output (target feature)	DC Power (t)

The predictive models estimated the DC power at the next time step (t) using the historical values of DC power along with other relevant features. Based on the difference between the predicted and actual outputs at each time step, a data sample is classified as either faulty or normal. For this purpose, five modelling approaches were employed: Linear Regression (LR), Random Forest Regressor (RFR), Artificial Neural Networks (ANNs), LSTM, and CS-LSTM. Figure 5.10 shows the combinations of the predictive models and fault detection methods that we considered for this dataset. In the given dataset, string 1 does not have faults apart from partial shadowing. We considered string 2 for our fault detection models. Further, string 1 does not influence string 2, so two separate fault detection models can be developed for each string. Here we are focusing on the model for string 2. As the dataset consists of 16 days of data, it is partitioned into 11 days of training data, 2 days of validation data, and 3 days of test data. The data samples where the irradiance is less than 100 are removed. Table 5.7 shows the number of data samples for each of the partitions after removal of low radiance samples. To train the predictive models, the faulty samples are removed. The distribution of train, validation, and test data after removing the faulty samples is shown in Table 5.8.

We developed two sets of models, the first set does not take into account the context feature,

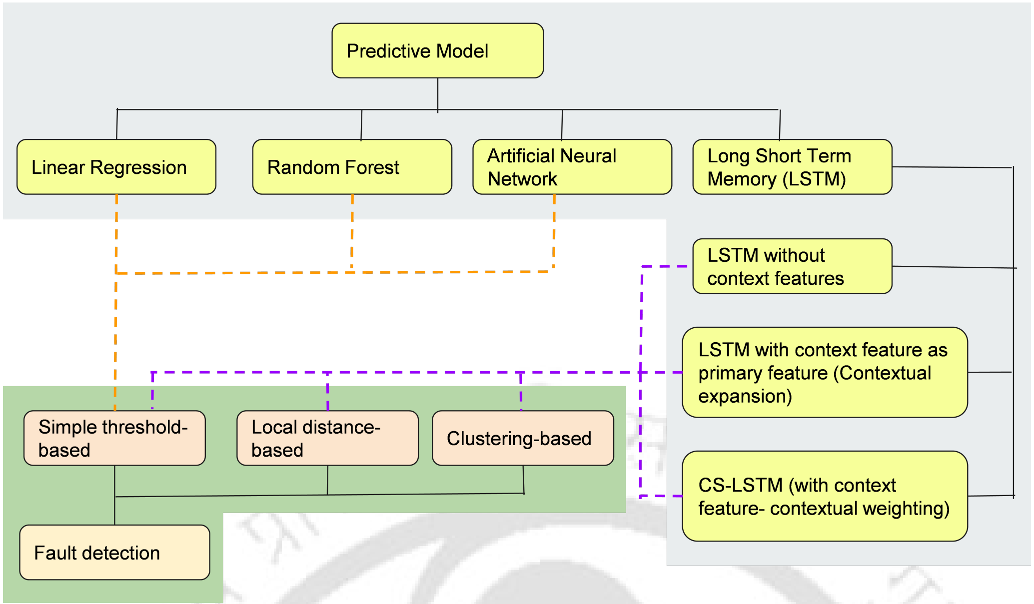


Figure 5.10: Various combinations of predictive models and fault detection methods

which is 'Time of the day', whereas the second set of models considers the context feature. For each of the models, the hyperparameters are selected using the validation set. Table 5.9 presents the hyperparameters used for tuning the various models. For LSTM and CS-LSTM, the number of layers is fixed to 1, and for the context integration, a polynomial basis function of degree 2 is considered. The best hyperparameters obtained after tuning the models and the prediction results are shown in Table 5.10.

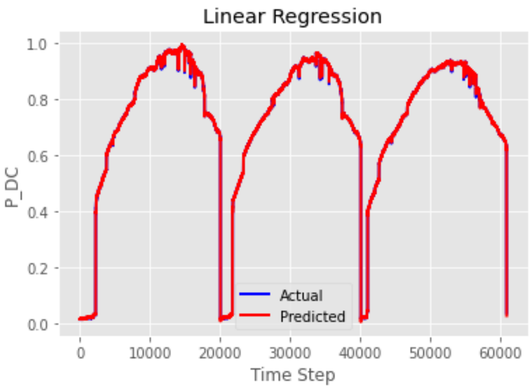
Table 5.7: Data partitions

Data	**No. of samples**
Training data	354696 (∼68.5%)
Validation data	66071 (∼12.5%)
Testing data	95191 (∼18.7%)
Total	515958

5.3.2.2 Power Prediction Results

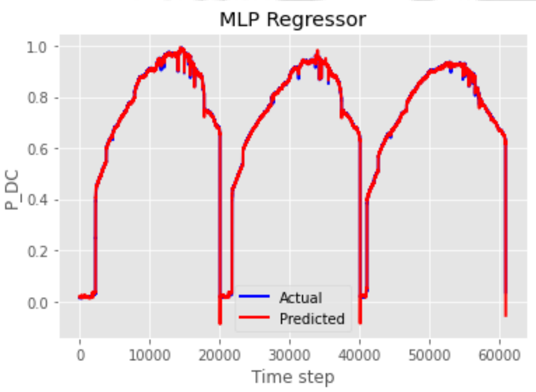
The predicted power is obtained with the test dataset, which only consists of normal data samples. Figure 5.11 shows the actual and predicted power using the test data for all the models, considering the best RMSE values.

From Table 5.10 and Figure 5.11, it is apparent that the models without the context features

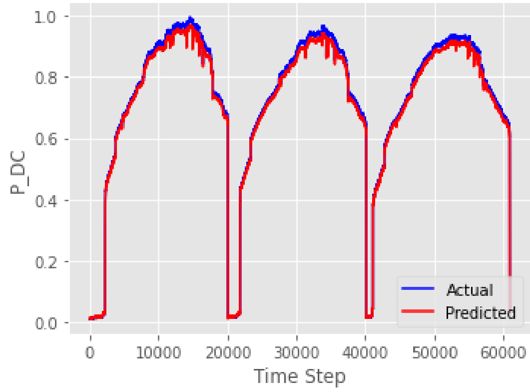


(a) Linear Regressor

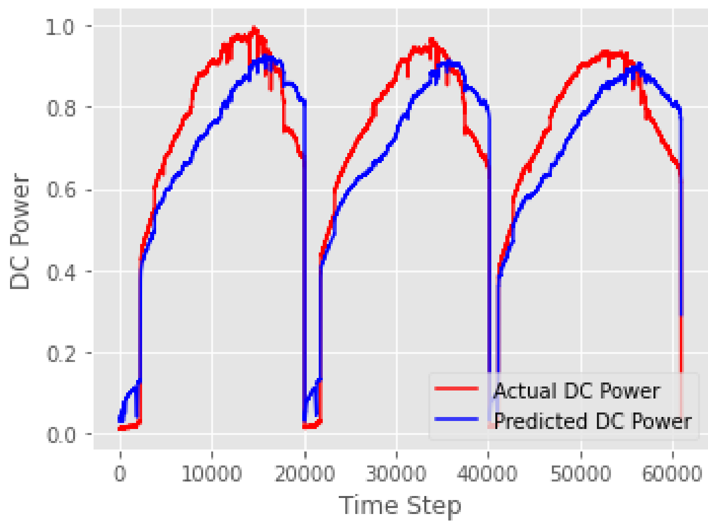
(b) Random Forest Regressor



(c) MLP Regressor



(d) LSTM



(e) CS-LSTM

Figure 5.11: Actual versus predicted power.

Table 5.8: Data without faulty samples

Normal Data	# samples
Training	209989
Validation	38269
Testing	60995

perform better compared to the models with context features. Further, it is interesting to see that linear regression is sufficient to capture the relation between input features and target with the least RMSE score. Also, adding context features to the classical models (LR, RFR) does not improve the performance. On the other hand, the CS-LSTM model performs better than LSTM with contextual features.

Table 5.9: Hyperparameter values

Model	Hyperparameter	Value
Random Forest Regressor (RFR)	Number of estimators	50
	Maximum depth	$\{20, 30, 40, 50\}$
ANN	Number of hidden units	$\{10, 20, 30, 40, 50\}$
	Maximum iteration	100
	Batch size	200 (default)
LSTM and CS-LSTM	Number of hidden units	$\{10, 15, 20, 25, 30\}$
	Sequence length	$\{2, 5, 10\}$
	Batch size	128
	Learning rate	loguniform(1×10^{-5}, 1×10^{-3})
	Optimizer	SGD, Adam, RMSProp

5.3.2.3 Fault detection

After developing the prediction models, the next step is to integrate an approach that can measure the deviation between predicted and actual power and indicate if the sample is faulty. We implemented three approaches: simple threshold-based, local distance-based, and clustering-based fault detection.

TH-4096_186151013

Table 5.10: Power prediction results (Test data without faults)

Model	**Hyperparameters**	**RMSE (normalised) (Without Context features)**	**RMSE (normalised) (With Context features)**
LR	-	0.0048	0.0049
RFR	Number of estimators = 30	0.0160	0.0164
ANN	Number of hidden layers = 30	0.0055	0.0079
LSTM	Number of hidden units = 20 Sequence length = 5 Learning rate = 0.0007	0.0136	0.2128
CS-LSTM	Number of hidden units = 20 Sequence length = 5 Learning rate = 0.007 Optimizer = RMSprop	-	0.0993

TH-4096_186151013

- Simple threshold-based fault detection: In this approach, based on the training error, the mean and the standard deviation (σ) is identified. Then the threshold is set as (3σ). In the test set, the error or deviation for each sample is calculated, and if it is more than the threshold, it is considered as a faulty sample.

- Local distance-based fault detection: First, for the errors, a window of size k is fixed, then for each data point (e_t), an anomaly score is determined depending on the sum of distances from e_t to all other data points in the window as shown in equation 5.7,

$$\text{score}(e_t) = \frac{1}{1+\exp(-d_t)} \tag{5.7}$$

  Where $d_t = \sum_{i=1}^{k} |e_t - e_i|$ is the sum of distances from e_t to all other points within the window. Note that the score can take the values between 0.5 and 1 as d_t will always be positive.

- Clustering-based fault detection: In the clustering-based approach, we use the class labels. First, the prediction errors from the training data are partitioned into two groups: errors belonging to the normal class and errors belonging to the fault class. Each group is clustered further to obtain the error patterns in normal and faulty prediction errors. For each cluster, we have the mean and standard deviation, which are used to predict the membership of the test prediction errors. For a prediction error e_t , the membership degree (τ_i) to each of the clusters is computed as:

$$\tau_i = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{e_t - \mu_i}{\sigma_i}\right)^2\right) \tag{5.8}$$

  Finally, the data point is assigned a label (Fault or normal) based on the maximum value of membership degree.

5.3.2.4 Fault Detection Results

Table 5.11 shows the results obtained from the three predictive models discussed previously. For the three models (LR, RFR, ANN), with simple threshold approach, the threshold is varied from

Table 5.11: Fault detection performance using simple threshold-based approach

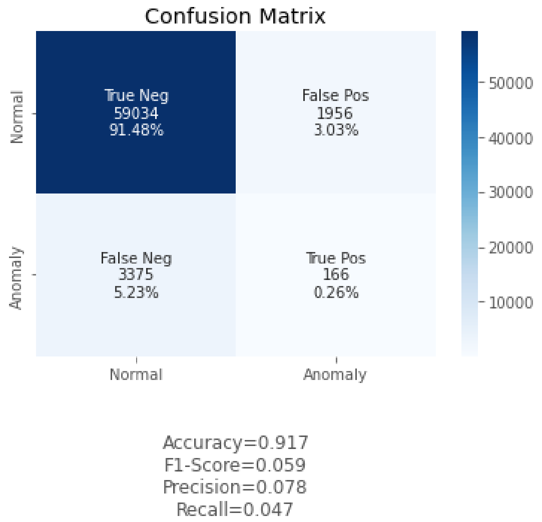
Predictive Model	Detection Approach	Accuracy (TP / Total Samples)	Recall (TPR) TP/(TP+FN)	Precision TP/(TP+FP)	F1-score
Linear Regression(LR)	Simple threshold	0.917	4.7	3.2	0.059
Random Forest Regressor(RFR)	Simple threshold	0.781	42.4	1.9	0.175
ANN	Simple threshold	0.895	1.2	5.3	0.012

($3\times\sigma$) to ($0.5\times\sigma$). The best results are achieved with ($0.5\times\sigma$). The corresponding confusion matrix is shown in Figure 5.12. As can be observed from the results presented in Table 5.11, the first three models are not able to capture the error deviation for fault detection, and therefore, the F-1 score is very low. The problem with these models is that they fit the data so well, evident from their RMSE values, that the variance in the errors is very low, as shown in Figure 5.13. It can be observed that the tail probability is very low.

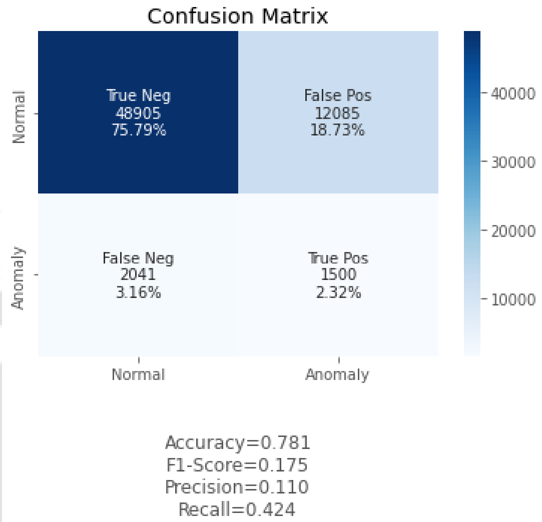
Considering the results from LR, RFR, and ANN with simple thresholding, we did not further continue with these models. Next, we considered LSTM and CS-LSTM models with fault detection approaches. Table 5.12 shows the results obtained from these models and Figure 5.14 shows the confusion matrices. The confusion matrices show the results of the best combination of predictive model and fault detection approach. For simple threshold-based approach, the threshold parameter is varied from ($2\times\sigma$) to ($0.5\times\sigma$), for the clustering-based approach, the number of clusters is varied from 5 to 15. Finally, for the distance-based approach, the window size of 10 minutes is considered, and the threshold is varied from 0.7 to 0.9. The best parameter values of each of the approaches is presented in Table 5.12 alongside the approach. It can be observed from Table 5.12 that for all the models, clustering-based fault detection provided the best F1-score. If the performance of LSTM and CS-LSTM is observed, the recall is comparable; however, the precision in LSTM is much higher compared to CS-LSTM. This means there are fewer false positives in the case of LSTM.

Figure5.15 highlights the portion of the plot where fault samples are detected, and the count of such faulty samples is more than 300 (5 mins). It is apparent from the plots that LSTM with clustering highlights most of the faulty regions. The small periods of normal operation in between the faults and also in between the partial shadowing could be identified by LSTM whereas CS-LSTM could not identify these as normal data. However, it is interesting to observe that both LSTM and CS-LSTM could do early detection of faults as indicated in Figure5.15(a)

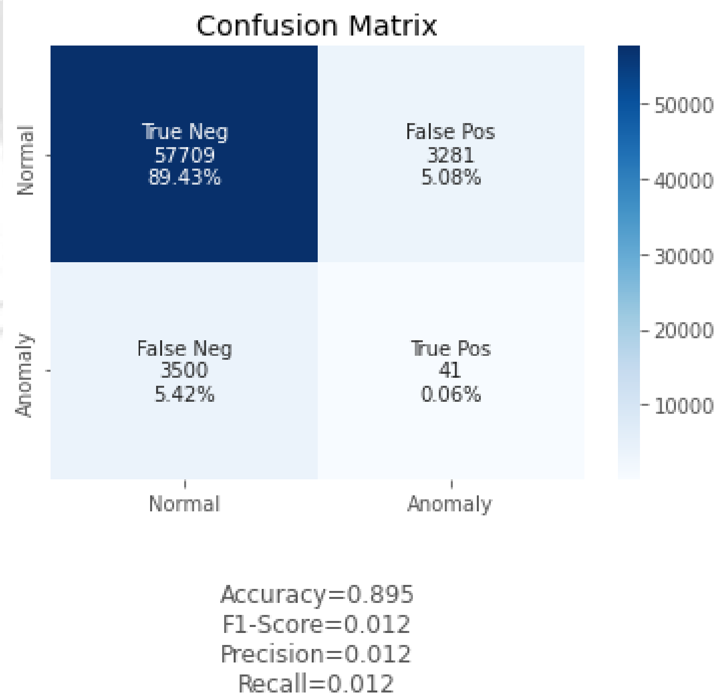
TH-4096_186151013



(a) LR



(b) RFR



(c) ANN

Figure 5.12: Confusion matrices for LR, RFR, and ANN

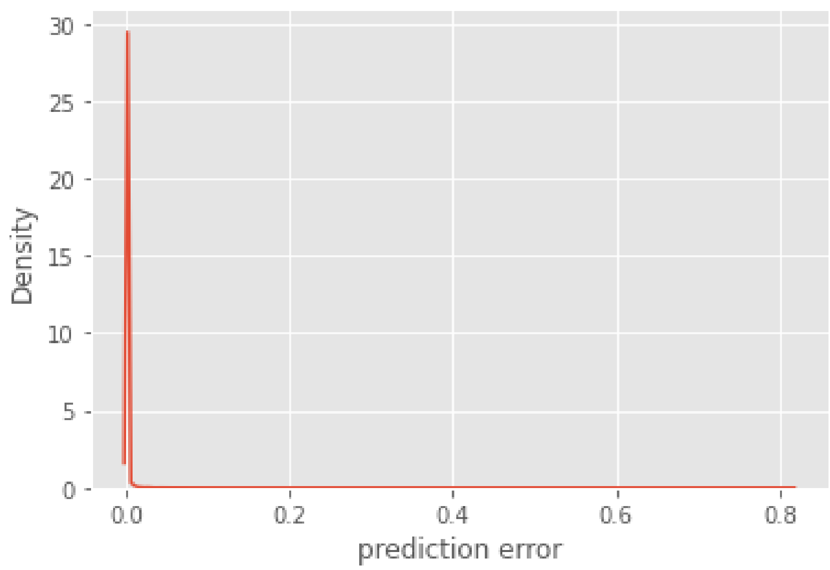
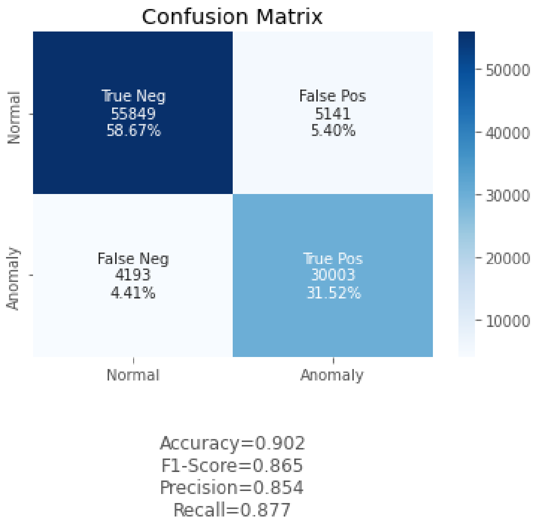


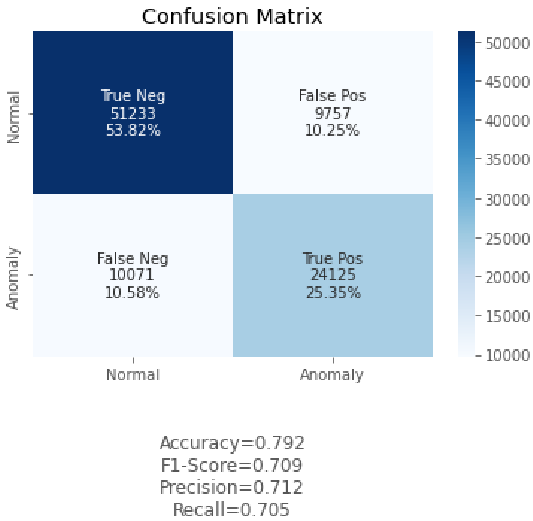
Figure 5.13: Probability density of prediction errors from ANN model (standard deviation = 0.006).

Table 5.12: Fault detection results (LSTM and CS-LSTM)

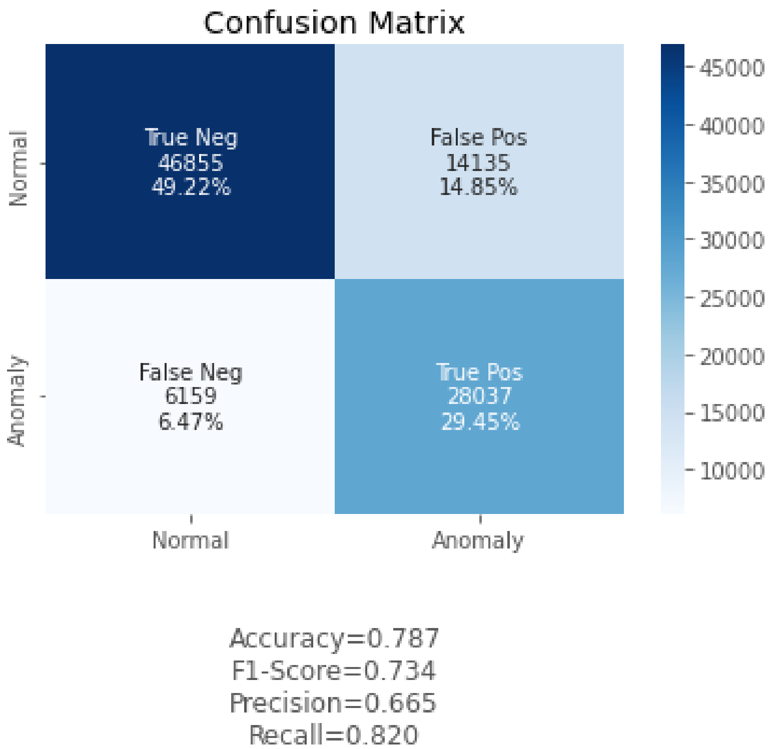
Predictive Model	Detection approach	Accuracy (%)	Recall (TPR)	Precision	F1-score
LSTM (without context features)	Simple threshold ($0.25 \times \sigma$)	90.2	0.923	0.055	0.103
	Local distance (threshold = 0.7)	90.2	0.691	0.712	0.701
	Clustering (#clusters = 8)	90.2	0.877	0.854	0.865
LSTM (with context features)	Simple threshold ($1.0 \times \sigma$)	79.2	0.869	0.066	0.122
	Local distance (threshold = 0.7)	79.2	0.618	0.701	0.657
	Clustering (#clusters = 15)	79.2	0.705	0.712	0.709
CS-LSTM	Simple threshold ($0.25 \times \sigma$)	78.7	0.999	0.063	0.118
	Local distance (threshold = 0.7)	78.7	0.744	0.467	0.574
	Clustering (#clusters = 8)	78.7	0.820	0.665	0.734



(a) LSTM (without context) and clustering



(b) LSTM (with context) and clustering



(c) CS-LSTM and clustering

Figure 5.14: Confusion matrices for LSTM, LSTM with context, and CS-LSTM with clustering-based fault detection.

and (c). The start of the shaded area indicates where the first fault is identified by the models before the actual start of the fault. The difference is around 1800 samples, which means 30 minutes ahead.

Further, we plotted the false positives and the false negatives for each of the two models (LSTM and CS-LSTM) with clustering-based fault detection as shown in Figure 5.16. It can be observed that most of the false positives are present during mid-day. During this period, multiple faults are introduced into the installation (degradation, open-circuit, and short-circuit). Also, more false negatives are observed when the partial shadowing starts to occur in the later part of the day. This is more visible in the case of CS-LSTM.

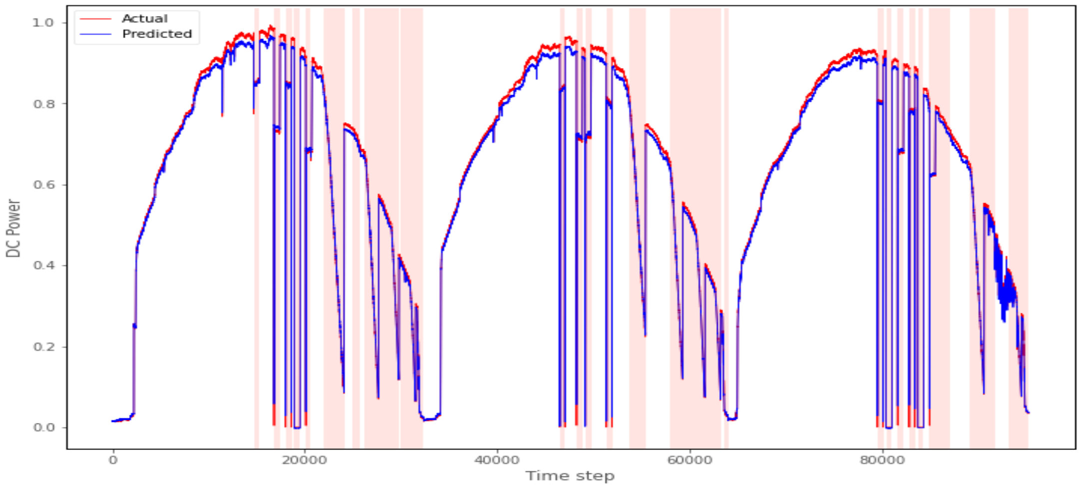
5.4 Sensitivity Analysis

In the CS-LSTM-based fault detection approach, the training-error distribution was used, and a threshold of 3*sigma was determined experimentally for the first case study using real-world SCADA data. The model performance was then evaluated using accuracy, true positive rate (TPR), and false positive rate (FPR), and CS-LSTM was shown to improve FPR compared with the baseline LSTM as shown in Table 5.5. Table 5.13 shows the sensitivity analysis of LSTM and CS-LSTM for a threshold of 3*sigma.

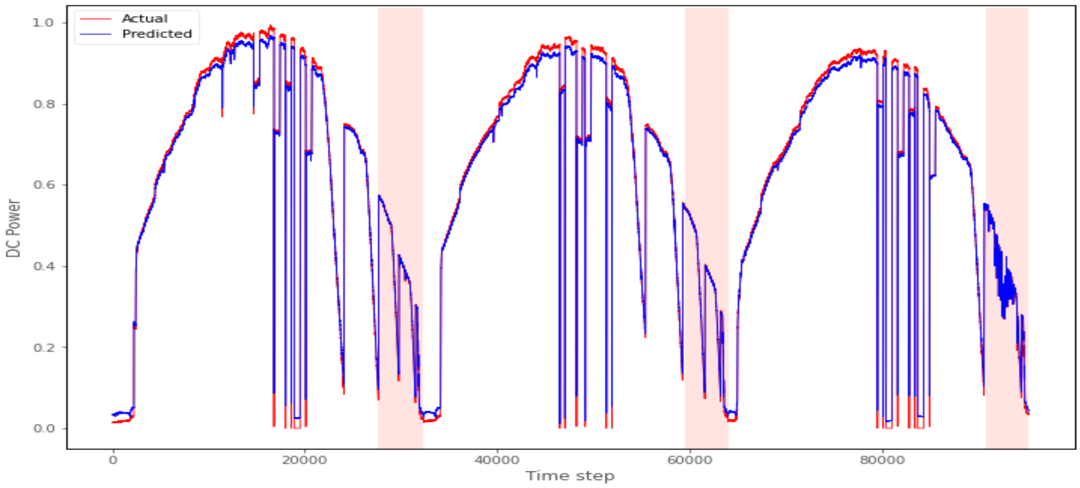
Table 5.13: Sensitivity analysis of LSTM and CS-LSTM for a threshold of 3*sigma.

Metric	**LSTM**	**CS-LSTM**
True Negative (TN)	26031	27045
False Positive (FP)	1740	726
False Negative (FN)	159	154
True Positive (TP)	125	130
Accuracy	93.23%	96.86%
Precision	6.70%	15.19%
Recall	44.01%	45.77%
F1-score	11.65%	22.85%

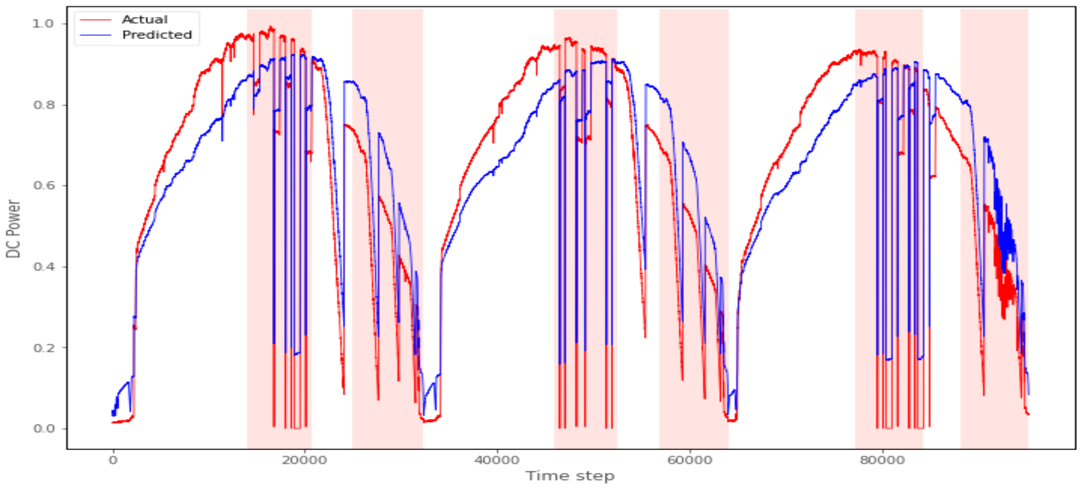
Moreover, for the second case study using a laboratory dataset, different fault-detection strate-



(a) LSTM (without context) and clustering



(b) LSTM (with context) and clustering



(c) CS-LSTM with clustering

Figure 5.15: Fault detection.

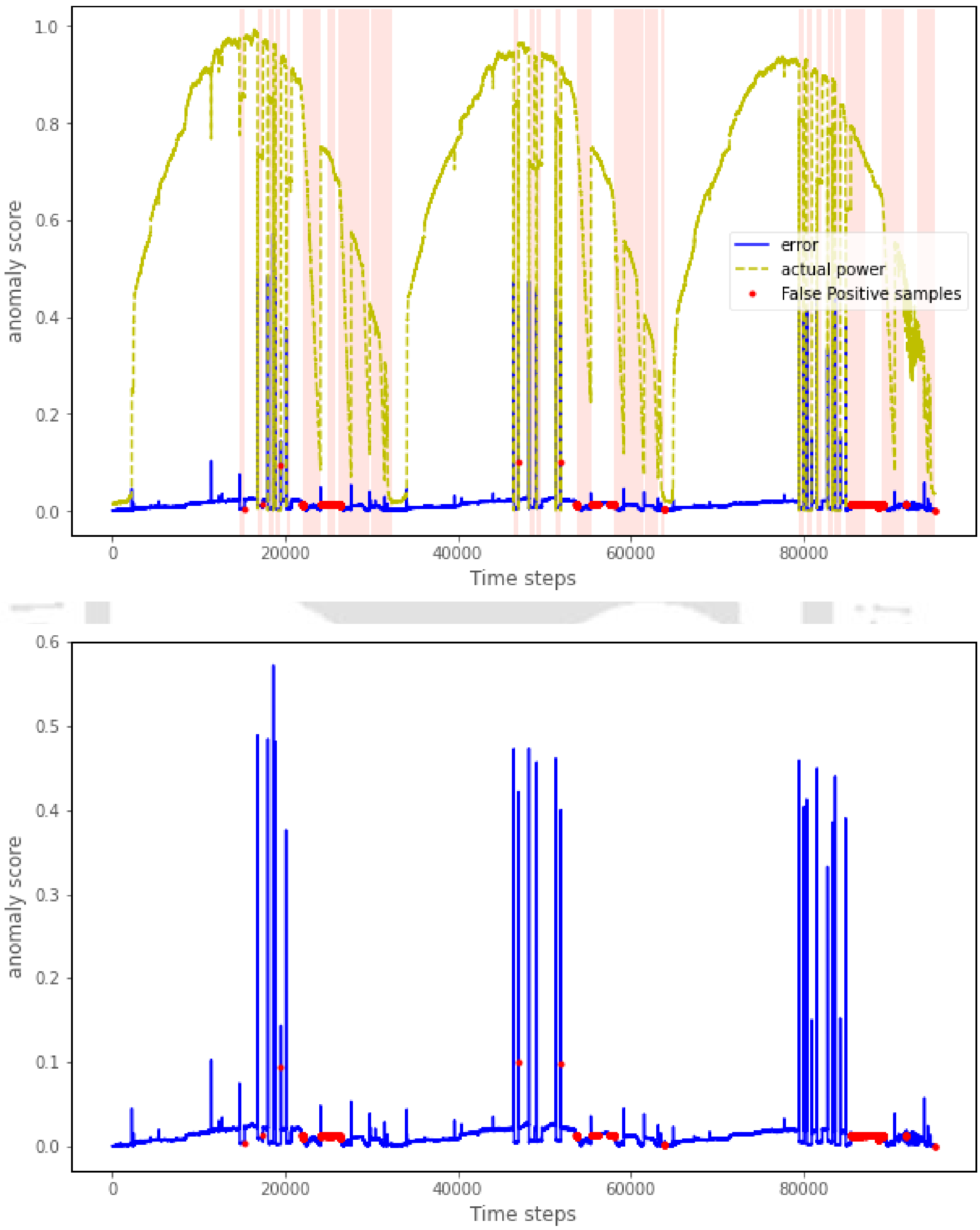


Figure 5.16: False positives (LSTM with clustering)

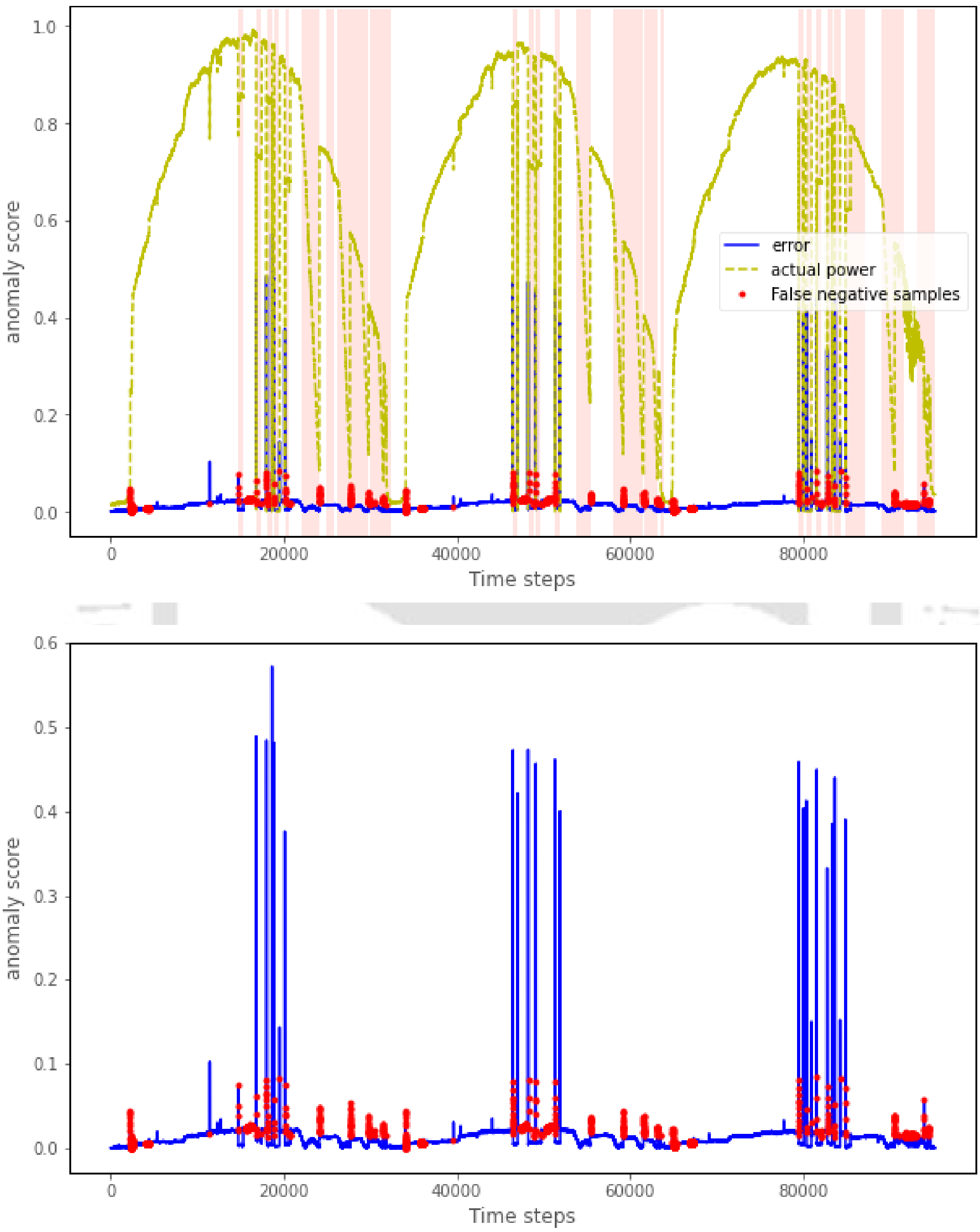


Figure 5.17: False negatives (LSTM with clustering)

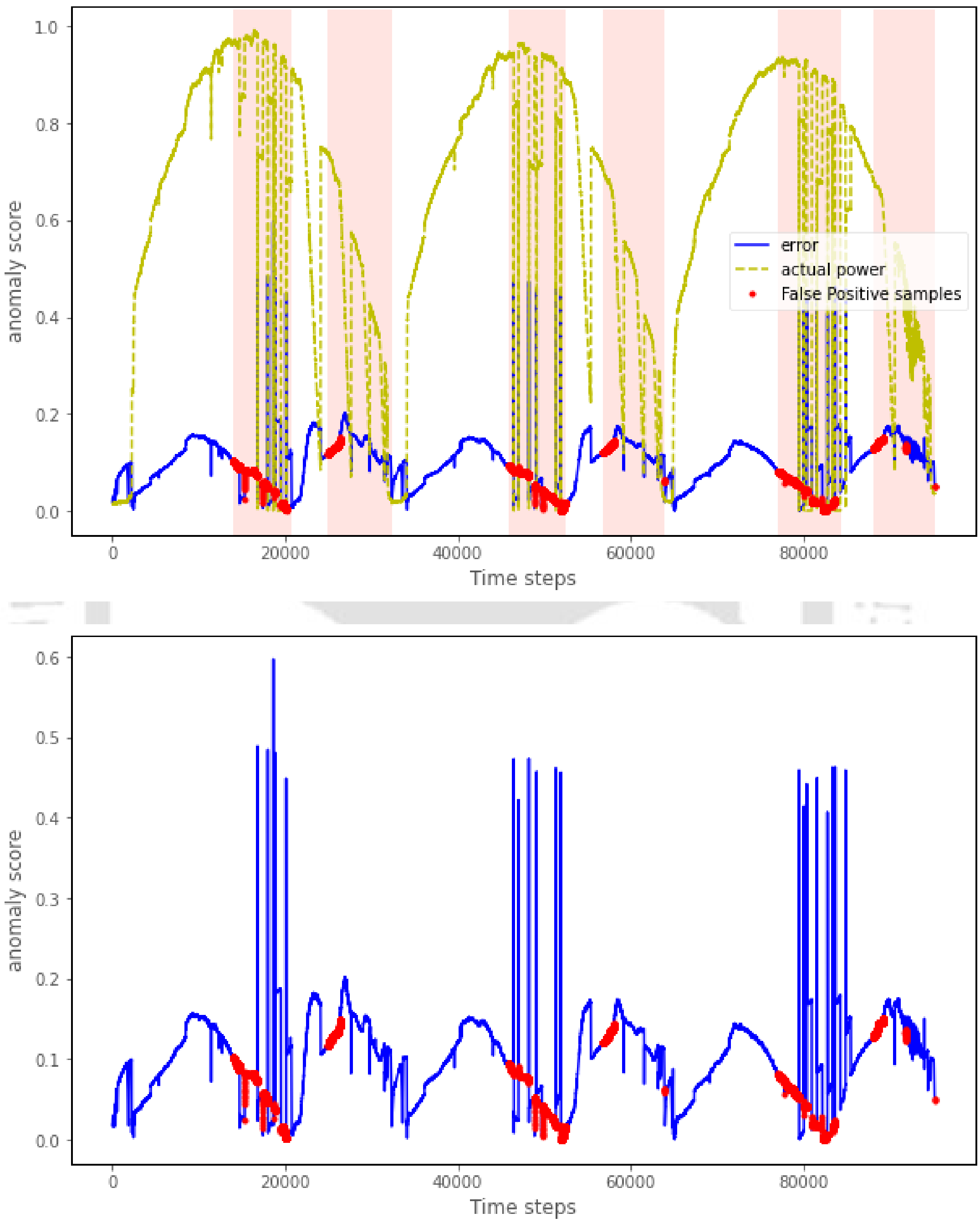


Figure 5.18: False positives (CS-LSTM with clustering)

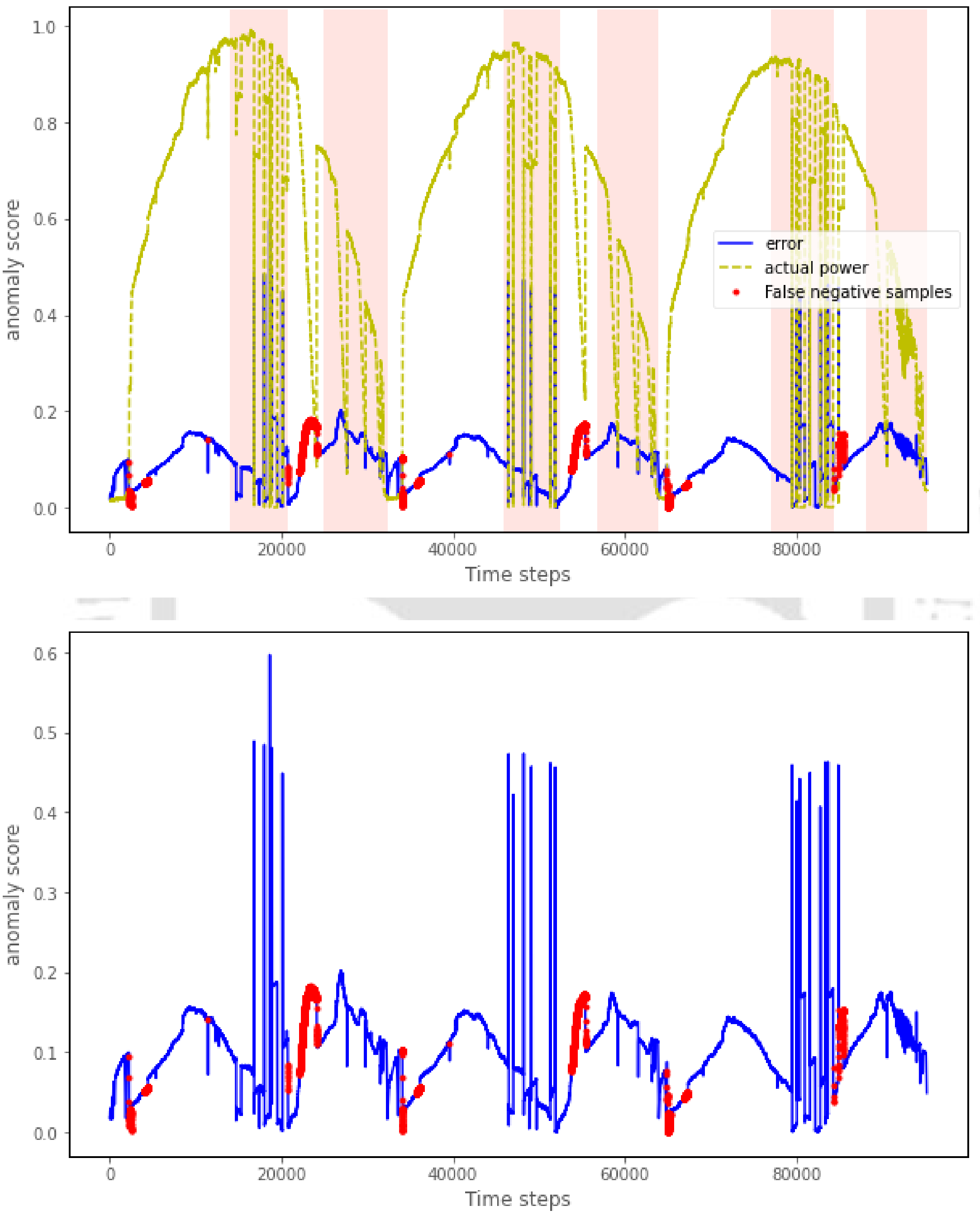


Figure 5.19: False negatives (CS-LSTM with clustering)

gies, including simple thresholding, local distance-based thresholding, and clustering-based detection were compared in terms of accuracy, recall, precision, and F1-score. Table 5.12 shows the sensitivity analysis of LSTM (without context features), LSTM (with context features), and CS-LSTM in combination with simple thresholding, local distance-based thresholding, and clustering-based detection. For simple threshold-based approach, the threshold parameter is varied from ($2\times\sigma$) to ($0.5\times\sigma$), for the clustering-based approach, the number of clusters is varied from 5 to 15. Finally, for the distance-based approach, the window size of 10 minutes is considered, and the threshold is varied from 0.7 to 0.9. The best parameter values of each of the approaches are also listed in Table 5.12 corresponding to each approach. Among all the models as shown in Table 5.12, clustering-based fault detection provided the best F1-score. Also, the recall is comparable for both LSTM and CS-LSTM, and the precision in LSTM is much higher than that in CS-LSTM. Thus, there are fewer false positives in the case of LSTM. Table 5.14 shows sensitivity analysis for LSTM (without context) with clustering, LSTM with context and clustering, and CS-LSTM with clustering. This study provided additional evidence that detection performance varied significantly with the selected fault-detection criterion.

Table 5.14: Sensitivity analysis for clustering based fault detection

Model	**TN**	**FN**	**FP**	**TP**	**Accuracy**	**F1-score**	**Precision**	**Recall**
LSTM (without context) with clustering	55849	4193	5141	30003	0.902	0.865	0.854	0.877
LSTM with context and clustering	51233	10071	9757	24125	0.792	0.709	0.712	0.705
CS-LSTM with clustering	46855	6159	14135	28037	0.787	0.734	0.665	0.820

5.5 Summary

This chapter investigated the role of contextual information in improving the accuracy and reliability of fault detection in solar PV systems. While contextual data such as time, season, environmental conditions etc. have been effectively utilized in domains like Natural Language Processing (NLP), computer vision, and recommender systems, their use in solar energy analytics has been limited. Recognizing that solar irradiance and, consequently, PV power output are influenced by these contextual factors, this chapter has emphasized their integration into machine learning-based fault detection models. The chapter first outlined various techniques for representing and incorporating contextual features into predictive models. Five different mod-

TH-4096_186151013

elling approaches were trained and compared to evaluate the impact of context inclusion. Building upon this, a new model called the CS-LSTM was proposed. In this model, input weights were dynamically adjusted based on contextual information during training, enabling the model to better adapt to variations in operating conditions. We demonstrated the applicability of CS-LSTM for power prediction and subsequently fault detection using two case studies. The first case study consisted of data from real operational plant, and the second case study used data from PV installation in a Lab, which was publicly available. The real dataset consists mainly of inverter-related faults, whereas the open dataset has three types of faults and partial shadowing. The faults are degradation, open circuit and short circuit. We considered regression-based fault detection. For the first dataset, we used LSTM and CS-LSTM and compared the performances. The power prediction with validation data resulted in similar performances. However, the CS-LSTM model performed better compared to the LSTM model in fault detection in terms of all the performance metrics, viz., accuracy, true positive rate (recall), and false positive rate as shown in Table 5.5. In the second case study, to realise the fault detection, we first developed and tested five predictive models: two classical ML models, Linear regression (LR) (baseline model) and Random Forest Regressor (RFR), and three deep models, ANN with multiple layers, LSTM, and CS-LSTM. Two variations of LSTM were implemented, one without context features and the other with context features. For fault detection, we considered three approaches: simple threshold-based, local-distance-based, and clustering-based. The experiments were conducted to see the influence of context on the given dataset. From the experimental results, we observed that for the given dataset, integration of context in predictive models had less influence on the models' outcomes. For fault detection, the performance of LSTM was found to be better than other models in terms of F1-score. However, if we consider only detecting the occurrence of a fault as event (not in terms of faulty samples), then both LSTM and CS-LSTM could detect early faults. The main challenge with this dataset was that the faults were introduced every 15 minutes, which made it unrealistic. Further, apart from the time of the day, no other contextual information could be added as the dataset was captured for a short period of time,(i.e.,16 days).

Thus, to summarise, the proposed CS-LSTM model which integrates explicit contextual information into the learning process is a key contribution of the thesis. The performance of the CS LSTM model is analysed with two datasets and the results obtained are compared with existing approaches. The first case study consists of data from real operational plant, and the second case study uses data from PV installation in a Lab which is publicly available. The real dataset

TH-4096_186151013

consists of mainly inverter related faults. In the first case study, for estimation of AC power, LSTM and CS-LSTM models are used and a threshold-based fault detection is applied. The CS-LSTM model achieves high accuracy and low false positive rate (FPR), which is desired, and outperforms its baseline counterpart (i.e., LSTM). Taking into account the real data set from the 10 MW PV plant, the accuracy, TPR, and FPR of CS-LSTM are 96.9%, 45.8% and 2.6%, respectively, compared to 93.2%, 44.0%, and 6.2%, respectively, for LSTM. Thus, it is evident that CS-LSTM performs marginally better than LSTM in terms of accuracy and TPR; however, a significant improvement can be observed in FPR. Thus, experimental evaluation on real SCADA dataset demonstrates that CS-LSTM improves fault detection performance in real-world data, particularly in terms of accuracy, recall (i.e., true positive rate (TPR)), and false positive rate (FPR).

For the second case study, data from a PV system installed in a Lab is used, and both conventional and sequential models without context and also with context are developed. The second dataset has three types of faults and partial shadowing. The faults are degradation, open circuit and short circuit. For the second case study, the fault detection models rely on regression models for estimation of DC Power at the next time step, and then based on the difference between the predicted and actual outputs at each time step, a data sample is classified as either faulty or normal. Using the second dataset, experiments are conducted with three different approaches for fault detection, namely threshold-based, local distance-based, and clustering-based techniques for fault detection. Finally, we compared the results of these models and observed that for this dataset, context has less influence on the results. One reason may be the frequent introduction of faults into the system (e.g., every 15 minutes), and also insufficient contextual information (e.g., there is only one contextual feature, i.e., "Time of day", and "Season" is not available because the data is collected only for 16 days. However, for this dataset also, both the LSTM and CS-LSTM models provide early detection of faults. If we can collect laboratory datasets for several months, then a fair comparison between SCADA datasets and laboratory datasets will be more meaningful.

This page was intentionally left blank.

TH-4096_186151013

Chapter 6

Predictive Maintenance Framework using Context Sensitive LSTM Model

In this chapter, a predictive maintenance framework based on contextually weighted deep learning is developed to identify potential faults in solar photovoltaic (PV) systems at an early stage. It combines the Context-Sensitive LSTM (CS-LSTM) model for power prediction with a Random Forest (RF) classifier for fault diagnosis, utilising both temporal and environmental data, including time, seasonal variations, temperature, and irradiance.

This chapter's main contribution lies in developing a predictive maintenance framework that is sensitive to context and is designed for detecting faults early in solar PV systems. The framework's initial component involves the development of a CS-LSTM regression model, which predicts the anticipated AC power output when the system is functioning normally. This model incorporates contextual data, such as seasonal changes and the time of day, to understand how these factors affect system performance. By improving the accuracy of power prediction, the proposed model establishes a reliable baseline against which deviations between predicted and measured power can be evaluated, facilitating the detection of anomalous system behaviour and the identification of potential faults. Second, a classification framework that employs Random

TH-4096_186151013

Forest has been developed, adding an additional "Fault Impending" category to facilitate early warnings. To train this classification model, the challenge of class imbalance has been addressed using the Synthetic Minority Over-sampling Technique (SMOTE) [126]. Finally, an integrated predictive maintenance framework is constructed by combining the proposed CS-LSTM regression model with an RF classifier. This framework is designed to aid in the detection, diagnosis, and classification of faults in solar PV systems, tackling issues such as Straight Joint Faults, Earth Faults, Insulated Gate Bipolar Transistor (IGBT) Faults, and Overtemperature Faults. By incorporating contextual information within the CS-LSTM model, the proposed framework enhances both fault prediction and diagnostic performance, resulting in improved identification and classification of abnormal patterns.

We begin with a description of the proposed framework in Section 6.1. We then discuss the experiments and results in Section 6.2, and finally, a summary is presented in Section 6.3.

6.1 Framework for Early Fault Detection and Diagnosis

Figure 6.1 depicts the proposed framework's architecture, which includes two primary components: a fault detection unit and a fault categorization (i.e., classification) unit.The fault detection unit utilizes the CS-LSTM regression model to estimate the expected AC power output during normal system operation. Deviations between the estimated and actual AC power values are monitored, and substantial discrepancies are treated as indicators of potential fault conditions. These flagged instances are then passed to the classification module, which identifies the fault type. To prevent false alarms, the system uses a decision logic layer that distinguishes between transient, non-critical fluctuations and consistent patterns that indicate actual faults.

As a key component of the framework, the data pre-processing module ensures that the input data is properly prepared both during model training and real-time operation. It cleans and formats the data to make it suitable for training, and during deployment, it processes incoming data to ensure compatibility with the model for accurate fault label prediction. SCADA data is often prone to inaccuracies due to sensor failures, which can lead to missing entries, duplicate values, or records falling outside expected limits. Such faulty data are filtered out before model development begins. Additionally, the data available from 6:00 AM to 7:00 PM, aligning with

TH-4096_186151013

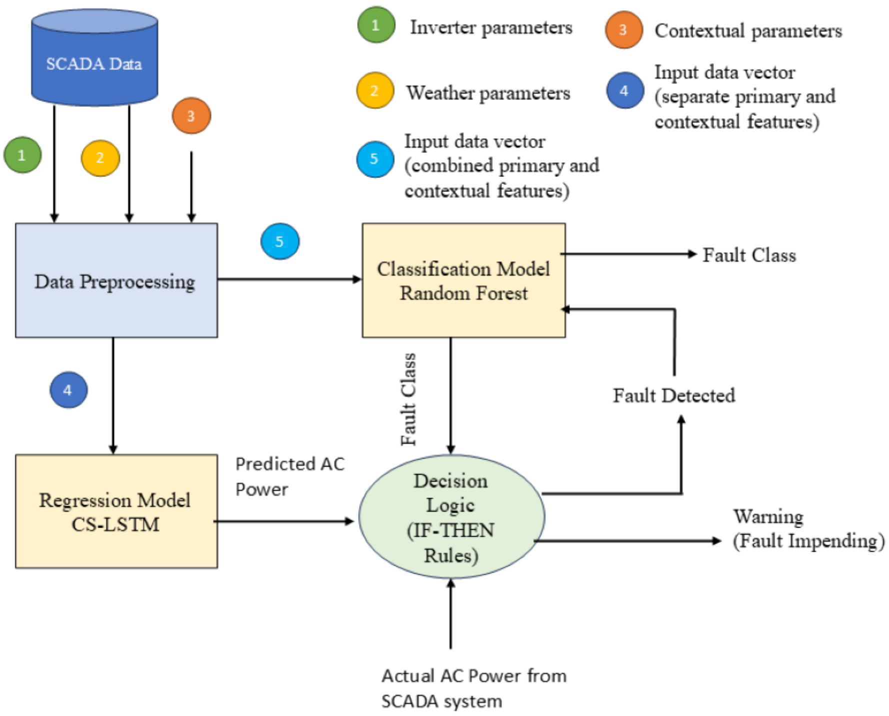


Figure 6.1: Predictive Maintenance Framework

the active operational hours of the PV system, is considered.

For training the regression model, only data corresponding to normal operating conditions is used. This is ensured by removing anomalous entries based on fault logs and retaining only those days with uninterrupted data within the selected time window. The cleaned data is then normalized to maintain consistent feature scaling. Unlike the regression model, the fault classification module is trained using samples representing both normal and abnormal operating states. The abnormal class encompasses several fault categories observed in practical PV installations, including Straight Joint Fault, Earth Fault, IGBT Fault, Overtemperature Fault, and Inverter Not Starting. To improve early detection capabilities, an additional novel class named as *Fault Impending* has been incorporated into the classifier. Samples occurring during the three observation intervals immediately preceding a recorded fault event are assigned to this class, allowing the classifier to learn incipient patterns that may indicate an approaching failure. This helps the model learn early indicators of failure. Normal operating instances are represented by a separate *No_Fault* class, resulting in a total of seven target categories. As fault occurrences are inherently uneven across classes, the training dataset exhibits significant class imbalance. To mitigate this issue, SMOTE is employed to generate representative samples for underrepresented fault categories. This resampling strategy improves class distribution and reduces the tendency of the classifier to favour majority classes, thereby enhancing its ability to recognize infrequent fault conditions. The classification framework utilises both operational(i.e., primary) and contextual attributes. Contextual variables are incorporated by augmenting the original feature space with additional context-related information, following the principle of contextual expansion [127]. Multiple classification algorithms are subsequently trained and evaluated using the prepared dataset, and the model yielding the most favourable performance is selected for deployment within the overall predictive maintenance framework.

The key components of the proposed framework are (i) the CS-LSTM model, (ii) the RF classification framework, and (iii) a decision logic for governing the interpretation of the model outputs. For clarity of presentation, these components are discussed below.

6.1.1 Context Sensitive LSTM (CS-LSTM) based Regression Model

The CS-LSTM model, originally introduced by Baruah *et al.* in [18], extends the conventional LSTM architecture by incorporating contextual information into the learning process. This enhancement enables the model to capture not only temporal dependencies present in sequential data but also the influence of contextual factors, thereby improving prediction performance. The proposed CS-LSTM architecture retains the standard LSTM cell as its fundamental computational unit while explicitly integrating contextual variables into the model.

The model operates on two categories of input variables: primary features and contextual features. Primary features are derived directly from the principal source of information, whereas contextual features originate from supplementary sources that provide additional information about the operating environment [113]. Both categories of features are collected from the same system environment and are jointly utilised during model training.

Within the CS-LSTM framework, the relative importance assigned to primary features is adjusted according to the prevailing context. This mechanism, referred to as *contextual weighting*, increases the influence of those primary attributes that are more informative under a particular contextual condition [127],[18]. Although predictive models can be trained using primary features alone, contextual features do not possess sufficient discriminatory capability when used independently. Nevertheless, the inclusion of contextual information can significantly enhance

TH-4096_186151013

the learning process by providing additional cues that improve model generalization and prediction accuracy.

The benefit of contextual information can be illustrated through speech recognition systems. In such applications, the acoustic spectrum constitutes the primary input used for recognition, while speaker-specific characteristics, such as accent, serve as contextual information. Although the accent alone cannot be used to recognize speech, its incorporation can assist the learning algorithm in interpreting the primary input more effectively and consequently improve overall system performance.

LSTM were developed to address the limitations of conventional Recurrent Neural Networks (RNNs) [96] in modelling long sequential dependencies. Traditional RNNs often encounter the vanishing and exploding gradient problems during backpropagation through time, making it difficult for the network to learn relationships between events that are separated by long temporal intervals. As a result, their effectiveness is limited when processing long time-series sequences. LSTM networks overcome these challenges by introducing a specialized memory mechanism that enables the preservation and selective updating of information over extended periods. Consequently, LSTMs have become one of the most widely adopted architectures for time-series forecasting, sequence modelling, and predictive analytics [96][113].

The memory cell is the core component of an LSTM network, acting as an internal storage unit capable of maintaining relevant information over multiple time steps. In an LSTM unit, apart from the memory cell, there are three gates: the forget gate, the input gate, and the output gate. These gates manage how information enters, circulates within, and exits the memory cell, allowing the network to decide which information should be retained, updated, or discarded during the learning process[113].

The forget gate is responsible for evaluating the significance of information stored in the previous cell state. It generates values in the range $[0,1]$, where values closer to zero indicate that the corresponding information should be discarded, while values closer to one imply that the information should be preserved[113]. This selective forgetting mechanism enables the network to eliminate irrelevant or outdated information that may otherwise degrade prediction performance.

The input gate regulates how fresh data is added to the memory cell. The gate decides which

TH-4096_186151013

information is significant enough to affect the updated cell state based on the current input and the prior hidden state. By regulating the amount of new information admitted into memory, the input gate prevents unnecessary fluctuations and helps the model focus on relevant temporal patterns.

The output gate regulates the generation of the hidden state, which is then transmitted to the subsequent time step and, if necessary, to the output layer. This gate determines the degree to which the information stored in the memory cell influences the current output. By integrating information from both the updated cell state and the preceding hidden state, the output gate ensures that only the most pertinent information is propagated forward.

Both short-term and long-term temporal dependencies can be efficiently captured by LSTM networks through the coordinated action of various gating processes. Because of this ability, they are especially well-suited for applications involving sequential data, like fault prediction in solar photovoltaic systems, load forecasting, speech recognition, and natural language processing, where past observations frequently contain important information for future predictions.

In contrast to the conventional LSTM architecture, the CS-LSTM model incorporates contextual information directly into the learning process by allowing the network parameters to vary according to the prevailing context. More specifically, contextual variables influence the weights of the input connections, thereby enabling the model to adjust its behaviour under various operating conditions. This mechanism allows the network to learn not only temporal dependencies but also the relationships between contextual factors and the target variable, thereby improving prediction accuracy in environments where system behaviour is strongly affected by external conditions[113] .

Figure 6.2 shows the structure of the CS-LSTM cell. The architecture receives two categories of inputs at each time step. The primary input vector, denoted by x_t, contains the main features used for prediction, while the contextual input vector, represented by z_t, contains auxiliary information describing the operating environment. Examples of contextual variables in solar photovoltaic systems include seasonal indicators, time of day, irradiance levels, and environmental conditions that may influence system performance.

Similar to a standard LSTM network, the CS-LSTM maintains an internal memory state c_t and a hidden state h_t to capture temporal relationships in sequential data. The mechanisms governing the update of the cell state and hidden state remain fundamentally unchanged from those of the conventional LSTM architecture. Consequently, the model preserves the ability of LSTM networks to learn long-term dependencies while avoiding the vanishing-gradient problem associated with traditional recurrent neural networks.

The key distinction of the CS-LSTM lies in the formulation of its gating functions. In a standard LSTM, the weights associated with the input connections remain fixed after training and are applied uniformly irrespective of the operating context. In the proposed CS-LSTM, however, these weights are dynamically adjusted as functions of the contextual input vector z_t. As a result, the influence of a primary feature on the network output can vary according to the prevailing context. Features that are more informative under a particular contextual condition receive greater emphasis, whereas less relevant features contribute proportionally less to the learning process. This context-dependent adaptation is consistent with the principle of contextual weighting and enables the model to generate more accurate predictions under varying operating conditions.

Accordingly, the activation functions associated with the forget gate, input gate, candidate memory state, and output gate are reformulated such that their input weights become dependent on the contextual variables. This modification allows contextual information to directly influence the information flow within the network, thereby enhancing the model's capability to capture complex interactions between temporal patterns and environmental factors. The resulting gate equations are presented below [113].

$$f_t = \sigma(w_{fh} \cdot h_{t-1} + w_{fx}(z_t) \cdot x_t + b_f) \tag{6.1}$$

$$i_t = \sigma(w_{ih} \cdot h_{t-1} + w_{ix}(z_t) \cdot x_t + b_i) \tag{6.2}$$

$$\tilde{c}_t = \tanh\left(w_{ch} \cdot h_{t-1} + w_{cx}(z_t) \cdot x_t + b_c\right) \tag{6.3}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{6.4}$$

TH-4096_186151013

$$o_t = \sigma(w_{oh} \cdot h_{t-1} + w_{ox}(z_t) \cdot x_t + b_o) \tag{6.5}$$

$$h_t = o_t * \tanh(c_t) \tag{6.6}$$

Assuming that $\mathbf{w}(z_t)$ denotes the weights of the input connections, where subscripts are omitted for simplicity of notation, each element of the parameter matrix denoted by $\mathbf{w}(z_t)$ having a dimension of $m \times n$ is defined as shown below [113]:

$$w_{ki} = \mathbf{B}_{ki}\mathbf{\Gamma}(z_t), \quad k = 1, \ldots, m, \quad i = 1, \ldots, n \tag{6.7}$$

In the proposed formulation, the context-dependent weights are parameterized through a coefficient vector represented by $\mathbf{B}_{ki} = [\beta_{ki1}, \beta_{ki2}, \ldots, \beta_{kil}]$. The vector of basis functions, denoted by $\mathbf{\Gamma}(z_t) = [\gamma_1(z_t), \gamma_2(z_t), \ldots, \gamma_l(z_t)]^T$ provides a functional representation of the contextual input and serves as a design-specific hyperparameter of the model.

In CS-LSTM, the context-dependent input weights are parameterized through a set of learnable coefficient vectors. For each weight element, the influence of the contextual variables is captured by a coefficient vector denoted as $\mathbf{B}_{ki} = [\beta_{ki1}, \beta_{ki2}, \ldots, \beta_{kil}]$. These coefficients determine how the corresponding weight varies under different contextual conditions and are learned during the training process through backpropagation.

The contextual information is represented through a set of basis functions collected in the vector $\mathbf{\Gamma}(z_t) = [\gamma_1(z_t), \gamma_2(z_t), \ldots, \gamma_l(z_t)]^T$ where each basis function maps the contextual input z_t into a transformed representation that can effectively influence the network parameters. The choice of these basis functions is specified during model design and therefore acts as a hyperparameter of the CS-LSTM architecture. By selecting appropriate basis functions, the model can capture complex and potentially nonlinear relationships between the contextual variables and the network weights.

The context-sensitive weights are obtained by combining the coefficient vectors with the basis-function representation of the contextual input. In this manner, the input weights are no longer fixed quantities but become adaptive parameters whose values change according to the prevailing context. Consequently, the contribution of a primary feature can vary dynamically depending on the operating conditions represented by z_t.

For compact representation, the coefficient vectors associated with all input connections can be organized into a matrix **B**. This arrangement provides a convenient matrix formulation of the context-dependent weight structure and facilitates efficient implementation of the CS-LSTM learning algorithm. The construction of the matrix **B** is given as follows:

To facilitate a more compact formulation of the context-sensitive weight parameters, the coefficient vectors are assembled into a matrix denoted by **B**. The k^{th} row of this matrix, denoted as $\mathbf{B}_k$, is obtained by performing concatenation of the vectors $\mathbf{B}_{k1}, \mathbf{B}_{k2}, \ldots, \mathbf{B}_{kn}$, resulting in

$$\mathbf{B}_k = [\mathbf{B}_{k1}, \mathbf{B}_{k2}, \ldots, \mathbf{B}_{kn}], \quad k = 1, 2, \ldots, m$$

Thus, **B** is composed of $\mathbf{B}_k$, k=1, 2, ..., m; and to obtain each $\mathbf{B}_k$, we concatenate the vectors $\mathbf{B}_{ki}$, i=1, 2, ..., n. Since $\mathbf{B}_{ki} = [\beta_{ki1}, \beta_{ki2}, \ldots, \beta_{kil}]$, the matrix **B** has a dimension of $(m \times nl)$. This matrix formulation enables the context-dependent parameters to be represented and manipulated efficiently during model training and inference.

In the Context-Sensitive LSTM, we can reformulate the gate activation equations by using matrix **B** as [113]:

$$f_t = \sigma(w_{fh} \cdot h_{t-1} + \mathbf{B}_f(x_t \otimes \mathbf{\Gamma}(z_t)) + b_f) \tag{6.8}$$

$$i_t = \sigma(w_{ih} \cdot h_{t-1} + \mathbf{B}_i(x_t \otimes \mathbf{\Gamma}(z_t)) + b_i) \tag{6.9}$$

$$\tilde{c}_t = \tanh\big(w_{ch} \cdot h_{t-1} + \mathbf{B}_c(x_t \otimes \mathbf{\Gamma}(z_t)) + b_c\big) \tag{6.10}$$

TH-4096_186151013

$$o_t = \sigma(w_{oh} \cdot h_{t-1} + \mathbf{B}_o(x_t \otimes \boldsymbol{\Gamma}(z_t)) + b_o) \tag{6.11}$$

In this case, the Kronecker product is represented by $\otimes$. During the network's training phase, the parameters found in the coefficient vectors $\mathbf{B}_{ki}$ are determined. Like a traditional LSTM, a loss function is first defined to measure the prediction error, and the corresponding coefficients are subsequently adjusted through gradient-based optimization using backpropagation.

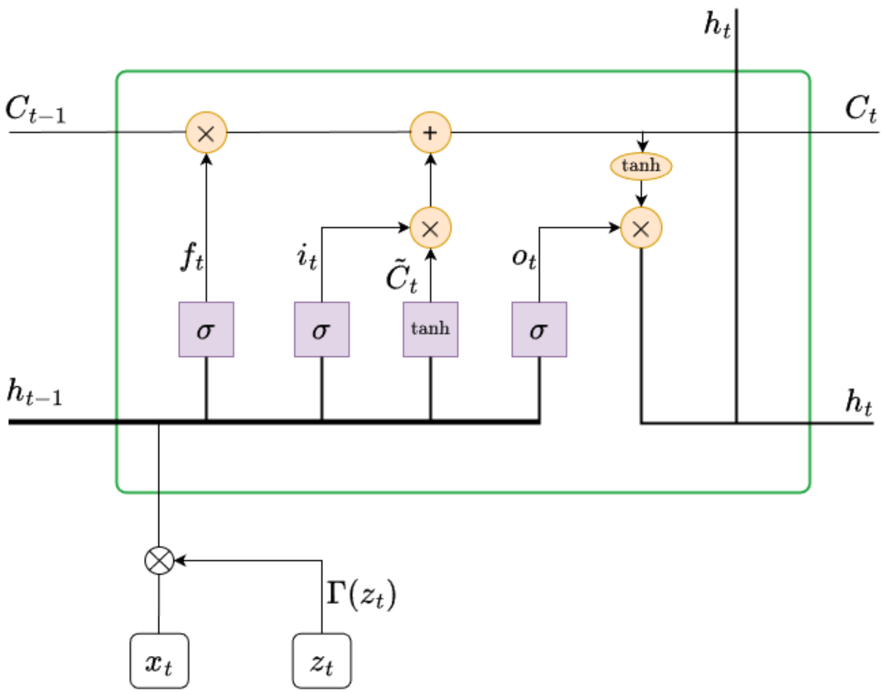


Figure 6.2: CS-LSTM Cell

Within the proposed predictive maintenance framework, the CS-LSTM regression model serves as the core component for early fault detection. The model is trained exclusively using data collected during healthy operating conditions to learn the normal relationship between the input variables and the corresponding AC power output. Once trained, the model generates an estimate of the expected AC power under normal system operation. The predicted power values are continuously compared with the actual measurements obtained from the photovoltaic plant. A significant and sustained deviation between the estimated and observed power outputs beyond a predefined threshold is interpreted as an indication of abnormal system behaviour and may suggest the presence of an impending or existing fault.

6.1.2 Fault Classification Framework

Random Forest-based supervised learning is used for developing the fault classification model, where the training dataset is balanced using SMOTE. The training dataset comprises various fault types commonly observed in PV systems. In particular, the following types of faults are considered for evaluation of the model: Earth Fault, Straight Joint Fault, IGBT Fault, Overtemperature Fault, and Inverter Not Starting. In the classification framework, the various faults are treated as class labels that are distinct, thereby enabling the model to learn to distinguish among them.

To enhance the prognostic capability of the framework, as a novel concept, a new class label denoted as *FaultImpending* has been introduced into the classification framework. This category corresponds to operating conditions immediately preceding a recorded fault event and is intended to capture the early signatures of fault development. In the present study, the three observations prior to any fault occurrence are assigned to this class. The introduction of the *FaultImpending* category enables the classifier to provide advance warning of potential failures in addition to identifying faults that have already occurred [113].

Lastly, normal operating conditions are represented by a *No_Fault* class. Seven different class labels are used in the categorization task overall. The classifier utilizes both primary and contextual attributes. For classification purposes, the contextual variables are appended to the original feature set and treated collectively with the primary features, following the contextual expansion approach described in [127]. Five distinct classifiers are ultimately trained, tested, and their performance evaluated. The model demonstrating superior performance has been chosen in order to be integrated into the predictive maintenance framework that has been proposed.

6.1.3 Decision Logic

This section discusses the decision logic which governs the interpretation of the model outputs. The decision-making component is a rule-based system that operates by analyzing discrepancy between the AC power predicted by the CS-LSTM model and the actual measured power. If the discrepancy surpasses a predetermined threshold, it is flagged as a possible anomaly and

TH-4096_186151013

categorised as a non-serious fault.

Simultaneously, the classification module keeps on evaluating the input and predicts a fault type. If the predicted class is "Fault Impending," it also triggers a non-serious fault alert. To differentiate between persistent issues and random fluctuations, the system uses a counter. This counter records how often the deviation stays above the threshold for consecutive time steps. When the count surpasses a predetermined limit, the event is classified as a serious fault, signalling a prolonged abnormal condition. In addition to the detailed fault types determined by the classifier, the serious faults are also recorded. This layered decision logic not only helps to enable early warnings for developing faults but also helps to identify critical recurring issues, thus aiding in effective and proactive maintenance scheduling. The full procedure is detailed in Algorithm 1 for individual inverter evaluation.

6.2 Experiments and Results

The experimental assessment of the proposed framework is outlined in this section, with a focus on two primary components: (i) regression-based AC power prediction, essential for identifying faults, and (ii) classification of detected faults. Further, the overall performance of the integrated framework is assessed through end-to-end experiments.

6.2.1 AC Power Prediction Models

The first phase of experimentation focuses on assessing the capability of various regression models to predict the generated AC power by the solar photovoltaic plant during healthy operating conditions. To illustrate the efficacy of the proposed CS-LSTM method, its results are evaluated against several popular machine learning and deep learning models, including Artificial Neural Network (ANN), Random Forest Regressor (RFR), Long Short-Term Memory (LSTM), and Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM). Each regression method is examined in two forms: a traditional version that uses only the main input features and a context-enhanced version that incorporates additional contextual data, indicated

TH-4096_186151013

Algorithm 1 Fault detection and classification decision logic

Require:

$\mathbf{x}$: Input vector for classification model

$\mathbf{y}$: actual AC power value

$\hat{\mathbf{y}}$: predicted AC power value

τ: Deviation threshold for anomaly detection

θ: Number of consecutive deviations to declare a serious fault (consecutive threshold)

$\mathbf{M}_c$: Trained classification model

N_{err} : Number of consecutive errors

Initialise

$\theta \leftarrow 0$

for each time step t **do**

    $\delta \leftarrow \left|\hat{\mathbf{y}}[t] - \mathbf{y}[t]\right|$

    Predict fault class using the classifier:

    $\mathscr{C} \leftarrow \mathbf{M}_c(x[t])$

    if $\mathscr{C}$ == "Fault Impending" **OR** $\delta > \tau$ **then**

        Issue warning for Impending Fault at the corresponding inverter ID (non-serious fault)

        $N_{\text{err}} \leftarrow N_{\text{err}} + 1$

    else

        $N_{\text{err}} \leftarrow 0$

    end if

    if $N_{\text{err}} == \theta$ **then**

        Raise `serious fault` alarm

        Exit

    end if

end for

by the suffix "-X". The standard LSTM model, which learns temporal dependencies without explicitly integrating contextual elements, serves as the baseline for assessing the advantages provided by the CS-LSTM architecture. The CNN-LSTM model merges the feature extraction abilities of convolutional layers with the sequence learning capabilities of LSTM units, allowing it to capture both spatial feature representations and temporal patterns within the data.

6.2.1.1 Model Training and Validation

For the power prediction experiments, the input variables are selected using a correlation-based feature selection approach. Based on the analysis, the primary feature set comprises *Ambient_Temperature*, *Tilt_Irradiation*, *Module Temperature*, *Horizontal_Irradiation*, and *Wind_Speed* with *AC_Power* serving as the target variable. To capture temporal dependencies and exploit historical system behaviour, the value of *AC_Power* from the preceding time step is also incorporated as an input attribute. However, for architectures involving convolutional neural networks (CNNs), feature selection is not applied, and the complete set of available input variables is utilized. This allows the convolutional layers to automatically learn and extract relevant feature representations directly from the raw input data. To train the contextual models, two additional contextual features, *season* and $time\_of\_day$, are incorporated alongside the primary inputs.

For the sequence-based models, namely CNN-LSTM, LSTM, and CS-LSTM, the data corresponding to each inverter was divided into training and validation subsets. The latest s observations from every inverter were set aside for validation purposes, with the rest of the samples being used for training the model. To meet the models' sequence-processing needs, the parameter s was chosen as an integer multiple of the sequence length. This strategy ensured that the validation dataset contained representative samples from all sixteen inverters. Furthermore, for performance evaluation, three days were randomly selected from each month for the training process, thereby forming an independent test set that captures seasonal variations throughout the year.

A range of hyperparameter combinations was explored to identify suitable model configurations. For the sequential architectures, hidden units ranged from 10 to 30, increasing by 5 each time, while the sequence length was investigated over the range of 5 to 15 time steps. The learning rate was treated as a tunable hyperparameter and sampled from a log-uniform distribution spanning (10^{-5}) to (10^{-3}). Every model was trained with the RMSProp optimisation algorithm, utilising a mini-batch size of 128. Owing to the computational complexity of custom implementation of the CS-LSTM architecture, a single recurrent layer was employed. Contextual input was represented through second-order polynomial basis functions. To investigate the impact of contextual information, both the CNN-LSTM and LSTM models were evaluated under two scenarios: one using only the primary features and another incorporating contextual attributes through the contextual expansion approach. In contrast, the CS-LSTM model integrated contextual information using a contextual weighting mechanism, allowing the influence of primary features to vary according to the prevailing context.

The entire training data was used for the ANN and RFR models without a separate validation set, as hyperparameter optimisation was carried out using GridSearchCV available in PyTorch, which internally performs cross-validation to evaluate different parameter combinations. For RFR, the number of trees was tested from 100 to 300, increasing by 50 each time. The maximum tree depth was set at 10, 15, 20, and 25. A single hidden-layer design was used for the ANN, with neurons ranging from 50 to 250, increasing by 50. The optimal hyperparameter settings selected for each regression model are summarized in Table 6.1.

6.2.1.2 Test Results

The models are assessed using common regression evaluation metrics such as Root Mean Squared Error (RMSE) and the R2 score. The performance results for each model are summarized in Table 6.2, with the corresponding plot provided in Figure 6.3.

The results show that incorporating contextual features significantly enhances prediction accuracy across all model types. The ANN model's RMSE drops from 79.223 to 49.137, while its R^2 score improves from 0.7797 to 0.8985 with the inclusion of contextual inputs. The Random Forest Regressor (RFR) shows an even more pronounced improvement with the reduction in RMSE from 85.739 to 33.789. Also, the R^2 score of RFR improves from 0.7419 to 0.9520. These advancements emphasise the importance of contextual information in helping models to better understand and capture the intricate patterns involved in solar power generation.

Among all the evaluated models, CS-LSTM delivers the most accurate results, achieving an RMSE of 25.93 and an R^2 score of 0.9741. These results highlight its superior ability to model

TH-4096_186151013

Table 6.1: Final Hyperparameters for Regression Models

Model	**Hyperparameter**	**Final Value**
RFR	Number of Estimators	200
	Max Depth	25
ANN	No. of Neurons (1 layer)	250
	Activation function	ReLU
	Learning rate	0.0001
	Optimizer	Adam
LSTM	Hidden units	25
	Number of layers	25
	Sequence length	5
	Learning rate	0.0003
CNN-LSTM	For CNN: Kernel size	3
	Convolution layers	2
	For LSTM: Hidden units	50
	Number of layers	2
	Sequence Length	5
	Learning Rate	0.0003
CS-LSTM	Hidden units	25
	Number of layers	1
	Sequence length	5
	Learning rate	0.0009

TH-4096_186151013

Table 6.2: Comparative performance of various regression models

Model	R2-Score	RMSE(Not-Normalised)	RMSE(Normalised)
ANN	0.7797	79.223	0.0989
ANN-X	0.8985	49.137	0.0667
RFR	0.7419	85.739	0.1070
RFR-X	0.9520	33.789	0.0459
LSTM	0.8217	63.179	0.0858
LSTM-X	0.9316	39.389	0.0535
CNN-LSTM	0.8013	69.530	0.0868
CNN-LSTM-X	0.9261	40.957	0.0556
CS-LSTM	0.9741	25.93	0.0397

both time-dependent behavior and contextual influences. CS-LSTM achieves a significant improvement over LSTM by decreasing the RMSE by 58.96%, indicating a substantial performance gain.

It is important to note that the models labelled with the "-X" suffix follow the contextual expansion approach, where contextual features are simply appended to the primary inputs and treated equally by the model. On the contrary, the CS-LSTM model employs a contextual weighting mechanism that uses contextual features to dynamically adjust the contribution of primary inputs via learned weights. Experimental findings confirm that the contextual weighting strategy in CS-LSTM leads to better predictive performance, emphasizing its effectiveness in improving the accuracy of solar PV power estimation.

Further, using SHapley Additive Explanations (SHAP), we determined the feature importance scores for CS-LSTM model. The top 15 features are shown in in Figure 6.4, based on a sequence length of 5 (from t_0 to t_4). Among these, primary features like $AC\_RealPower$, $Tilt\_Irradiation$, and $Ambient\_Temperature$ consistently emerged as the most influential across all time steps. Notably, their contribution tends to decline as the sequence approaches the most recent time point. The corresponding temporal distribution of feature influence is illustrated in Figure 6.5.

The SHAP analysis of the CS-LSTM model indicates that primary sensor features account for a slightly higher share of the model's predictive power (55.7%) compared to contextual inputs

TH-4096_186151013

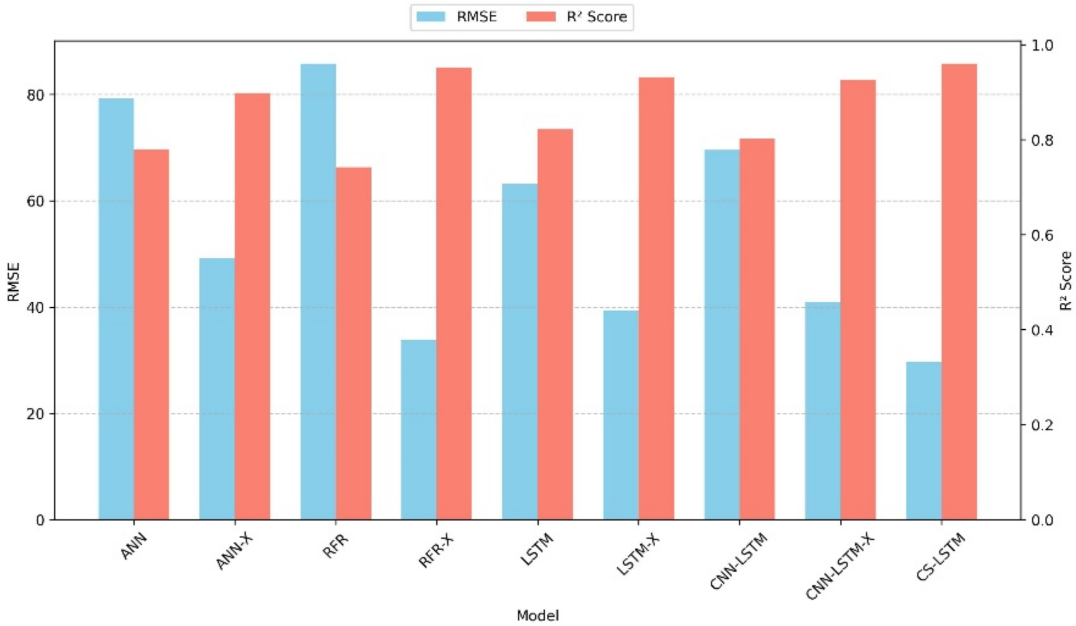


Figure 6.3: RMSE and R2 score of all the regression models

(44.3%). However, contextual factors, especially time of day, have a notable influence on the model's output, as illustrated in Figures 6.6 and 6.7. Within the contextual features, time of day proves to be nearly twice as impactful as the seasonal indicator, suggesting that the model is more responsive to short-term temporal dynamics than to long-term seasonal variations. The influence of individual primary features is further detailed in Figure 6.8.

6.2.2 Fault Classification Models

In the second phase of the experiments, the objective was to determine the most suitable classifier for incorporation into the proposed framework, with the goal of accurately detecting existing faults and predicting potential issues via the "Fault Impending" class. Several classification methods, such as Artificial Neural Networks (ANN), k-Nearest Neighbours (k-NN), Decision Tree (DT), and Random Forest (RF), were tested on a labelled dataset containing normal, faulty, and pre-fault conditions, and their performance was assessed using accuracy, recall, precision, and F1-score metrics. The objective was to select the classifier that best balances early fault detection with resilience to false alarms.

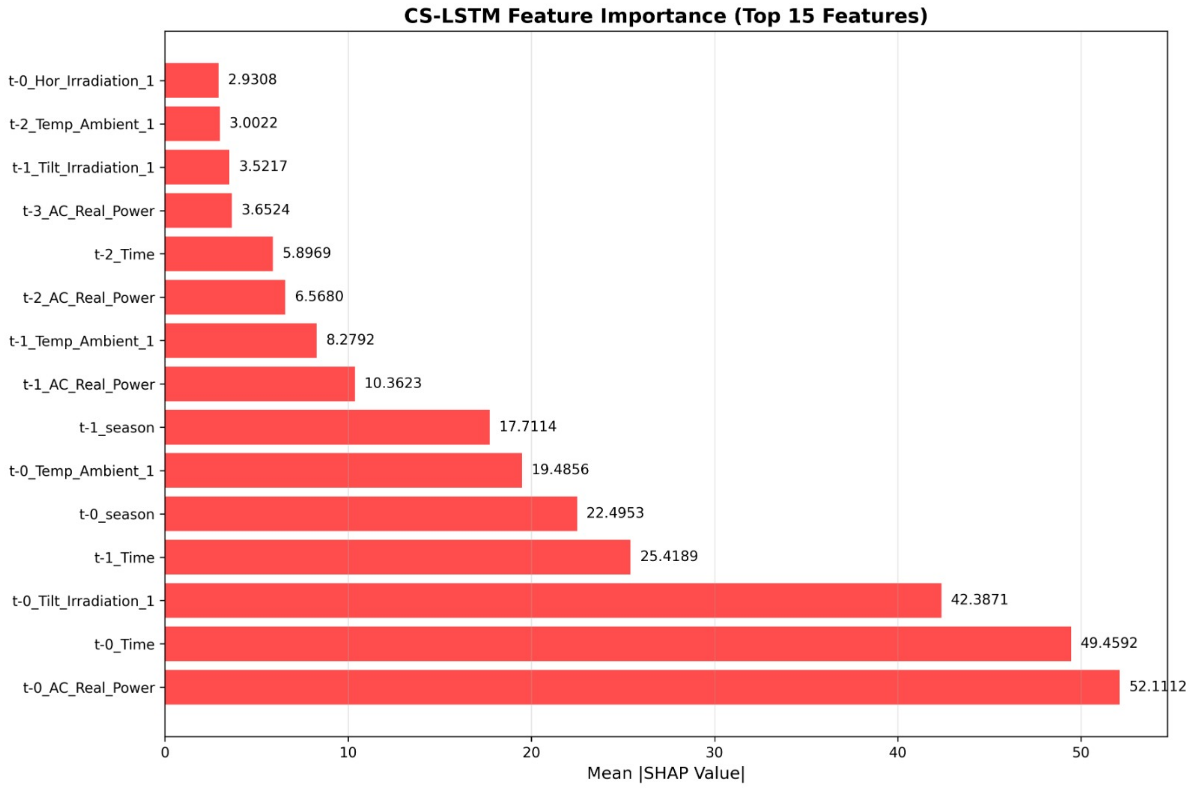


Figure 6.4: Feature Importance of CS-LSTM (Top 15 features)

6.2.2.1 Model training and validation

For the classification task, XGBoost [128] is employed to rank features based on their importance, retaining only those with positive importance scores for model development. The final feature set includes AC_Power, DC_Power, $AC\_Current$, $DC\_Current$, $Ambient\_Temperature$, $Tilt\_Irradiation$, $Module\_Temperature$, $Horizontal\_Irradiation$, and $Wind\_Speed$. The dataset undergoes a random shuffle with a fixed seed to maintain reproducibility, followed by a division into training and testing sets at a 70:30 ratio. Hyperparameter tuning is performed using GridSearchCV, which thoroughly explores a specified range of parameter values. This process includes k-fold cross-validation to assess model performance over various data splits, minimizing the potential bias from a single train-test division. The model that yields the highest average performance based on metrics such as F1-score or accuracy is chosen as the optimal estimator to ensure reliability in deployment.

The number of neighbours was systematically varied from 3 to 9 in the k-NN classifier. Both weighting schemes ’distance’ and ’uniform’ were employed, along with distance metrics such as ’euclidean’ and ’minkowski’. For the DT classifier, optimization involved adjusting the

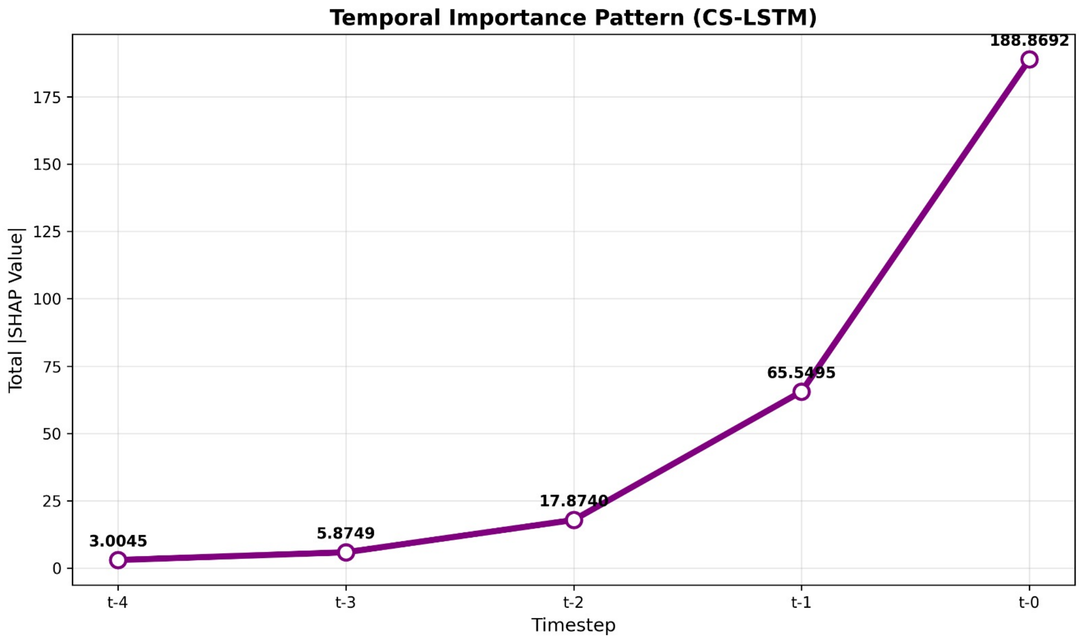


Figure 6.5: Temporal Importance of CS-LSTM

splitting criteria, specifically 'entropy' and 'gini', and altering the maximum depth, with options including none, 10, 20, and 30. The RF classifier extended this tuning by varying the number of estimators from 50 to 200 and testing maximum depths of 10 and 20. For AdaBoost, learning rates ranging from 0.01 to 1.0 were evaluated, along with estimator counts between 50 and 200. The Multi-Layer Perceptron (MLP) classifier explored both single and two layer architectures, with neuron counts between 50 and 100, activation functions including 'relu' and 'tanh', and learning rate strategies such as 'adaptive' and 'constant'. Table 6.3 summarizes the final hyperparameter settings selected for each model.

6.2.2.2 Test Results for Classification

The evaluation of classification models was conducted using metrics such as precision, recall, accuracy, and F1-score. Table 6.4 presents the results, while Figure 6.9 provides a comparative visualization of accuracy and F1-score across the models. The RF classifier delivered the best results among all models, achieving 99.24% accuracy, 99.25% precision, and an F1-score of 99.24%, effectively distinguishing both faulty and non-faulty instances. The DT classifier followed closely, reaching 98.99% accuracy, reaffirming the robustness of tree-based approaches. The k-NN model exhibited impressive results, achieving an accuracy and F1-score of 98.67%, which suggests that distinct local patterns in the dataset are effectively captured

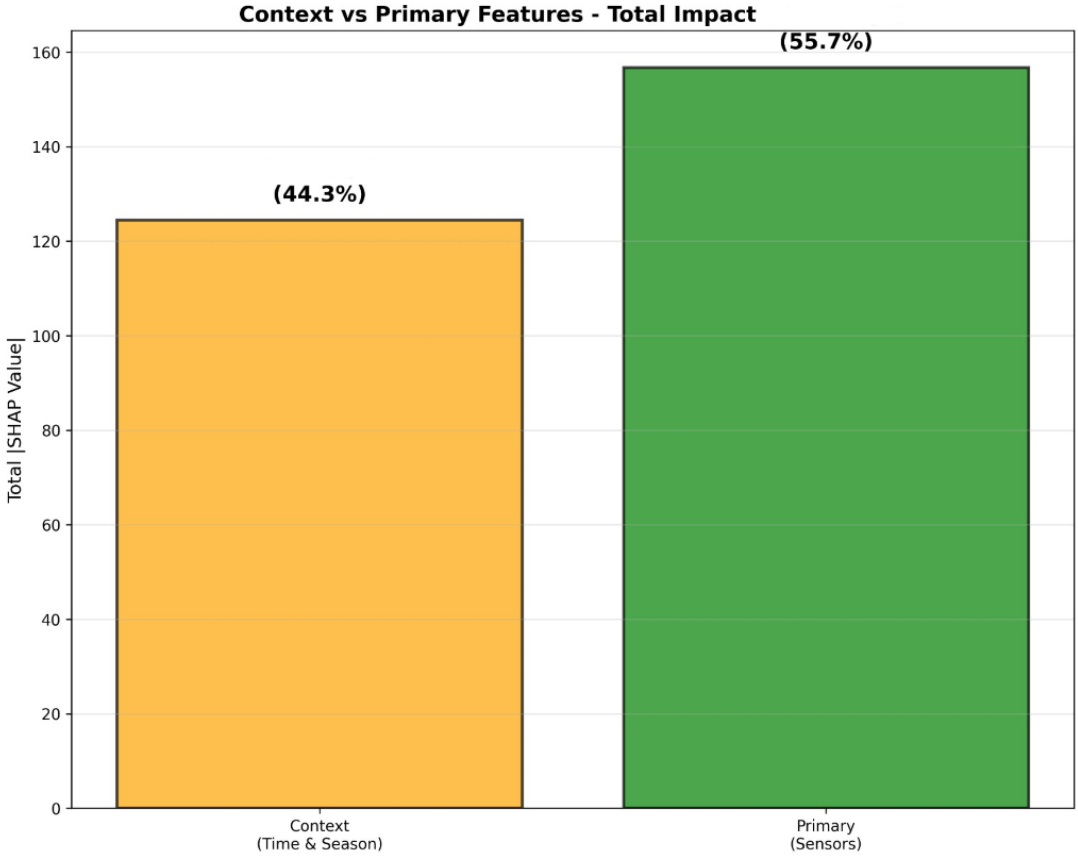


Figure 6.6: Context vs. Primary Features: Comparison of Total Impact

through proximity-based voting. In contrast, the MLP attained an accuracy of 94.21% with 94.08% as F1-score. Meanwhile, AdaBoost had the lowest performance, with an accuracy of 64.16% and an F1-score of 59.73%. As evident from the results, all classifiers except AdaBoost show minimal false positives and false negatives.

Figure 6.10 shows the confusion matrix of the RF model. The matrix indicates that the 'Earth Fault' class accounts for the majority of both false positives and false negatives, highlighting the difficulty in accurately identifying this specific fault. To investigate further, Figure 6.11 presents the signal patterns for earth faults from two inverters. These plots suggest that the signals associated with earth faults exhibit minimal distinction from normal operational data, which poses a challenge for conventional classifiers to reliably detect them.

Table 6.3: Final Hyperparameters for Classification Models

Model	**Hyperparameter**	**Value**
k-NN	Neighbors	5
	Distance Metric	euclidean
Decision Tree	Criterion	entropy
	Max Depth	none
Random Forest	Estimators	100
	Max Depth	null
	Criterion	entropy
AdaBoost	Learning Rate	0.1
	Estimators	50
MLP Classifier	Layers	100
	Activation	relu
	Learning Rate	0.001
	Optimizer	adam

Table 6.4: Classification Performance of Different Models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Random Forest	99.2	99.2	99.2	99.2
Decision Tree	98.9	98.9	98.9	98.9
k-Nearest Neighbors	98.7	98.7	98.7	98.7
MLP Classifier	94.2	94.6	94.2	94.0
AdaBoost	64.2	57.0	64.2	59.7

TH-4096_186151013

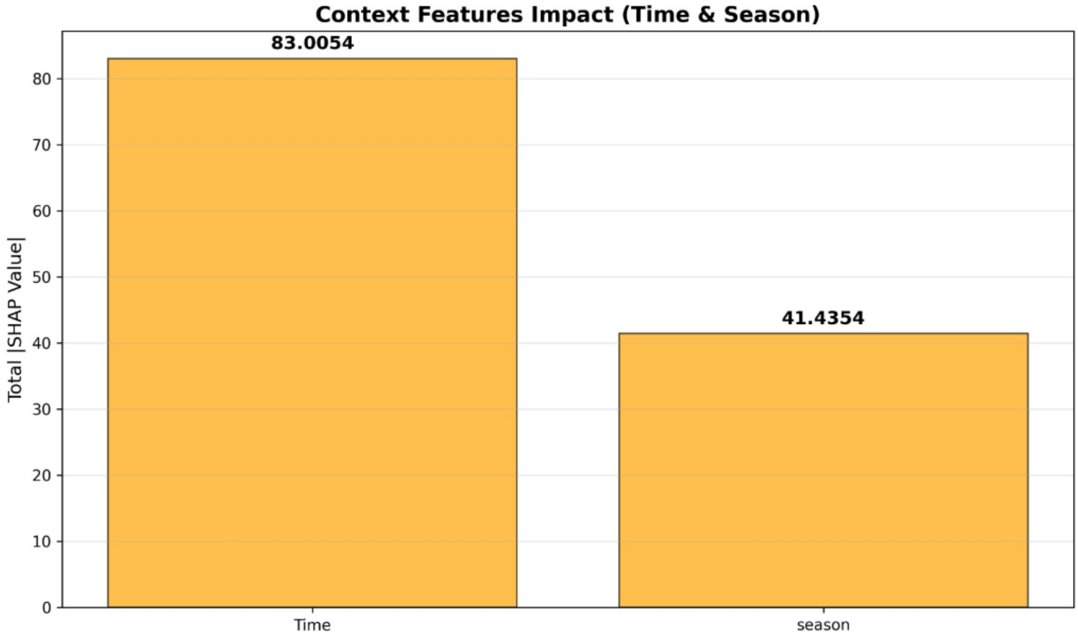


Figure 6.7: Context Features Impact

6.2.3 Integrated Fault Detection and Classification

The system for detecting and classifying faults utilises a CS-LSTM regression model, which is paired with a hierarchical method for classification. As detailed in Section 6.1.2, each input sample is first processed to estimate the expected AC power. The framework then calculates the deviation between this estimated value and the actual measured AC power. Simultaneously, a classification model assigns a fault label to the input. Using both the deviation and the predicted label, the decision logic determines whether the situation represents a non-serious (warning) or serious fault. The purpose of this evaluation is to assess the framework's effectiveness in identifying and classifying faults. The test dataset comprises data from normal operations on specific days, along with fault scenarios documented in the fault log. This log contains data spanning 15 days, detailing faults in 9 different inverters. However, data from three days in July—when Inverter 3 experienced continuous failure—were excluded from the analysis.

6.2.3.1 Results and Discussion

Table 6.5 shows comparison between the actual fault times and the instances where the proposed framework identified potential faults (non-serious faults). The detection of a non-serious fault

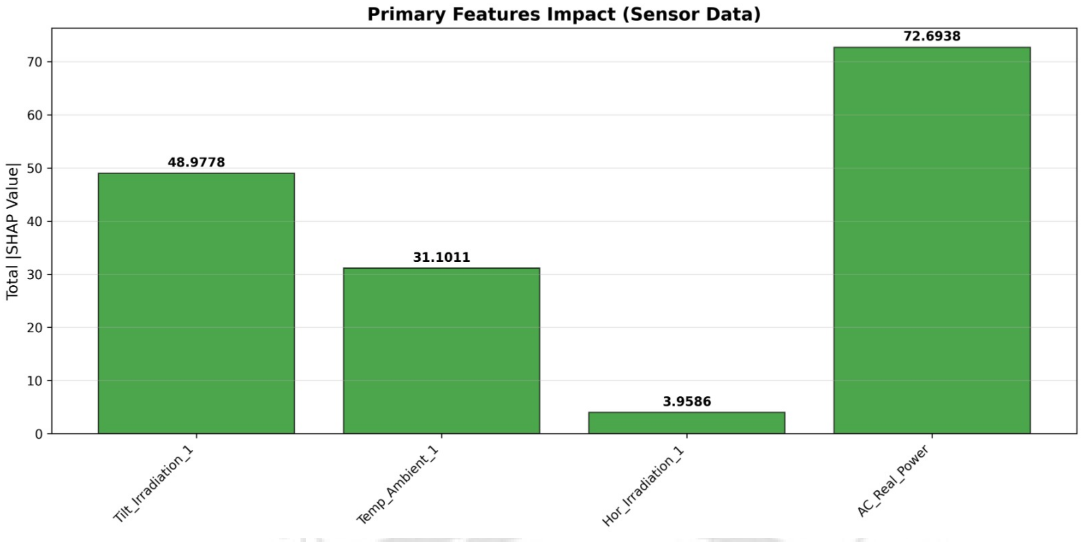


Figure 6.8: Primary Features Impact

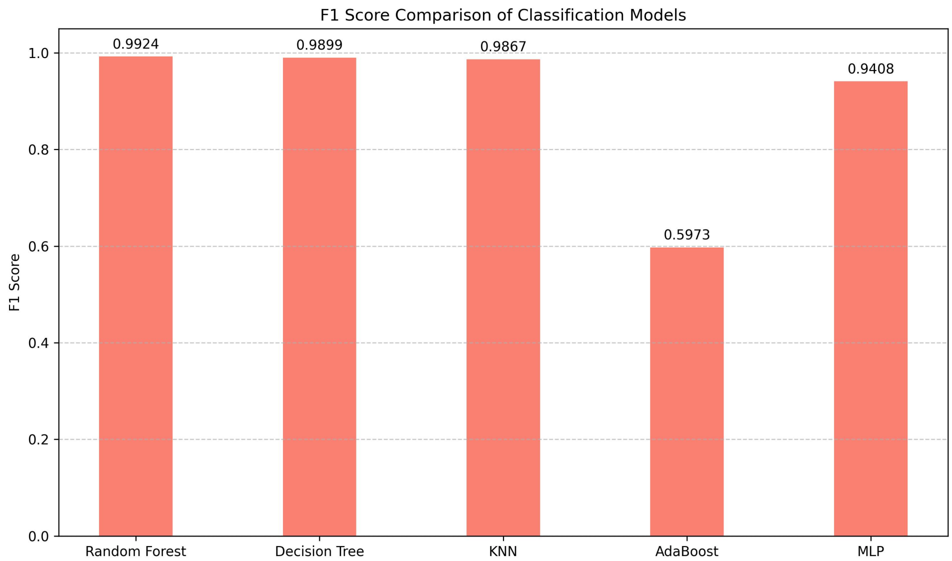


Figure 6.9: F1-scores of the classifiers

signals an impending issue, indicating a likely future fault. The framework demonstrated early warning capabilities in most cases, correctly predicting 7 out of 12 fault events in advance. On average, it provided alerts approximately two hours before the actual fault occurred. However, early detection was not achieved in all cases—particularly for faults that took place in the early morning, such as those in Inv-14 and Inv-4. In the case of Inv-8, the framework detected the fault at the exact time it occurred, offering no advance warning. For faults detected after their occurrence, the average delay was around 56 minutes, with these delays mainly linked to early morning events. Figure 6.12 shows two examples from Inverter 9: in (a), the system successfully flagged the fault early, while in (b), it missed the prediction. The missed detection

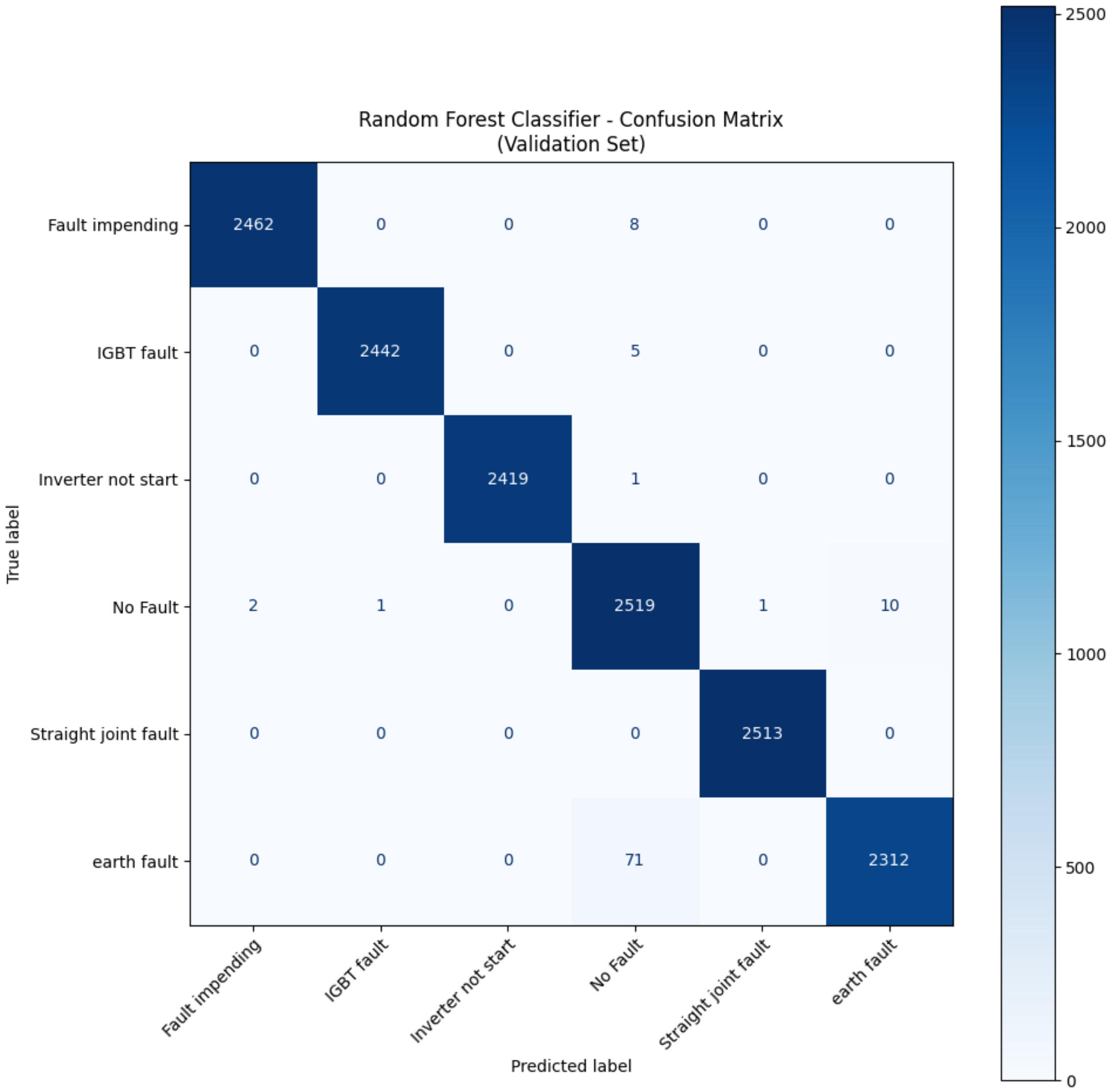


Figure 6.10: Confusion matrix for RF classifier

in the second instance was due to both estimated and measured AC power values being low during early hours, resulting in an insufficient deviation to trigger an alert.

Table 6.6 presents a comparison between the actual class labels with those predicted by the framework for serious faults. A fault is classified as serious if a non-serious fault occurs in two successive intervals in time. Upon confirmation of a serious fault, the classification model is employed to ascertain the fault type. The experimental findings reveal that the system accurately identified the fault type in 7 out of 12 instances. However, in two cases involving earth faults, the framework completely failed in detecting the serious faults. In a different instance (Inv-15), the system identified the fault but could not ascertain its type. For the other three cases, the

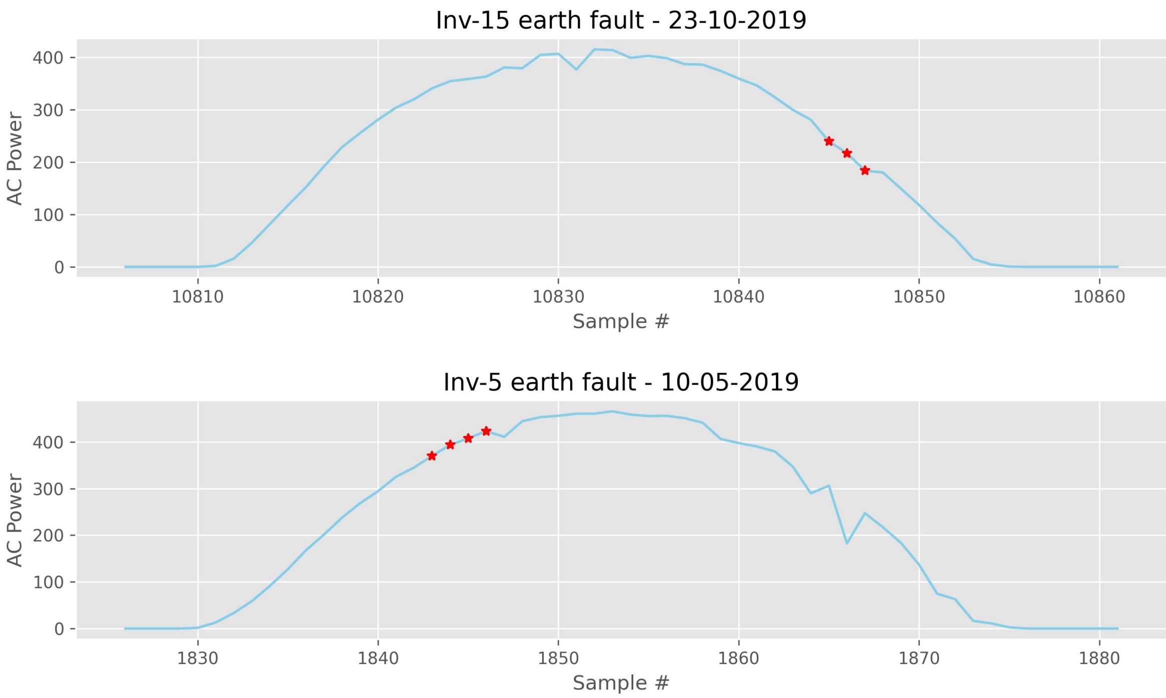


Figure 6.11: Earth fault in two inverters (Inv-15 and Inv-5)

model failed to correctly determine the type of fault. It is worth noting that classification is event-driven rather than applied to each individual time step—once a serious fault is detected, the first classification outcome is used, and all subsequent samples are treated as faulty until the issue is resolved.

It was seen that IGBT faults that occurred in the early morning could not be consistently classified accurately. To investigate this phenomenon, both a correctly identified case and a misclassified instance were analyzed. In the accurately classified event of Inv-7 on 16 September, 2019, depicted in Figure 6.13, there was a quite visible decline in power during the period of fault. In contrast, the misclassified case on 09 July, 2019 (Inv-9), illustrated in Figure 6.12(b), shows a continuous increase in power output despite the presence of a fault. This trend resembles the typical morning rise in power due to increasing irradiance, which likely obscured the fault indicators and led to a misclassification by the model[113].

To further evaluate the framework's robustness, it was tested on data from five randomly selected days when the PV system was operating normally, with the goal of assessing false positive rates. Each data point was individually analysed to check for incorrect fault detection and classification. The results, summarised in the confusion matrix in Figure 6.14, show that out of 3,479 samples, the system correctly identified 3,361 as normal, resulting in an accuracy of

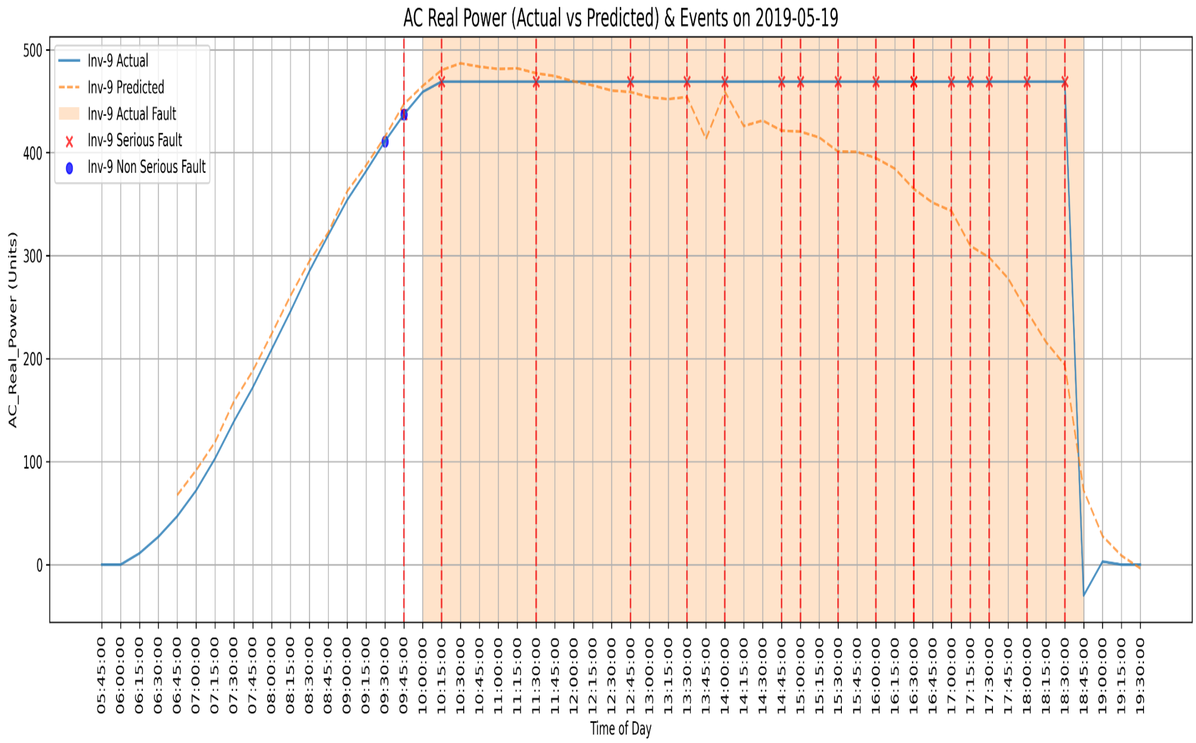
Table 6.5: Predicted Time of Faults versus Actual Time of Faults

Fault Day	Inverter	PredictedTime of Fault	ActualTime of Fault	Lead Time	EarlyPrediction
10-05-2019	Inv-5	**09:15**	10:00	0:45 hr	**Yes**
19-05-2019	Inv-9	**09:30**	10:00	0:30 hr	**Yes**
26-06-2019	Inv-6	**11:45**	14:30	2:45 hrs	**Yes**
05-07-2019	Inv-3	**11:00**	14:00	3 hrs	**Yes**
07-07-2019	Inv-14	07:00	06:15	-0:45 hr	**No**
09-07-2019	Inv-9	07:00	06:30	-0:30 hr	**No**
10-08-2019	Inv-7	09:30	07:45	-1:45 hr	**No**
16-09-2019	Inv-7	**09:30**	14:45	5:15 hrs	**Yes**
25-09-2019	Inv-8	08:00	08:00	0 hrs	**No**
04-10-2019	Inv-4	07:00	06:15	-0:45 hr	**No**
23-10-2019	Inv-15	**15:15**	15:30	0:15 hr	**Yes**
07-02-2020	Inv-15	**08:45**	12:00	3:15 hrs	**Yes**

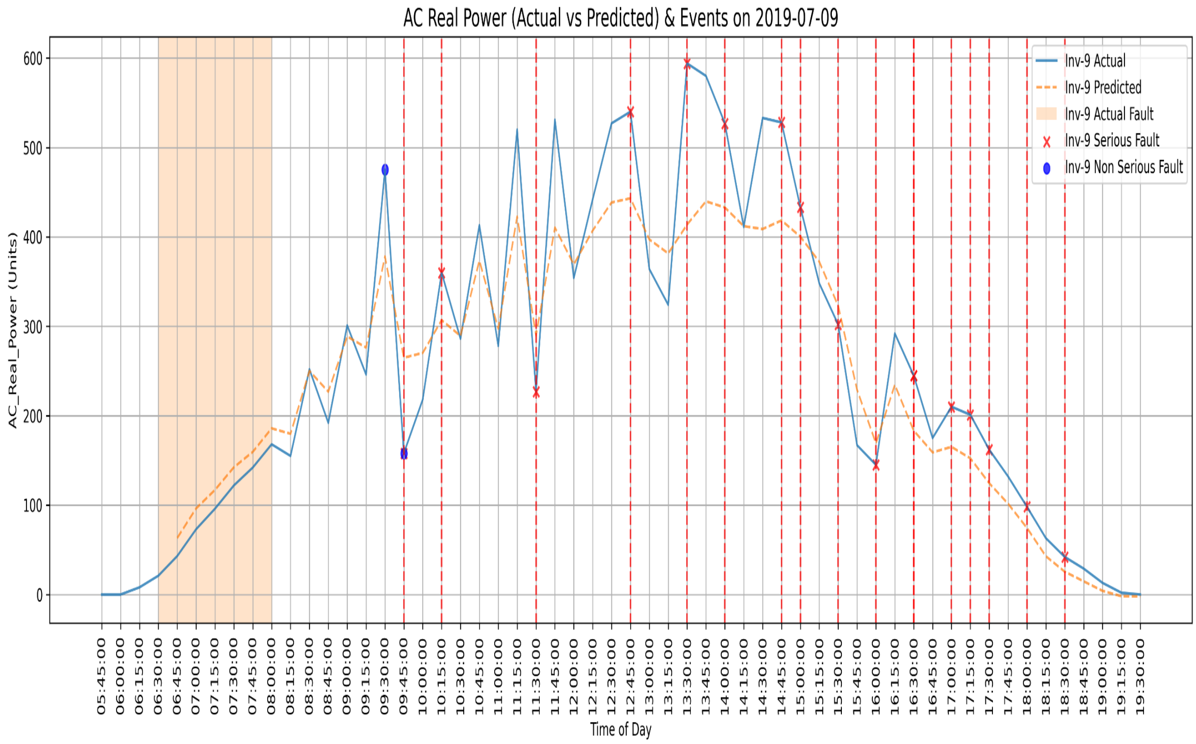
Table 6.6: Actual Faults versus Predicted Serious Faults

Inverter	Fault Day	StartTime	StopTime	Actual Fault	Fault Predicted
Inv-5	10-05-2019	10:00	10:45	Earth fault	No serious faultdetected
Inv-9	19-05-2019	10:00	18:45	**Straight joint fault**	**Straight joint fault**
Inv-6	26-06-2019	14:30	15:30	**Earth fault**	**Earth fault**
Inv-3	05-07-2019	14:00	15:00	Earth fault	No serious faultdetected
Inv-14	07-07-2019	06:15	09:00	**Overtemperaturefault**	**Overtemperature fault**
Inv-9	09-07-2019	06:30	08:00	IGBT fault	Earth fault
Inv-7	10-08-2019	07:45	11:30	Inverter not start	Earth fault
Inv-7	16-09-2019	14:45	15:15	**IGBT fault**	**IGBT fault**
Inv-8	25-09-2019	08:00	08:45	**IGBT fault**	**IGBT fault**
Inv-4	04-10-2019	06:45	12:00	**Inverter not start**	**Inverter not start**
Inv-15	23-10-2019	15:30	16:00	Earth fault	Fault Impending
Inv-15	07-02-2020	12:00	12:45	**Earth fault**	**Earth fault**

TH-4096_186151013



(a) Early Detection of Fault in Inv-9



(b) Delayed Fault Detection in Inv-9

Figure 6.12: Early and Delayed Fault Detection in Inv-9

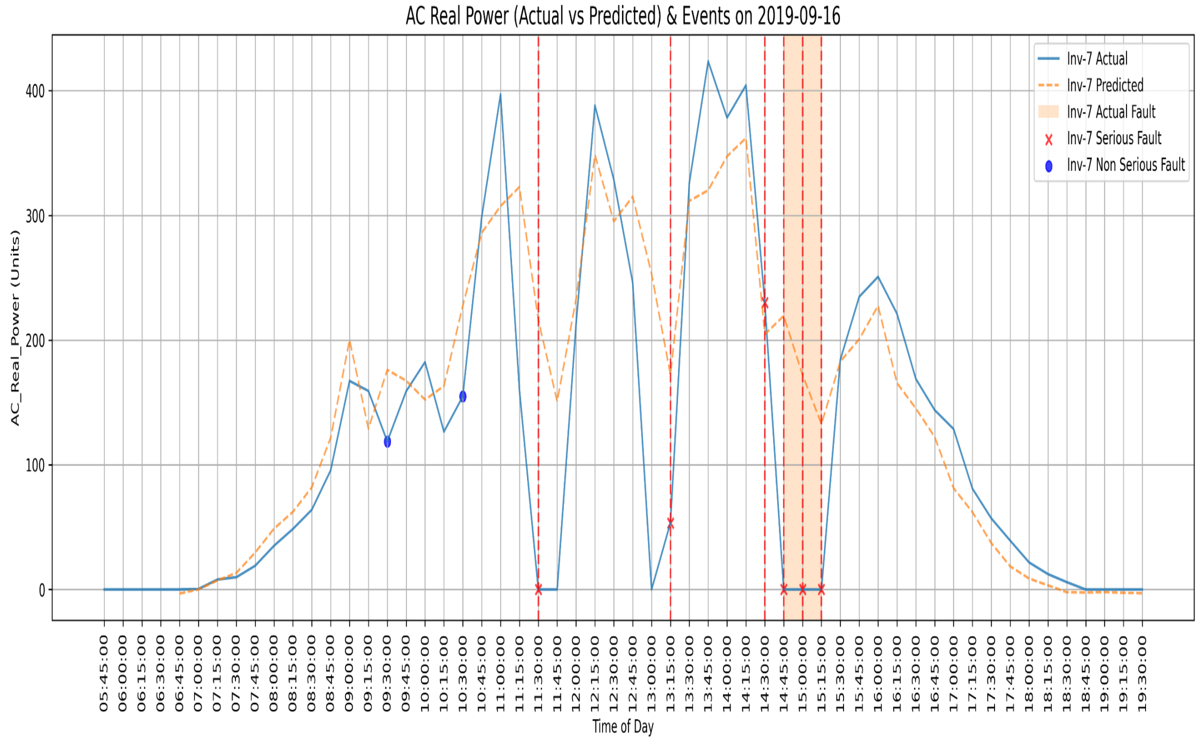


Figure 6.13: Inv-7 IGBT Fault

approximately 96%. However, 101 samples were misclassified as earth faults, and 17 samples were wrongly labelled as overtemperature faults. Despite these errors, the model achieved a recall of 0.966, a precision of 1.0, and an F1-score of 0.983, demonstrating high reliability with less misclassification[113].

6.3 Summary

This chapter presented a context-aware framework for predictive maintenance in solar PV systems. This framework combines a novel regression model, known as CS-LSTM, with a classification technique that employs an RF classifier. A major contribution of this work lies in the integration of contextual variables—such as environmental and operational factors—into both predictive and diagnostic components of the system. The CS-LSTM model incorporates these contextual features through a weighting mechanism that modulates the influence of primary inputs, enabling more accurate estimation of AC power. Anomalies are flagged based on deviations from the predicted power, while a separate classification module, trained on a class-balanced dataset using SMOTE, identifies the type of fault and provides early warnings through

TH-4096_186151013

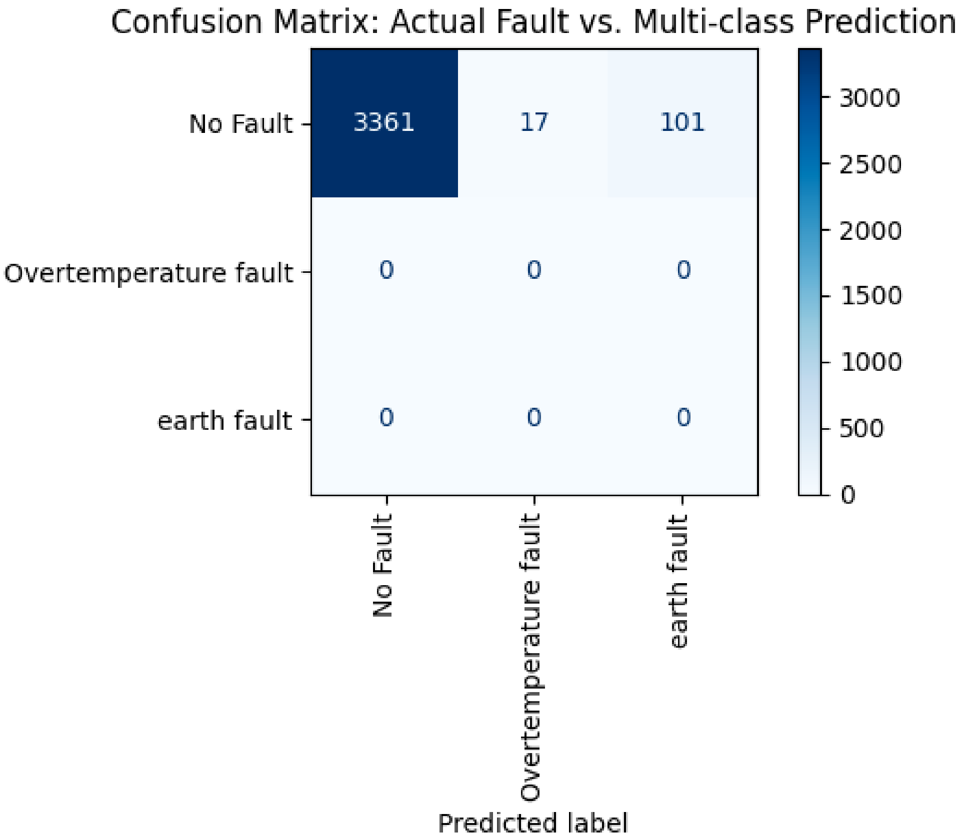


Figure 6.14: Confusion matrix for normal operation

a dedicated "Fault Impending" class.

Experimental assessment on real-world SCADA data confirms the effectiveness of the approach, with the CS-LSTM outperforming both the CNN-LSTM and baseline LSTM models. Additionally, the RF classifier is found to yield the highest classification metrics. The system demonstrated the capability to detect faults up to two hours in advance, enhancing real-time monitoring and predictive maintenance. However, limitations were observed in early-morning fault detection under low-irradiance conditions and in instances with ambiguous fault signatures. Future work will aim to address these challenges by incorporating domain-specific heuristics, leveraging federated learning for scalable deployment, and exploring transfer learning techniques to adapt the framework to diverse PV installations.

Chapter 7

Summary, Conclusion, Limitations, and Future Work

This thesis investigated various Artificial Intelligence (AI) techniques including traditional Machine Learning (ML) models, Recurrent Neural Network (RNN) based sequential models, context integrated ML models, and contextual Deep Learning (DL) models to predict and classify faults in solar PV systems. The conventional ML models are not effective for early prediction of faults in solar PV systems. However, significant improvements are observed in early fault prediction with our proposed methodologies with RNN-based sequential models, and Context Sensitive Long Short Term Memory (CS-LSTM) models when contextual features are incorporated with these models. It is anticipated that the contributions made in this thesis can be applied for early fault prediction of PV systems well in advance. This approach is part of predictive maintenance in order to schedule maintenance activities before the occurrence of faults.

The research started with an analysis of inverter data obtained from a PV plant with a capacity of 10 MW connected to a grid, designed and developed by Azure Power (Haryana) Pvt. Ltd., which is operational in the state of Gujarat in India. Initially, traditional ML models were developed for fault detection. As the research progressed further, we delved into contextual ap-

TH-4096_186151013

proaches for detecting PV faults and predicting faults early. Finally, we developed a predictive maintenance framework for solar PV systems. This chapter presents an overall summary of the research and concludes with a future direction of research.

7.1 Summary

With the aim of early prediction of faults in PV systems, various AI-based data-driven techniques for the last 15 years (from 2010 to 2025) were reviewed in Chapter 2. As part of the review, data driven approaches for diagnostics of faults in solar PV systems using traditional AI based techniques were reviewed in great detail according to the following categories: (i) based on Artificial Neural Network(ANN), (ii) based on Support Vector Machines (SVM), (iii) based on Gaussian Mixture Model (GMM), (iv) based on Collaborative Filtering (CF), (v) based on Fuzzy Logic (FL), (vi) based on Decision Tree (DT) and other advanced ML techniques, (vii) based on Deep Learning (DL). Some of the data-driven approaches that were reviewed on diagnostics of faults in Solar PV systems using DL techniques were based on Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM), Deep Belief Networks (DBN), and Autoencoders (AE). Based on our review, it was observed that there was very limited use of context features for fault prediction in solar PV systems. Considering this limitation as the main motivation, we considered real solar PV SCADA data obtained from Azure Power(Haryana) Pvt. Ltd. We also considered SCADA data obtained from laboratory for evaluation of the various models. This work focused on developing data-driven approaches for anomaly and fault detection in grid-connected solar PV systems using SCADA data. The study was carried out over multiple phases and datasets, with the overall objective of advancing predictive maintenance strategies for grid-connected PV systems.

System description and data analysis have been described in Chapter 3. Initially, an extensive analysis of real solar PV SCADA data obtained from Azure Power (Haryana) Pvt. Ltd, comprising 16 inverters, was conducted. The analysis highlighted temporal and seasonal variations in AC power generation and revealed distinct fault patterns associated with inverter-related faults in few cases. Both real-world operational data and a publicly available laboratory dataset were analyzed to understand differences between naturally occurring and artificially introduced

TH-4096_186151013

faults.

In Chapter 4, we developed baseline models for predicting AC power and detecting faults using traditional ML and DL methods. The contributions of chapter 4 can be summarised as shown below:

- Developing basic models for fault prediction using regression for all the inverters for comparative analysis.

  We developed six models using various ML techniques for estimation of AC Power supply, namely, Linear regression (LR), Random Forest Regressor (RFR), Support Vector Regression (SVR), Multi-Layer Perceptron (MLP), Ensemble voting Regressor (EVR), and Ensemble Stacking Regressor (ESR). Among all the models, RFR was found to be the best performing model, and therefore, was retained for fault detection for the next stage.

- Developing basic models for fault prediction using classification for all the 16 inverters for comparative analysis. We developed five approaches to train the classifiers, namely, Logistic Regression (LGR), Support Vector Machine (SVM), Multi Layer Perceptron (MLP), Decision Trees (DT), and Random Forest (RF).

  In these set of experiments, SVM performed better than all other classifiers, and after SVM, MLP provided better scores. However, overall the classifier models did not give satisfactory results.

- To capture temporal dependencies in SCADA data, sequential models such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) networks were developed for predicting AC power and detecting faults. Among the sequential models, GRU produced the best results as compared to RNN and LSTM in terms of RMSE and R2-score for prediction of AC Power. When the models are used for threshold-based fault detection, GRU provided the highest Recall while LSTM provided the highest precision. However, sequential models suffered from error accumulation, leading to increased false positives under faulty conditions. We further analysed the false positives and false negatives of GRU model and observed that GRU-based models showed marginally better performance in AC power prediction and higher recall in fault detection, while LSTM provided better F1-scores.

The main objective of Chapter 5 was to investigate the role of contextual information in improving the accuracy and early fault detection capability in solar PV systems. In this chapter, first we investigated all context handling strategies for managing contexts such as contextual normalisation, contextual expansion, adjustment of contextual classification, selection of a contextual classifier, and contextual weighting. This chapter describes how different contextual approaches are integrated into model architectures, how they are trained on the real dataset, and assessment of the various models using common metrics. For all the contextual approaches, we have used regression-based models for fault detection. For evaluating the effectiveness of the various context-handling strategies, five regression models were developed, namely, Linear Regression (LR), Multilayer Perceptron (MLP), Support Vector Regressor (SVR), Random Forest Regressor (RFR), and Long Short-Term Memory (LSTM) networks. To reflect different levels of contextual integration, each regression model was trained under three distinct settings. Only primary features were used for training each model in the first setting, and the resulting models were used as baseline models. In the second setting, we applied contextual normalisation such that context-sensitive primary features were normalised with respect to contextual variables to minimise their dependence on contexts. In the third setting, we have used contextual expansion, where contextual features were appended to the primary features, allowing the models to learn directly from the combined feature space. For contextual model selection, a context-specific modelling strategy was explored by training four separate models, each corresponding to a distinct seasonal context. Among all the models, RFR gave the best performance. Both contextual expansion and contextual normalisation performed well. Across all models, contextual expansion and contextual normalisation strategies yielded slightly better results.

As a key contribution of this thesis, we have also proposed a novel approach using contextual weighting, named as Context Sensitive Long Short Term Memory (CS-LSTM) in Chapter 5. To validate the CS-LSTM model, we experimented with two SCADA datasets, one obtained from a real grid-connected solar PV plant with a capacity of 10 MW, and the other one obtained from a PV system installed in a Lab. For the CS-LSTM model, we have used threshold based approach for fault detection using the real dataset from 10 MW PV plant, and compared its performance with LSTM model, and observed that the CS-LSTM model outperforms the baseline LSTM model across all performance metrics such as accuracy, true positive rate (TPR), and false positive rate (FPR).

TH-4096_186151013

For the second case study considering the dataset obtained from PV system installed in a lab, we have used five predictive models, namely, Linear Regression (LR), Random Forest Regressor (RFR), Artificial Neural Networks (ANNs), LSTM, and CS-LSTM for estimating the DC power at the next time step (t) using the historical values of DC power along with other relevant features. We developed two sets of models, the first set did not take into account the context feature, which is 'Time of the day', whereas, in the second set of models, we considered the context feature. Based on experiments, we found that adding context features to the classical models, namely LR, and RFR did not improve the performance considering RMSE score. Also, LSTM with context also did not perform well compared to LSTM. There is no counterpart of the CS-LSTM without context, because CS-LSTM itself is based on contextual weighting. We selected LR, RFR, and ANN for the next set of experiments for fault detection using simple thresholding, and found that these models could not capture the error deviation because of very low F1 score, and therefore, we did not continue with these models further. Next, we considered LSTM and CS-LSTM for fault detection, and evaluated them using (i) simple threshold based, (ii) local distance based, and (iii) clustering based approaches for fault detection using contextual features and without using contextual features. Based on experiments, it was observed that clustering based approaches provided best F1 scores for all the models, both with contexts and without contexts.

In Chapter 6, we have introduced a predictive maintenance framework for early prediction of PV faults employing a deep learning method that is sensitive to context. This framework combines an innovative CS-LSTM regression model with a supervised classification system utilizing Random Forest. This integration effectively uses both contextual and temporal characteristics, including factors like seasonal changes, time of day, irradiance, and temperature etc. For the purpose of training and assessing the model, SCADA data were obtained from a 10 MW PV plant situated in the western region of India. The CS-LSTM model introduced by Baruah et al. in [18] was employed to estimate AC power under standard conditions, with significant deviations from actual measurements being identified as potential faults. For fault categorization into various types, a supervised learning framework was implemented using RF classifier. This model was trained on data that had been balanced through the application of the Synthetic Minority Oversampling Technique (SMOTE). The novelty of this framework was introduction of an additional "Fault Impending" class into the Random Forest based classification module in order to provide early warning.

TH-4096_186151013

To evaluate the performance of the CS-LSTM model for estimating the AC power output of the solar PV plant under normal operating conditions, eight other regression models are developed. Out of these eight, four models, namely, ANN, RFR, CNN-LSTM, and LSTM, are the baseline configurations. Each baseline configuration has an enhanced version that integrates contextual information (indicated by the “-X” suffix). Thus, the other four developed models are: ANN-X, RFR-X, LSTM-X, and CNN-LSTM-X. Among all the evaluated models for AC power prediction, CS-LSTM is found to be the best-performing model. As a second stage of the experiment, various classification methods such as Decision Trees (DT), Random Forest (RF), k- Nearest Neighbours (k-NN), Multilayer Perceptron (MLP), and AdaBoost were developed for integrating the most effective classifier into the proposed framework with the objective of accurately detecting faults and predicting potential issues via the "Fault Impending" class. Among all classification models, RFR delivered the best results, and therefore RFR was combined with CS-LSTM for developing the comprehensive framework.

7.2 Conclusion

This study demonstrates that traditional ML and DL models can effectively support fault detection and predictive maintenance in solar PV systems. Among all the traditional regression models such as LR, RFR, SVR, MLP, EVR, and ESR, it can be observed that the performances of RFR and ESR are better than those of other models in terms of RMSE and R2-score for AC power prediction, with RMSE and R2-score of RFR being 0.0658 and 0.9154, respectively, as compared to 0.0655 and 0.9160, respectively for ESR. The RF model is selected for further fault detection, as it is computationally faster than ESR models. For fault detection using the RFR, threshold value is kept at 2*sigma, 3*sigma, and 4*sigma, and the accuracy obtained was 0.923, 0.960, and 0.975, respectively. The highest recall(0.373) is obtained with a threshold value of 2*sigma. In our case, false negatives are the anomalies that are tagged as normal, therefore higher recall means less number of false negatives. Based on experiments, it is observed that faults that are caused due to “Earth fault” or “Inverter not start” cannot be detected by the model, as these faults cause a dip in the AC power.

Among all the traditional classification models for fault detection such as LGR, SVM, MLP,

TH-4096_186151013

DT, and RF, SVM is the best performing model in terms of recall as compared to other models with a recall of 0.89. Experiments demonstrate that conventional regression and classification models are capable of identifying certain fault types; however, their performance is limited when faults do not strongly affect AC power output.

Among the deep learning (sequential) models such as RNN, LSTM, and GRU, the performance of GRU is better than RNN and LSTM in terms of RMSE (46.32)[not normalized] and R2-score (0.964) for prediction of AC Power. When the models are used for threshold-based fault detection, GRU provides the highest recall (0.424), while LSTM provides the highest precision (0.222), F1-score (0.265), and accuracy (0.699). The results demonstrate that sequential models suffer from error accumulation, leading to increased false positives under faulty conditions. Analysis of false positives and false negatives of GRU model demonstrates that GRU shows marginally better performance in AC power prediction and higher recall in fault detection. However, the accumulation of prediction errors remains a challenge in sequential fault detection.

As part of the research, to investigate the research questions (RQ1-RQ2) as outlined in the problem definition, contextual features are utilized for early prediction of PV faults. Five strategies for integrating explicit contextual features (i.e., contexts) into classification tasks, namely, contextual normalisation, contextual expansion, contextual model selection, adjustment of contextual classification, and contextual weighting are explored, and four of these (except adjustment of contextual classification) are implemented in this thesis. Contextual normalization normalizes primary features using contextual features to reduce context sensitivity, contextual expansion expands the feature space by including both primary and contextual features, contextual classifier selection uses contextual features to choose a specialized classifier from a set and then applies it to the primary features, adjustment of contextual classification performs initial classification using primary features and then refines it with contextual features, and contextual weighting assigns weights to primary features based on contextual features emphasizing their relevance within specific contexts [121]. These strategies highlight how contextual information can be effectively integrated into machine learning models to enhance their predictive capabilities. The following models were trained for comparing the various context-handling strategies: LR, MLP, RFR, SVR, and LSTM networks. Our training process involved four distinct strategies for managing context. The first model, referred to as the Base model, was trained

TH-4096_186151013

exclusively with primary features. In the second model, the primary features incorporated were normalised with contextual features. The third model utilised a contextual expansion approach, treating contextual features as primary features. Additionally, four separate models were trained for each season, implementing a contextual model selection strategy whereby a model is chosen based on the specific context of the input data.

Based on experiments, the results demonstrate that across all models, incorporating contextual information generally leads to improved prediction accuracy to predict AC power one time-step ahead (i.e., 15 minutes) as compared to the base models. RFR achieves the lowest RMSE overall, indicating its superior ability to capture nonlinear patterns in the data. Specifically, RFR performs best with Contextual Expansion (RMSE = 0.0589), followed by Contextual Normalisation (RMSE = 0.0590), and Contextual Model Selection (RMSE=0.0665) as compared to the base model (RMSE=0.0658). LR and SVR also benefit from the incorporation of contextual features, particularly under Contextual Normalisation, where RMSE is reduced from 0.1026 to 0.0932 for LR and from 0.0984 to 0.0803 for SVR with reference to the base models. The MLP model shows modest improvement with Contextual Normalisation (RMSE = 0.08155), while the LSTM model displays inconsistent behaviour with its performance declining with Contextual Normalisation (RMSE=0.09678) and Contextual Model Selection (RMSE=0.1012), but improving with Contextual Expansion (RMSE=0.0801) as compared to its base model (RMSE=0.0811). The RFR is the best performing model for AC power prediction, making it suitable for fault detection. Fault detection relies on establishing an appropriate threshold to assess the difference between measured and predicted AC power. It is observed that a threshold of 2*sigma provides better fault detection results, as it leads to fewer false negatives and false positives. It is observed that out of 13 faults, 6 faults are detected in advance, while the remaining 7 faults are detected after their actual occurrences. The faults that are detected later than their actual occurrences are dominantly associated with faults occurring in the early morning hours. Overall, the analysis confirms that context-aware strategies enhance model reliability for most approaches, though the degree of improvement varies. Based on the performance of various models for managing contexts, including contextual normalisation, contextual expansion, and contextual model selection, our findings establish that integrating contextual information can improve the performance of the predictive models. Furthermore, our proposed methodologies for fault detection demonstrate the capability for early fault prediction, thereby affirmatively answering the research questions (RQ1-RQ2) mentioned in the problem definition.

TH-4096_186151013

In this thesis, a new approach is proposed for integrating contextual information with LSTM network with the aim of improving the model performance. The contextual information is used in such a way that during learning, the input weights are adapted according to the context, a process known as contextual weighting. The proposed model is named as Context Sensitive-LSTM (CS-LSTM). A key contribution of this thesis is the development of the CS-LSTM model, which integrates explicit contextual information into the learning process. The performance of the CS-LSTM model is analysed with two datasets and the results obtained are compared with existing approaches. The first case study consists of data from real operational plant, and the second case study uses data from PV installation in a Lab which is publicly available. The real dataset consists of mainly inverter related faults. In the first case study, for estimation of AC power, LSTM and CS-LSTM models are used and a simple threshold-based fault detection is applied. The CS-LSTM model achieves high accuracy and low false positive rate (FPR), which is desired, and outperforms its baseline counterpart (i.e., LSTM). Taking into account the real data set from the 10 MW PV plant, the accuracy, TPR, and FPR of CS-LSTM are 96.9%, 45.8% and 2.6%, respectively, compared to 93.2%, 44.0%, and 6.2%, respectively, for LSTM. Thus, it is evident that CS-LSTM performs marginally better than LSTM in terms of accuracy and TPR; however, a significant improvement can be observed in FPR. Thus, experimental evaluation on real dataset demonstrates that CS-LSTM improves fault detection performance in real-world data, particularly in terms of accuracy, recall (i.e., true positive rate(TPR)), and false positive rate (FPR). For the second case study, data from a PV system installed in a Lab is used, and both conventional and sequential models without context and also with context are developed. The second dataset has three types of faults and partial shadowing. The faults are degradation, open circuit and short circuit. For the second case study, the fault detection models rely on regression models for estimation of DC Power at the next time step, and then based on the difference between the predicted and actual outputs at each time step, a data sample is classified as either faulty or normal. Using the second dataset, experiments are conducted with three different approaches for fault detection, namely threshold-based, local distance-based, and clustering-based techniques for fault detection. Finally, we compared the results of these models and observed that for this dataset, context has less influence on the results. One reason may be the frequent introduction of faults into the system (e.g., every 15 minutes), and also insufficient contextual information (e.g., there is only one contextual feature, i.e., "Time of day", and "Season" is not available because the data is collected only for 16 days). However, for this dataset also, both the LSTM and CS-LSTM models provide early detection of faults. Experiments on real SCADA data obtained from PV plants and publicly available data from laboratory validate that context-aware modelling enhances early fault prediction capability and reduces false alarms. This confirms that contextual deep learning is a promising direction for PV system monitoring, especially under realistic and diverse operating conditions. To be precise, the work carried out in this thesis bridges an important gap in the literature by moving beyond fault detection and identification towards early fault detection and predictive maintenance, an area that has been unexplored in solar PV research for so long. This contribution affirmatively answers the research questions (RQ1-RQ2) posed in the problem definition as outlined in Chapter 1.

In this thesis, a novel CS-LSTM+RF framework for predicting PV faults early has also been proposed. The main contribution of this framework is the integration of contextual information, such as operational settings and environmental conditions, into both regression and classification, thereby improving the accuracy of prediction. In the CS-LSTM model, the weights of the primary features are influenced by the integration of contextual features. For effectively estimating AC power, the CS-LSTM regression model is used, while the same model also indicates potential anomalies when deviations become greater than a defined threshold. The Random Forest-based classification module, which is trained with a balanced dataset through SMOTE, successfully identifies different fault types and forecasts potential faults by using a "Fault Impending" category.

Experimental validation demonstrates that incorporating context greatly enhances prediction accuracy, with the CS-LSTM model attaining an R^2-score of 0.9741 and exhibiting better RMSE performance with RMSE value of 0.0397 over all baseline models like ANN, RFR, LSTM, and CNN-LSTM, and their corresponding versions with contextual expansion such as ANN-X, RFR-X, LSTM-X, and CNN-LSTM-X. The results show that incorporating contextual features significantly enhances prediction accuracy across all model types. The ANN model's RMSE (not normalized) drops from 79.223 to 49.137, while its R^2 score improves from 0.7797 to 0.8985 with the inclusion of contextual inputs. The Random Forest Regressor (RFR) shows an even more pronounced improvement, with RMSE (not normalized) decreasing from 85.739 to 33.789 and the R^2 score increasing from 0.7419 to 0.9520. Among all the evaluated regression models, CS-LSTM delivers the most accurate results, achieving an RMSE (not normalized) of 25.93 and an R^2 score of 0.9741. These gains highlight the importance of contextual information in enabling the models to better capture complex patterns in power generation in solar PV systems. These results highlight superior ability of CS-LSTM to model both time-dependent behavior and influence of contextual features. In comparison to LSTM, CS-LSTM reduces the RMSE by 58.96%, indicating a significant performance gain. The SHapley Additive Explanation (SHAP) analysis of the CS-LSTM model shows that primary features account for 55.7% of the model's predictive power as compared to 44.3% for contextual inputs. Contextual feature, especially time of day, have a notable influence on the model's output. Within the contextual features, time of day proves to be nearly twice as impactful compared to the seasonal indicator.

The Random Forest (RF) classifier gives the best results with highest F1-score (0.9924), accuracy (99.24%), and precision (0.9925) in fault classification tasks, and it is the best-performing model considering all the classification metrics. The CS-LSTM+RF-based predictive maintenance framework correctly predicts 7 out of 12 faults in advance, and on average, it provides alerts 2 hours before the occurrence of actual fault. However, in some cases, early warning fails in case of faults occurring in the early morning hours such as those in inverter 4 and inverter 14. In some cases (e.g. in inverter 8), the framework detects faults exactly at the same time the fault occurs, thereby offering no advance warning. For faults detected after their occurrences, the average delay is approximately 56 minutes, with most of these occuring in the early morning hours. The missed detection in the early morning hours is mainly due to both the estimated and measured AC power being very low in the early morning hours, resulting in an insufficient deviation to trigger an alert. However, the comprehensive predictive maintenance framework is capable of accurately forecasting multiple faults as early as two hours before they occur, demonstrating its potential application in the real-time surveillance of solar PV systems. Thus, incorporating contextual information into the CS-LSTM+RF-based predictive maintenance framework significantly enhances early fault detection capability, thus affirmatively answering the primary research question (RQ2) as mentioned in the problem definition.

In this thesis, the robust findings include the usefulness of contextual information such as time of day and seasonal variations, the effectiveness of the CS-LSTM model for context-aware AC power prediction, and the potential of combining regression-based fault detection with classification-based diagnosis. One of the key contributions of the thesis is the development of the CS-LSTM framework which is validated on a real SCADA data obtained from 10 MW PV plant and a laboratory benchmark dataset. For the real SCADA data, context has significant

TH-4096_186151013

influence on the results as compared to the laboratory datasets. Context has greater influence on the results obtained using the real SCADA dataset of 10 MW PV plant, because plant output depends on season, time of day, irradiance variability, and operating conditions; whereas for the laboratory dataset, faults were introduced artificially every 15 minutes under controlled conditions and only limited context (i.e., "Time of day") was available, thus reducing the benefit of contextual modelling. As already mentioned in Chapter 5, for the laboratory dataset, we have used only "Time of day" as the contextual feature, and "Season" is not applicable because the data is available only for 16 days. If we can collect laboratory datasets for several months, then a fair comparison between SCADA datasets and laboratory datasets will be more meaningful. The results obtained from CS-LSTM regression model and the comprehensive framework for predictive maintenance consisting of the CS-LSTM regression model and the Random Forest-based classification model may be considered robust findings for early fault prediction. The context-dependent findings based on experiments include the exact detection threshold, the model performance for specific fault types, the early-warning time, and the transferability of the trained model to other PV installations without retraining or adaptation.

7.3 Limitations

The limitations of the research work are discussed below:

- **Poor performance in detecting faults in early morning:** During early morning hours, because of low solar irradiance, PV current is low resulting into low power output. Moreover, signal to noise ratio is less during early morning hours, and therefore, most PV faults such as partial shading, degradation, open-circuit, hotspot, line-to-line faults are difficult to identify because they are detected using electrical parameters like voltage, current, power etc., and faulty and normal operations look almost similar during early morning. In early morning hours, even healthy PV systems generate significantly low current, and thus low current produced because of faults may become indistinguishable from normal operation during early morning hours. This observation is also discussed in Chapter 5. In the comparison of actual and detected fault occurrence times, it is noted that several late detections are associated with faults that occurred during early morning hours. A

TH-4096_186151013

similar explanation is also provided in Chapter 6 for the integrated fault detection and classification framework. For example, in one misclassified IGBT fault case, the power output increased steadily despite the fault. This pattern resembled the normal morning increase in power caused by increasing irradiance, thereby obscuring the fault signature and leading to misclassification by the model.

In summary, the limitation arises mainly because early morning PV output is low and changes gradually with irradiance. Under such conditions, the absolute deviation between predicted and measured power can be small and comparable to normal variations. Residual-threshold models may therefore classify weak fault signatures as normal, producing false negatives. Early morning faults are difficult to detect because irradiance and power levels are low, and the fault-induced deviation between predicted and actual power may remain below the anomaly threshold. The smooth increase in power during early morning hours can also resemble normal operating behaviour. Consequently, the model tends to under-detect faults during low-irradiance periods.

- **Generalization and transferability of models:** Generalization and transferability of models are essential key requirements for developing reliable fault prediction models for photovoltaic (PV) systems operating under different environmental and varying operational conditions. Generalization represents the capability of a fault prediction model to sustain accurate and reliable performance when evaluated on unseen data that follow the same or a similar operational distribution as the training data. In PV systems, attaining effective generalization is particularly difficult because PV electrical behaviour is strongly affected by continuously changing environmental and contextual conditions, including solar irradiance, ambient and module temperatures, humidity, partial shading, soiling, cloud movement, and seasonal fluctuations. These factors significantly influence the voltage, current, and power output characteristics of the PV array, thereby modifying the observable manifestations of faults. Consequently, the same fault type may exhibit different electrical signatures under different operating conditions. For example, power reduction caused by partial shading during low irradiance periods may appear similar to degradation-related or open-circuit fault conditions. Therefore, a robustly generalized PV fault prediction model should be capable of learning fundamental fault-specific characteristics while remaining insensitive to normal environmental and operational variability, in-

TH-4096_186151013

stead of merely memorizing patterns specific to the training dataset. Transferability refers to the ability of a PV fault prediction model trained on one dataset or PV installation to maintain reliable performance when applied to a different operational environment, PV plant, or climatic condition with minimal retraining. It is an important characteristic for real-world deployment because PV systems differ significantly in terms of geographical location, environmental conditions, hardware configuration, sensor quality, and operational behaviour. In practical PV applications, models are commonly trained using data collected from a specific PV system under particular operating conditions. However, when the same model is deployed on another PV installation, its performance may degrade due to differences in feature distributions between the source and target domains. As a result, identical faults may exhibit different electrical signatures across different PV installations. For example, a fault pattern learned from a utility-scale PV plant in a desert climate may not directly transfer to a rooftop PV system operating in a humid tropical environment due to substantial differences in irradiance variability, temperature profiles, and soiling conditions. Poor transferability limits the scalability and practical applicability of intelligent PV fault prediction systems because retraining models for every individual PV plant requires large volumes of labelled fault data, which are often difficult and expensive to obtain. In this research work, the proposed models were not tested for generalization and transferability of models.

- **Remaining Useful Life (RUL) Estimation:** Estimation of the Remaining Useful Life (RUL) of critical PV system components, including PV arrays, inverters, and other Balance of System (BoS) components, has emerged as an important area of research for enhancing predictive maintenance and improving system reliability. Accurate RUL estimation enables proactive maintenance scheduling, minimises unexpected failures, and reduces operational and maintenance costs. Although significant progress has been made in fault detection and diagnosis of PV systems, the present work is primarily focused on fault prediction and classification. Although estimation of critical PV components such as PV arrays, inverters and other balance of system (BoS) components is also gaining popularity among researchers, however, the current work did not focus on estimation of RUL of PV components. Future research may extend the proposed framework by integrating degradation modelling, prognostics, and machine learning-based health assessment techniques to estimate the RUL of PV system components under varying operating and

TH-4096_186151013

environmental conditions.

7.4 Future Work

Based on the findings of this research work, several directions for future research have been identified:

- **Inverter-specific modelling :** Individual predictive models will be developed for each inverter to improve fault detection accuracy and capture inverter-specific operational behaviour.

- **Enhancing fault detection for early morning hours:** The comprehensive framework for predictive maintenance consisting of CS-LSTM regression model and Random Forest based classification framework is highly effective in early fault prediction well in advance upto 2 hours, however, the framework encounters challenges in forecasting faults that occur in the early morning when irradiance is low, particularly in cases where fault signatures are overlapping. Early morning faults are difficult to detect because irradiance and power levels are low, and the fault-induced deviation between predicted and actual power may remain below the anomaly threshold. The smooth increase in power during early morning hours can also resemble normal operating behaviour. Consequently, the model tends to under-detect faults during low-irradiance periods. Future work will emphasise on improving early-morning fault prediction through domain-specific rules, more specifically time-of-day-specific decision rules, and irradiance-dependent/adaptive thresholds. Moreover, additional features that explicitly represent ramp-rate and low-light operating conditions can also be explored.

- **Extension of CS-LSTM architecture to improve transferability:** The CS-LSTM model will be further refined to improve the true positive rate (TPR) and evaluated using additional real-world data sets from multiple PV plants. Transferability is a critical requirement for developing scalable, robust, and deployment-ready PV fault prediction frameworks capable of operating reliably across diverse environmental and operational conditions. To improve transferability, several advanced learning approaches have been proposed. Transfer learning is one of the most widely adopted techniques, where knowledge learned from a source PV system is adapted to a target system using limited additional training data. In this approach, lower-level feature extraction layers learned from large datasets are reused, while only the final layers are fine-tuned for the target domain. Another promising approach is federated learning. Federated learning enables multiple PV plants to collaboratively train a global model without sharing raw operational data, thereby improving robustness and privacy preservation. Integration of federated learning will be explored in future for scalable deployment across multiple PV plants.

- **Deployment in PV plants:** In future, the proposed frameworks can also be considered for deployment in real solar PV plants for real-time monitoring as pilot projects.

- **Explore richer benchmark datasets with additional contextual features:** More benchmark datasets with richer contextual information including environmental and operational parameters can be explored in future. For example, humidity can also be included as contextual feature, but its influence is very negligible as compared to time of day and season, because the power output is directly proportional to the amount of solar irradiation. However, humidity is also a relevant environmental context for PV operation because it may influence irradiance attenuation and module operating conditions. Though humidity is scientifically relevant; however, it should be included only when it is reliable and consistently available in the dataset. For the real 10 MW SCADA dataset used in our research, humidity was not available consistently with the SCADA variables used for model training. Therefore, humidity was not included in the final CS-LSTM framework. For the laboratory dataset, relative humidity is available as a weather variable, but the selected modelling features are irradiance, PV temperature, past DC power, and time of day. Future work may include humidity as an additional contextual feature when reliable plant-level humidity measurements are available, and feature-selection confirms its contribution. Therefore, humidity can be identified as a useful future contextual feature, and all the experiments will have to be performed again considering humidity as an additional contextual feature. In the future, as more data from different PV plants become accessible, we plan to investigate their integration with additional contextual attributes. It should be noted that real data present challenges that affect model performance. We also aim to delve deeper into contextual aspects and expand the work to address a greater num-

TH-4096_186151013

ber of faults. Moreover, the proposed approaches can be validated using various publicly available datasets in order to establish generalizability and robustness across multiple PV installations.

- **Transformer-based Models:** For modelling long-term temporal dependencies, transformer-based architectures can also be explored for early fault prediction in PV systems.

This page was intentionally left blank.

TH-4096_186151013

Bibliography

[1] S. R. Madeti and S. Singh, "Monitoring system for photovoltaic plants: A review," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 1180–1207, 2017, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.09.088. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032116305792.

[2] M. Muntean et al., "Fossil co2 emissions of all world countries - 2018 report," *EUR 29433 EN, Publications Office of the European Union, Luxembourg, 2018, ISBN 978-92-79-97240-9, doi:10.2760/30158, JRC113738.*, 2018. DOI: https://doi.org/10.2760/30158.

[3] S. P. Sukhatme and J. K. Nayak, *Solar Energy : Principles Of Thermal Collection And Storage, 3Ed.* MCGRAW-HILL EDUCATION (INDIA) LTD, 2010, ISBN: 0070260648. [Online]. Available: https://www.amazon.com/Solar-Energy-Principles-Thermal-Collection/dp/0070260648?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0070260648.

[4] A. Teta et al., "Fault detection and diagnosis of grid-connected photovoltaic systems using energy valley optimizer based lightweight cnn and wavelet transform," *Scientific Reports*, vol. 14, no. 1, p. 18 907, 2024.

[5] 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, ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2017.08.069. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X17307508.

[6] A. Mellit, G. Tina, and S. Kalogirou, "Fault detection and diagnosis methods for photovoltaic systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 1–17, 2018, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2018.03.062. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032118301370.

TH-4096_186151013

[7] A. Modi, F. Bühler, J. G. Andreasen, and F. Haglind, “A review of solar energy based heat and power generation systems,” *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 1047–1064, 2017, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.09.075. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032116305664.

[8] A. Chokor, M. El Asmar, and S. Lokanath, “A review of photovoltaic dc systems prognostics and health management: Challenges and opportunities,” *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, vol. 2016-October, pp. 43–54, 2016.

[9] “National solar mission of india, website: Https://www.indiascienceandtechnology.gov.in/st-visions/national-mission/jawaharlal-nehru-national-solar-mission-jnnsm,” (Date last accessed 7 January-2026). [Online]. Available: https://powermin.gov.in.

[10] “Installed capacity of power in india, website: Https://powermin.gov.in/en/content/generation-capacity,” (Date last accessed 7 January-2026). [Online]. Available: https://powermin.gov.in.

[11] H. Tian, F. Mancilla-David, K. Ellis, E. Muljadi, and P. Jenkins, “Cell-to-module-to-array detailed model for photovoltaic panels,” *Solar Energy*, vol. 86, pp. 2695–2706, Sep. 2012. DOI: 10.1016/j.solener.2012.06.004.

[12] C. S. Solanki, *Solar Photovoltaics : Fundamentals, Technologies and Applications, Third Edition*. PHI Learning Pvt. Ltd., 2015, ISBN: 9788120351110.

[13] A. Goetzberger and C. Hebling, “Photovoltaic materials, past, present, future,” *Solar Energy Materials and Solar Cells*, vol. 62, no. 1, pp. 1–19, 2000, ISSN: 0927-0248. DOI: https://doi.org/10.1016/S0927-0248(99)00131-2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927024899001312.

[14] D. M. Chapin, C. S. Fuller, and G. L. Pearson, “A new silicon p-n junction photocell for converting solar radiation into electrical power,” *Journal of Applied Physics*, vol. 25, no. 5, pp. 676–677, 1954. DOI: 10.1063/1.1721711. eprint: https://doi.org/10.1063/1.1721711. [Online]. Available: https://doi.org/10.1063/1.1721711.

[15] Y. Zhao, J. de Palma, J. Mosesian, R. Lyons, and B. Lehman, “Line–line fault analysis and protection challenges in solar photovoltaic arrays,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 3784–3795, 2013.

TH-4096_186151013

[16] D. S. Pillai, F. Blaabjerg, and N. Rajasekar, "A comparative evaluation of advanced fault detection approaches for pv systems," *IEEE Journal of Photovoltaics*, vol. 9, no. 2, pp. 513–527, 2019.

[17] A. Livera, M. Theristis, G. Makrides, and G. E. Georghiou, "Recent advances in failure diagnosis techniques based on performance data analysis for grid-connected photovoltaic systems," *Renewable Energy*, vol. 133, pp. 126–143, 2019, ISSN: 0960-1481. DOI: `https://doi.org/10.1016/j.renene.2018.09.101`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0960148118311753`.

[18] D. Baruah, S. Subbiah, and S. Chouhan, "Exploiting context for fault detection in solar photovoltaic system using context-sensitive recurrent neural network," in *2023 IEEE International Conference on Power Electronics, Smart Grid, and Renewable Energy (PESGRE)*, 2023, pp. 1–6. DOI: `10.1109/PESGRE58662.2023.10404256`.

[19] M. Mansouri, M. Trabelsi, H. Nounou, and M. Nounou, "Deep learning-based fault diagnosis of photovoltaic systems: A comprehensive review and enhancement prospects," *IEEE Access*, vol. 9, pp. 126 286–126 306, 2021. DOI: `10.1109/ACCESS.2021.3110947`.

[20] 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.

[21] Y. Zhao, L. Yang, B. Lehman, J. de Palma, J. Mosesian, and R. Lyons, "Decision tree-based fault detection and classification in solar photovoltaic arrays," in *2012 Twenty-Seventh Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2012, pp. 93–99.

[22] A. Triki-Lahiani, A. [.-B. Abdelghani], and I. Slama-Belkhodja, "Fault detection and monitoring systems for photovoltaic installations: A review," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 2680–2692, 2018, ISSN: 1364-0321. DOI: `https://doi.org/10.1016/j.rser.2017.09.101`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S1364032117313618`.

[23] Zhehan Yi and A. H. Etemadi, "A novel detection algorithm for line-to-line faults in photovoltaic (pv) arrays based on support vector machine (svm)," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 2016, pp. 1–4.

TH-4096_186151013

[24] D. S. Pillai and N. Rajasekar, "A comprehensive review on protection challenges and fault diagnosis in pv systems," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 18–40, 2018, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2018.03.082. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032118301758.

[25] J. Flicker and J. Johnson, "Electrical simulations of series and parallel pv arc-faults," in *2013 IEEE 39th Photovoltaic Specialists Conference (PVSC)*, 2013, pp. 3165–3172.

[26] K. A. Kim and P. T. Krein, "Hot spotting and second breakdown effects on reverse i-v characteristics for mono-crystalline si photovoltaics," in *2013 IEEE Energy Conversion Congress and Exposition*, 2013, pp. 1007–1014.

[27] K. A. Kim and P. T. Krein, "Photovoltaic hot spot analysis for cells with various reverse-bias characteristics through electrical and thermal simulation," in *2013 IEEE 14th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2013, pp. 1–8.

[28] M. Simon and E. L. Meyer, "Detection and analysis of hot-spot formation in solar cells," *Solar Energy Materials and Solar Cells*, vol. 94, no. 2, pp. 106–113, 2010, ISSN: 0927-0248. DOI: https://doi.org/10.1016/j.solmat.2009.09.016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927024809003420.

[29] L. Cristaldi, M. Faifer, M. Lazzaroni, M. M. A. F. Khalil, M. Catelani, and L. Ciani, "Diagnostic architecture: A procedure based on the analysis of the failure causes applied to photovoltaic plants," *Measurement*, vol. 67, pp. 99–107, 2015, ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2015.02.023. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263224115000810.

[30] A. L. Rosenthal, M. G. Thomas, and S. J. Durand, "A ten year review of performance of photovoltaic systems," in *Conference Record of the Twenty Third IEEE Photovoltaic Specialists Conference - 1993 (Cat. No.93CH3283-9)*, 1993, pp. 1289–1291.

[31] A. Golnas, "Pv system reliability: An operator's perspective," *IEEE Journal of Photovoltaics*, vol. 3, no. 1, pp. 416–421, 2013. DOI: 10.1109/JPHOTOV.2012.2215015.

TH-4096_186151013

[32] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers Industrial Engineering*, vol. 137, p. 106 024, 2019, ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2019.106024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835219304838.

[33] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2213–2227, 2019.

[34] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 812–820, 2015.

[35] E. Sezer, D. Romero, F. Guedea, M. Macchi, and C. Emmanouilidis, "An industry 4.0-enabled low cost predictive maintenance approach for smes," in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2018, pp. 1–8. DOI: 10.1109/ICE.2018.8436307.

[36] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," Nov. 2019.

[37] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, 2018, ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2017.11.024. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327017306064.

[38] J. Campos, "Development in the application of ict in condition monitoring and maintenance," *Computers in Industry*, vol. 60, no. 1, pp. 1–20, 2009, ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2008.09.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166361508000997.

[39] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006, ISSN: 0888-3270. DOI: https://

doi.org/10.1016/j.ymssp.2005.09.012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327005001512.

[40] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Computers Industrial Engineering*, vol. 150, p. 106 889, 2020, ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106889. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835220305787.

[41] G. Smith, J. B. Schroeder, S. Navarro, and D. Haldeman, "Development of a prognostics and health management capability for the joint strike fighter," in *1997 IEEE Autotestcon Proceedings AUTOTESTCON '97. IEEE Systems Readiness Technology Conference. Systems Readiness Supporting Global Needs and Awareness in the 21st Century*, 1997, pp. 676–682.

[42] H. Meng and Y.-F. Li, "A review on prognostics and health management (phm) methods of lithium-ion batteries," *Renewable and Sustainable Energy Reviews*, vol. 116, p. 109 405, 2019, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2019.109405. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032119306136.

[43] "Ieee standard framework for prognostics and health management of electronic systems," *IEEE Std 1856-2017*, pp. 1–31, 2017.

[44] K. Javed, R. Gouriveau, and N. Zerhouni, "State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels," *Mechanical Systems and Signal Processing*, vol. 94, pp. 214–236, 2017, ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2017.01.050. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327017300468.

[45] M. Pecht and R. Jaai, "A prognostics and health management roadmap for information and electronics-rich systems," *Microelectronics Reliability*, vol. 50, no. 3, pp. 317–323, 2010, ISSN: 0026-2714. DOI: https://doi.org/10.1016/j.microrel.2010.01.006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026271410000181.

TH-4096_186151013

[46] Y. Zhao, Q. Liu, D. Li, D. Kang, Q. Lv, and L. Shang, "Hierarchical anomaly detection and multimodal classification in large-scale photovoltaic systems," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 3, pp. 1351–1361, 2019.

[47] D. An, N. H. Kim, and J.-H. Choi, "Practical options for selecting data-driven or physics-based prognostics algorithms with reviews," *Reliability Engineering System Safety*, vol. 133, pp. 223–236, 2015, ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2014.09.014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832014002245.

[48] Y. Peng, M. Dong, and M. Zuo, "Current status of machine prognostics in condition-based maintenance: A review," *International Journal of Advanced Manufacturing Technology*, vol. 50, pp. 297–313, Sep. 2010. DOI: 10.1007/s00170-009-2482-0.

[49] Y. Zhao, R. Ball, J. Mosesian, J. de Palma, and B. Lehman, "Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays," *IEEE Transactions on Power Electronics*, vol. 30, no. 5, pp. 2848–2858, 2015.

[50] L. L. Jiang and D. L. Maskell, "Automatic fault detection and diagnosis for photovoltaic systems using combined artificial neural network and analytical based methods," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–8.

[51] C. Coonick, *Fire and solar pv systems – investigations and evidence, bre national solar centre*, (Date last accessed 13 January-2026), 2018. [Online]. Available: https://assets.publishing.service.gov.uk/media/5c90a30840f0b633ff9a3537/Fires_and_solar_PV_systems-Investigations_Evidence_Issue_2.9.pdf.

[52] S. Firth, K. Lomas, and S. Rees, "A simple model of pv system performance and its use in fault detection," *Solar Energy*, vol. 84, no. 4, pp. 624–635, 2010, International Conference CISBAT 2007, ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2009.08.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X0900187X.

[53] A. J. Hanson, C. A. Deline, S. M. MacAlpine, J. T. Stauth, and C. R. Sullivan, "Partial-shading assessment of photovoltaic installations via module-level monitoring," *IEEE Journal of Photovoltaics*, vol. 4, no. 6, pp. 1618–1624, 2014.

TH-4096_186151013

[54] S. Atcitty, J.E.Granata, M.A.Quintana, and C. A. Tasca, "Utility scale grid-tied pv inverter reliability workshop report," 2011, (Date last accessed 26 July-2020). [Online]. Available: https://energy.sandia.gov/wp-content/gallery/uploads/Inverter_Workshop_FINAL_072811.pdf.

[55] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and Ubiquitous Computing*, H.-W. Gellersen, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 304–307, ISBN: 978-3-540-48157-7.

[56] N. Salazar-Peña, A. Tabares, and A. González-Mancera, "Performance assessment and dynamic fault detection in photovoltaic systems using artificial intelligence," *Energy*, vol. 330, p. 136 759, 2025, ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2025.136759. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544225024016.

[57] S. Voutsinas, D. Karolidis, I. Voyiatzis, and M. Samarakou, "Development of a machine-learning-based method for early fault detection in photovoltaic systems," *Journal of Engineering and Applied Science*, vol. 70, no. 1, p. 27, Apr. 2023, ISSN: 2536-9512. DOI: 10.1186/s44147-023-00200-0. [Online]. Available: https://doi.org/10.1186/s44147-023-00200-0.

[58] M. Aghaei et al., "Autonomous intelligent monitoring of photovoltaic systems: An in-depth multidisciplinary review," *Progress in Photovoltaics: Research and Applications*, vol. 33, no. 3, pp. 381–409, 2025. DOI: https://doi.org/10.1002/pip.3859. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pip.3859. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/pip.3859.

[59] G. M. El-Banby, N. M. Moawad, B. A. Abouzalm, W. F. Abouzaid, and E. A. Ramadan, "Photovoltaic system fault detection techniques: A review," *Neural Computing and Applications*, vol. 35, no. 35, pp. 24 829–24 842, Dec. 2023, ISSN: 1433-3058. DOI: 10.1007/s00521-023-09041-7. [Online]. Available: https://doi.org/10.1007/s00521-023-09041-7.

[60] M. Islam, M. R. Rashel, M. T. Ahmed, A. K. M. K. Islam, and M. Tlemçani, "Artificial intelligence in photovoltaic fault identification and diagnosis: A systematic review," *En-*

TH-4096_186151013

ergies, vol. 16, no. 21, 2023, ISSN: 1996-1073. DOI: 10.3390/en16217417. [Online]. Available: https://www.mdpi.com/1996-1073/16/21/7417.

[61] J. F. Gaviria, G. Narváez, C. Guillen, L. F. Giraldo, and M. Bressan, "Machine learning in photovoltaic systems: A review," *Renewable Energy*, vol. 196, pp. 298–318, 2022, ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2022.06.105. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148122009454.

[62] K. Masita, A. Hasan, T. Shongwe, and H. A. Hilal, "Deep learning in defects detection of pv modules: A review," *Solar Energy Advances*, vol. 5, p. 100 090, 2025, ISSN: 2667-1131. DOI: https://doi.org/10.1016/j.seja.2025.100090. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667113125000038.

[63] D. Riley and J. Johnson, "Photovoltaic prognostics and heath management using learning algorithms," in *2012 38th IEEE Photovoltaic Specialists Conference*, 2012, pp. 001 535–001 539.

[64] W. Chine, A. Mellit, V. Lughi, A. Malek, G. Sulligoi, and A. [Pavan], "A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks," *Renewable Energy*, vol. 90, pp. 501–512, 2016, ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2016.01.036. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148116300362.

[65] H. Mekki, A. Mellit, and H. Salhi, "Artificial neural network-based modelling and fault detection of partial shaded photovoltaic modules," *Simulation Modelling Practice and Theory*, vol. 67, pp. 1–13, 2016, ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2016.05.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X16302040.

[66] Syafaruddin, E. Karatepe, and T. Hiyama, "Controlling of artificial neural network for fault diagnosis of photovoltaic array," in *2011 16th International Conference on Intelligent System Applications to Power Systems*, 2011, pp. 1–6.

[67] M. N. Akram and S. Lotfifard, "Modeling and health monitoring of dc side of photovoltaic array," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1245–1253, 2015.

[68] A. Betti, M. L. L. Trovato, F. S. Leonardi, G. Leotta, F. Ruffini, and C. Lanzetta, "Predictive maintenance in photovoltaic plants with a big data approach," *CoRR*, vol. abs/1901.10855, 2019. arXiv: 1901.10855. [Online]. Available: http://arxiv.org/abs/1901.10855.

[69] M. Charkhgard and M. Farrokhi, "State-of-charge estimation for lithium-ion batteries using neural networks and ekf," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 4178–4187, 2010.

[70] W. Rezgui, L. Mouss, N. K. Mouss, M. D. Mouss, and M. Benbouzid, "A smart algorithm for the diagnosis of short-circuit faults in a photovoltaic generator," in *2014 First International Conference on Green Energy ICGE 2014*, 2014, pp. 139–143.

[71] M. A. Patil et al., "A novel multistage support vector machine based approach for li ion battery remaining useful life estimation," *Applied Energy*, vol. 159, pp. 285–297, 2015, ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2015.08.119. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261915010557.

[72] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, and K. Dietmayer, "Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods," *Journal of Power Sources*, vol. 239, pp. 680–688, 2013, ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2012.11.146. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775312018605.

[73] Y. Zhao, D. Li, T. Lu, Q. Lv, N. Gu, and L. Shang, "Collaborative fault detection for large-scale photovoltaic systems," *IEEE Transactions on Sustainable Energy*, pp. 1–1, 2020.

[74] 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*, 2011, pp. 1341–1346.

[75] L. Bonsignore, M. Davarifar, A. Rabhi, G. M. Tina, and A. Elhajjaji, "Neuro-fuzzy fault detection method for photovoltaic systems," English, *Energy Procedia*, vol. 62, no. Complete, pp. 431–441, 2014. DOI: 10.1016/j.egypro.2014.12.405.

TH-4096_186151013

[76] M. Abdelsattar, A. AbdelMoety, and A. Emad-Eldeen, "Advanced machine learning techniques for predicting power generation and fault detection in solar photovoltaic systems," *Neural Computing and Applications*, vol. 37, no. 15, pp. 8825–8844, May 2025, ISSN: 1433-3058. DOI: 10.1007/s00521-025-11035-6. [Online]. Available: https://doi.org/10.1007/s00521-025-11035-6.

[77] H. N. Noura, Z. Allal, O. Salman, and K. Chahine, "Explainable artificial intelligence of tree-based algorithms for fault detection and diagnosis in grid-connected photovoltaic systems," *Engineering Applications of Artificial Intelligence*, vol. 139, p. 109 503, 2025, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2024.109503. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197624016610.

[78] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[79] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527. [Online]. Available: https://doi.org/10.1162/neco.2006.18.7.1527.

[80] M. A. Ranzato, Y.-L. Boureau, and Y. LeCun, "Sparse feature learning for deep belief networks," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07, Vancouver, British Columbia, Canada: Curran Associates Inc., 2007, pp. 1185–1192, ISBN: 9781605603520.

[81] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06, Canada: MIT Press, 2006, pp. 153–160.

[82] Y. Bengio, "Learning deep architectures for ai," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009, ISSN: 1935-8237. DOI: 10.1561/2200000006. [Online]. Available: https://doi.org/10.1561/2200000006.

[83] S. Pouyanfar et al., "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, Sep. 2018, ISSN: 0360-0300. DOI: 10.1145/3234150. [Online]. Available: https://doi.org/10.1145/3234150.

TH-4096_186151013

[84] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539. [Online]. Available: https://doi.org/10.1038/nature14539.

[85] W. J. Zhang, G. Yang, Y. Lin, C. Ji, and M. M. Gupta, "On definition of deep learning," in *2018 World Automation Congress (WAC)*, 2018, pp. 1–5.

[86] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014, ISSN: 1932-8346. DOI: 10.1561/2000000039. [Online]. Available: http://dx.doi.org/10.1561/2000000039.

[87] Wikipedia contributors, *Deep learning — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=958523497, [Online; accessed 25-May-2020], 2020.

[88] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016, ISBN: 0262035618.

[89] T. Huuhtanen and A. Jung, "Predictive maintenance of photovoltaic panels via deep learning," in *2018 IEEE Data Science Workshop (DSW)*, 2018, pp. 66–70.

[90] F. Aziz, A. Ul Haq, S. Ahmad, Y. Mahmoud, M. Jalal, and U. Ali, "A novel convolutional neural network-based approach for fault classification in photovoltaic arrays," *IEEE Access*, vol. 8, pp. 41 889–41 904, 2020.

[91] Y. Tan, K. Liao, X. Bai, C. Deng, Z. Zhao, and B. Zhao, "Denoising convolutional neural networks based dust accumulation status evaluation of photovoltaic panel," in *2019 IEEE International Conference on Energy Internet (ICEI)*, 2019, pp. 560–566.

[92] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[93] X. Li, Q. Yang, Z. Lou, and W. Yan, "Deep learning based module defect analysis for large-scale photovoltaic farms," *IEEE Transactions on Energy Conversion*, vol. 34, no. 1, pp. 520–529, 2019.

[94] R. Priyadarshini, P. Manoharan, and S. Roomi, "Efficient net-based deep learning for visual fault detection in solar photovoltaic modules," *Tehnički vjesnik*, vol. 32, no. 1, pp. 233–241, 2025.

[95] N. V. Sridharan and V. S. and, "Convolutional neural network based automatic detection of visible faults in a photovoltaic module," *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 47, no. 1, pp. 6270–6284, 2025. DOI: 10.1080/15567036.2021.1905753. eprint: https://doi.org/10.1080/15567036.2021.1905753. [Online]. Available: https://doi.org/10.1080/15567036.2021.1905753.

[96] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990, ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. [Online]. Available: https://www.sciencedirect.com/science/article/pii/036402139090002E.

[97] H. Salehinejad, J. Baarbe, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *CoRR*, vol. abs/1801.01078, 2018. arXiv: 1801.01078. [Online]. Available: http://arxiv.org/abs/1801.01078.

[98] A. Y. Appiah, X. Zhang, B. B. K. Ayawli, and F. Kyeremeh, "Long short-term memory networks based automatic feature extraction for photovoltaic array fault diagnosis," *IEEE Access*, vol. 7, pp. 30 089–30 101, 2019.

[99] G. Liu and W. Yu, "A fault detection and diagnosis technique for solar system based on elman neural network," in *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, 2017, pp. 473–480.

[100] X. Liu et al., "Resgru: A novel hybrid deep learning model for compound fault diagnosis in photovoltaic arrays considering dust impact," *Sensors*, vol. 25, no. 4, 2025, ISSN: 1424-8220. DOI: 10.3390/s25041035. [Online]. Available: https://www.mdpi.com/1424-8220/25/4/1035.

[101] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5695–5705, 2018.

[102] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[103] M. Massaoudi, H. Abu-Rub, S. S. Refaat, M. Trabelsi, I. Chihi, and F. S. Oueslati, "Enhanced deep belief network based on ensemble learning and tree-structured of parzen estimators: An optimal photovoltaic power forecasting method," *IEEE Access*, vol. 9, pp. 150 330–150 344, 2021.

[104] C. Tao, X. Wang, F. Gao, and M. Wang, "Fault diagnosis of photovoltaic array based on deep belief network optimized by genetic algorithm," *Chinese journal of electrical engineering*, vol. 6, no. 3, pp. 106–114, 2020.

[105] Wikipedia contributors, *Autoencoder — Wikipedia, the free encyclopedia*, `https://en.wikipedia.org/w/index.php?title=Autoencoder&oldid=957588290`, [Online; accessed 25-May-2020], 2020.

[106] A. Hadrawi, Y. S. Akil, D. Utamidewi, et al., "A deep autoencoder-based method for detecting mismatch faults in photovoltaic modules," in *2024 International Conference on Electrical Engineering and Computer Science (ICECOS)*, IEEE, 2024, pp. 1–6.

[107] A. Dairi, F. Harrou, Y. Sun, and S. Khadraoui, "Short-term forecasting of photovoltaic solar power production using variational auto-encoder driven deep learning approach," *Applied Sciences*, vol. 10, no. 23, p. 8400, 2020.

[108] A. Seghiour, H. A. Abbas, A. Chouder, and A. Rabhi, "Deep learning method based on autoencoder neural network applied to faults detection and diagnosis of photovoltaic system," *Simulation Modelling Practice and Theory*, vol. 123, p. 102 704, 2023, ISSN: 1569-190X. DOI: `https://doi.org/10.1016/j.simpat.2022.102704`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1569190X22001733`.

[109] A. E. Lazzaretti et al., "A monitoring system for online fault detection and classification in photovoltaic plants," *Sensors*, vol. 20, no. 17, p. 4688, 2020.

[110] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[111] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[112] B. Schmidt, D. Galar, and L. Wang, “Context awareness in predictive maintenance,” *Lecture Notes in Mechanical Engineering*, pp. 197–211, Jan. 2016. DOI: 10.1007/978-3-319-23597-4_15.

[113] D. Baruah, N. Das, S. Chouhan, and S. Subbiah, “A comprehensive framework for predictive maintenance in solar photovoltaic systems using contextual deep model,” *IEEE Access*, vol. 13, pp. 191 924–191 940, 2025. DOI: 10.1109/ACCESS.2025.3627866.

[114] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. [Online]. Available: https://aclanthology.org/D14-1179.

[115] P. Turney, “The management of context-sensitive features: A review of strategies,” in *13th International Conference on Machine Learning (ICML96), Workshop on Learning in Context-Sensitive Domains, Bari, Italy, July*, 1996, pp. 60–66.

[116] X. Wang and Z. Zhu, “Context understanding in computer vision: A survey,” *Computer Vision and Image Understanding*, vol. 229, p. 103 646, 2023.

[117] T. Mikolov and G. Zweig, “Context dependent recurrent neural network language model,” in *2012 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2012, pp. 234–239.

[118] S. Kulkarni and S. F. Rodd, “Context aware recommendation systems: A review of the state of the art techniques,” *Computer Science Review*, vol. 37, p. 100 255, 2020.

[119] F. Shaik, S. S. Lingala, and P. Veeraboina, “Effect of various parameters on the performance of solar pv power plant: A review and the experimental study,” *Sustainable Energy Research*, vol. 10, no. 1, p. 6, 2023.

[120] P. Turney, “The identification of context-sensitive features: A formal definition of context for concept learning,” in *13th International Conference on Machine Learning (ICML96), Workshop on Learning in Context-Sensitive Domains,Bari, Italy*, 1996, pp. 53–59.

TH-4096_186151013

[121] D. Baruah, R. Roy, R. Ahmed, S. Subbiah, S. Chouhan, and K. Angappan, "Contextual approaches to data-driven fault detection in solar photovoltaic system," in *2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 2024, pp. 1–7. DOI: 10.1109/EAIS58494.2024.10570026.

[122] P. D. Turney, "Exploiting context when learning to classify," in *European Conference on Machine Learning*, Springer, 1993, pp. 402–407.

[123] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735.

[124] R. D. Baruah and M. M. Organero, "Integrating explicit contexts with recurrent neural networks for improving prognostic models," in *2023 IEEE Aerospace Conference*, 2023, pp. 1–8.

[125] R. Dutta Baruah and M. M. Organero, *Explicit context integrated recurrent neural network for sensor data applications*, 2023. [Online]. Available: https://arxiv.org/abs/2301.05031.

[126] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002, ISSN: 1076-9757.

[127] P. D. Turney, *The management of context-sensitive features: A review of strategies*, 2002. arXiv: cs/0212037 [cs.LG]. [Online]. Available: https://arxiv.org/abs/cs/0212037.

[128] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794, ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. [Online]. Available: https://doi.org/10.1145/2939672.2939785.

TH-4096_186151013