IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Solar PV Module Fault Classification using Artificial Intelligence and Machine Learning Techniques

Jitamitra Mohanty
Krupajal Engineering College

Itun Srangi
Krupajal Engineering College

Jagadish Chandra Pati
Krupajal Engineering College

Abstract

Fault analysis in solar photovoltaic (PV) arrays is essential to increase reliability, and improve efficiency and safety in PV systems. Conventional fault protection methods are usually employed to overcome the challenge however conventional protection is only effective in isolating faulty circuits in time of large current flow and remains inactive in case of low fault currents and may cause problems in long run. The model of the different faults emulates the different PV fault conditions which are essential for a healthy PV power system analysis. The model is a solution to classify the potential faults during fault conditions to cut down on the time and cost invested in fault analysis through human analysis. The model is achieved through the use of Artificial Intelligence and Machine learning techniques. The model performance is matched along with specified vectors to check the accuracy using the confusion matrix to ensure good performance in the design. The simulated results determined that the fault diagnosis scheme can correctly classify faults with high efficiency making the power plant troubleshooting process easier. The entire plant characteristics are got from the fed data and the model is trained to capture the entire system behaviour for future instance classification.

Keywords: Faults, Photovoltaics, Simulink, Simulation, Confusion matrix, Power characteristics, Machine learning, Decision tree, Classifiers.

Introduction

Photovoltaic systems provide a promising solution to the world's energy problem. The solar energy industry is currently on a rise in popularity following the maturity of the technologies and consequently the reduced material costs due to the better technologies. However, the capital cost and maintenance costs for PV panels are still high because they are mostly installed outside where they suffer both mechanical and electrical stress resulting in additional power losses, hot-spots formation, and different complications in PV modules like fire outbreaks, in turn, leading to reduced PV efficiency even complete breakdown in production. The PV systems if not monitored the faults may propagate within the modules and cause a complete failure of the PV array. The fault detection methods for the PV system are generalized in visual analysis, thermal testing, and electrical techniques. The electrical fault analysis is more effective and promising for efficient monitoring and diagnostics of PV systems. Today the electricity is predicted to be largely

supplied by solar power. It is reasonable to focus on the design of smart systems to monitor such solar power systems and classify the fault type that might be present for reliability. PV systems provide several advantages over other conventional energy systems. The energy provided is modular in that the capacity to be generated depends on the amount required it also provides easy options to expand the power system to meet the demand. Regardless of the massive initial cost of setting up a PV power system; there is no cost on machineries like transformers, generators, and transmission equipment. Overall maintenance of a PV system is more modular and easily accessible. The above attributes have resulted in an expansion of photovoltaics, and India has invested enough to improve the sector as shown in Figure 1.

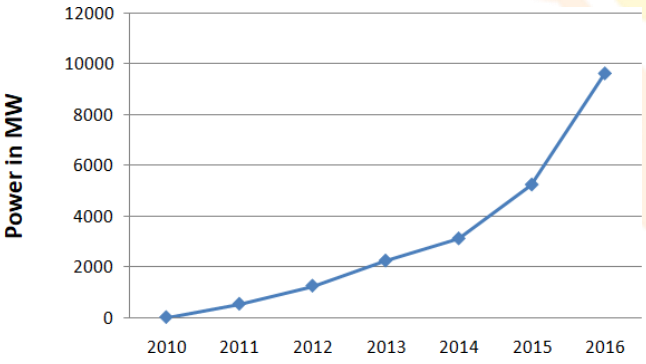


Figure 1: Solar market in India by installed capacity

There has been a progressive increase in the installation of solar power plants. With the continued rate of installation, a future with more clean and reliable energy will be guaranteed to improve the energy sector. The monitoring systems that can capture real-time analysis of power plants are been designed to improve the reliability and stability of power systems improving energy utility by the industries and avoiding the risk of fires or any other hazards.

Modeling and Simulation of PV Modules and Data Generation:

Solar cell exhibits a non-linear output characteristic as in Figure 2 and the curve varies with irradiance and temperature levels. The solar cells are connected in a series and parallel combination to form a module. If modules are connected in series the system voltage increases and if connected in parallel the system current increases as in Figure 2. Every solar cell design should account for parameters that affect the amount of generated current like irradiance, temperature, and type of semiconductor material.

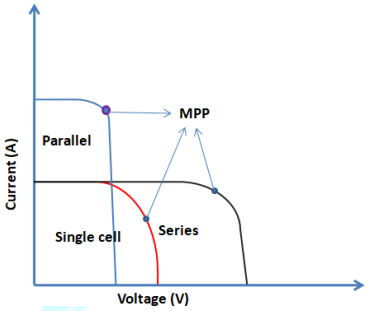


Figure 2: I-V characteristics of a solar cell

The PV module usually is composed of several solar cells with identical characteristics and the modules in a series and parallel combination form a PV array. In real working conditions, PV modules may work at different irradiance, ambient temperature and even under different fault conditions, this makes the I-V curve of a PV array completely different from the ideal case of Figure 2.

There are many models for solar cells that have been designed to meet different conditions. However, the best model should simply be accurate enough to account for most solar cell parameters. The numerical approach to model the PV module using the equations that define its basic working. Several circuit software can be used to design this model but this uses Simulink in Matlab to build a 1.3KW PV system. The approach to building the system is shown in the sequence of Figure 3. The 1.3KW plant is then introduced to different fault configurations for data generation for future fault prediction on the plant.

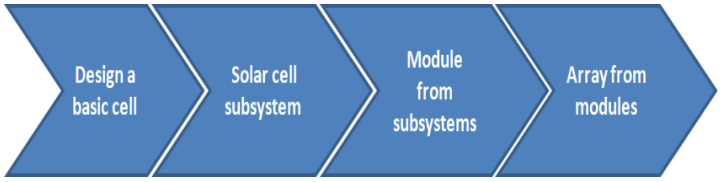
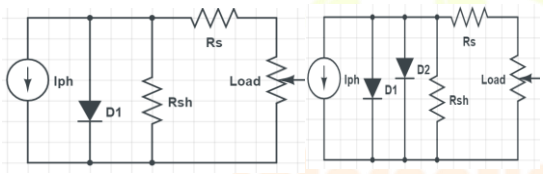


Figure 3: The sequence for modeling the 1.3KW PV system

Solar Cell Models

Solar cells have non-linear I-V characteristics that vary with irradiance and so it isn't suitable to model a solar cell to be a constant voltage source instead solar cells are modelled as a current source. Among the different circuit designs, the single-diode and double-diode models are the most used to describe the characteristics of the solar cell. The Rs describes the ohmic losses in the contacts solar cell contacts metal-semiconductor interfaces. It is assumed for the sake of simplicity that there is no recombination around the junction region for the single-diode model. Especially for semiconductor materials with larger bandgaps, this assumption leads to deviations between actual and simulated characteristic curves of the solar cell but the double-diode model attempts to incorporate the recombination in the junction. The equivalent circuits for the single-diode model and the double-diode model are shown in Figure 4(a) and Figure 4(b).



a. **b.**

Figure 4: Solar cell circuit models (a) Single-diode model (b) Double-diode model

For the single-diode model in Fig. 4(a), the solar cell current equation (1) indicates that more current flows through the load if Iph is more which depends on the irradiance levels. Iph is the generated current and becomes maximum during noontime. V is the working voltage of the cell and any system design must meet this voltage level for the smooth working of the PV system. I represent the load current indicating the demand drawn by the load. Io is the saturation current and depends on the semiconductor material used. The parameters in equation (1) are shown in Table 1.

$$I = Iph - Io\ (e^{\frac{(V+IRs)}{nKNs}} - 1) - (\frac{V+IRs}{Rsh})$$

Equation (1)

Table 1: Solar cell parameters

Symbol	Parameter
I	Solar cell current (A)
V	Solar cell voltage (V)
Iph	Light-generated current (A)
Ish	Shunt resistance current (A)
Io	Saturation current of the diode (A)
Rsh	Solar cell shunt resistance (ohms)
Rs	Series resistance (ohms)
n	diode ideal factor
k	Boltzmann's constant =1.38×10-23 J/K
q	Electron charge = 1.6×10-19 C
T	ambient temperature (K)

The double-diode model incorporates a second diode as in equation (2) which represents the losses due to recombination on the surface and the junction of the solar cell. Equation (2) provides a more accurate description of a solar cell but requires a higher computation power.

$$I = Iph - Io1\ (e^{\frac{(V+IRs)}{nKNs}} - 1) - Io2\ \left(e^{\frac{(V+IRs)}{nKNs}} - 1\right) - \frac{V+IRs}{Rsh}$$

Equation (2)

The single-diode model of a solar cell for simulation because the model is accurate enough and converges faster than the double-diode model. The design is achieved in Matlab by using an inbuilt solar module by "1-Soltech" to develop a 1.3KW PV system.

Simulation in MATLAB/Simulink

The proposed solar model of Figure 4 is implemented in Matlab using the inbuilt solar module by "1-Soltech" by simulation in Sims-cape toolbox which can offer an open and flexible interface for modeling numerical and electrical systems. Figure 5 shows the design in Matlab for the 1.3KW power system. The 1.3KW power system is made of two strings with three modules in series in each string. Each module generates an optimum current of 7.47 Amps and at an optimum voltage of 29.3 V, the short circuit rating of the module is 7.79 Amp and the open-circuit voltage is 36.6 V. The panel has an efficiency of 14% and the maximum working voltage of 600 V as shown in Table 2.

Figure 5: The 1.3KW power system circuit diagram

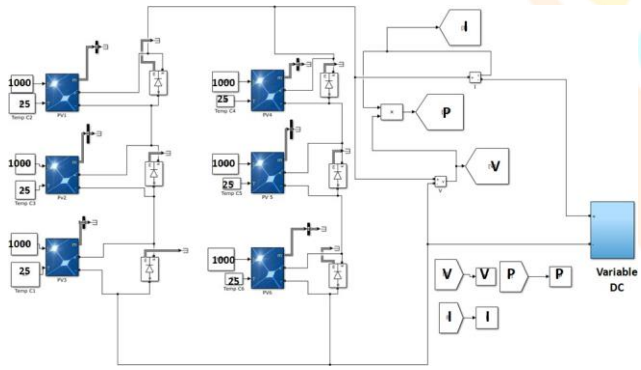
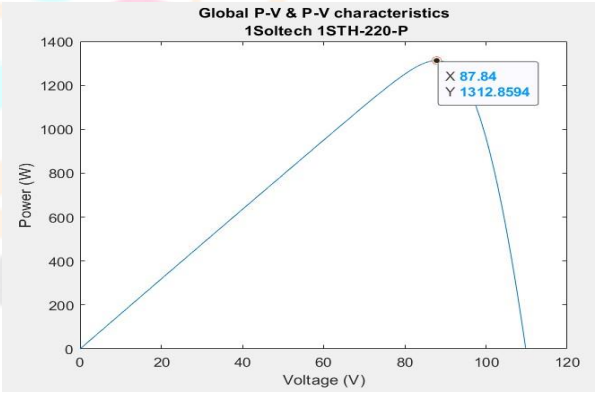


Table 2: The PV panel by 1-Soltech specifications

Parameter	Value
Open Circuit Voltage Voc (V)	36.6
Short Circuit Current Isc (A)	7.79
Voltage Vmp (V)	29.3
Current Imp(A)	7.47
Panel Efficiency	14.0%
Fill Factor	0.754
System Voltage Vmax(V)	600

The PV system demonstrated above is tested in MATLAB/Simulink using the simulation model suggested. The PV modules can have different I-V curves for different irradiance levels and for different module conditions a feature that is essentially useful for fault studies. The approach here we note the healthy operating points and it's important to note that for the developed PV system and irradiance of 1000 kilowatts per meter squared the system parameters are Imp=14.93(A), Vmp=87.91(V), and Pmax =1312 (W) as in the simulation results obtained in Figure 6.

a



b

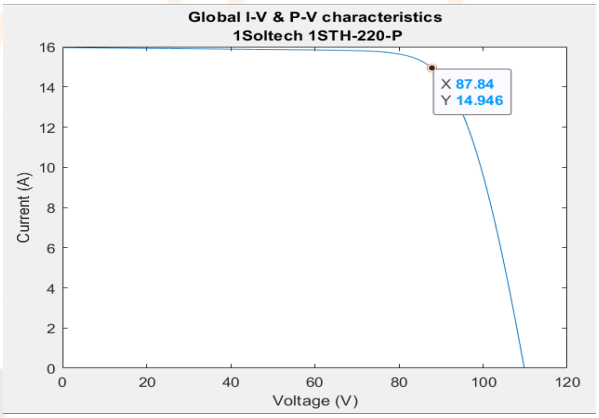


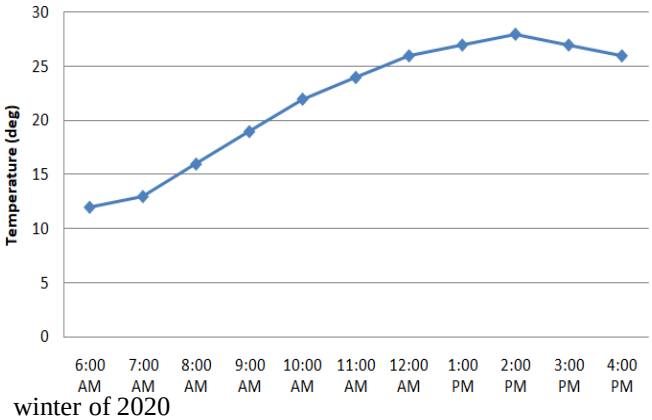
Figure 6: The 1.3KW healthy I-V and P-V characteristics (a) I-V characteristic (b) P-V characteristics

The capacities indicate the normal working conditions of the plant and any deviations from this operating point suggest a defect in the system. Data indicating the possible deviation margin from these conditions under different fault conditions were generated. The different introduced conditions show the plant behavior during fault and this data is exported for model design. The model designed understands the characteristics of the plant during fault and can isolate the type of fault in future instances using its previous learning experience hence can predict present fault conditions. The simulation process to capture the system fault configuration was done incorporating real-time weather. To achieve authenticity, irradiance and temperature were varied according to the levels received in Puri time zone and weather conditions.

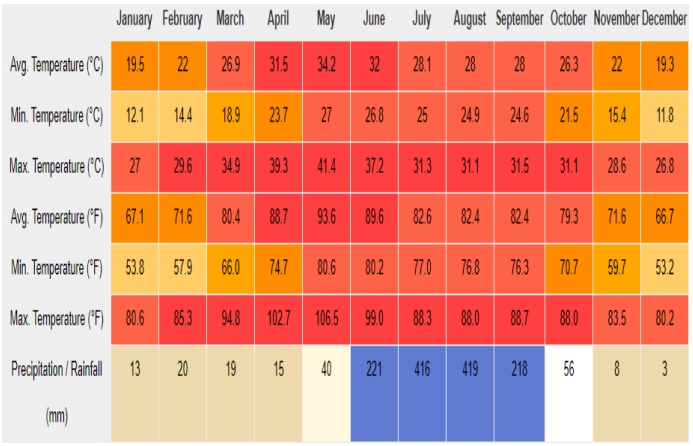
Data Generation

To capture the real-time characteristics of the 1.3KW PV system, real weather conditions were introduced based on the winter of 2020-2021 in Puri where the plant is to be set. Average weather measurements for winter were recorded for the simulation purpose. The first step is to identify the sunrise and sunset hours to consider only generation periods of the power system and for the winter of 2020-2021 in Puri the average sunrise hour was 7 am and sunset at 4 pm and obtained average temperatures for the winter of 2020-2021 are shown in Figure 3. Average measurements may not exactly give the actual behavior of the plant but give an overall and general behavior of the plant. The measurements were recorded for two months and average irradiance levels corresponding to the time and and temperatures of the day recorded. The irradiance was low during morning hours and maximum at noontime. The different irradiance levels were used in simulating all fault configurations. Temperatures were varied in the simulation following the recorded levels.

Figure 7: The measured average temperatures in Puri in winter of 2020



The measured temperatures were used for characterizing the system from healthy conditions to different fault conditions for data generation. Different fault configurations were introduced into the power system to capture the system characteristics as much as possible. A dataset with over 1062 (6*177) data points was generated to be used in machine learning to train the model that can fully characterize the system and classify the possible faults in future instances of the plant. The temperatures over two months were recorded and the average temperature noted for simulation in Simulink. The temperatures recorded only respond to the generation period of the plant and these temperatures were used for the different fault conditions to generate data. The hourly temperatures were recorded and the average calculated as in Figure 8



	January	February	March	April	May	June	July	August	September	October	November	December
Avg. Temperature (°C)	19.5	22	26.9	31.5	34.2	32	28.1	28	28	26.3	22	19.3
Min. Temperature (°C)	12.1	14.4	18.9	23.7	27	26.8	25	24.9	24.6	21.5	15.4	11.8
Max. Temperature (°C)	27	29.6	34.9	39.3	41.4	37.2	31.3	31.1	31.5	31.1	28.6	26.8
Avg. Temperature (°F)	67.1	71.6	80.4	88.7	93.6	89.6	82.6	82.4	82.4	79.3	71.6	66.7
Min. Temperature (°F)	53.8	57.9	66.0	74.7	80.6	80.2	77.0	76.8	76.3	70.7	59.7	53.2
Max. Temperature (°F)	80.6	85.3	94.8	102.7	106.5	99.0	88.3	88.0	88.7	88.0	83.5	80.2
Precipitation / Rainfall (mm)	13	20	19	15	40	221	416	419	218	56	8	3

Figure 8- Average temperatures in winter

PV System Fault Classification

The energy demand is exponentially increasing and PV energy production is by far the fastest-growing energy technology to meet the demand. The PV industry is guaranteed by the reduced cost of materials hence the production cost of energy. In the last few years, the industry has also seen a qualitative improvement regarding growth in grid connectivity. The PV system suffers loss more for a variety of faults. The different common faults include ground fault, the line to line fault, hotspots, bypass mismatch, and arc faults which all result in high current inflows with the potential to cause a fire. Fault analysis and protection besides improving the efficiency and reliability of the PV system, if ignored, can lead to a reduction in the power generated and breakdown of the power system.

Classification of faults in PV system

There so many types of faults either electrical or non-electrical that affect a power system. The idea of broadly classifying the possible faults is close to impossible, and throughout the years, many different types of research have been carried to isolate different fault types as much as possible. Some of the common faults like the mismatch faults, ground faults, line to line faults, bypass diode faults, and arcing are explained here as they are common PV systems

Table 3: Classification of various types of faults in a PV system

Type of fault	Subclass of fault	Description
Mismatch faults	Partial shading	Caused by shielding of blockage on the panels
	The uneven irradiance distribution	Due to varying intensity levels at different times of the day
	Soiling	Due to dust and dirt on the solar panels
	Hotspot	Partial shading of one part of the solar panel
Ground faults	Upper ground fault	shorting of the last two modules of the PV string to the ground
	Lower ground fault	shorting of second and third modules of the PV string to the ground resulting in the substantial back-feed current
Arc faults	Series arc fault	discontinuity in any current-carrying conductor
	Parallel arc fault	Insulation failure in the current-carrying conductors
The line to line fault		Accidental short-circuiting of two strings of solar cells
Bypass diode fault		Short-circuiting due to incorrect connection
Degradation fault		Delamination and yellowing of modules, insertion of bubbles in the modules, cracking, and defects in the anti-reflection coating of the panels
Open circuit fault		Unplugging of connection wires in the junction box
Inverter faults		Failure in the components
Outage		Blackout due to weather conditions like lightning, storm, hurricanes

Mismatch faults in the solar panels

Mismatch faults are most common in the PV arrays resulting in power loss and permanent damage to the modules. Mismatch faults in PV modules occur

when electrical parameters of some panels are significantly different from the others or mismatched. The possible reason for the mismatch is the varied irradiance levels (Figure 9) on different panels or different temperature levels. Mismatch faults are further classified into two types:

Temporal mismatch faults: Caused by shading of panels from structures, clouds, foliage, dust on panels, and anything else that block radiation.

Permanent mismatch faults: Mainly caused by hotspots and aging of the modules. The shading effect results in uneven distribution of the irradiance on the PV array as shown in Figure 9 causing reduced power production

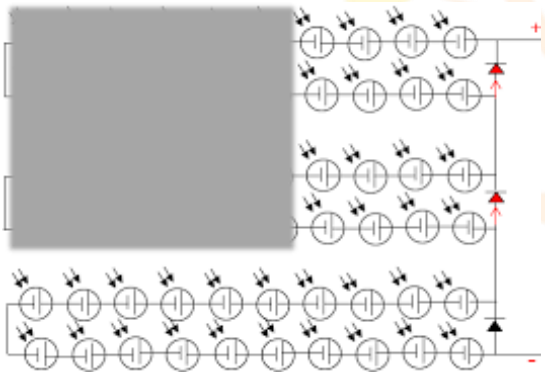


Figure 9: Shading of the PV array

Ground faults in Photovoltaic systems

A PV panel design comprises noncurrent-carrying (NCC) metals (e.g.., module frame, and the metal enclosures) to provide mechanical support during normal operation of the panels .The conduits can accidentally short-circuit the current-carrying wires of the panel due to various reasons. To prevent a short circuit all NCC conductors are connected to an equipment grounding conductor (EGC) to the ground and the conducting conductors are well insulated. The potential reasons for ground faults are.

i. Degradation and liquid entry leading to a short circuit between EGC.

ii. Animal infestation resulting in damaged cable insulation.

iii. Insulation damage to cables due to aging, corrosion due to water, damaged panels, or incorrect installation.

iv. Short circuits in the PV combiner box

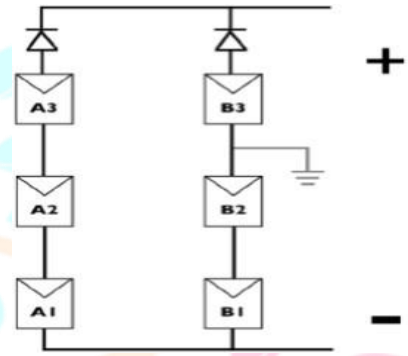


Figure 10: Ground fault

Ground faults are easily detected by monitoring the direction of the inverter current continuously and once a reverse current is detected the fault alarm becomes active and no additional sensors are required. Although different ground fault detection devices include Ground Fault

Detection Interruption (GFDI) a fuse, and Residual Current Device (RCD) is also used.

Arc faults

Factors cause arcs within the module and these persistent burns for a long time interval cause massive damages in the PV system. Arcs burn at very high temperatures which depends on the

available energy and thermal characteristics of the panel. The current carrier generation in a module depends on the irradiance and therefore can aid in stable ignition conditions for electrical arcs. If not put off within a short period, the direct radiation on the arc may start a fire.

Arc faults occur in a variety of locations for example in a fuse, terminals, inverters, bypass diodes and also within the PV modules at joint locations. Classification of arc faults:

i. Parallel arc fault to the ground

ii. Cross-string parallel arc fault

iii. Intra-String parallel arc fault

The line to line faults

A line to line fault is an accidental short circuit between two or more random points in an array that are operating at different potentials. The line to line faults is more difficult to isolate with any conventional fault clearing devices. The line to line faults depicts distinct behavior under low irradiance and at night time to day time .A line to line fault can be represented as in Figure 11 shorting two points There are two different techniques for analysis of faults in PV arrays;

i. steady-state analysis

ii. The transient fault analysis

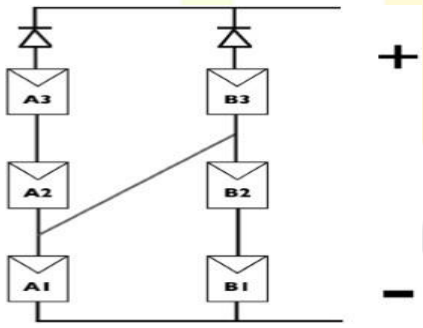


Figure 11: Line to Line fault

Bypass Diode Faults

During shading conditions, bypass diode bypasses the non-generating group of panels at low voltages and this reduces the risk of hotspots and minimizes the shading effect. Any damage to the diode results in local hotspots causing heating and damaging the solar cells and the panel effectively. Bypass diodes play a very essential role without which the entire panel breaks down over time. The diodes are connected across each module or a group of panels in systems with so many panels.

Results

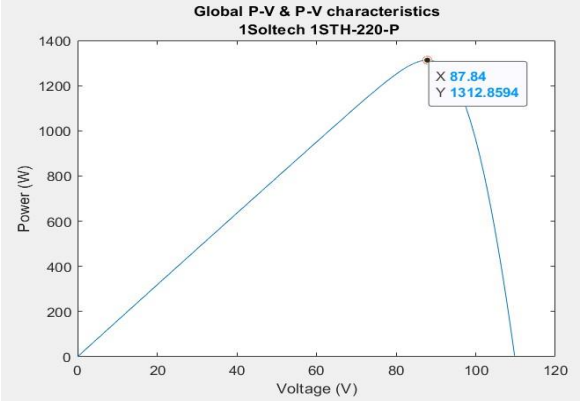
Successful fault classification in PV systems is essential for reliability in power production. Any fault configuration results in a shift from the optimal operating conditions of the power plant resulting in a reduced capacity and potentially into a total power system breakdown. Fast identification and isolation of the fault will ensure satisfactory customer service but the key depends on pinpointing the exact type of fault the system is suffering from at the time. Some example simulated conditions on the plant are:

Healthy condition: The power capacity during normal operation of the "1-Soltech" solar panel is 1.3KW as in Figure 12 (a) the generation resides around this capacity during good sunshine hours and its desired to operate at these conditions.

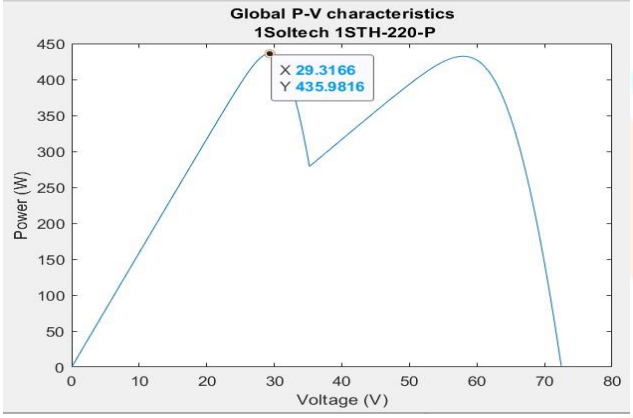
Shading faults: The shading effect on the PV capacity depends on how much shade is affecting the panels and incase where all generating units are blocked and no radiation reaching the panels the production is zero. The shading effect shifts the PV system operating point affecting the efficiency of the system. Figure 12(b) shows the simulation results for 30% shading on the plant and the observed capacity reduces to 435.98W.

The line to Line fault: The line to line fault results in an additional current path in the system reducing the current to the load ultimately shifting the operating point of the PV system. The capacity reduces and line fault risk heating of NOC if in

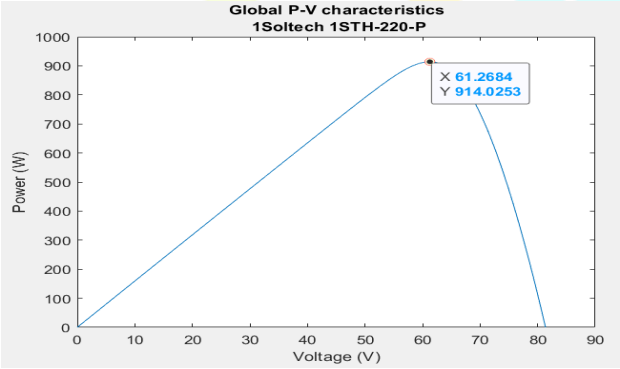
contact and may cause a fire and damage to insulation. All line to line simulated on the 1.3KW system resulted in the reduced capacity to 914W as Figure 12(c).



a



b



C

Figure 12: The different P-V characteristics of the 1.3KW PV plant

(a) Healthy (b) Shaded condition (c) Line to Line fault.

Artificial Intelligence and Machine Learning

Artificial intelligence (AI) deals with the ability of a computer system to perform tasks commonly similar to human intelligence. AI involves developing systems endowed with the intellectual processes that mimic human characteristics like the ability to reason, understand, generalization, and learn from experience. Despite all continuing advances in the technology, there are still challenges in processing the speed and memory capacity of the systems and there haven't been programs that can match human flexibility in general to meet the human level of intelligence. In recent developments, some programs have attained the performance levels of human experts and professionals at performing certain tasks, so in that regard, we can say artificial intelligence is limited in the sense that it is efficient in applications of specialization as diverse as medical applications, search engines, and voice or handwriting recognition which are and not necessarily in the generalization based tasks as humans.

The classification task

Classification refers to categorizing a given set of data into classes and can be performed on both unstructured and structured data. The target is predicting the class of given data based on familiar knowledge or experience. The classes are often referred to as a target or label. The classification modeling process is the process of approximating some function from a set of input variables unto output variables. The goal is to identify the category of the new data which is the best fit. The ability to do so depends on the learning experience of the classifier therefore the most important part of classification is selecting the best form of the learning process for the classifier.

To understand the different type of learners available and the main types are:

Lazy Learners: Lazy learners just store the training data and await the testing data to appear. The classification depends on how close the stored data is related. They have more predicting time compared to eager learners. An example of such is the k-nearest neighbour.

Eager Learners: Learners construct a general classification approach based on a particular training data to design models with the ability to predict correctly on future instances. The model commits to a strong hypothesis that will work for the entire domain. More time is invested in training and less time on the testing. Examples of such learners are, ANNs, Decision Tree, and Naive Bayes.

Classification terminologies

The modeling process highly depends on the data visualization process. A successful data analysis depends on the knowledge of data mining and statistics hence knowledge of terminologies and definitions is important for our analysis. After analysis of the data, a classifier is selected depending on the purpose of the model design. The most important part of the classification is selecting the best classifier.

Classifier: It is an algorithm that is used to map the input data to a specific category by isolating the clusters or patterns in the data.

Classification Model: A design that predicts or draws a conclusion on the inputted data from the given training data and makes predictions on the category in which the new data best fits in.

Feature: A feature is a measurable or observable property of the phenomenon being observed. Refer to the parts or properties that form the entire system.

Binary Classification: An isolation process with only two outcomes in the results.

Multi-Class Classification: An isolation process with more than two classes and each sample is assigned to one and only one label.

Training: A process of feeding data to an algorithm (F(x,y)) with the ability to learn the data.

Prediction: The process of decoding future instances based on the training obtained from previous data.

Evaluate: This means the analysis of the model by checking performance parameters.

Machine Learning Models (ML)

Supervised Learning is a based technique that deals with the prediction of outcomes on data based on previous data. It involves teaching a model to learn patterns and functions that help map the desired outcome in a future instance. The learned model is simply a numerical model designed based on the labelled data. The algorithm uses this previous knowledge by the model in predicting outcomes of future data. Supervised Learning utilizes regression strategies to fabricate these models. Classification algorithms are just used on discrete or categorical targets. Model design for any machine learning requires a well-labelled data and selection of the best algorithm for design. The selected model is then fed with data for training and exploring the patterns in the data. The performance is improved by tuning the parameters of the classifier. After the design new data is fed to the trained model to give a prediction and the accuracy is cross-validated through analysis. The machine learning classification models in this work include Random Forest, Artificial Neural

K-Nearest Neighbor

K-nearest neighbor (KNN) is one of the lazy learner-based algorithms. This means that the model makes no assumptions on the data

distribution, the classifier design underlines the dataLazy learning algorithms keep all the training data for prediction on the outcome of future instances. The premise of KNN is the fact that similar objects appear close to each other. Objects are classified based on a majority vote from their neighbors and are assigned tothe category closest to their neighbors. K is a positive integer parameter passed to the KNN for tuning to improve accuracy.

Random forest

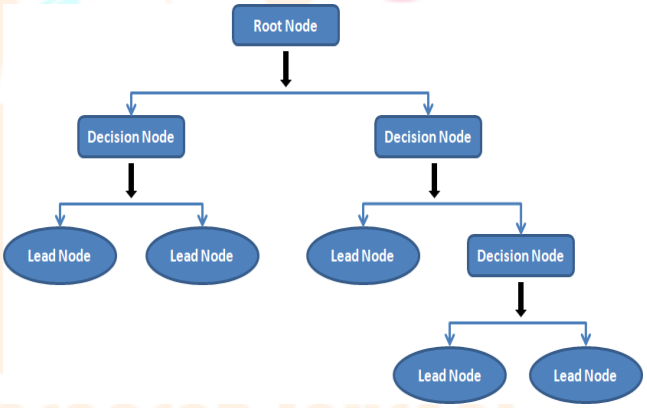
Random forest algorithms use ensemble algorithms that create several decisions called trees as in Figure 14 from the training dataset to predict outcomes of future instances. Once data is fed on to the algorithm it sets rules from it which are used for the prediction on future outcomes on the new data. The classifiers have an up to down approach analysis on the nodes starting from the root node applying a binary split first on the most predictive features creates nodes further down through the process. This continues until leaf nodes are generated with no possibilities of further splitting. This process is based on the calculation of entropy. The model predicts a target class for each leaf node upwards to the actual class.

Random forests as the name ts create several trees called a forest of decisions based on the training data depicting patterns and behavior contained in the data. The random forests called a bagging classifier. Decision trees pick a random set of features from the fed data and differ from in that regard.

Support Vector Machines

Support vector machines (SVM) are good for non-linearly data that has clusters. The premise of an SVM model is locating a hyperplane in a multidimensional space of features that can be able to separate or categorize future instances easily. For example, in the case of a 2-dimensional space, the hyperplane is a line that can isolate the clusters with high accuracy. This line is at an equal distance away from the SVMs. Support vectors refer to the closest points to the hyperplane. SVMs explore the different possibilities of the several lines then select the farthest from the support vector. In the case of a non-linear classification data, the algorithm utilizes several kernels that map the points unto different planes to isolate the clusters accurately. In the work, the SVM classifier applies the OVR strategy to build a binary classifier for every class. The classifier focuses on the current class and treats it as a positive leaving the rest negative. A cluster is treated as a single class and is fit against the rest of the clusters

Decision Tree



Decision trees are very common algorithms and their learning methods with a wide range of applications New data is classified by sorting from up to down of the root node following attributes tested in the previous nodes. Each branch below a node indicates a possible value for an attribute. . The process is repeated until a leaf node hence reaching the classification of the instance.

Construction of a decision tree starts with the selection of a suitable attribute to put in the root node and followed by the creation of branches for every possible value of the attribute dividing the sample set into subsets. The process is recursive for each branch until all instances at a node result

in the same classification and tree development stops. The key to the development of a great decision tree is the selection of an optimal method for data splitting the data, which means, selecting an attribute that is most useful for classifying the samples. Recursive visits at each decision node, selecting the optimal split until no further splits are possible this is the basic premise in decision trees. Decision trees use the concept of entropy reduction to optimize the splitting process.

Entropy measures how good an attribute isolates the training examples according to their target or label based on the measure of their Information.
In a binary classification setup, the entropy in set X is calculated using equation (3) which gives information
in the message.

$(x) = -\mathrm{P_p} log_2 (P_p) - \mathrm{P_n} log_2 (P_n)$ -- Equation (3)

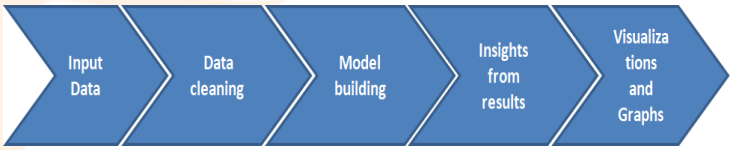
Pp represents the positive examples in X and Pn the
negatives in X. The entropy can be calculated as the weighted sum of the entropies for the subsets as in equation (4) and shows the weightage of information. Information Gain measures a reduction in entropy by partitioning the training data. The unit of information is called a bit. As a measure of the average uncertainty in X, the entropy is always non-negative indicating the average number of bits required to describe the random variable. A higher entropy indicates more information the variable contains. The key to applying the concept of entropy to visualization problems depends on the proper specification of the random variable X and the definition of the probability function P(x) which depends on individual applications.

Hs (T) =∑ (PH (T)) ---------Equation (4)

Design and Predictions

After capturing the PV system characteristics under the different fault conditions the data is exported to design a smart model to predict future fault conditions in the PV system. The design is dependent on previous experiences and the more the data the better the learning experience for the model. The data is split into training and testing points to ensure the model predicts correctly. The problem statement is the ability of the designed system to correctly isolate the type of fault occurring in the system to initiate quick troubleshooting to cut down on power outage. The data analysis is achieved using anaconda's Jupyter notebook.

Model design sequential steps



Feature Selection and Engineering

Successful feature extraction on the power system data is followed by an analysis of the features to isolate those with high weightage on the classification process. For better classification features that tend to be noise or highly correlated are removed. This is called feature selection and aids to isolate any redundancy or irrelevant variables in the data preserving information in the data. The process reduces the overfitting of data essential for higher performance and accuracy. The feature selection process involves a quick scan of the data for pattern identification. The feature selection techniques are classified into:

Wrapper methods: Wrapper methods technique focuses on training features with the more functional subset by giving priority to those only.

This process is recursive and the decision to isolate the most significant subset to be set aside repeats to either add or remove a feature. The process repeats until the desired feature subsets are attained. The feature subsets are cross-validated for their performance using the right learning classifier. These methods are extensive in searching and can easily find the best features for the training model. But the selected features using wrappers are only great for a particular data on which the model was trained and may not perform perfectly on future instances risking overfitting on the data.

Filter methods: Filter methods find relevant features based on analyzing the relationship between the target class and the features. This technique gives a rank to the entire dataset on top of selecting the best features. Ranking helps during performance enhancement and analysis which improves performance. Filters are not good at providing the best feature subsets instead they give a general model design. The performance of a model based on this technique is mostly less than those designs implemented using wrappers with less computational requirements and are free from overfitting. These methods can sometimes be used for pre-processing before application on wrapper methods.

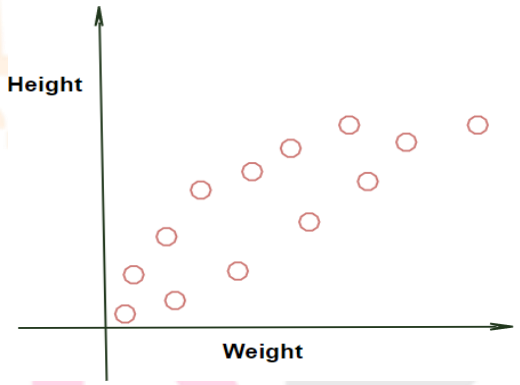
Embedded methods: Embedded technique is a hybrid made from both filter and wrapper methods. Unlike the filters and wrappers, the technique features both feature selection and incorporation of a learning algorithm on the selected features. The method begins with selecting the most significant features at every iteration based on the intrinsic nature of the method. The process repeats until no further improvement in the performance of the model. The technique is less prone to overfitting than wrappers.

Data Analysis and Interpretation

To fully understand the nature of the data and its meaning we employ Jupyter notebook using the Anaconda IDE. The different libraries are employed to display data and interpretation. The different features are checked parallel to understand how they are related to each other and interpret their significance in the system. To get a general description of the data we use the "describe" command and Table 4 gives the summary of the data.

Table 4: General description of the dataset

	Irradiance	Temperature	Imp (A)	Vmp (V)	Pmax (W)
Count	176.0	176.0	176.0	176.0	176.0
Mean	645.5	21.5	7.0	40.2	288.3
Std	206.7	5.4	3.1	20.4	216.5
Min	300	12	1.3	27.9	41.7
25%	500	16	4.5	30.4	152.6
50%	600	24	6.8	31.5	228.2
75%	800	26	8.9	32.3	326.4
Max	1000	28	14.9	93.6	1302.2



The correlation is an important element employed in feature selection to avoid overfitting in the model design process. The highly correlated features suggest dropping some of the features to improve the machine's ability to retain accuracy. The correlation in the data is shown in Table 5 and we observe that the highest correlation is 0.844 between temperature and irradiance which confirms the strong relationship between the two parameters. Current is highly dependent on the irradiance and this can be confirmed by the correlation of 0.733. In situations where the

dataset is massive, the feature engineering involves dropping off the highly correlated features as they contain almost the same meaning on the system characteristics. Temperature and voltage show a negative correlation that means they are inversely proportional, a rise in temperature reduces the cell voltage with less effect on the generated current. Both voltage and current show a high correlation with the maximum power (Pmax) of 0.726 and 0.671 respectively.

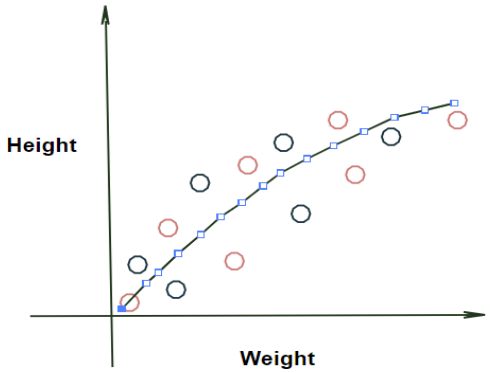
Table 5: Correlation between the different features

	Irradiance	Temperature	Imp	Vmp	Pmax
Irradiance	1	0.844	0.733	-0.032	0.408
Temperature	0.844	1	0.621	-0.040	0.340
Imp	0.733	0.621	1	0.084	0.671
Vmp	-0.032	-0.040	-0.084	1	0.726
Pmax	0.408	0.340	0.671	0.726	1

Model training and Testing

After all the data cleaning and a full understanding of the data, we start designing the model using the different packages and the different Algorithms potentially able to execute the work. The cleaned data is encoded for all non-numerical features. The classification process involves splitting the data frame into the target element versus the rest of the features (mapping between the target and the selected feature). The data is then split into training and testing data as shown in Figure for the classifier.

(Data before splitting)



Data after splitting into training and testing points

After splitting the data the appropriate library and algorithm are employed to train and test the model. In this project, the algorithms used are Decision tree, ANN, Support vector machine, and Random forest. To understand the working of the classifier Figure shows how the linear and polynomial based classifiers follow the testing points. Observe that the linear classier is has a high bias but is likely to predict correctly on the future samples avoiding overfitting. The polynomial has high variance and gives 100 percent accuracy on the testing data but fails terribly on future samples hence giving a poor model. In general, 80% of the data is used for training, and 20% used for testing the trained model. The accuracy of each model depends on the data hence data visualization is a very important part of the design process to ensure the best algorithm for that particular dataset is selected. The learning process ensures that the model explores the data and capture hidden patterns and characteristics in the data

Classifier performance in predicting the testing data

Linear classifier and Polynomial classifier

Accuracy Check and Results

After designing the model we select the best fit model by checking the accuracy of each using the accuracy score command. This is the last step in the design process and determines whether the

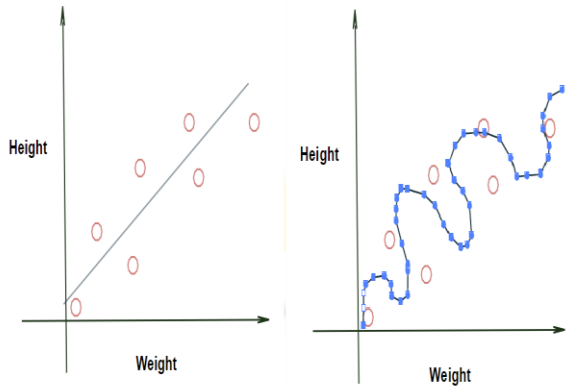
model is successful and if not the process is repeated after further feature engineering and analysis.

Confusion Matrix ((i, j)): Also referred to as the Error matrix and uses a table in describing the efficiency of a model based on data with known or true outcomes. It gives a clear description of the errors that a particular model makes representing the classifier's efficiency in the prediction process. For a binary classification problem the confusion matrix is represented as:

	Predicted Class 1	Predicted Class 0
Actual Class 1	True Positive	False Negative
Actual Class 0	False Positive	True Negative

Table 6: General Confusion matrix for a binary classifier

Analysis: The diagonals indicate the correctly predicted parameters and the off diagonals indicate the misclassified classes, Table 7.1 shows a typical confusion matrix. The elements i and j in nij of equation (5) indicate the row and the column and



show the cases of class i identified as class j. Hence, the diagonal elements nii are the correctly classified classes, while the off-diagonals the misclassified classes. The total cases (N) is given by equation (5) and the complete parameters of the matrix are shown in Table 7.2.

$$N = \sum_{i=0}^{M} \sum_{i=0}^{M} n_{ij} \quad \text{Equation (5)}$$

Despite the confusion matrix containing all information about the possible outcome of a classifier, it's not ideal for reporting on the brain-computer interface (BCI) field as they are not only difficult to compare but to interpret too. Hence, only, some parameters are considered from the confusion matrix for analysis.

Table 7.1: Typical Confusion matrix

Symbol	Formula
X	$X1 + X2 + X3$
W	$W1 + W2 + W3$
Y	$Y1 + Y2 + Y3$
A	$X1 + Y1 + W1$
B	$X2 + Y2 + W2$
C	$X3 + Y3 + W3$
T	$A + B + C$
T	$X + Y + W$

Table 7.2: Confusion matrix different parameters

		Predicted Class			
		Class A	Class B	Class C	Total
Actual Class	Class A	X_1	Y_1	W_1	A
	Class B	X_2	Y_2	W_2	B
	Class C	X_3	Y_3	W_3	C
	Total	X	Y	W	T

The confusion matrix, (i, j), is a square (n*n) matrix with rows and columns referring to the actual and predicted class on the data respectively. It follows that the diagonals (i=j) indicate the correct classification decisions. For many applications, normalization of the confusion matrix is useful for easy analysis and understanding of data. This can be achieved i by number of ways, first of which involves dividing each

element of $CM(i, j)$ by the total number of samples in the dataset or the sum of the elements in the matrix as in equation (6):

$$CMn\ (i, j) = \frac{\mathrm{CM(i,j)}}{(\sum_{i=1}^{Nc}) \sum_{i=1}^{Nc} \mathrm{CM(i,j)}} \quad \text{--- Equation (6):}$$

The second type of normalization is done row-wise by dividing each element of the confusion matrix by the sum of elements of the respective row (the true population of the class that has been mapped on that row). After normalization has taken place we can now discard the information that is related to the size of each class. In this form, all classes are considered to be of equal size and the dataset is now class-balanced. This normalization as in equation (7) gives the elements a per unity analysis

$$CMn\ (i, j) = \frac{\mathrm{CM(i,j)}}{\sum_{n=1}^{Nc} \mathrm{CM(i,n)}} \quad \text{Equation (7)}$$

Analysis of the Confusion matrix before normalization, we can extract three useful performance measures, namely the overall accuracy (Acc) of the classifier, which represents the fraction of samples of the dataset that have been correctly classified. The overall accuracy (Acc) in equation (8) can be computed by dividing the sum of the diagonal elements by the total sum of the elements of the matrix (T).

$$Acc = \frac{(\sum_{m=1}^{Nc} CM(m,m)}{(\sum_{m=1}^{Nc}) \sum_{m=1}^{Nc} CM(m,m)} \text{--} \quad \text{---- Equation (8)}$$

Two other class-specific measures that describe how well the classification algorithm performs on each class include recall (Re (i)) and Precision (Pr (i)). The recall is defined as the proportion of data with true class labels i that were correctly assigned to class i and is computed as in equation (9)

$$Re\ (i) = \frac{\mathrm{CM(i,i)}}{\sum_{m=1}^{Nc} \mathrm{CM(m,i)}} \quad \text{Equation (9)}$$

Where $\sum_{m=1}^{Nc} \mathrm{CM(i,i)}$ is the total number of samples belonging to class i. If the confusion matrix is row-wise normalized, then $\sum_{m=1}^{Nc} \mathrm{CM(i,m)}$=1 giving Re (i) = CM (i, i), implying that the diagonal elements of the matrix are recall values.

Pr(i) is a fraction of samples that are correctly classified to class i taking into account all the samples that are classified to that class. Precision is a measure of accuracy on a class basis and is defined according to the equation (10):

$$Pe\ (i) = \frac{\mathrm{CM(i,i)}}{\sum_{m=1}^{Nc} \mathrm{CM(m,i)}} \quad \text{Equation (10)}$$

Where $\sum_{m=1}^{Nc} CM\ (m, i)$ represent the total number of samples that were classified to class i note that, if all classes contain the same number of samples, i.e. if all classes then all three performance measures can be computed from any version of the confusion matrix either with or without normalization. Otherwise, if the classes are not balanced, the second normalization method will yield different performance results from the first, standard normalization scheme. An important measure that combines the values of precision and recall is the F measure, which is computed as the harmonic mean of the precision and recall values as equation (11).

$$F\ (i) = \frac{2\mathrm{Re(i)Pr(i)}}{\mathrm{Pr(i)+Re(i)}} \quad \text{Equation (11)}$$

Following Table 7 for an easier understanding equation (12) to equation (14) adopts the nomenclature and we can see that $X1 = \mathrm{CM(i,i)}$ and X = $\sum_{m=1}^{Nc} \mathrm{CM(m,i)}$ and the performance measures are:

i. $Pe\ (i = 1) = \frac{\mathrm{CM(i,i)}}{\sum_{m=1}^{Nc} \mathrm{CM(m,i)}} = \frac{X1}{X}$ Equation (12)

ii. $Re\ (i = 1) = \frac{\mathrm{CM(i,i)}}{\sum_{m=1}^{Nc} \mathrm{CM(i,m)}} = \frac{X1}{X1+Y1+W1}$ Equation (13)

iii. $Acc = \frac{(\sum_{m=1}^{Nc} \mathrm{CM(m,m)}}{(\sum_{m=1}^{Nc}) \sum_{n=1}^{Nc} \mathrm{CM(m,n)}} = \frac{X1+Y2+W3}{T}$ Equation

(14)

Prediction Results:

The dataset of 1062 (6*177) data points was split with 80% of the data for training and around 20% for training. 20% of the data gives about 250 data points and this confirms with confusion matrix of (16*16) obtained for the prediction on the testing data points as in Table 8. To calculate the accuracy equation (14) is used on all the four algorithms then the algorithm with the highest score is selected. Data from the confusion matrix of the other three algorithms are reported in Table 9 and Table 8 shows the detailed confusion matrix for decision tree since it gave the highest score.

Table 8: The obtained confusion matrix using Decision tree classifier

Parameter	Decision Tree	Random Forest	ANN	SVM
$(\sum_{m=1}^{Nc} \mathbf{CM(m,m)}$	31	11	14	19
$\sum_{m=1}^{Nc} \mathbf{CM(m,i)}$	5	25	22	17
$(\sum_{m=1}^{Nc}) \sum_{n=1}^{Nc} \mathbf{CM(m,n)} = T$	36	36	36	36
$\frac{(\sum_{m=1}^{Nc} \mathbf{CM(m,m)})}{(\sum_{m=1}^{Nc}) \sum_{n=1}^{Nc} \mathbf{CM(m,n)}}$	0.8611	0.3056	0.3888	0.5277

Using equation (14) for the obtained matrix, $\sum_{m=1}^{Nc} CM(m,m)$ =31 indicating the sum of the diagonal elements and $\sum_{m=1}^{Nc}) \sum_{n=1}^{Nc} CM(m,n)(T)$ =36 which is the total elements and the obtained Accuracy score is $Acc = \frac{31}{36}$ = 0.8611. Similarly, Table 9 data is used to calculate the accuracy of the other classifiers

Table 9: Data obtained from the confusion matrix from the classifiers

4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Using the equations (14) we calculate the accuracy score of each classifier from data in Table 9 as shown in Table 10 and Table 11 shows the summary of the performance by each classifier used for the classification process with decision tree giving the highest score of 86% which is good enough for model designing.

Table 10: Tabulation of the performance score of the classifiers from the confusion matrix

Element	Random Forest	ANN	SVM
Diagonal	[0,2,2,3,0,0,0,0, 0,0,0,4, 0,0,0,0]	[0,2,2,4,0,0,0,1,0,0,0, 5,0, 0,0,0]	[0,2,2,4,0,0,1,1,1 ,1,1, 5,0,0,1]
Non-zero off-diagonals	[1,2,2,3,1,1,1,1, 1,3,1,2,3, 1,1,1]	[1,1,2,1,1,2,2,3,2,3,1, 1,1,1]	[2,3,3,1,2,2,4]

Table 11: Accuracy score by different classifiers used

Model	Testing Accuracy
Decision tree	86%
Random Forest	31%
ANN	38%
Support Vector Machine (SVM)	53%

Conclusion and Future Work

The data was successfully generated following the weather conditions in Puri during the winter period. The dataset captured the system behavior under fault. The different fault configurations were incorporated and the

data was visualized in Jupyter notebook using python to infer the meaning hidden about the 1.3KW plant. Four different algorithms were used and the most accurate with an efficiency of 86% was the decision tree and was used to implement the design. The biggest challenge was exhausting all the possible fault configuration to capture the entire system behavior, for example, it was impossible to simulate the ground fault on the designed DC power system. This suggests that the model may not effectively classify instances that may represent fault conditions that were not incorporated during model training. This is the biggest throwback of using generated synthetic data rather than real-time data from a plant collected over the years. Future work can include expanding the dataset to incorporate all year seasonal conditions and possibly the implementation of fault classification techniques on real plant data and improve the model model design through tunning of classifier parameters using new python packages.

Bibliography

1) Ministry of Renewable energy government of India available at https//mnre.gov.in
2) Alternative Energy available at http://www.alternative-energy.com
3) RECP, “Global Market Outlook - For Solar Power/2017-2021,” 2017
4) S. K. Firth, K. J. Lomas, and S. J. Rees, "A simple model of a PV system
5) Alternative energy tutorials athttps://www.alternative-energggy-tutorials.com
6) ”K Nearest Neighbor- sklearn”. [Online]. Available: https://scikit-learn.org/stable/ modules/generated/sklearn.neighbors.KNeighborsClassifier.html
7) ”Random Forest-sklearn”.[Online].Available: https://scikit-learn.org/stable/ modules/generated/sklearn.ensemble.RandomForestClassifier.html
8) PVGIS for assessment of solar PV energy potential of Odisha. Int J Renew Energy Res 2016; 6:61–72.
9) Short RD, Fukunaga K. The optimal distance measure for nearest neighbor classification. IEEE Transactions on Information Theory 1981; 27:622-7. 10.1109/TIT.1981.1056403
10) C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/cjlin/libsvm, 2001.
11) Rossi D, Omana M, Giaffreda D, Metra C. Modeling and detection of hotspot in shaded photovoltaic cells. IEEE Trans Very Large Scale Integr Syst 2015; 23:1031–9. doi:10.1109/TVLSI.2014.2333064.
12) Johnson J, Montoya M, Fresquez A, Gonzalez S, Granata J, Mccalmont S, et al. Differentiating Series and Parallel Photovoltaic Arc- Faults Arc-Fault Types 2012.
13) Zhao Y, Yang L, Lehman B, de Palma J-F, Mosesian J, Lyons R. Decision tree-based fault detection and classification in solar photovoltaic arrays. 2012:93–9. doi:10.1109/APEC.2012.6165803.
14) Yi Z, Etemadi AH. A novel detection algorithm for Line-to-Line faults in Photovoltaic (PV) arrays.