JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Quality Analysis of Solar Panel

Shradha Jadhav[1] , A.S.Deokate[2], Aishwarya Melgiri[3], Somesh Kamble[4], S.S.Savkare[5]

Department of Electronics and Telecommunications, SKNCOE, SPPU, Pune

anjujadhav2000@gmail.com

d.archana455@gmail.com

aishwaryamelgiri007@gmail.com

someshkamblesrk@gmail.com

swati_savakre@yahoo.com

Abstract- **the system was able to detect all faults if studied over a two-week period. These findings show that a machine-learning model may be used to simulate the projected output of a solar panel system (using meteorological data) and use it to determine if the system is producing as much power as it should be. Adding meteorological the rising use of solar panels around the world emphasizes the need of being able to detect defects in solar-panel-based systems. In this study, historical solar panel power output (kWh) was integrated with meteorological data to build a machine-learning model that could predict the projected power production of a solar panel system. A comparison of the expected and actual power outputs was done using the expected power output to determine if the system was faulty. As a result, while simulating a failure (a 50% reduction data, enhancing the precision of the meteorological data, and training the machine learning model on more data are some of the alternatives for improving the system**.

Keywords – Python, Sckit-learn.

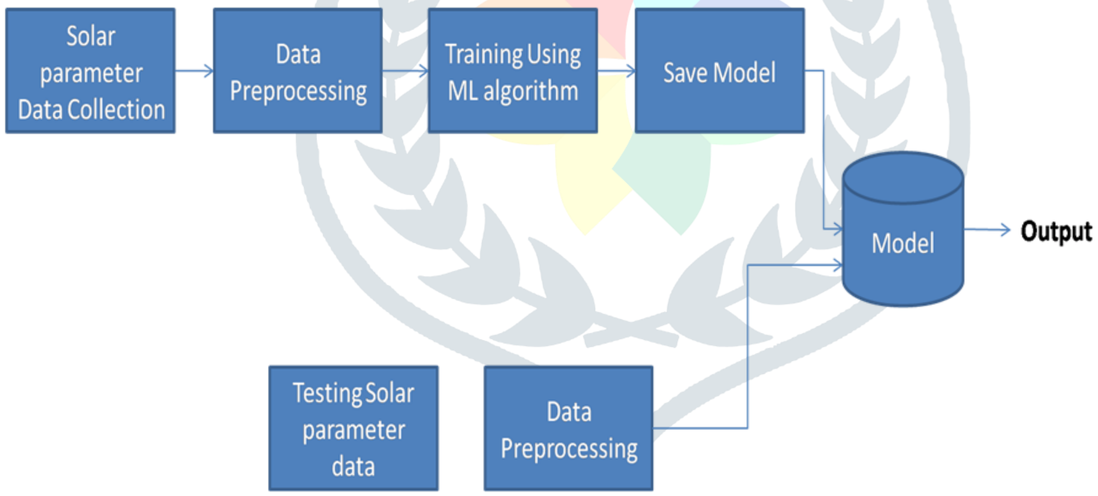
I INTRODUCTION

The usage of solar panels, also referred to as photovoltaic (PV) modules, has seen an almost exponential increase globally during the last decade [19]. How a system of PV modules perform depends heavily on weather [9], dirt covering the panels [2] and a multitude of other factors. Some factors that decrease the energy production of PV modules are natural and cannot be prevented by an owner, such as the angle of the sun, clouds and other weather related causes. There are other sorts of energy production decreases that an owner could actually stop from happening. Examples of these could be a PV module breaking down, leaves covering the panels or other factors of similar nature. The focus of this project was to develop a software based fault detection system that can detect decreases of this kind, in a photovoltaic system. To detect severe decreases, hereinafter referred to as faults, in a PV system the historical energy production of the system and meteorological data was utilized. The meteorological data is gathered in close geographical and temporal proximity to the PV system. The energy production of a PV system over a given time period (e.g. an hour) will in this report be referred to as power output measured in the unit of kWh. The power output and meteorological data was then used to train a machine-learning model, which predicts the expected power output for a given PV system. The fault detection system uses the expected, or predicted, power output and the actual power output to detect if the PV system is faulty or not. A fault detection system of this kind will make it easy to diagnose and detect afaulty PV system seeing as it can be done remotely. This could lead to faster reparations or necessary maintenance and more produced energy. The aim of creating a fault detection system that can detect faults in PV systems by using machine learning together with historical power output and meteorological data was achieved. However, the extent of what the system can detect turned out to be dependent on three parameters: the size of the power decrease, the threshold and the time horizon. By simulating a fault (50% power output decrease) it was found that, the fault detection system can detect all of the faults while analysing over a two-week period. However, simulating lesser faults, decreasing the time horizon or decreasing the threshold may negative impact the result which is elaborated on later in the report

II LITERATURE SURVEY

suggested the generic video camera dependent convolutional neural network (CNN) based air-writing framework has been suggested by Gestures are made with a fixed-color marker in front of a generic video camera, then color-based segmentation is used to identify the marker and track the marker tip's trajectory. The gesture is then classified using a pre-trained CNN. With the newly collected data, transfer learning improves recognition accuracy even more. Due to color-based segmentation, the system's performance is highly dependent on lighting conditions. The system can distinguish isolated unistroke numbers of several languages in a less variable illumination setting. In human independent evaluations, the proposed methodology achieved 97.7%, 95.4%, and 93.7% recognition rates for English, Bengali, and Devanagari numerals, respectively Grigoris Bastas et al investigate deep learning architectures for the air-writing recognition issue, in which a person freely creates text in three dimensions. The handwritten numerals 0 to 9 are constructed as multidimensional time-series obtained from a Leap Motion Controller (LMC) sensor. To predict the motion trajectory, we look at both dynamic and static methodologies. Several state-of- the-art convolutional and recurrent architectures are trained and compared. We used a Long Short-Term Memory (LSTM) network, as well as its bidirectional counterpart (BLSTM), to map the input sequence to a fixed-dimensionality vector, which was then transferred to a dense layer for classification among the target air-handwritten classes. This system uses 1D Convolutional Neural *Networks (CNNs) to encode the input features before feeding them to an LSTM neural network in the second design (CNN-LSTM).* shows how the Kinect sensor's colour and depth images are used to recognise the user's air- writing, which includes Persian digits and integers. We suggest a simple yet effective method termed slope variations detection to extract a feature vector from the trajectory, which is robust to changes in the trajectory's size, translation, and rotation. In addition, a novel analytical classifier for mapping a vector to a character is proposed. This classifier outperforms classic classifiers like SVM, HMM, and K Nearest Neighbors in terms of speed and accuracy. The average recognition rate for digits and numerals in Persian is 98%, which is pretty adequate for a practical system, according to experimental result.

III PROPOSED SYSTEM



Fig(1): Proposed Methodology

The block diagram of this research work has shown in Fig, for a better understanding of the proposed work. The system consists of three modules: Data gathering, data preprocessing and fault detection. The first module of the system is the Data Gathering module. This is where the system retrieves meteorological data and combine it with PV system data for a given PV system. PV system has their own file containing data for the PV system corresponding meteorological data for each data point. The compiled file is then forwarded to the Data Preprocessing module. Solar Panel parameter data collection - The dataset of the fault detection in solar panel is collected from the online dataset. https://data.mendeley.com/ provides the dataset. Grid- connected PV System Faults (GPVS-Faults) data are collected from lab experiments of faults in a PV microgrid system. There are 16 data files in „.mat" and also „.csv", each for one experiment scenario, including photovoltaic array faults; inverter faults; grid anomalies; feedback sensor fault; and MPPT controller faults of various severity. PVS-Faults data can be used to design/ validate/ compare various algorithms of fault detection/ diagnosis/ classification for PV system protection and reactive .

SOFTWARE SPECIFICATION :

Python : Python is a high-level programming language extensively used for programming. Python, an interpreted language, supports several programming scripts and a syntax that allows you to use programs in most languages such as C ++ or Java. The language provides constructions designed to permit clear programs at each scale. Python is easy and simple to know, the python code is way easier than alternative languages.

Machine Learning Library: Scikit-learn : Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

The faults were introduced manually halfway during the experiments. The high-frequency measurements are noisy; with disturbances and variations of temperature andinsolation during and between the experiments; MPPT/IPPT modes have adverse effects on the detection of lowmagnitude faults. After critical faults, the operation is interrupted array current measurement. Vpv: PV array voltage measurement. Vdc: DC voltage measurement. ia, ib, ic: 3-Phase current measurements. va, vb, vc: 3- Phase voltage measurements. Iabc: Current magnitude. If: Current frequency. Vabc: Voltage magnitude. Vf: Voltage frequency.

Data Preprocessing :

The Data Preprocessing module is responsible for two key areas. Firstly, cleaning the data retrieved by the Data Gathering module. Cleaning data is the process of detecting and correcting corrupt or incorrect data. Secondly, construct and add more features to the data set. Constructing features refers to the process of converting existing features to another form. After these steps the data can now be used in the Fault Detection module.

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

Training and Testing using Machine learning mode :

The proposed system uses machine learning algorithm for classification problem. The system uses Support Vector Machine (SVM), K Nearest Neighbor (KNN) and decision tree (DT) algorithm for classification of input PV parameters into faulty and normal. The detail of the machine learning algorithm is explained in the section below.

Support Vector Machine:

The SVM algorithm was first developed in 1963 by Vapnik and Lerner. SVM [12] is a binary classifier based on supervised learning which gives better result than other classifiers. SVM classifies between two classes by constructing a hyper-plane in high-dimensional feature space which can be used for classification. SVM is a classification algorithm, which is based on different kernel methods. SVM is classified in two groups. Support Vector Machine (SVM)

Table(1): Comparative Parameters

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Time	Ipv	Vpv	Vdc	ia	ib	ic	va	vb	vc	Iabc	If	Vabc	Vf
2	2.82E-05	1.572327	101.3489	144.1406	-0.13513	0.490112	-0.35499	41.74454	-149.873	109.0646	1	50	1	50
3	0.000128	1.503265	101.4587	143.5547	-0.10828	0.510254	-0.38855	46.83151	-150.717	105.83	1	50	1	50
4	0.000228	1.492859	101.5747	143.5547	-0.1687	0.496826	-0.33484	51.07468	-152.019	102.5431	1	50	1	50
5	0.000328	1.558136	101.3123	143.2617	-0.13513	0.510254	-0.3617	55.84824	-152.585	98.14326	1	50	1	50
6	0.000428	1.631927	101.1414	143.8477	-0.20227	0.50354	-0.32142	60.05524	-152.609	94.26173	1	50	1	50
7	0.000528	1.60733	101.0132	143.8477	-0.1687	0.510254	-0.34827	64.86496	-154.152	90.27573	1	50	1	50
8	0.000628	1.500427	100.9338	143.8477	-0.23584	0.50354	-0.30799	69.5903	-154.815	86.59912	1	50	1	50
9	0.000728	1.445557	101.0437	143.8477	-0.19556	0.510254	-0.32142	74.48441	-154.851	83.19173	1	50	1	50
10	0.000828	1.51178	101.3916	143.8477	-0.26941	0.510254	-0.27442	77.47391	-154.538	78.57086	1	50	1	50
11	0.000928	1.541107	101.3123	143.5547	-0.22913	0.496826	-0.30127	81.84967	-154.948	73.76114	1	50	1	50
12	0.001028	1.524078	101.4587	143.5547	-0.29627	0.516968	-0.25428	85.8638	-155.056	69.46172	1	50	1	50
13	0.001128	1.474884	101.532	143.2617	-0.2627	0.490112	-0.27442	89.91409	-154.345	65.14221	1	50	1	50
14	0.001228	1.502319	101.532	143.5547	-0.32983	0.510254	-0.23414	93.67508	-154.152	60.90307	1	50	1	50
15	0.001328	1.487183	101.3489	143.5547	-0.28284	0.483398	-0.24085	97.83386	-152.742	55.99289	1	50	1	50
16	0.001428	1.594086	101.2817	143.8477	-0.34998	0.516968	-0.20057	101.2332	-152.597	51.43229	1	50	1	50
17	0.001528	1.525024	101.3367	143.2617	-0.29627	0.483398	-0.20057	105.9706	-151.621	46.82749	1	50	1	50
18	0.001628	1.465424	101.2756	143.8477	-0.3634	0.510254	-0.18042	109.1289	-150.379	44.32821	1	50	1	50
19	0.001728	1.528809	101.3855	143.2617	-0.30298	0.476685	-0.167	112.4197	-149.33	41.87714	1	50	1	50
20	0.001828	1.585571	101.3	143.8477	-0.38355	0.50354	-0.15357	115.132	-149.608	33.22205	1	50	1	50

IV CONCLUSION

The aim of the project was to create a system that can detect faults in PV systems with the use of machine learning and meteorological data. This was achieved by creating an expected output of a PV system which is compared to the actual output. PV systems are subject to various faults and failures, and early fault detection of those faults and failures is very important for the efficiency and safety of the PV systems. ML-based fault detection models are trained with data and provide prediction results with very high accuracy. However, data-based fault detection models for PV systems can sometimes give false predictions, especially when the environmental parameters are not taken into consideration. This approach develops an intelligent fault detection model for PV arrays based on SVM and KNN for accurately classifying the fault types. The model was trained with a large dataset containing different data values under different environmental conditions in the summer and the winter season

REFERENCES

1. M. Santhakumari and N. Sagar, "A review of the environmental factors degrading the performance of silicon wafer-based photovoltaic modules: Failure detection methods and essential mitigation techniques," Renewable and Sustainable Energy Reviews, 2019, 110, pp. 83-100.

2.G. Masson and I. Kaizuka, Trends in Photovoltaic Applications. International Energy Agency, Aug. 2019, pp. 75, 94, ISBN: 978-3-906042-91-6.

3. B. Guo, W. Javed, B. W. Figgis, and T. Mirza, "Effect of dust and weather conditions on photovoltaic performance in doha, qatar," in 2015 First Workshop on Smart Grid and Renewable Energy (SGRE), 2015, pp. 1–6.

4.B. A. Alsayid, S. Y. Alsadi, J. S. Jallad, and M. H. Dradi, "Partial shading of PV system simulation with experimental results," Smart Grid and Renewable Energy 04(06):429-435, 2013.

5.V. S. B. Kurukuru, A. Haque, M. A. Khan and A. K. Tripathy, "Fault classification for Photovoltaic Modules Using Thermography and Machine Learning Techniques," 2019 International Conference on Computer and Information Sciences (ICCIS), 2019, pp. 1-6, doi: 10.1109/ICCISci.2019.8716442.

7.Prof. Vishal R. Shinde, Mr.Vipinkumar Gautam, Mr.Chinmay Gawande, Miss.Pooja Mahale, "Solar Data Faults Detection Using Machine Learning," International Journal for Research in Engineering Application & Management (IJREAM), 2019.

8. Barun Basnet, Hyunjun Chun, Junho Bang, "An Intelligent Fault Detection Model for Fault Detection Photovoltaic Systems", Journal of Sensors, vol. 2020, Article ID 6960328, 11 pages, 2020.

9. Michael Oberdorf, Sunil Rao, Andreas Spanias, Elias Kyriakides SenSIP Center, “Machine Learning for Rooftop PV Fault Detection” School of ECEE, Arizona State University, KIOS Center, University of Cyprus.

10. GPVS-Faults: Experimental Data for fault scenarios in grid-connected PV systems under MPPT andIPPT modes Available Online:https://data.mendeley.com/datasets/n76t439f65/