EPiC Series in Computing

Volume 63, 2019, Pages 21–33

Proceedings of 32nd International Conference on Computer Applications in Industry and Engineering

Machine Learning as an Efficient Diagnostic Tool for Fault Detection and Localization in Solar Photovoltaic Arrays

Masoud Alajmi, Sultan Aljahdali, Sultan Alsaheel, Mohammed Fattah, and Mohammed Alshehri

College of Computers and Information Technology, Taif University

Al-Hawiya, Taif 21974, Saudi Arabia

ms.alajmi@tu.edu.sa, aljahdali@tu.edu.sa, sultanalsaheel@gmail.com, fattah6335@gmail.com, eng.mohammed20@outlook.sa

Abstract

Solar energy, one of many types of renewable energy, is considered to be an excellent alternative to non-renewable energy sources. Its popularity is increasing rapidly, especially because fuel energy consumes and depletes finite natural resources, polluting the environment, whereas solar energy is low-cost and clean. To produce a reliable supply of energy, however, solar energy must also be consistent. The energy we derive from a photovoltaic (PV) array is dependent on changeable factors such as sunlight, positioning of the array, covered area, and status of the solar cell. Every change adds potential for the creation of error in the array. Therefore, thorough research and a protocol for fast, efficient location and correction of all kinds of errors must be an urgent priority for researchers.

For this project we used machine learning (ML) with voltage and current sensors to detect, localize and classify common faults including open circuit, short circuit, and hot-spot. Using the proposed algorithm, we have improved the accuracy of fault detection, classification and localization to 100%. Further, the proposed method can execute all three tasks (detection, classification, and localization) simultaneously.

1 Introduction

Electrical energy generated by fuel and gas exists only as non-renewable power that consumes finite resources and causes pollution in the environment: not only land pollution and oil spills, but also detrimental atmospheric effects, including acid rain and, potentially, catastrophic climate change. Renewable energy sources solve these problems by generating pure, clean energy through harnessing of the power of sunlight, wind, and water. Over the past 10 years the market for solar photovoltaic cells has experienced huge gains as research on photovoltaic array has improved and solved production

Q. Yuan, Y. Shi, L. Miller, G. Lee, G. Hu and T. Goto (eds.), CAINE 2019 (EPiC Series in Computing, vol. 63), pp. 21–33

problems [1]. Still, PV systems (like all systems) have some predictable faults. Short circuit, open circuit and hot-spot fault can affect the efficiency of generating power in the system, so it is essential to monitor and to control the performance of PV in efficient uncompromising ways in order to prevent all possible faults. We will use Machine Learning, one of the newest and most promising research fields in the world, to achieve this end.

2 Related Work

Reference [2] focuses only on open-circuit and short-circuit faults. Researchers employed graph-based, semi-supervised Machine Learning along with the variance and the mean of these graphs to detect and localize the faults. Other researchers [3] tested several types of ML techniques in order to predict the output power for PV systems. They concluded that Artificial Neural Network (ANN) is best suited to predict output power in a PV system. In [4], the authors used Machine Learning Techniques (Model-Based, Data-Driven and Hybrid) to detect and to classify the faults in PV array. They divided the application of these techniques into four methods: Detection, Identification, Diagnostic and Prognostic. The researchers in [5] used Machine learning based on a Support Vector Machine (SVM) to detect just one type of fault (Line-to-Line) that may produce excessively high levels of power and that may also create fire hazard conditions, performance issues and subsequent revenue losses. The SVM method shows high accuracy for detecting this type of fault. In [6], research by Alajmi and Abdel-Qader designed a new approach to detect faults in a PV array. They compared current-voltage values to maximum measured signal values generated under normal operation conditions. Their results successfully identified the the specific type and location of faults (open-circuit, short-circuit, and hotspot). On the other hand, they neglected to examine any temporary effects caused by environment. If the proposed algorithm detects any of those, it may produce a false report of hotspot fault. Also, they did not classify various types of faults for which the technology is best suited. In [7], Alajmi's research group implemented hybrid methods of data analysis and a network of voltage-current sensors to detect, localize, and classify the faults with accuracy reaching to 100%. Our proposed algorithm and method are based on the same model used in their study. We have, however, now developed a new system to detect, localize, and classify faults by applying the Machine Learning method of Support Vector Machine in one execution time.

3 Framework of the Proposed Structure

3.1 Solar cell connections

The proposed method uses 36 cells for one module, with an array of 8 modules per branch, connected in series. Each series of 8 PV modules is called a branch. We will divide it into 4 zones, each zone containing 2 modules. In this paper, we use 8x3 model which means that 3 branches connected in parallel and each branch contains 8 modules connected in series as illustrated in Figure 1.

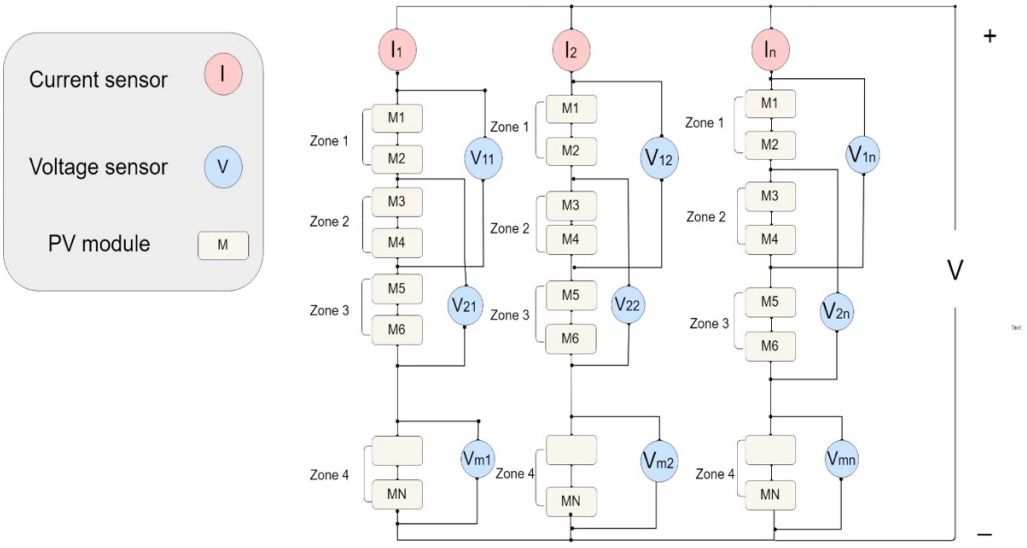

Figure 1: PV System array model (NxM) used to generate training data.

The data we obtained from the proposed model (NxM) shown in Figure 1, where N represents the number of PV modules connected in series and M represents the number of branches connected in shunt. The array contains 9 voltage sensors which are connected with PV modules in every branch to form zones. It also contains 3 current sensors. We can detect, classify and localize the faults according to these zones.

3.2 Fault detection, classification and localization using Machine Learning

In Figure 2, we tested the system on one branch with all possible fault scenarios to obtain the training data for the ML-model, we then trained our ML model using the classification learner app in MATLAB with the training data. Next, we tested our model on every branch in the system. We calculated margins 5% from above and below to be sure the prediction in the ML trained model is accurate. We then trained the ML-model using the training data. Finally, we tested the ML model using the PV array 8x3 and reported the condition of the PV array.

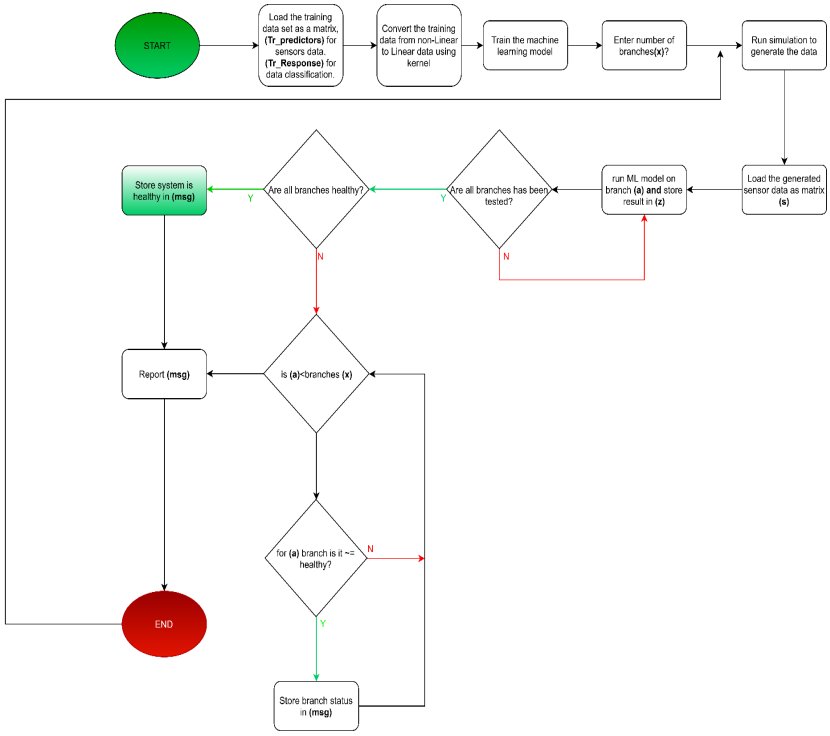

Figure 2: Flowchart of the Proposed Algorithm.

3.3 Training data

We created a fault in every zone and took measurements of these zones in all possible fault scenarios, then we calculated the margin at 5% from above and below. We then have three readings from each possible scenario.

3.4 Cross validation

We divided the training data set into four folds then we used every fold as a validation test once and as training data three times, so that we could test every fold. We also measured the accuracy of every fold to obtain the highest accuracy without overfitting. See Figure 3.

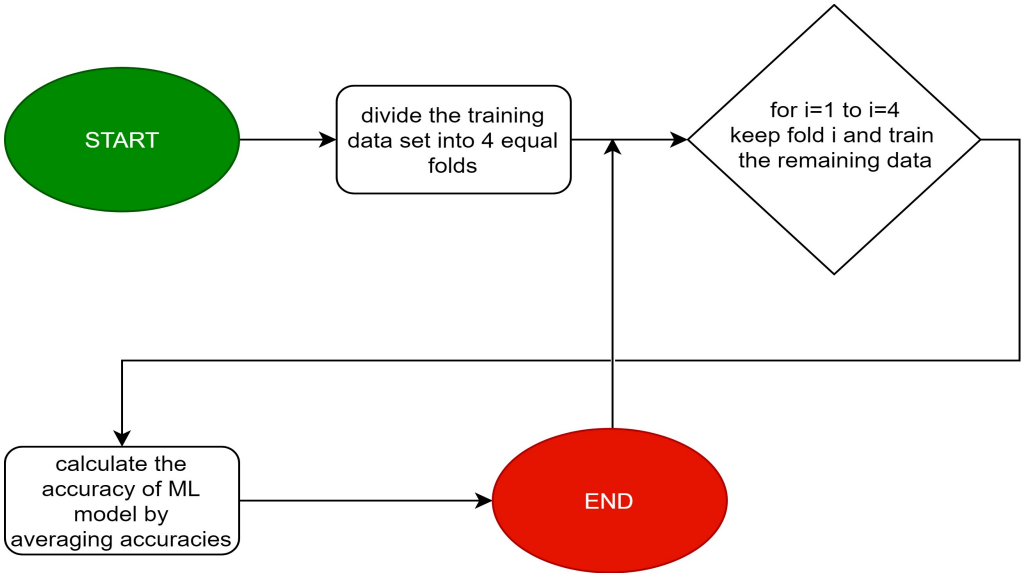

Figure 3: Flowchart of Cross Validation

3.5 Linear kernel

Linear kernel is very useful and a powerful tool for creating text classification and faster training SVM using linear kernel. Also, when training SVM using linear kernel we only need to optimize the C regularization parameter. Instead, if training with another kernel having more than an optimize C regularization parameter, it needs to optimize the γ parameter. Linear kernel converts the data from non-linearly separable to linearly separable as constructed in (1) [8]:

$$K(x, y) = w^T y + b \tag{1}$$

Where $K(x, y)$ is the point on the feature space and w^T is the transformation vector in feature space, y, then, is the vector in the feature space and b is a scalar that is equivalent to the bias that intercepts the y-axis.

3.6 Hyperplane

Hyperplane is a subspace of two or three dimensions. We use it to create classes by separating the data. We can then evaluate hyperplane according to (2) [8].

$$w.x + b = 0 \tag{2}$$

Where w and x are vectors placed in the two-dimensional plane and b is the scalar that is equivalent to the bias that intercepts the y-axis that is equivalent to the c in the line equation $y = ax + c$.

3.7 One versus one approach

In order to classify our data, we used one vs. one (OVO) approach by training one classifier for pair classes. According to K(K-1)/2, we trained 78 classifiers to classify our data using coding matrix (C) with the size NxM as shown in Figure 4, where N denotes number of classes and M is number of classifiers. The principle of the process of detection, localization, and classification is based on three scenarios. The first scenario occurs when C is equal to 0 that indicates to ignore all observations in the corresponding classes. The second case is at C is equal to 1 that represents the observations in the corresponding classes are positive. The third scenario is at C is equal to -1 that indicates the observations in the corresponding classes are negative. Those scenarios are shown clearly in Figure 4 and Table 1.

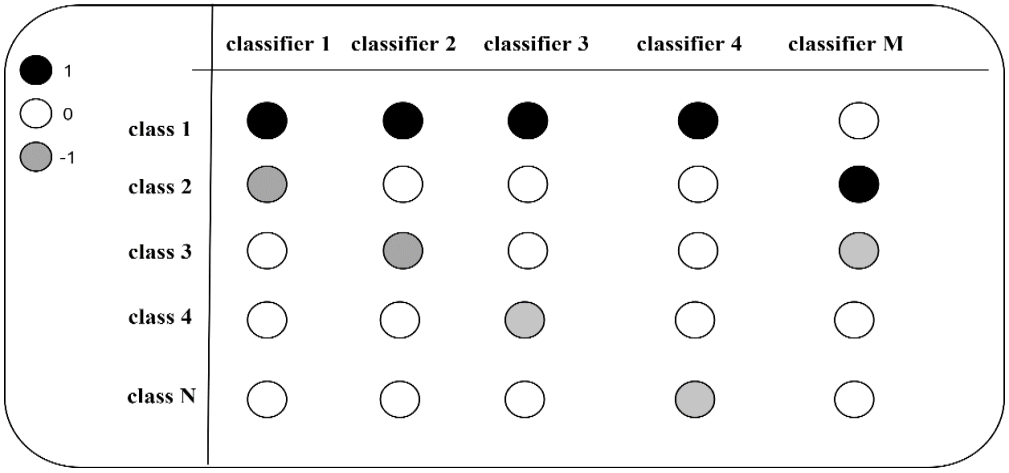

Figure 4: Coding Matrix (C) shows the principle of the process of the detection, localization, and classification.

Table 1: Samples of coding matrix that shows the classes and their classifiers.

	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Classifier 5
Healthy System	0	0	0	0	0
Short Circuit at zone 2	-1	0	0	0	0
Short Circuit at zone 3	0	-1	0	0	0
Open Circuit at zone 1	0	0	0	-1	0

Table 1 illustrates a sample under healthy system scenario. Every class has a unique codeword which is different from other classes. When a new input is being processed, it will be compared with all codewords in C to find an identical codeword in the best case or to calculate the distance between them and choose the shortest distance as predicted class according to (3)[8].

$$\bar{k} = argmin_k \frac{\sum_{l=1}^{L} |c_{kl}| g(c_{kl}, s_l)}{\sum_{l=1}^{L} |c_{kl}|} \quad (3)$$

Where $\bar{k}$ is a new observation assigned to the class, C is the coding design matrix with elements c_{kl} and s_l is the predicted classification score for the positive class of learner l.

If the distance between two codewords is the same, we can correct it by using $(d_r - 1)/2$. The distance between two codewords can be constructed as shown in (4)[9]

$$d_r = \min \{ \sum_{j=1}^{N} (1 - sign(c_{i1}^{j} \cdot c_{i2}^{j}))/2 \} \quad (4)$$

Where d_r is the distance between two codewords and c_{i1}^{i} is the class, where $i_1 \neq i_2$ and jth is the position of the codeword and i is the codeword.

3.8 Confusion matrix

We tested our ML model 109 times and ended up having 109 true prediction classes. We applied a confusion matrix to calculate the accuracy of prediction in our ML model and reached 100% accuracy according to (5)[20] as shown Figure 5.

$$\text{ACC} = \frac{true\ positive + true\ negative}{total\ number\ of\ observations} X100 \quad (5)$$

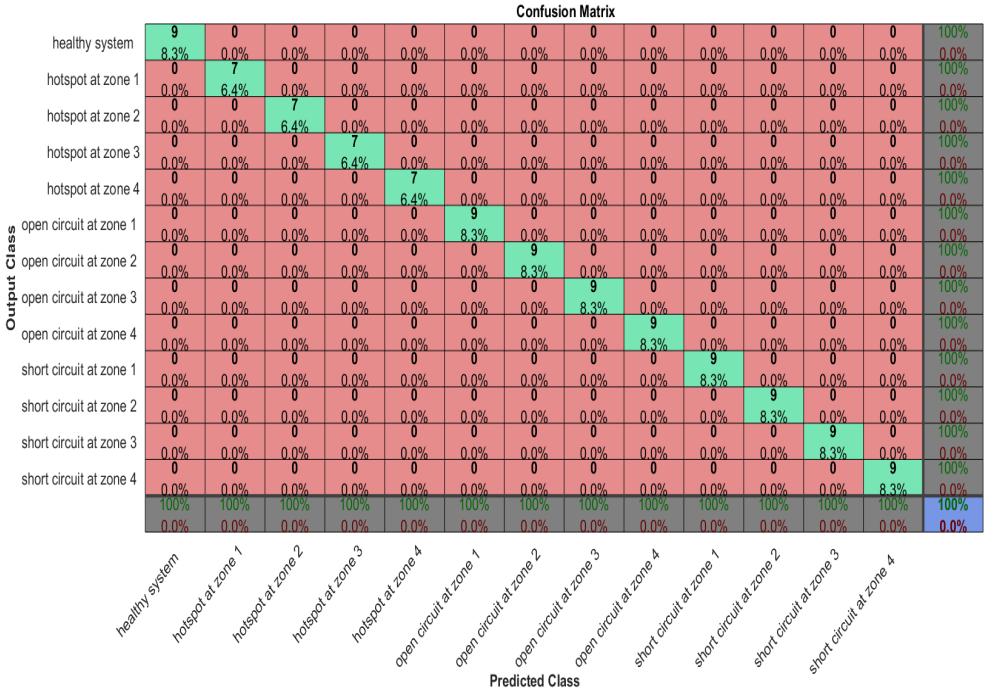

Figure 5: Confusion matrix with the observation, the true class, and the predict class.

4 Simulation Results

In order to obtain the training data, we simulated an 8x3 PV array using a Simulink model as illustrated in Figure 6. This model has been executed under standard test conditions (STC) with the simulation parameters shown in Table 2.

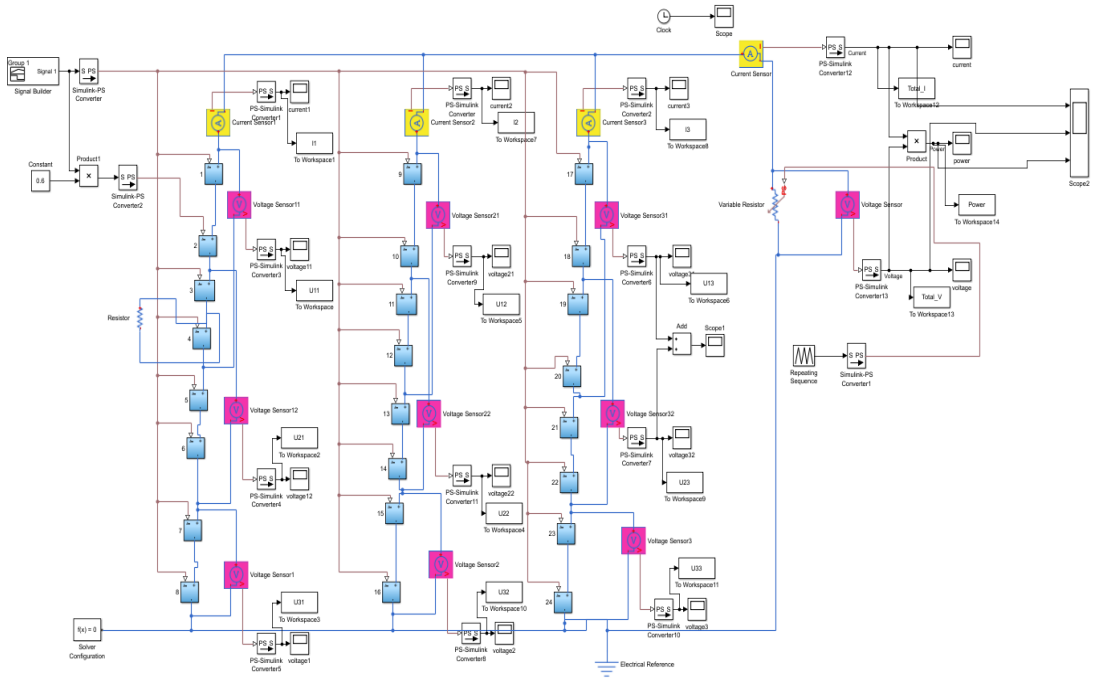

Figure 6: Simulink model of 8x3 photovoltaic array used to generate training data.

Table 2: Standard Test Conditions (STC)

Performance STC	
Operating temperature	25℃
Incident solar irradiance level	$1000\,{}^{W}/_{m^2}$
Open-circuit voltage	$v_{oc} = 0.6\,V$
Short-circuit current	$I_{SC} = 7.34\,A$
Series resistance	$R_s = 0\Omega$
Shunt resistance	$R_{sh} = 200\Omega$
Threshold voltage	$0.0014\,V$

4.1 Short-circuit fault

We can know whether there is a short-circuit fault if there are any changes in voltage sensors of that branch. Figure 7 shows the current-voltage (I-V) curve of the 8x3 PV array when it has short-circuit fault compared with healthy condition. Also, the output power is decreasing as shown in Figure 8. Figure 9 shows the message box that displays the status of PV array under short-circuit fault circumstances.

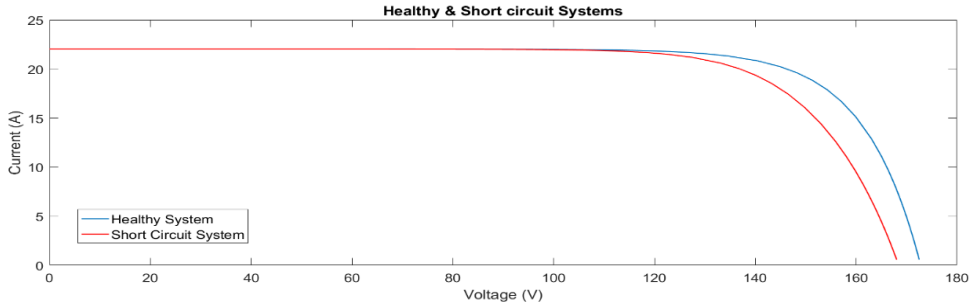

Figure 7: I-V Curve of the 8x3 PV array model shown under healthy and short-circuit fault.

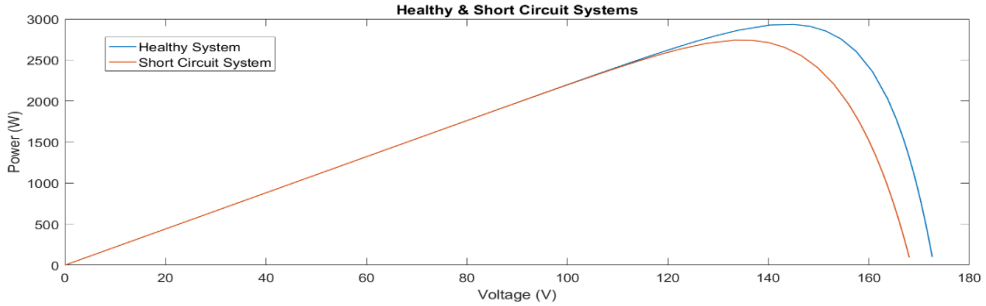

Figure 8: P-V curve of 8x3 PV array depicted under healthy system and short-circuit fault.

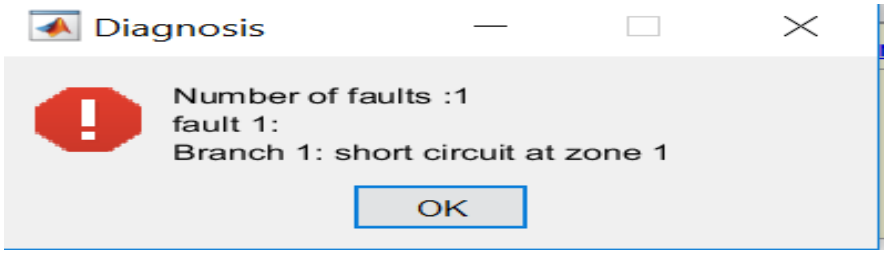

Figure 9: The message box that expresses the number and type of faults.

4.2 Open-circuit fault

We need to ascertain whether there is an open-circuit when there is any change in the current sensors: usually its measurements will be ~0 in that branch. Figure 10 depicts the current-voltage (I-V) curve of 8x3 PV array when it has open-circuit fault compared with healthy condition. Also, the output power is decreasing as shown in Figure 11. Figure 12 shows the message box that summarizes the condition of PV array under open-circuit fault showing the total number of faults, type of fault, and its location.

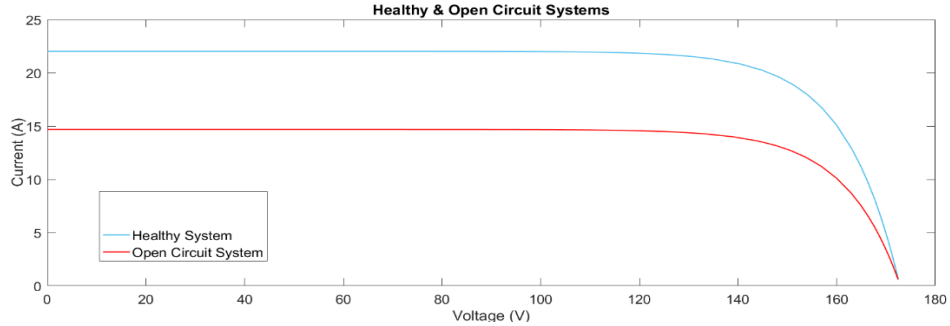

Figure 10: I-V curve in healthy and open-circuit fault of the 8x3 PV array model.

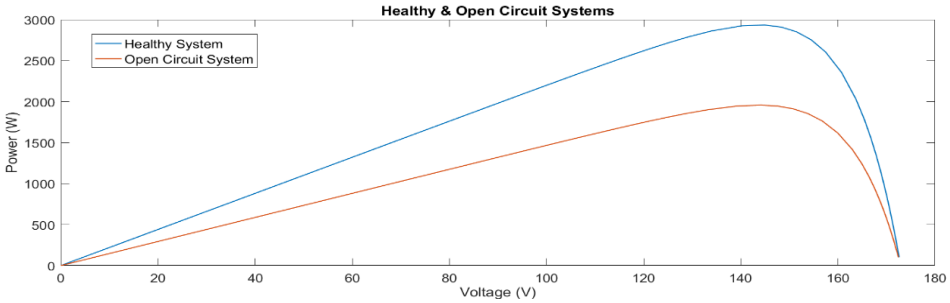

Figure 11: P-V curve in healthy and open-circuit fault of 8x3 PV array model.

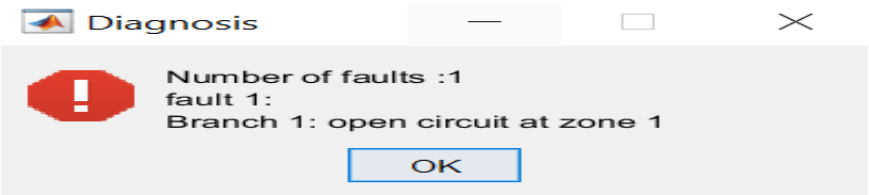

Figure 12: The message box indicates the total number of faults that occurred provides the class and exact location of the fault.

4.3 Hot-spot fault

Hotspot fault can be detected if there are any changes in the voltage sensors and the current sensor of the faulty branch. Figure 13 shows the I-V curve relation of the proposed model under healthy and

hotspot fault conditions. Figure 14 depicts the P-V curve relation of the proposed model under healthy and hotspot fault. Figure 15 presents the message box that indicates the diagnosis of the system.

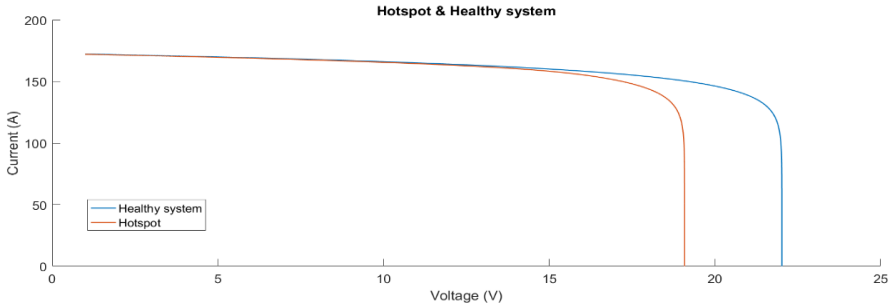

Figure 13: I-V curve showing healthy and hotspot of the 8x3 PV array model.

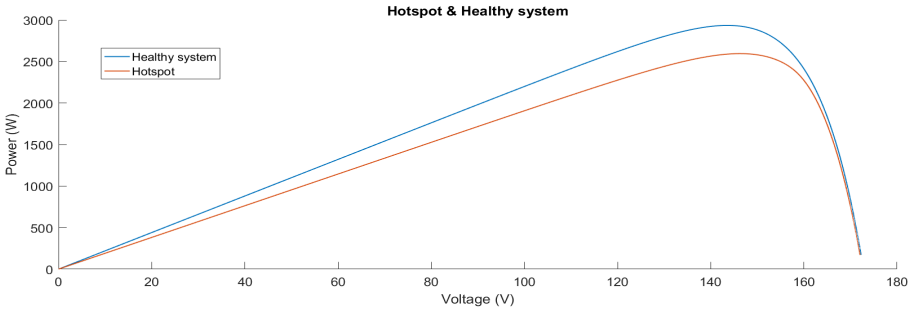

Figure 14: P-V curve showing healthy and hotspot of the 8x3 PV array model.

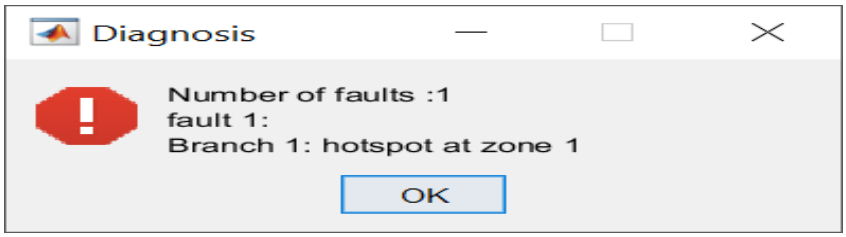

Figure 15: The message box that reports the number, location, and type of faults.

4.4 Healthy system

The system is healthy if and only if there is no indication that any fault occurred. Figure 16 and Figure 17 show I-V and P-V curves under normal operating conditions (healthy system) while Figure 18 shows the displayed message box under healthy conditions.

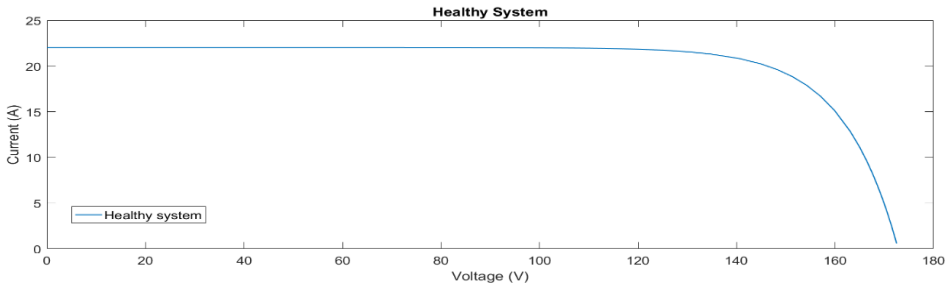

Figure 16: I-V curve of the 8x3 PV array model operating under healthy system circumstances.

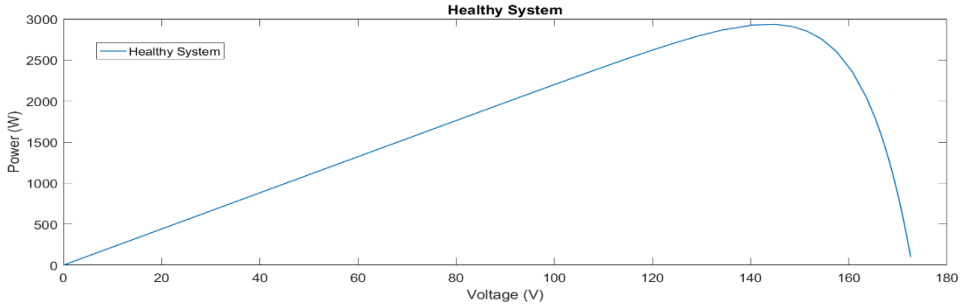

Figure 17: P-V curve of the 8x3 PV array operating under healthy system circumstances.

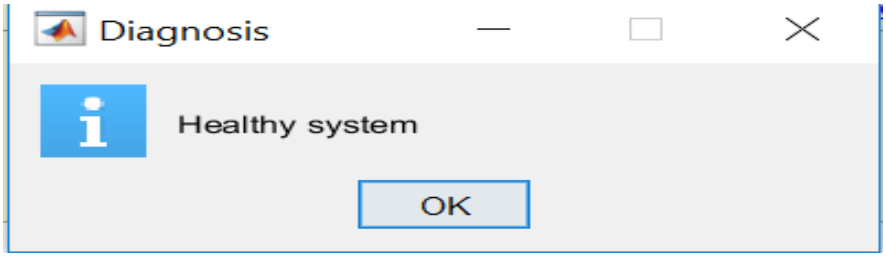

Figure 18: Message box indicating that the system is healthy.

5 Conclusion and Future Work

5.1 Conclusion

This project has presented a developed application for existing fault detection technology. We designed a more efficient and reliable method through the use of train data which was obtained from a Simulink voltage-current sensors model. After that, we created a model for healthy system and faulty system that reports both types of data from the voltage-current sensor. We used it as training data, then created a new model and trained all data on every branch and zone that we had collected from previous models. In this way we were able to test our system under different scenarios. Our results demonstrate a measurable improvement in the accuracy and efficiency of fault detection, localization, and classification in PV array. Also, the proposed algorithm is capable of processing multiple measurements simultaneously.

5.2 Future work

Using the proposed monitoring and fault detection system, we will continue to test the proposed method at the Taif University campus. Additional emphasis will be placed on practical applications as we test our system under real circumstances. We are also planning to apply isolation method to improve the accuracy of results in future investigations.

References

[1] K. Stefan (2019). *What Are the Effects of Non-Renewable Resources on Living Organisms?* Retrieved from | *https://www.livestrong.com/article/157263-what-are-the-effects-of-non-renewable-resources-on-living-organisms*

[2] Y. Zhao, R. Ball, J. Mosesian, J.-F. de Palma, and B. Lehman, "Graph-Based Semi-supervised Learning for Fault Detection and Classification in Solar Photovoltaic Arrays," *IEEE Trans. Power Electron.*, vol. 30, no. 5, pp. 2848–2858, 2015.

[3] S. Theocharides, G. Makrides, E. George, and A. Kyprianou, "System Power Output Prediction," *2018 IEEE Int. Energy Conf.*, pp. 1–6, 2018.Prognostic.

[4] S. Rodrigues and H. G. Ramos, "Machine Learning in PV Fault Detection , Diagnostics and Prognostics : A Machine Learning in PV Fault Detection , Diagnostics and Prognostics : A Review," February 2018, 2017.

[5] Z. Yi and A. H. Etemadi, "A novel detection algorithm for Line-to-Line faults in Photovoltaic (PV) arrays based on support vector machine (SVM)," *IEEE Power Energy Soc. Gen. Meet.*November, 2016.

[6] M. Alajmi and I. Abdel-Qader, "Fault detection and localization in solar photovoltaic arrays using the current-voltage sensing framework," *2016 IEEE Int. Conf. Electro Inf. Technol.*, pp. 0307–0312, 2016.

[7] M. Alajmi, O. Aljasem, N. Hassan, A. Alqurashi, and I. Abdel-Qader, "Fault Detection and Localization in Solar Photovoltaic Arrays Framework : Hybrid Methods of Data-Analysis and a Network of Voltage-Current Sensors," *2018 IEEE Int. Conf. Electro/Information Technol.*, pp. 404–410, 2018.

[8] S. Escalera, O. Pujol, and P. Radeva, "Separability of ternary codes for sparse designs of error-correcting output codes," *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 285–297, 2009.

[9] A. Kowalczyk, "Support Vector Machine Succinctly," *Tradingeconomics*. Retrieved from https://www.syncfusion.com/ebooks/support_vector_machines_succinctly

[10] Classifier evaluation with imbalanced datasets. (2019). *Basic evaluation measures from the confusion matrix.* Retrieved from *https://classeval.wordpress.com/introduction/basic-evaluation-measures/*