ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

Preparing for Machine Learning Applications: Signal-to-Image Methods for Photovoltaic Fault Diagnosis

[1] A. Suman Kumar, [2] V. Likitha, [3] S. Lava Kishore, [4] B. Venu Kumar, [5] Mr. Paindla Thirupathi, [6] Mr. V. Pradeep Kumar,

[1,2,3,4,] UG Scholar, Dept. of CS, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

[5] Assistant Professor, Dept. of CSE, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

[6] Assistant Professor, Dept. of EEE, Narsimha Reddy Engineering College, Maisammaguda, Kompally, Secunderabad, India.

Abstract—

In order to use machine learning models developed for 2D data, this research investigates several methods for converting 1-dimensional time-series data into 2-dimensional pictures. Markov transition, Poincaré plots, spectrograms, heatmaps, direct plots, phase space transformation, and Gramian angular fields are among the eight approaches that are presented. We put these techniques to the test by modeling a grid-connected photovoltaic (PV) system with a shorted string defect and one without. All approaches use the same fixed window size of 256 sample points to record the fault and no-fault replies. Python 3 programming on a laptop with little computational capabilities is used to evaluate all transformation methods. A one-channel grayscale picture or a three-channel RGB image may be produced by each modification. You have the option to raise or reduce the dimensions of the created picture when storing it. Various approaches to converting 1D time-series data into 2D visuals for use in machine learning have resulted in distinct visual representations of the shorted string fault and a no-fault.

Topics covered include 2D images, signal modification, problem diagnostics, and machine learning/deep learning.

I. INTRODUCTION

Machine learning (ML) and deep learning (DL) are growing in popularity as AI finds more and more uses in many fields. There is a notable use of AI in the power system's many sub-disciplines, particularly those dealing with microgrids, RES, and the energy utility grid [1]. In these domains, AI finds use in optimization methods to improve power quality generation, control strategies, and failure and fault diagnostics [1]. Power generation's CO_2 emissions have been steadily declining as a result of developed and developing nations' joint commitment to the Kyoto Protocol [1]. For this reason, renewable energy sources (RES) are gradually replacing more conventional power generating methods that rely on coal and other fossil fuels. Among these RES, you may find solar, biofuel, wind, and geothermal power, among others [2]. Photovoltaic (PV) generation systems for solar energy are one of the fastest-growing options because of their many benefits, such as being silent, inexpensive, easy to integrate, low maintenance, pollution-free, dependable, and having an almost infinite supply [3]. Over time, PVS will have the same malfunctions as any other system. Converters, inverters, connectors, protective devices, and PV modules are all prone to a number of faults that might lead to these failures [1, 4]. The primary reasons for these problems may be attributed to external operating circumstances, such as dust or dirt in the modules, converter and/or inverter failures, shadowing, manufacturing incompatibilities, and module aging [1, 4]. Tragic flaws in PVS may be categorized into four primary types: line-to-line faults, arc faults, ground faults, and mismatch faults [1, 4]. These problems, if left unattended, may lead to fires, decreased profits, and even death [1, 4]. In order to know what to do in the event of a mistake, it is essential to be able to identify and categorize them. Understanding the nature of the issue and putting preventative measures in place to ensure it does not happen again is another benefit of PV fault diagnostics. Furthermore, methods to address these shortcomings may also be devised. It also helps in making the PV system last longer and more efficiently. Several approaches to fault diagnosis in photovoltaic (PV) systems using ML and DL have been covered in the review given in [1]. The use of DL models such VGG16, VGG19, 3DCNN, ResNet, AlexNet, GoogleNet, and vision transformers are among the strategies used for PV defect diagnostics

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

[5, 6]. Two- or three-dimensional pictures captured from above or on-site with regular cameras or thermal imaging devices (infrared cameras) are common inputs for these DL designs [1]. In addition to taking pictures with regular or infrared cameras, you may also use sensors to measure various electrical characteristics of the PV system, including DC voltage (V), current (I), and power (P) [1]. The responses, however, are voltage, current, or power measurements plotted against time; these parameters are, nonetheless, given as time-series signals. The information gathered in these formats is, therefore, a 1D signal. It is feasible to utilize a 1D signal with some current DL models, however using 2D pictures as inputs to the DL models stated previously is more simple and easy. Furthermore, a 2D picture retains far more data when converted from a 1D signal of the same duration in terms of the sample. As an example, think about making a 2D picture from a 1D power signal that has 512 sample points. The resultant picture dimensions are 512 pixels by 512 pixels in this instance. By adding additional dimensions, we can get a more accurate picture, which improves the characteristics we can learn from the data. To add insult to injury, gathering RGB or IR image data with UAVs is weather dependent and could miss concealed defects that are often only detectable using electrical means. In order to overcome this restriction, the current study presents several ways for converting 1D data into 2D pictures. These approaches may find use in photovoltaic (PV) problem detection.

II. DESCRIPTION OF METHODS

A. General Framework of PV Fault Diagnosis Using 2D Data

Sensors installed on the PV system may collect several kinds of data, including power, current, and voltage.

Making a replica of the PV system under study is another option for gathering information [1]. After that, we'll model several scenarios that reflect the various kinds of errors. Importing 1D signals into 2D photos while preserving and improving the information within the signals is the first step in using some or all of these data as input to a preferred DL model that accepts 2D images. When converting the signal to a picture. One consideration should be the amount of sample points or window size that provides the relevant information. You should pay close attention to this location. If the information needed to classify the existing defect is not included in the transformed picture. So, the results of the detection or classification process can end up being erroneous. The produced information may be saved in any format that can produce 2D array features, such as.jpg,.png,.tiff, or any similar format. Annotating or labeling the data based on the sorts of flaws that correlate to each picture is the next step after image generation. If you're looking for an open-source labeling tool, there are plenty of options available online. It is also possible to label data while preparing it to be used in a chosen DL model or when transforming it from signal to picture. To summarize, Fig. 1 shows the overall flow diagram of the approach for utilizing the data for PV problem diagnostics.

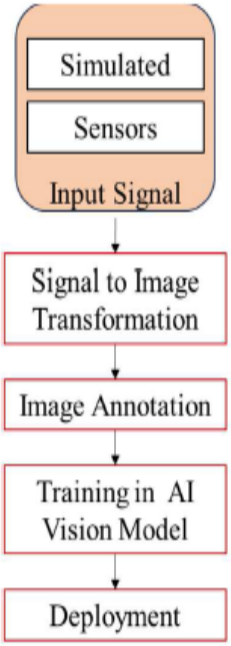


Figure 1. Overall process flow of using 1D image signal as input into DL accepting 2D data.

B. Methods of Signal to Image Transformation

1) Recurrence Plot

One technique to visualize time-series data [7] like V, I, or P is via a recurrence graphic. It is a graphical representation used for time-series data analysis. Furthermore, a recurrence plot is useful for showing patterns that appear more than once. To understand recurrence plots, it's helpful to think of a timeseries _ _ ___,…, __. By using a recurrence plot, we can extract and see the distances between the trajectories of each member in set _. Presented below are the extracted trajectories:

$$t_i = \left(t_i, t_{i+\epsilon}, \ldots, t_{i+(m-1)\varepsilon}\right), \forall_i \in \{1, \ldots, n-(m-1)\varepsilon\} \quad (1)$$

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

where m is the number of trajectories' dimensions and ε is the time delay. Each pixel in the final picture, which is [7], is represented by the pairwise distance between each trajectory in _ and that in __ by the recurrence plot, _,_.

$$R_{i,j} = \theta(\rho - ||t_i - t_j||). \forall_{i,j} \in \{1, \ldots, n - (m-1)\varepsilon\} \quad (2)$$

the Heaviside function, the norm operation, and the threshold are all represented by || ||. Alteration of the Gramian Angular Field The cosine similarity between each pair of data points in a time series is preserved by the Gramian matrix, which is formed by the gramian angular field (GAF) transformation [8]. The next step is to transform this Gramian matrix into a picture, where the brightness of each pixel stands for the degree of similarity between the respective data points. Two kinds of GAF transformations are the Gramian angular difference fields (GADF) and the Gramian angular summation fields (GASF). The mathematical expression of the GAF transformation process is as follows. Suppose you have a time series * _ _+_, +,, +-,…, +._ with N points in the sample. Using Eqs. 3 for GASF and 4 for GADF, we shall rescale or normalize each element in X within the range of /_1, 10 for GASF and /0,10 for GADF, respectively.

$$\tilde{x}_{-1}^{i} = \frac{(x_i - \max(X) + (x_i - \min(X))}{\max(X) - \min(X)} \quad (3)$$

$$\tilde{x}_{0}^{i} = \frac{(x_i - \min(X))}{\max(X) - \min(X)} \quad (4)$$

Now we can use the arccosine function to encode each angle value and time stamp as the radius r, and we can project the rescaled *= into the polar coordinate system.

$$\begin{cases} \emptyset_i = arccosine(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ r = \frac{t_i}{N}, t_i \in \mathbb{N} \end{cases} \quad (5)$$

In the polar coordinate system, the timestamp is denoted by _, the angle by ∅, and N is a constant factor that normalizes the range. To define a trigonometric sum or difference between any two coordinate values, one may use Equation (5) to get the modified polar coordinates. An expression for the matrix expressing the temporal correlation for each point within a specified time range of GASF is given by Equation (6) [8].

$$GASF = \begin{bmatrix} \cos(\emptyset_1 + \emptyset_1) & \cdots & \cos(\emptyset_1 + \emptyset_N) \\ \vdots & \ddots & \vdots \\ \cos(\emptyset_N + \emptyset_N) & \cdots & \cos(\emptyset_N + \emptyset_N) \end{bmatrix}$$

The boundary of each element in Eq. 4 is /0, W0 when GASF (3) is used, but it is in the range of /0, X,0 when GADF is used.
3. The Transformation of the Markov Transition Field The Mean Time Fluctuation (MTF) transformation involves visualizing a time-series by considering the changes in values or states within its components [9]. Think about the time-series data, denoted as * __+_, +,, +-,…, +_, where the quantile bins Q are determined and each xi Î X is given a matching bin qj (where j is a subset of Q). Building an MTF matrix M using the components shown in Eq. 7 [9].

$$M = \begin{bmatrix} m_{11|x_1 \in q_i, x_1 \in q_j} & \cdots & m_{1Q|x_1 \in q_i, x_Q \in q_j} \\ m_{21|x_2 \in q_i, x_1 \in q_j} & \cdots & m_{2Q|x_2 \in q_i, x_Q \in q_j} \\ \vdots & \ddots & \vdots \\ m_{Q1|x_Q \in q_i, x_1 \in q_j} & \cdots & m_{QQ|x_Q \in q_i, x_Q \in q_j} \end{bmatrix} \quad (7)$$

The element mij represents the transition probability of qi → qj in equation (7). Consequently, the typical Markov transition matrix is spread out according to the input signal's temporal locations. After the multi-span transition probabilities of the time series are encoded, each element in (7) may be written as &__|___de, where k is the time interval.
4) Graphics display
A spectrogram is the end result of converting a signal to a picture using the short-time Fourier transform (STFT). In order to determine the STFT of a certain signal, one may use the formula [10].

$$STFT\left(x_{(t)}, f, \tau\right)(\omega, \tau) = \int_{-\infty}^{\infty} x_{(t)} w(t - \tau) e^{-i\omega t} dt \quad (8)$$

The signal +_H~ with respect to frequency f and time shift g is represented by the Short-Time Fourier Transform L_M_ _+_H , f, g_, STFT in Eq. 8, where h is the frequency variable. The complex exponential capturing the frequency component at h is represented by F_kH, and the term j__ _ g corresponds to the window function. The STFT is obtained by dividing the signal +_H into overlapping windows j__ _ g that cover the whole time. The Fourier transform is then applied to each window that follows. The use of j__ _ g to taper the window edges reduces spectral leakage. When shown visually, the resultant STFT becomes the spectrogram. The short-

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

time Fourier transform (STFT) is obtained by introducing sliding windows into the input signal at brief intervals. The spectrogram's frequency and temporal resolution are affected by the window's size and overlap. Reducing the size of the window improves frequency resolution but increases the size of the temporal resolution, and vice versa. The spectrogram equivalent L_f, _ of STFT _M_ _+_H~, f, g_ is represented by Equation 9 [10].

$$S(f,t) = |STFT(x(t), f, \tau)|^2 \quad (9)$$

In each element of STFT, the square of the magnitude of STFT, denoted as | |, will become L_f, _. Methods for Converting a 1D Signal into a 2D Image According to the literature, the most popular forms of signal-to-image conversions are the ones mentioned above. The good news is that there are solutions for the problem of signal-to-image conversion. A few of the approaches are listed below. a) Signal characteristics plotted directly. Implementing this strategy does not need complex mathematics or coding abilities. The basic premise is to use pre-existing functions and libraries in any programming language, such Python or C++, to directly record the signal's behavior over a certain time period or number of sample points. Direct picture conversion of the collected signal's behavior follows [11]. b) Poincare diagrams A Poincare plot is a graphical representation of the connection between two subsequent data or sample points in a time-series signal [12]. By charting the signal's history value against its present value, this is generated in the form of a scatter plot. One possible output of any DL design is a scatter plot. (c) A heatmap A time-series signal may be more complexly shown using a heatmap. The picture is created by assigning a color to each sample point's output value (I, V, or P) and then plotting them on a two-dimensional grid. The value of the quantity at a certain instant is shown by the color of each pixel in the heatmap [13]. d) Reconstructing the Phase Space In order for phase space reconstruction to function, the initial assumption is that each time-and interval-varying variable's value represents the coordinates of a single point in an m-dimensional phase space [14]. A nonlinear system may be graphically represented by explaining a set of data points in the phase space shown above.

III. RESULTS

In order to demonstrate that each approach is capable of successfully converting the signal into a picture. The methods are tested using time-series data from the research by YY. Hong and R. Pula (2022, 2023), which includes a no-fault condition (N) and a shorted string fault (SS). Based on real-world PV systems in operation, the data come from simulated grid-connected systems [1, 5]. Figure 2 shows the results of the time series analysis of PV DC power. Python 3 is the target of the transformation process. We conducted all the changes we covered before on a PC with an IntelR Core™ i7-8550U CPU @ 1.80GHz, with a maximum speed of 1.99 GHz, 32.0 GB of RAM loaded, and the graphics processing unit (GPU) turned on if desired. This study, however, does not make advantage of GPUs.

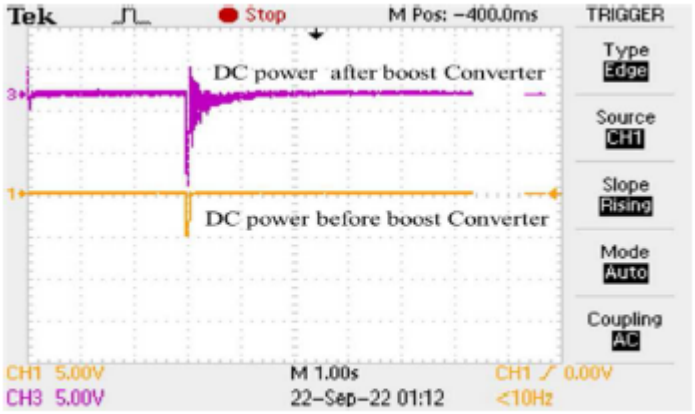


Figure 2. String fault illustration of the data from oscilloscope [1, 5].

Several ways for transforming signals into images are sent into the DC power that follows the boost converter (Figure 2). Starting at 3 seconds and continuing until 3.05 seconds, a photovoltaic (PV) system is simulated to operate in both the no-fault state (N) and a shorted string fault (SS). Three cycles of the typical 60Hz power cycle correspond to this period. The analysis is conducted using a single cycle, which corresponds to 256 sample points. As a result, adding 0.05/3 seconds to the original 3 seconds increases the window size for collecting the answer to the shorted string error. Figures 3a–3D show the results of the no-fault and shorted string fault scenarios as seen using the recurrence plot and GASF, respectively.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

A. Recurrence Plot and Gramian Angular Field Output

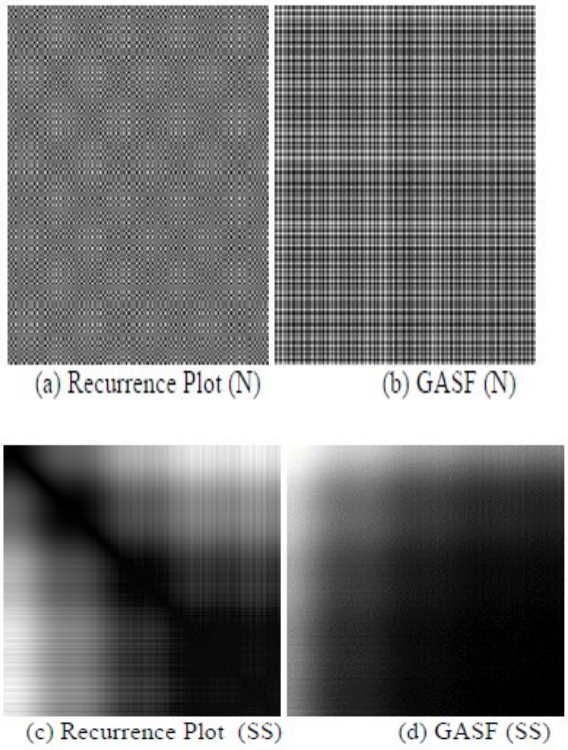


Figure 3. Signal to image of no-fault condition using recurrence plot (a) and GASF (b); shorted string fault using recurrence plot (c) and GASF (d).

Equation 2 yields a.png picture with a size of 256 × 256 pixels showing the shorted string fault of recurrence plot. Values for RGB colors might be included in the output. A 1-channel grayscale picture, however, is sufficient, according to the author's experience. As shown in Figure 3a (N) and Figure 3c (SS), the value of a pixel in the produced picture increases as its color becomes lighter. Per pixel, there is a value between zero and two hundred fifty. Figure 3b (N) and Figure 3d (SS) show the produced images from GASF; these images may be saved as either a grayscale picture with one channel or an RBG image with three channels, much like a recurrence plot.

Section B. Maximum Fluctuation Theory and Spectrogram The results

Figure 4 displays the N and SS conditions for the MTF and spectrogram that were obtained. Figure 4a and 4b show the no-fault circumstances for the MTF and spectrogram, while Figure 4c and 4d show the pictures with the shorted string fault. \

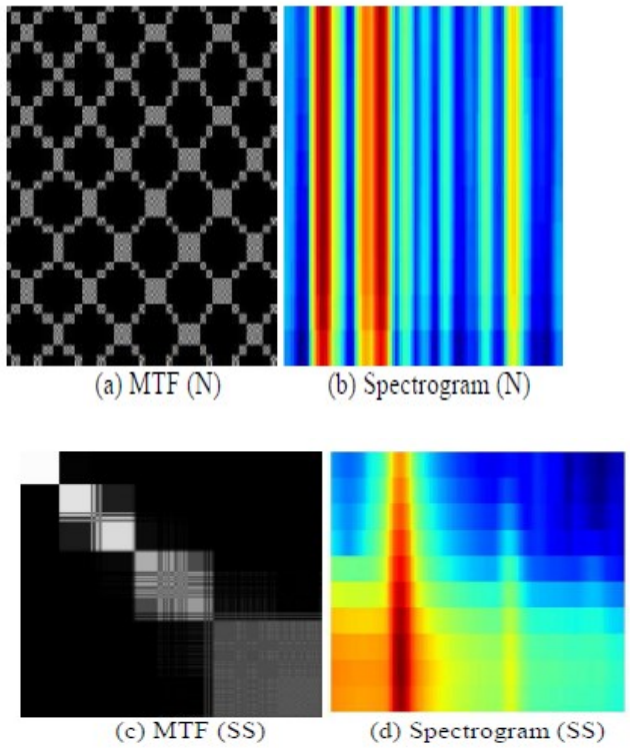
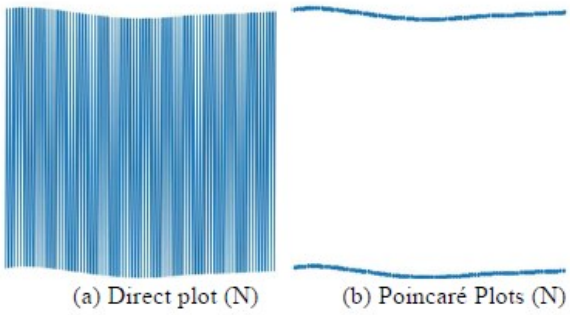


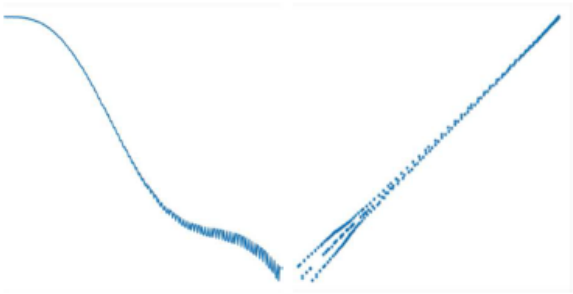
Figure 4. Signal to image of no-fault condition using MTF (a) and spectrogram (b); shorted string fault using MTF (c) and spectrogram (d).

Figure 4a (N) and Figure 4c (SS) show that the pixel-value and appearance characteristics of the MTF-generated images are similar to those of the recurrence plot and GASF. Figure 5b (N) and Figure 5d (SS) show that the spectrogram-generated images have a 3-channel RGB color format, where darker reds indicate higher pixel values and lighter colors lower pixel values, creating a visual representation where the intensity of the color is directly proportional to the pixel value.

C. Direct Plot and Poincaré Plots Output



ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

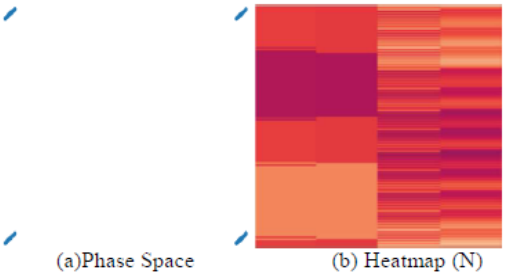


(c) Direct plot (d) Poincaré Plots

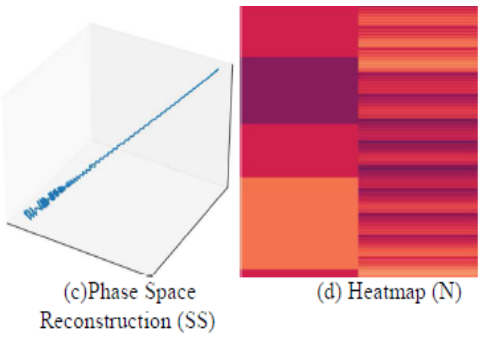
Figure 5. Signal to image of shorted string fault using (a) Direct plot and (b) Poincaré Plots

Figures 5a and 5c show the produced images, which are the real reaction of the 1D signal under no-fault and shorted string fault conditions, respectively, using a direct plot at the moment of the fault that covers 256 sample points. In Figures 5b and 5d, you can see poincare plots that show the results of a series of shorted faults. It is quite easy to convert a 1D signal to a 2D picture by simply charting and preserving the size of the recorded window. The straight plot approach is, without a doubt, the quickest to do. You can see that the RGB format is used for the plotted picture. However, it is not necessary in this sort of transformation as the color of the generated picture is not a large element in distinguishing the characteristic of a supplied signals. Therefore 1-channel grayscale picture is adequate to this type of modification. The pictures that are produced using poincare plots appear in Figure 5b and Figure 5d as scattered plots, but with a different representation. Just like in Figures 5a and 5c, the color does not play a significant role in this transformation either.

D. Phase Space Reconstruction and Heatmap



(a)Phase Space Reconstruction (N) (b) Heatmap (N)



(c)Phase Space Reconstruction (SS) (d) Heatmap (N)

Figure 6. Signal to image of shorted string fault using (a) Phase Space Reconstruction and (b) Heatmap

Figure 6c shows the outcome of a 3D cube comprising the sinusoidal features of a shorted string fault signal, which is used to project the picture of phase space reconstruction. Fig. 6c shows the final product, a three-channel RGB picture that includes the color information from the 3D-projected cube's backdrop. If a more complicated defect, such an arc fault, is present in a PV system. It is possible that this kind of change may be rather helpful. The heatmap produces a three-channel RGB picture, as seen in Figure 6b. In the event when every pixel's red, green, and blue channel values are identical. The characteristics of the final image can be captured with just one channel image. Unfortunately, in both the no-fault (Fig. 6b) and shorted string (Fig. 6d) output images, there is an issue. Notable visual traits are present in the final photos. When comparing the no-fault condition depicted in Fig. 6a using phase-space reconstruction to the shorted string fault, one can notice a distinct feature. The sample points are situated in the four corners of the image, as can be seen. Depending on how complicated the faults are, different transformations can be used for PV fault diagnosis. Mel-Frequency Cepstral Coefficients (MFCC) and other methods primarily used in biological data can also be investigated for potential use in other fields, like power system analysis. There are some limitations to the work that has been presented. A significant drawback is that the 1D data is not preprocessed in any way, prior to the 2D transformation. This includes neither filtering or removing noise. Hence, signal noise will be carried over into the processed 2D pictures and might affect the classification step that follows. Nevertheless, it should be mentioned that noise reduction may be handled before to the transformation step. You can use filters and other traditional noise removal

techniques, or you can use deep learning's autoencoder-decoder models. The results that were shown also only considered the most basic parameters for the methods that were introduced. This means that the study did not investigate all possible configurations that may have improved the results. However, it should be mentioned that this limitation is not impossible to overcome, since there are already Python libraries that offer functionalities that are similar to the ones presented in this work. This makes it easy to conduct additional configuration optimization and exploration. Not to mention that no-fault and shorted string fault were the only kinds of situations that were considered in this study. While these conditions were the primary focus, there exist various other fault conditions that could impact the performance and generalizability of the presented methods. Future research may consider expanding the scope to encompass a broader range of fault conditions for a more comprehensive evaluation.

IV. CONCLUSION

This research presents a number of new techniques for converting 1D data into 2D pictures. After collecting and experimenting with the resultant photographs from each change, they were finally shown. With an emphasis on the shorted string fault in a grid-connected PV system model developed from an actual installation, each transformation was tested using real data from both fault and no-fault scenarios. Each transformation has the option to produce a 1-channel grayscale or 3-channel RGB color picture as its output. When applied to certain machine learning (ML) or deep learning (DL) architectures, the output picture size is also customizable to fit personal preferences and system requirements. Several ML models use the produced pictures of no-fault and shorted string faults, taking advantage of their unique peculiarities.

But this uniqueness may not always be the case; prior research has shown that when 1D signals are translated into 2D pictures, certain defects seem quite similar [1, 5]. However, DL models have shown to be useful in properly diagnosing fault types from photos, even in cases when the visual features are almost similar.

V. FUTURE WORKS

Carrying out tests utilizing data collected from PV system sensors and simulations is an important part of the current study. Several deep learning (DL) techniques that can take 2D pictures as input will be compared in this study. We will combine the electric-based method with more advanced DL techniques to test their efficiency in signal-to-image transformation. Future studies will also investigate the possibility of incorporating other state-of-the-art technologies, such as quantum computing and edge computing. Given the state of the art in computing, the new area of hybrid quantum machine learning offers an interesting prospect for research. This field combines advanced DL models with quantum computing technology.

REFERENCES

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

[2] A. Rahman, O. Farrok, and M. M. Haque, "Environmental impact of renewable energy source based electrical power plants: Solar, wind hydroelectric, biomass, geothermal, Tidal, ocean, and Osmotic, " Renewable and Sustainable Energy Reviews, vol. 161, p. 112279, 2022.

[3] M. Victoria et al., "Solar photovoltaics is ready to power a sustainable future," Joule, vol. 5, no. 5, pp. 1041–1056, 2021.

[4] M. Waqar Akram, G. Li, Y. Jin, and X. Chen, "Failures of photovoltaic modules and their detection: A Review," Applied Energy, vol. 313, p. 118822, 2022.

[5] Y.-Y. Hong and R. A. Pula, "Detection and classification of faults in photovoltaic arrays using a 3D convolutional neural network," Energy, vol. 246, p. 123391, 2022.

[6] Z. Zhang, Z. Gong, and Q. Hong, "A survey on: Application of transformer in computer vision," The Proceedings of the 8th International Conference on Intelligent Systems and Image Processing 2021, 2021.

[7] B. Goswami, "A brief introduction to nonlinear time series analysis and recurrence plots," Vibration, vol. 2, no. 4, pp. 332–368, 2019.

[8] Y.-Y. Hong and R. A. Pula, "Detection and classification of faults in photovoltaic arrays using a 3D convolutional neural network," Energy, vol. 246, p. 123391, 2022.

[9] Y.-Y. Hong and R. A. Pula, "Diagnosis of PV faults using digital twin and convolutional mixer

with Lora Notification System," Energy Reports, vol. 9, pp. 1963–1976, 2023.

[10] Mathuranathan, "Spectrogram analysis using Python," GaussianWaves, https://www.gaussianwaves.com/2022/03/spectrogram-analysisusingpython/#:~: text=

[11] R. Horvath , "Plot with Pandas: Python data visualization for beginners," Real Python, https://realpython.com/pandas-plot-python/(accessed Nov. 19, 2023).

[12] G. D'Addio et al., "Extracting features from Poincare plots to distinguish congestive heart failure patients according to NYHA classes," Bioengineering, vol. 8, no. 10, p. 138, 2021.

[13] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Home: Heatmap output for future motion estimation," 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021.

[14] G. Li, Y. Li, and F. Roozitalab, "Midterm load forecasting: A multistep approach based on phase space reconstruction and Support Vector Machine," IEEE Systems Journal, vol. 14, no. 4, pp. 4967–4977, 2020. doi:10.1109/jsyst.2019.2962971