

# Machine Learning Algorithm for Assessing Photovoltaic Panels Partial Shading Losses based on Inverter Data

**Tiago Francisco Rodrigues Pires**

Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 PORTO, Portugal ([up201905970@up.pt](mailto:up201905970@up.pt))  
ORCID [0009-0001-8802-7552](https://orcid.org/0009-0001-8802-7552)




**Armando Luís Sousa Araújo**

Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 PORTO, Portugal ([asa@fe.up.pt](mailto:asa@fe.up.pt)) ORCID  
[0000-0003-0317-7111](https://orcid.org/0000-0003-0317-7111)

## Author Keywords

Partial Shading, Solar Panels, Photovoltaic Systems, Machine Learning, K-Means Clustering, LSTM Neural Networks, Renewable Energy.

Type: Research Article

 Open Access  
 Peer Reviewed  
 CC BY

## Abstract

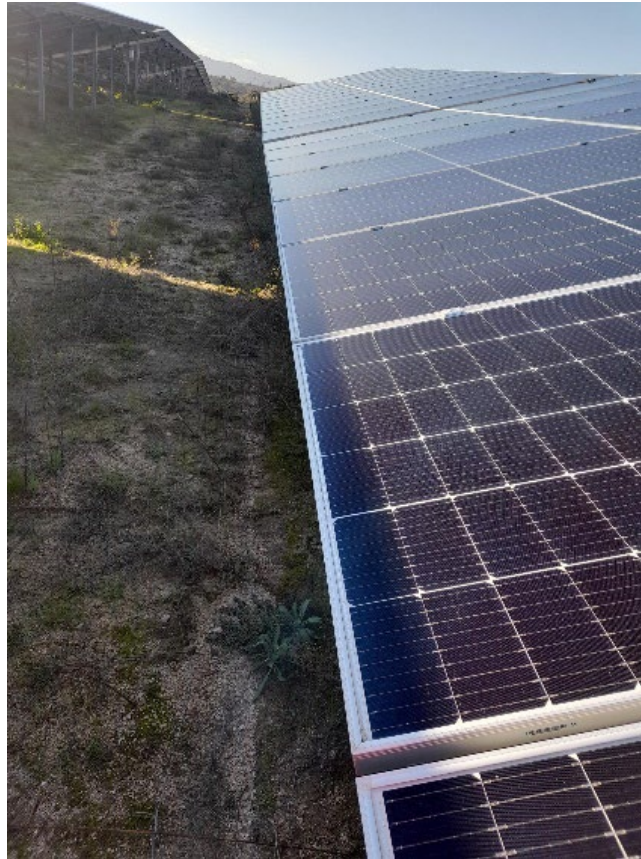
Partial shading is one type of fault where photovoltaic panels cast shadows between each other, reducing their production and hastening their ageing.

In this paper, we document and describe two distinct Machine Learning models that aim to identify and assess the impact of partial shading in a real case study. These algorithms recognise similarities and patterns using expected and measured power data. The predicted power is calculated using the measured panel irradiance, current, and voltage using a photovoltaic panel electric circuit model. The first Machine Learning model employs K-means clustering to analyse the differences between expected and measured power, grouping data based on these deviations. The second Machine Learning model leverages the outputs of the K-means model as labels for a Long Short-Term Memory neural network, which classifies periods of partial shading. Experimental data from both models are presented, with the K-means model achieving a closer approximation to the reference values. However, the Long Short-Term Memory model demonstrated flexibility and scalability without requiring prior dataset knowledge from the end user.

## 1. Introduction

The need for clean and renewable energy has drawn attention, mainly due to the climate crisis and economic interest in solar energy, thus attracting, in recent decades, the scientific community's attention to this subject ([Polman et al. 2016](#)). Therefore, over the last few decades, monitoring tools have been designed to guarantee standardised indicators for monitoring and maintaining photovoltaic energy (PV) generation systems. These tools were designed to be cheap and automated and can help avoid the average 20% energy losses that photovoltaic systems experience ([Ortega et al. 2017](#)), ([Li et al. 2021](#)). Given the large volumes of data that PV monitoring systems can generate, several Machine Learning (ML) based algorithms for dealing with these systems can be found in the literature ([Zsiborács et al. 2021](#)), ([De Benedetti et al. 2018](#)), ([Hopwood et al. 2020](#)), ([Ibrahim et al. 2022](#)). These automated applications are of great interest to help operators of these systems in the processing of big data, as they can allow the detection of anomalies in PV power plants that can lead to poor performance or even cause damage to equipment.

For example, at the plant under study in the south of Portugal, there was a 20% loss in energy production in a single day due to partial shading of the photovoltaic modules. Such shading can be seen in Figure 1.



**Figure 1:** PV module shaded by other modules

According to (Ortega et al. 2017), several factors, such as dust accumulation or cracked panels, can lead to failures in PV modules, yielding a lower revenue due to loss of productivity. The three performance indicators for a PV system presented here are final yield ( $Y_F$ ), reference yield ( $Y_R$ ) and performance ratio (PR). The PR is the ratio between  $Y_F$  and  $Y_R$ .  $Y_F$  is calculated by dividing the energy delivered to the load by the rated output power.  $Y_R$  is the energy available to the load during the defined period. During the engineering phase of a PV power plant, when choosing the array row pitch or distance between rows of modules, a trade-off must be considered between the produced energy per land area ( $kWh/m^2$ ) and the production per rated kilowatt ( $kWh/kW$ ). Increasing the produced energy per land area ( $kWh/m^2$ ) might lower the production per rated kilowatt ( $kWh/kW$ ), due to losing energy for partial shading. Reducing the production per rated kilowatt will lower the PR (Deline et al. 2014). When clients commission a PV power plant, they usually demand more energy production per land area, overlooking production per rated kilowatt. It is also common for clients and Engineering, Procurement and Construction (EPC) companies to have established PR values in their maintenance contracts. The EPC company may face monetary penalties when power plants perform below those PR values. However, some circumstances can exempt the EPC company from such penalties. The problem with partial shading losses is that they are hard to explain to clients, and the issue arises due to client demands during the PV field's conception phase. When clients face significant losses and a lower PR, they demand explanations. One way to explain shading losses is to provide an on-site picture, such as **Figure 1**, and present estimates of the loss values. These explanations are often not well received

and are met with scepticism. Another source of additional expenses is maintenance routines issued to search for correct these losses. Given that shading losses exist by design, these routines exist only per obligation and are, therefore, wasteful of resources and capital.

To identify and evaluate the impact of partial shading in a real case study, this work explores two different methodologies for a photovoltaic plant in southern Portugal. These two approaches can obtain and process factory source data and identify periods of partial panel shading in the monthly production of a string. Furthermore, they can evaluate the losses associated with these periods. The first model is based on a K-Means algorithm, proposed independently by several different researchers (Steinhaus 1957), (MacQueen 1967) and (Lloyd 1982). The second model is based on Long Short-Term Memory Neural Networks (LSTM), first proposed by (Hochreiter and Schmidhuber 1997). These models will be discussed in more detail in **Chapter 2**.

## 2. State of the Art

The potential for ML applications, related to the development of tools capable of contributing to the progress of advanced systems associated with renewable energy, is virtually unlimited. Therefore, numerous players, in this sector, are exploring inventive solutions to increase the efficiency of their systems through fault identification (Li et al. 2021). According to (Tina et al. 2021), common issues in PV systems can be automatically identified through ML approaches, primarily employing three distinct methodologies. Firstly, there is the analysis of string/panel current and/or voltage, including measurements at the inverter, exploiting external factors such as environmental variables. This is often achieved through established ML methods like Artificial Neural Networks (ANN), Fuzzy Logic (FL), Decision Tree (DT), and Random Forest (RF) algorithms. The second approach involves image analysis, predominantly applying infrared (IFR) images captured by Unmanned Aerial Vehicles (UAVs). Deep Learning (DL), particularly with various Convolutional Neural Networks (CNNs) types, has proven effective in this context. The third methodology revolves around clustering-based techniques that detect anomalies using unlabelled data. Key methods encompass k-nearest Neighbour (kNN), one-class Support Vector Machine (1-SVM), as well as more recent algorithms such as Isolation Forest (IS) or Local Outlier Factor (LOF). Within the literature, the predominant focus is centred around four categories of failures in PV systems: short circuit (SC), open circuit (OC), partial shading (PS), and abnormal ageing. (De Benedetti et al. 2018) propose an algorithm that can predict equipment faults several days or weeks before they occur. To achieve this, an ANN estimates the power output by simulating the behaviour of the panels, plus inverters groups, for the current conditions. The inverters will communicate their output in real-time, which is then compared to the simulation values. If the growing gap between production and simulation is a trend, adequate maintenance procedures are issued. An approach for unsupervised fault detection has been proposed by (Harrou et al. 2019). This approach combines the benefits of model-based techniques with the classification capacity of clustering algorithms, such as one-class SVM. The approach uses the one-diode-based simulation model to describe the nominal behaviour of a PV array by using real environmental measurements as inputs. It then applies one-class SVM to the generated residuals from the simulation model to detect faults. Another approach was presented by (Hopwood et al. 2020) in which three neural network architectures, 1D Convolutional Neural Networks, Single Headed Long Short Term Memory (LSTM) and multi-headed LSTM, have their performance in anomaly detection tested. These algorithms must be able to detect anomalies and classify Current versus Voltage (IV) curves from collected data into baseline/normal, partially soiled and cracked. The authors found that multi-headed LSTM and 1D CNN architectures produced accuracies greater than 99 % on

average, the best of the three. While 1D CNN was marginally more accurate, multi-headed LSTM took half the time to train. Three machine-learning schemes were also tested by (Ibrahim et al. 2022). Auto Encoder LSTM (AE-LSTM), Facebook-Prophet and Isolation Forrest were used to evaluate and distinguish healthy from abnormal PV system behaviour. It was observed that only AE-LSTM correctly identified all the anomalies and the healthy signal. In contrast, the other two models identified the anomalies but labelled positive points as anomalies.

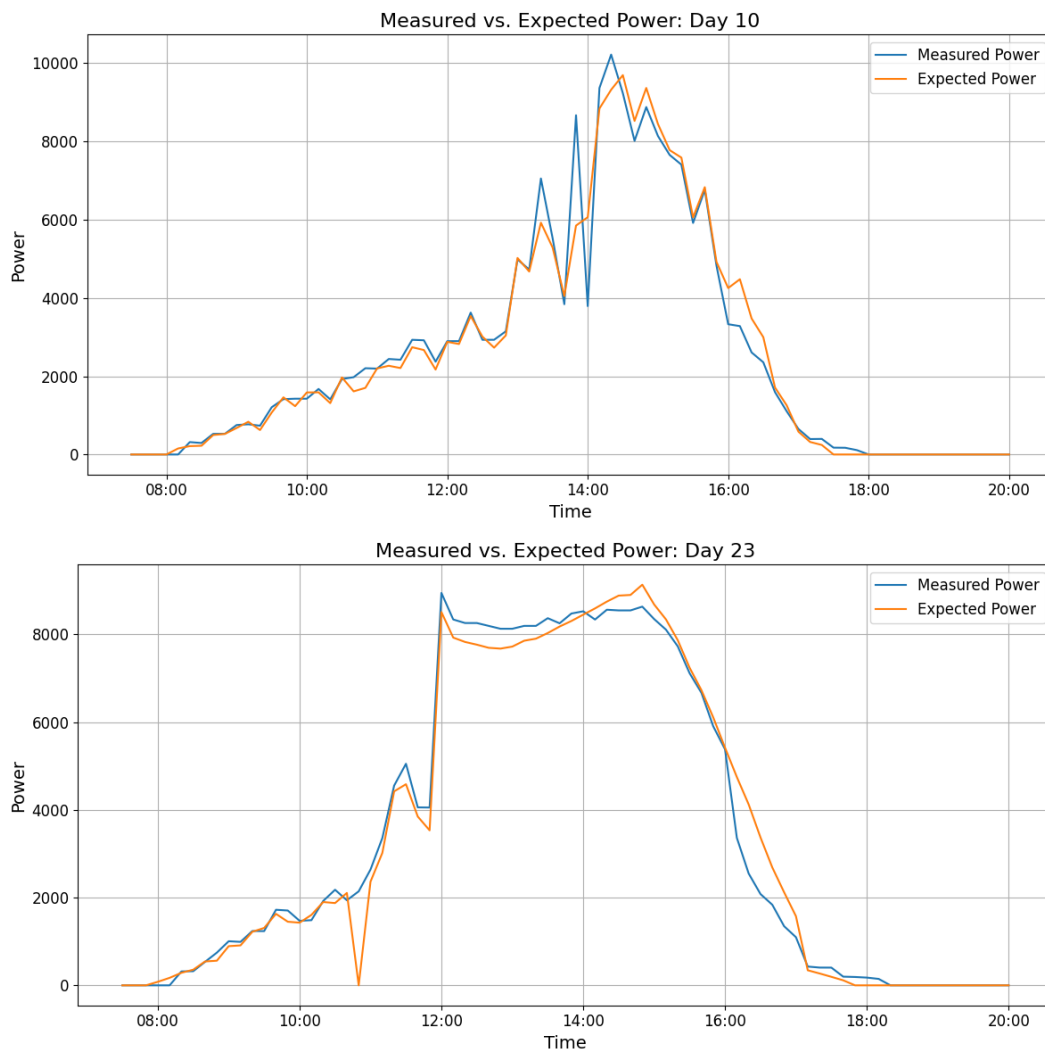
### 3. Methodologies and Implementation

The previously mentioned case study is a state-of-the-art PV power plant in southern Portugal, equipped with DC/AC inverters capable of recording historical data, which is then used for monitoring purposes. These measurements are taken every ten minutes, with each inverter registering current and voltage data. The local irradiance and temperature inside and outside solar cells are registered at the plant's weather station. When developing the models, the first data analysis phase consisted of understanding and processing said data, identifying patterns and translating it into a format that could be later used in ML applications. During this phase, it was necessary to model the system's expected power to calculate the difference between expected and measured power. This difference would then be used as input for the models. This analysis would shape how the models acquire the data, which will be reported in this chapter. The second phase was the conception of the algorithms themselves and the usage of the data processed in phase one. The K-Means-based approach was developed with MatLab (MATLAB 2024). The LSTM-based approach was developed with Python in VS Code (Rossum and Drake 2010) and visualised with MatLab. The algorithms and the challenges met during their development will be discussed in sections 2.1 and 2.2. The real datasets from the case-study plant were used in the data analysis phase. The available data was recorded by two inverters from areas with different slopes. The data was recorded during July of 2023 and January of 2024. When analysing and pre-processing the datasets, it became apparent that there were instances when the measured irradiance would be zero while the inverters still registered power production. Likewise, zero output in the inverters with irradiance being registered also happened. The distance between the inverters and the weather station means clouds could shade them at different intervals. Equipment faults could also cause this phenomenon.

To model the expected power production without partial shading, Equation 1 was employed:

$$P_{exp} = n \frac{P_{max} \cdot I}{I_{STC}} \cdot \frac{100 + Coef_{Voc} \cdot (T - T_{STC})}{100} \quad (1)$$

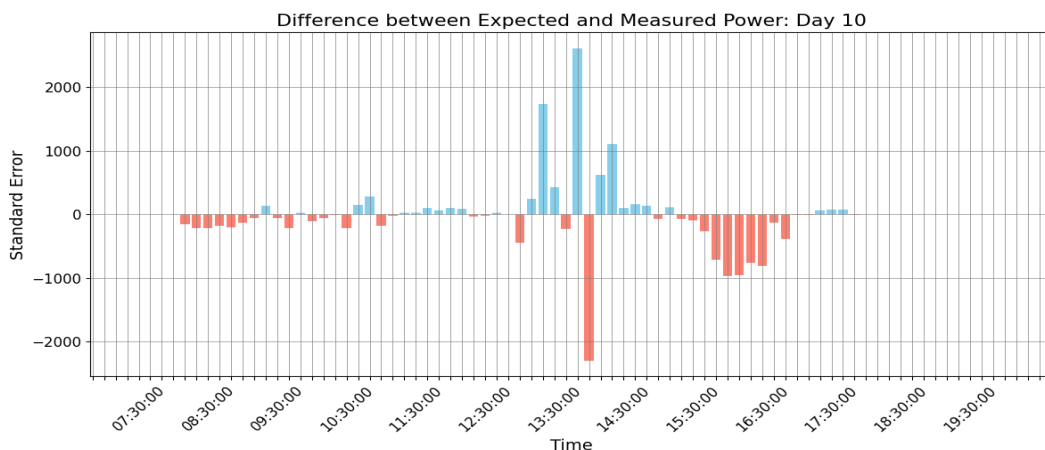
This equation based on an equation found in (Xydis 2013), calculates the expected power output,  $P_{exp}$ , of a PV system by considering the number of panels ( $n$ ), the maximum power output of each panel under standard conditions ( $P_{max}$ ), the actual irradiance ( $I$ ), temperature conditions ( $T$ ), and the temperature coefficient of the open circuit voltage ( $Coef_{Voc}$ ). The adjustments for irradiance and temperature ensure that the expected power output reflects the real operating conditions rather than the Standard Test Conditions (STC). STC typically means an irradiance ( $I_{STC}$ ) of  $1000 \text{ W/m}^2$  and a cell temperature ( $T_{STC}$ ) of  $25^\circ\text{C}$ . The  $P_{max}$ ,  $I_{STC}$ ,  $T_{STC}$  and  $Coef_{Voc}$  values were taken from the solar panel datasheet (Longi). After calculating the expected power, it was possible to compare the results with the measured power, as can be seen, for two different days in Figure 2.

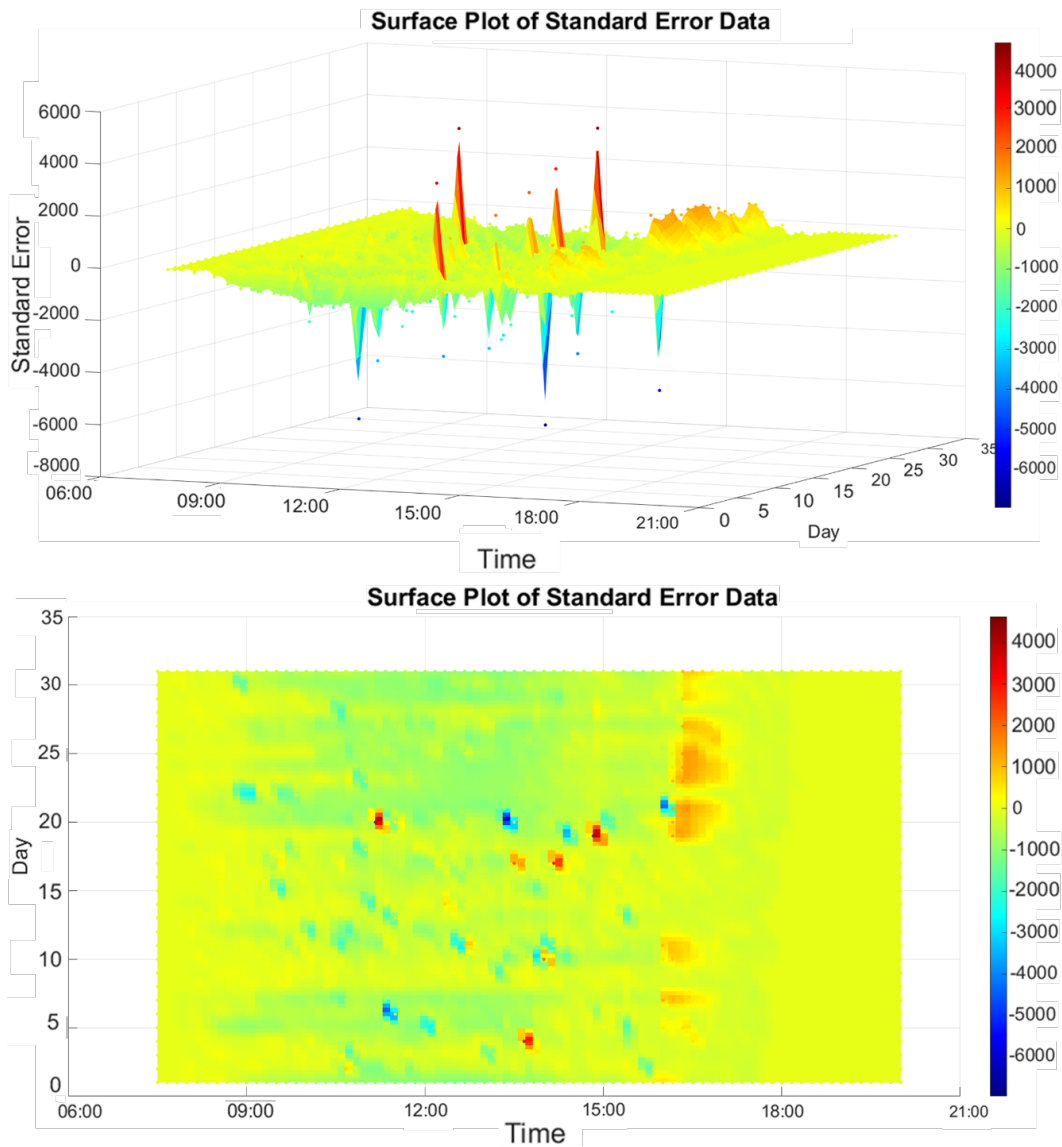


**Figure 2:** Measured vs Expected Power for string 2 on days 10 (top) and 23 (bottom) in January

**Figure 2** shows a noticeable gap between the expected and measured power from approximately 3 to 5 pm over the two days. This phenomenon was recognised for this string in 15 of the 31 days in the month. The other strings exhibited similar shading profiles.

It was then calculated and plotted this difference between the measured and expected power values to convert this data into a convenient format for the models.





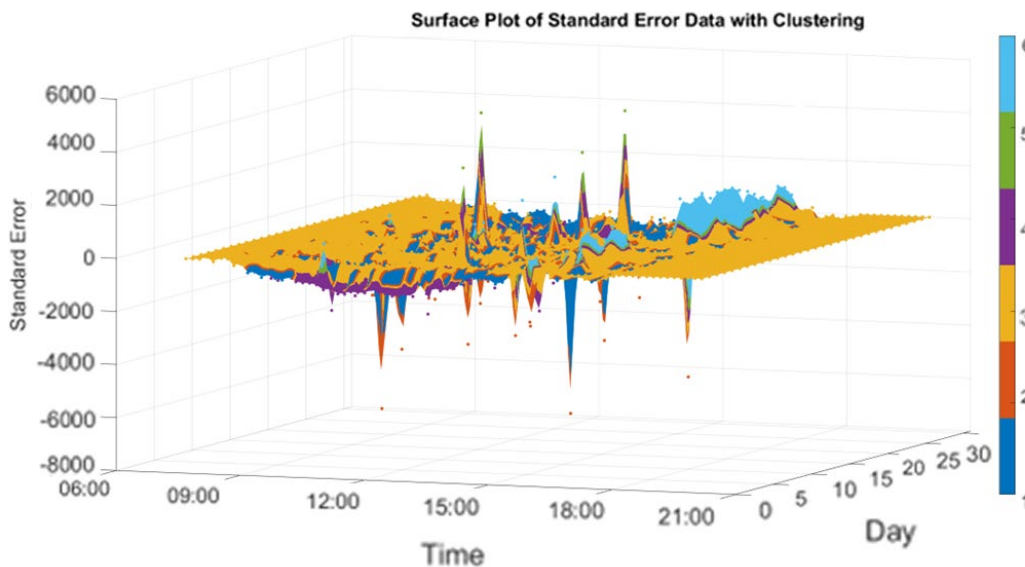
**Figure 3:** Difference between Expected and Measured Power: Day 10, string 2. Blue bars represent higher Measured than Expected Power. Red bars represent higher Expected than Measured Power.

The previously mentioned gap between the expected and measured power on day 10 becomes evident in the graph of **Figure 3**.

#### 4. Methodologies - K-Means

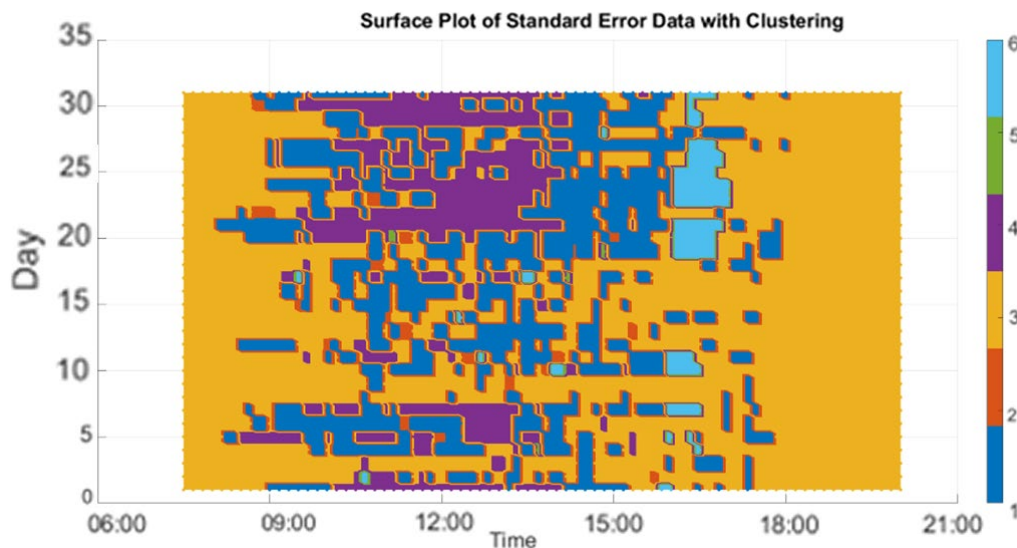
After performing feature engineering on the data, we began developing the algorithms, particularly by computing the difference between expected and actual power.

A first tentative version of the K-Means approach was made with Python in VS-Code. However, the time-series nature of the data proved to be challenging. The algorithm performed clustering based on the proximity of the timestamps as a sequence, prioritising the standard error value and the error's periodicity to a lesser extent. To try to solve this problem, the data was converted into a matrix and plotted in a 3-D mesh with MatLab, as seen in the example for string 15 in **Figure 4**.



**Figure 4:** Difference between expected and measured power in the month of January for string 15 (TOP: 3D Perspective View of Surface Plot, BOTTOM: Top-down View of 3D Surface Plot)

Within the depicted diagram **Figure 4**, it is observable that the losses occurring between 15:00 and 17:00 are consistent, suggesting that these shading losses can be attributed to the design of the plant, as anticipated. The values along the Z-axis have been inverted for improved clarity and interpretation of the outcomes. The K-Means algorithm was applied to the string. The results can be seen in **Figure 5**.



**Figure 5:** K-means clustering applied to the loss profile of string 15 in January (TOP: 3D Perspective View of Surface Plot, BOTTOM: Top-down View of 3D Surface Plot)

Each color in the graph indicates a different cluster, with light blue, cluster 6, belonging to partially shaded timestamps. **Table 1** shows each cluster's total power in *kW* in the column Total Power, and the number of timestamps belonging to each cluster in the column Count. The asterisk (\*) indicates that the values are contained in the time interval between 15:00 and 18:00. According to this model, this string lost in January, was 46.6 *kW*.

Cluster	Total Power in kW	Timestamp Count	Total Power *	Timestamp Count *
1	-234.2	560	-49.9	135
2	-71.2	24	-12.2	4
3	-68.0	1390	-33.3	387
4	-286.8	313	-5.0	5
5	15.6	4	0.0	0
6	53.8	65	46.6	58

**Table 1:** Total power and timestamp count per cluster for string 15 in January.

This model could group the timestamps, according to their similarity, by their power difference value, time of day and day of the month. Despite the model correctly identifying partially shaded periods, there were also false positives. These can be easily found in **Figure 5**, as there are light blue clusters before 15:00. Since no labelled data is available for this case, verifying false positives after 15:00 is impossible. Developing this model allowed us to confirm one of the most well-known disadvantages of K-means clustering: inputting the total number of clusters before the clustering occurs, which requires trial and error determination of this number and previous knowledge of the data set.

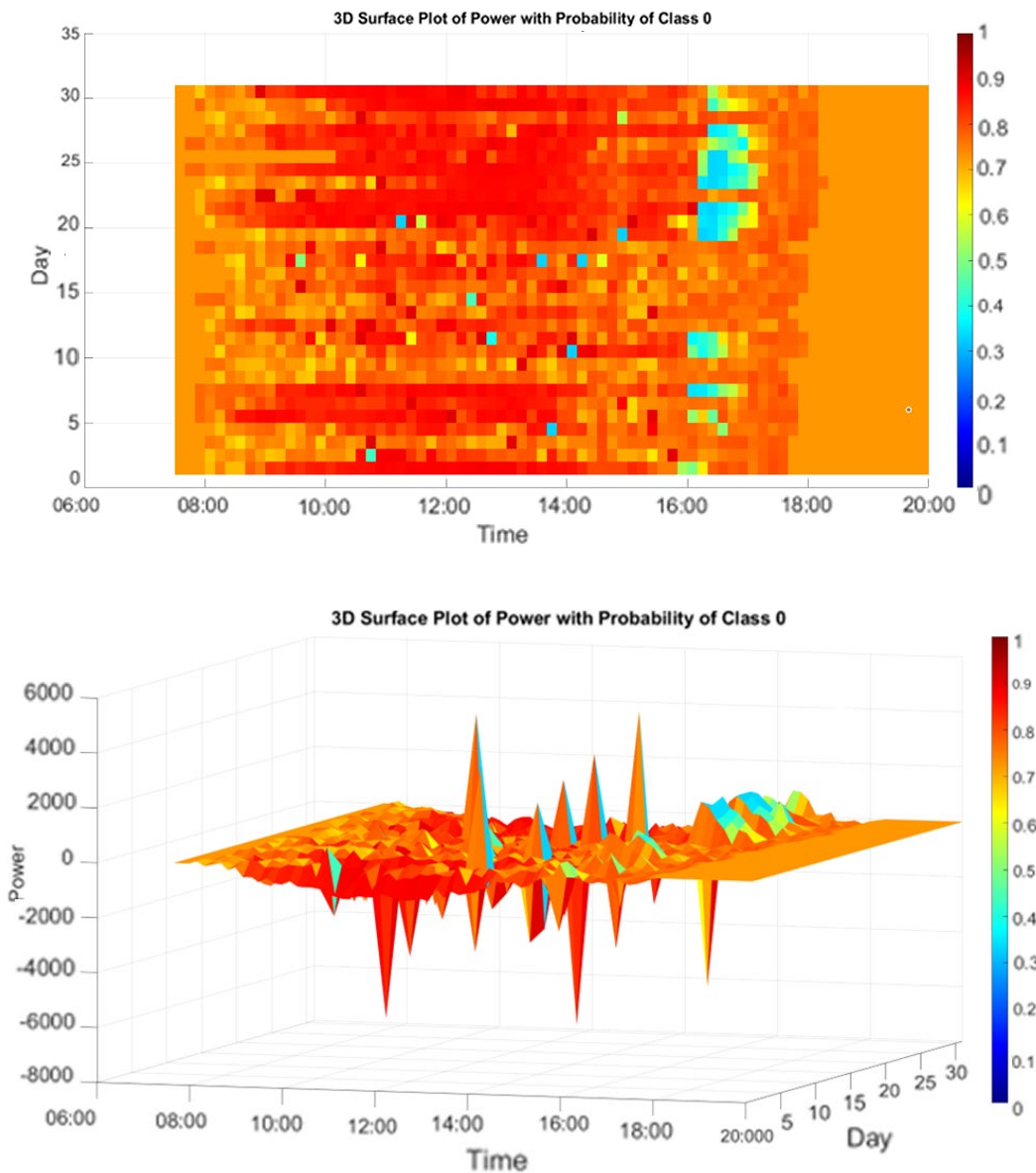
#### 4.1. Methodologies - LSTM

From the previously conducted literature research, LSTM neural networks consistently had the best results out of several ML models. Using the PyTorch framework (Paszke et al. 2019), an LSTM NN model was developed and tested in Python for the second part of our methodology.

Unlike the previously implemented unsupervised model, the biggest challenge faced when developing this supervised ML algorithm was the need for labelled data. To solve this problem, the first developed model was used to generate labels. In the results, relating to string 7, the labels for all timestamps were converted to 0, except for the ones between 15:00 and 18:00. All labels different from 6 were also converted to 0 to have only two classes and those with values equal to 6 were changed to 1. Class 0 meant the timestamp was not part of a partially shaded period, and class 1 meant the time stamp was included in a shaded period. Neural Networks do not favor unbalanced data sets when training (unbalanced data sets are those where some classes have disproportionately higher expression than the rest). Since the partial shading affected periods are contained between 15:00 and 18:00, the data from strings 7, 2 and 8 were filtered accordingly. The matrix relative to a string's monthly production had dimensions 31 by 76. To achieve this shape, the 25 by 31 matrices that resulted from filtering were concatenated and padded. It was then necessary to define the LSTM model architecture. According to (De Benedetti et al. 2018), it is general knowledge that using a more comprehensive hidden layer instead of several hidden layers diminishes the complexity of the model and processing times. The example in this work was used as a first basis to determine the batch size, number of neurons, and learning rate. These values were then tuned through a trial-and-error process, validated by comparisons with the K-means model, and visual analysis of the resulting graphs. In graph analysis, the following features were favourably weighted towards the architecture: clearly defined shaded periods and correct attribution of class probability to the timestamps to shaded periods, with a higher probability of class 0 for non-shaded periods and a lower probability of class 0 for shaded periods. Class 0 was chosen for data visualisation out of convenience. When defining the batch size, there was a trade-off between variety in results and consistency, with smaller batch sizes supporting the first and larger sizes favouring the latter. Iterations were made with the batch size ranging from 1 to



64. The size that better aligned with the required features and offered the best results was 32. The exact process was followed to determine the number of neurons and learning rate. The neurons were incremented in steps of 1 from 1 to 20. The optimal solution that did not lead to under or overfitting while clearly defining shaded periods was 4 neurons. The learning rate was incremented from 0.001 to 0.01 in steps of 0.001, with the final value for the model being 0.008. Given the random initialisation of the weights and the stochastic nature of training in neural networks, this architecture produced different models for each training iteration for string 15's data set. **Figure 6** and **Table 2** show the resulting graphs and recorded values for the model that better satisfied the previously established guidelines. Once again, the asterisk (\*) indicates that the values are contained to the time interval between 15:00 and 18:00.



**Figure 5:** LSTM classifier applied to the loss profile of string 15 in January (TOP: 3D Perspective View of Surface Plot, BOTTOM: Top-down View of 3D Surface Plot)

Probability Interval	Sum of Standard Error in kW	Timestamp Count	Sum of Standard Error in kW *	Timestamp Count*
0.0-0.1	0.0	0	0.0	0
0.1-0.2	0.0	0	0.0	0
0.2-0.3	0.0	0	0.0	0
0.3-0.4	49.3	33	28.7	26
0.4-0.5	13.0	20	11.7	18
0.5-0.6	9.2	22	7.6	18
0.6-0.7	12.6	104	4.5	31
0.7-0.8	-116.8	1411	-50.5	374
0.8-0.9	-507.1	746	-46.8	88
0.9-1.0	-509.9	20	-6.7	3

**Table 2:** Total power per probability of Class 0 interval for string 15 in January

As with the previous models, all the timestamps classified as partially shaded, in this case, with a probability of class 0 below 0.5, before 15:00, are guaranteed to be false positives. According to this model, the assessed partial shading losses between 15:00 and 18:00, and with a probability of class 0 of 0.5, or below, were 40.4W.

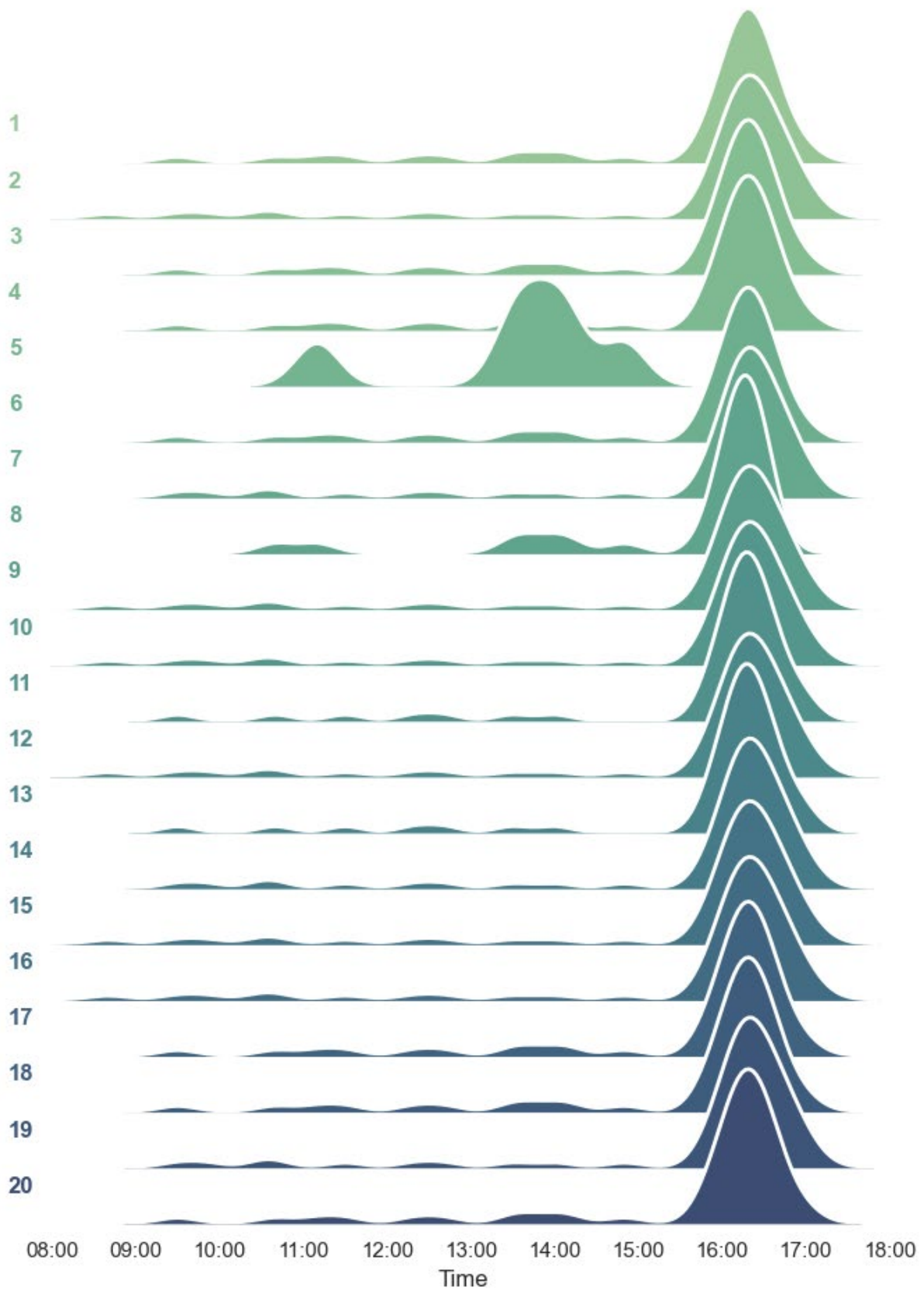
## 5. Validation and Results

### 5.1. K-Means-based Models Results

Despite the K-means algorithm not being stochastic, initialising the K cluster centres is random, introducing stochasticity into the process. As such, the results of 20 runs of this program were recorded and studied to understand its behavior. This program randomly attributes a number to the clusters, which means, for example, that what is shown as cluster 6, in **Figure 5**, could have any number between 1 and 6 in a different run. To align the cluster numbers between the 20 runs, it was necessary to apply the Kuhn–Munkres algorithm, also known as the Hungarian method (Kuhn 2010). The mentioned method is a combinatorial optimisation algorithm that solves assignment problems.

It provides a way to consistently label clusters across multiple runs of the K-means algorithm, enabling meaningful comparisons and evaluations of clustering stability and performance.

The distribution over time of timestamps, with label 1, can be seen in **Figure 7**:



**Figure 6:** Cluster 1 distribution over time, by model

Analysing **Figure 7** allowed us to easily identify that run 5 had 100 % of the label 1 timestamp before 15:00. So, this model could classify timestamps, as partially shaded, in the correct

period, in 19 out of 20 runs. An average of 19.7 % of the classified timestamps were false positives. Developing this model, as already presented, allowed us to confirm that, for K-means clustering, it is necessary to input the total number of clusters before the clustering occurs. So, a priori trial and error-based estimation is needed.

### **5.2. LSTM-based Models Results**

Previously, it was mentioned that the training process is stochastic. This architecture was initialised and trained 20 times with the combined dataset from strings 7, 2 and 8, and each model was then used to process the data from string 15 in January. **Figure 8** illustrates the distribution of 558 timestamps between 15:00 and 18:00, with each model's probability values represented by density plots. The density plots provide a representation of the distribution of probability values. From this figure, it is possible to infer that 3 models out of 20 could classify timestamps as partially shaded. Model 1, seen in **Figure 6**, classified 44 timestamps (40.4 kW) as partially shaded, whereas Model 2, also seen in **Figure 6**, classified 11 timestamps (14.1 kW) as partially shaded and Model 9, again in **Figure 6**, classified 8 timestamps as partially shaded (10.6 kW).

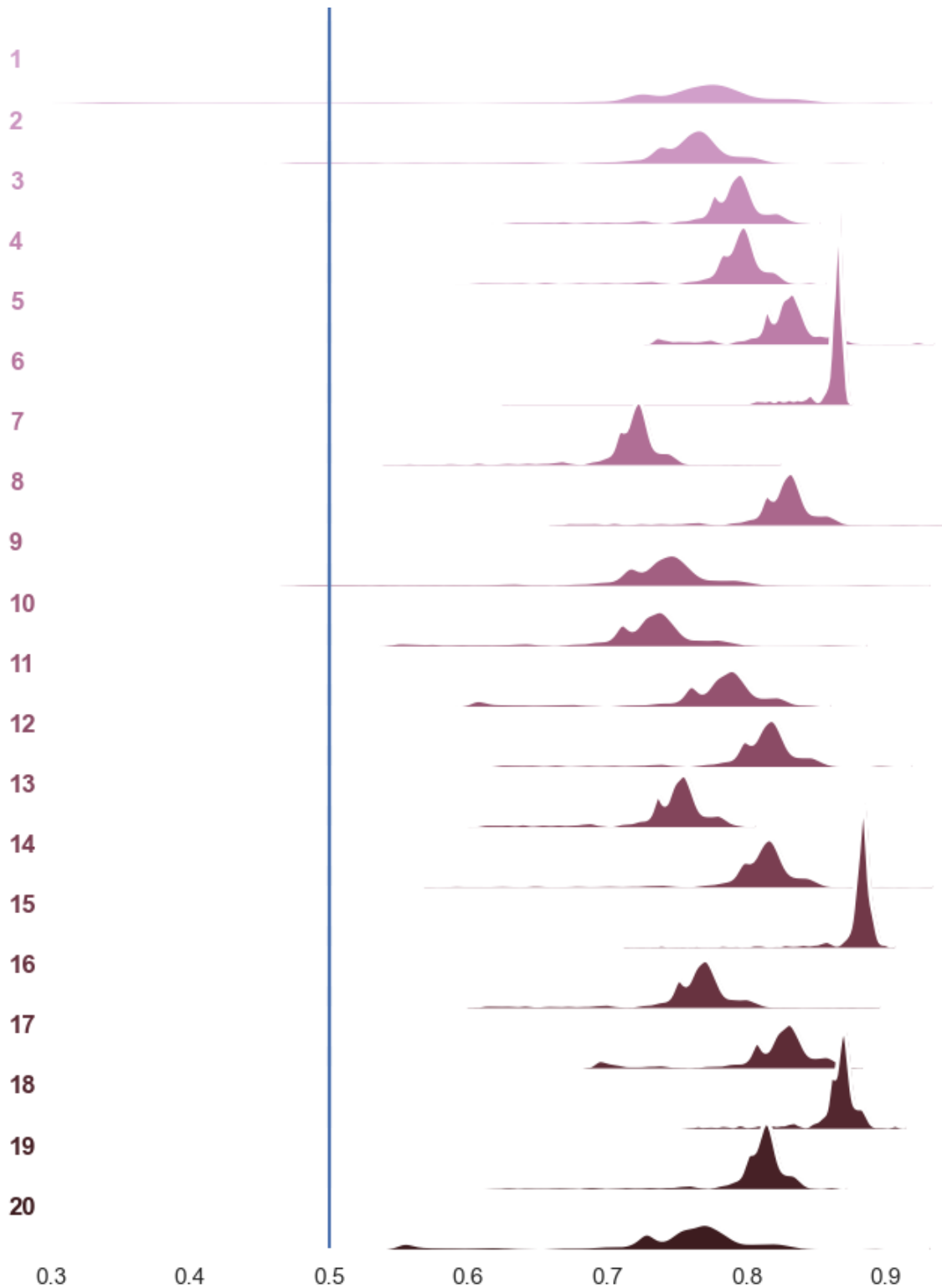


Figure 7: Probability distribution, by model

### 5.3. Model Results Comparison

On January 15, 2024, the EPC company conducted an experiment between 14:00 and 16:00 to assess the impact of shadow growth on panel productivity. The increase in shadow presence over the panels during this time frame was noted to have ranged from 1 cm to 10 cm on average. Furthermore, the study revealed a 20 % decrease in expected production during this period.

With this reference and having calculated the difference between measured and expected power for that period, the results for the models can be seen in **Table 3**.

	Total Expected Power in kW	Calculated Losses in kW	Losses as percentage	False positives
Reference	N/A	N/A	20,0%	N/A
Calculated	301.6	53.1	17.6%	N/A
K-Means	301.6	46.6	15.4%	7
LSTMModel1	301.6	40.4	13.4%	9
LSTMModel2	301.6	14.1	4.7%	6
LSTMModel9	301.6	10.6	3.5%	9

**Table 3:** Results for the studied models. False positives are only counted before 15:00

The K-Means-based model was the closest to the reference values from these results while having fewer false positives than the best LSTM Model. It must be noted that, when weighing these models, the K-Means-based approach holds all the disadvantages mentioned in **Section 2.1**. The LSTM models, on the other hand, do not require previous knowledge of the dataset and do not demand choosing a cluster number.

## 6. Conclusions

The main objective of this work was to develop and study ML models capable of assessing losses due to partial shading. Two distinct approaches were shown: an unsupervised ML approach, based on K-Means clustering, and a supervised ML approach, aided by unsupervised ML, based on LSTM Neural Networks. Assessing and attributing these losses to this fault can be key to future penalty negotiations in O&M. The models described in this paper could draw similarities and differences between the expected and measured power. This diagnostic tool might also be valuable for informing all parts involved in the PV power plant conception and engineering phases. The models were shown to have validation errors. The biggest challenge for the LSTM models was the lack of labelled data for this case. With more data for training purposes, better future results can be achieved, and the validation means can be improved. Future work might include finer tuning of hyperparameters through the development of parameter optimisation models to improve the current capabilities of this system (De Benedetti et al. 2018). It would also be interesting to try new and different neural network architectures, such as using an Autoencoder in conjunction with the existing LSTM. Additional features, such as the estimation of how the registered partial shading periods affect the ageing of the solar cells and calculating the estimated losses in revenue and equipment substitution, would also be of interest from a commercial application standpoint.

## References

- De Benedetti, Massimiliano, Fabio Leonardi, Fabrizio Messina, Corrado Santoro, and Athanasios Vasilakos. 2018. 'Anomaly Detection and Predictive Maintenance for Photovoltaic Systems'. *Neurocomputing* 310 (October):59–68. <https://doi.org/10.1016/j.neucom.2018.05.017>.
- Deline, Chris, Bill Sekulic, Josh Stein, Stephen Barkaszi, Jeff Yang, and Seth Kahn. 2014. 'Evaluation of Maxim Module-Integrated Electronics at the DOE Regional Test Centers'. In *2014 IEEE 40th Photovoltaic Specialist Conference (PVSC)*, 0986–91. Denver, CO, USA: IEEE. <https://doi.org/10.1109/PVSC.2014.6925080>.
- Harrou, Fouzi, Abdelkader Dairi, Bilal Taghezouit, and Ying Sun. 2019. 'An Unsupervised Monitoring Procedure for Detecting Anomalies in Photovoltaic Systems Using a One-Class

- Support Vector Machine'. *Solar Energy* 179 (February):48–58. <https://doi.org/10.1016/j.solener.2018.12.045>.
- Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. 'Long Short-Term Memory'. *Neural Computation* 9 (8): 1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Hopwood, Michael W., Thushara Gunda, Hubert Seigneur, and Joseph Walters. 2020. 'Neural Network-Based Classification of String-Level IV Curves From Physically-Induced Failures of Photovoltaic Modules'. *IEEE Access* 8:161480–87. <https://doi.org/10.1109/ACCESS.2020.3021577>.
- Ibrahim, Mariam, Ahmad Alsheikh, Feras Awaysheh, and Mohammad Alshehri. 2022. 'Machine Learning Schemes for Anomaly Detection in Solar Power Plants'. *Energies* 15 (3): 1082. <https://doi.org/10.3390/en15031082>.
- Kuhn, Harold W. 2010. 'The Hungarian Method for the Assignment Problem'. In *50 Years of Integer Programming 1958-2008*, edited by Michael Jünger, Thomas M. Lieblich, Denis Naddef, George L. Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey, 29–47. Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-68279-0\\_2](https://doi.org/10.1007/978-3-540-68279-0_2).
- Li, B., C. Delpha, D. Diallo, and A. Migan-Dubois. 2021. 'Application of Artificial Neural Networks to Photovoltaic Fault Detection and Diagnosis: A Review'. *Renewable and Sustainable Energy Reviews* 138 (March):110512. <https://doi.org/10.1016/j.rser.2020.110512>.
- Lloyd, S. 1982. 'Least Squares Quantization in PCM'. *IEEE Transactions on Information Theory* 28 (2): 129–37. <https://doi.org/10.1109/TIT.1982.1056489>.
- Longi Green Energy Technology Co., Ltd. LGiLE-T-TMD-059-108 LR5-72HBD 530-550M (35-30&15 frame dual certification) V14. Shaanxi, China. Accessed July 24, 2024. [https://static.longi.com/L\\_Gi\\_LE\\_T\\_TMD\\_059\\_108\\_LR\\_5\\_72\\_HBD\\_530\\_550\\_M\\_35\\_30\\_and\\_15\\_V14\\_4c79e9b9a7.pdf](https://static.longi.com/L_Gi_LE_T_TMD_059_108_LR_5_72_HBD_530_550_M_35_30_and_15_V14_4c79e9b9a7.pdf).
- Ortega, E., G. Aranguren, M.J. Saenz, R. Gutierrez, and J.C. Jimeno. 2017. 'Study of Photovoltaic Systems Monitoring Methods'. In *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, 643–47. Washington, DC: IEEE. <https://doi.org/10.1109/PVSC.2017.8366523>.
- Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, and Trevor Killeen, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In *Advances in Neural Information Processing Systems* 32, 8024-8035. Curran Associates, Inc., 2019. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- Polman, Albert, Mark Knight, Erik C. Garnett, Bruno Ehrler, and Wim C. Sinke. 2016. 'Photovoltaic Materials: Present Efficiencies and Future Challenges'. *Science* 352 (6283): aad4424. <https://doi.org/10.1126/science.aad4424>.
- Rossum, Guido van, and Fred L. Drake. 2010. *The Python Language Reference*. Release 3.0.1 [Repr.]. Python Documentation Manual / Guido van Rossum; Fred L. Drake [Ed.], Pt. 2. Hampton, NH: Python Software Foundation.
- The MathWorks Inc., MATLAB version: 24.1.0 (R2024a), (Natick, Massachusetts, United States: The MathWorks Inc., 2022), <https://www.mathworks.com>.
- Tina, Giuseppe Marco, Cristina Ventura, Sergio Ferlito, and Saverio De Vito. 2021. 'A State-of-Art-Review on Machine-Learning Based Methods for PV'. *Applied Sciences* 11 (16): 7550. <https://doi.org/10.3390/app11167550>.

- Xydis, George. 2013. 'The Wind Chill Temperature Effect on a Large-scale PV Plant—an Exergy Approach'. *Progress in Photovoltaics: Research and Applications* 21 (8): 1611–24. <https://doi.org/10.1002/pip.2247>.
- Zsiborács, Henrik, László Zentkó, Gábor Pintér, András Vincze, and Nóra Hegedűsné Baranyai. 2021. 'Assessing Shading Losses of Photovoltaic Power Plants Based on String Data'. *Energy Reports* 7 (November):3400–3409. <https://doi.org/10.1016/j.egy.2021.05.038>.