

Explaining the Efficiencies of Mass-Produced p-Type Cz-Si Solar Cells by Interpretable Machine Learning

Sven Wasmer,^{*} Kati Hübener, and Bernhard Klöter

It is shown how at Q CELLS, interpretable machine learning algorithms are used to understand the energy conversion efficiencies of mass-produced Q.ANTUM solar cells based on p-type Czochralski silicon (Cz-Si) wafers. For this end, the data of a 1 week ramp-up of over half a million cells acquired utilizing our TRA.Q single-wafer-tracking system are used. These consist of over 300 single information points per cell and feature inline measurements, path- and tool-related information, process data as well as the final I - V characteristics. In the analysis, using machine learning algorithms, it is focused on understanding how the many different input features influence the solar cell efficiency over time by using additive feature impacts. As sources of variation are spread over several features, special attention is paid to correlated features in a hierarchical clustering approach. Finally, with the model achieving a high Pearson's coefficient of correlation of 0.84 between true and predicted values for the holdout validation data set, subtle hourly cell efficiency fluctuations in the order of 0.01%_{abs} are explained. Features are identified which are relevant for short-term changes in the efficiency and others which influence the efficiency on a general level and do not show temporal changes.

1. Introduction

Increasing digitalization and accompanying data acquisition paves the way into industry for modern machine learning algorithms. Especially, the photovoltaics industry is predestined for applying these, as every product is assigned a unique quality-related parameter at the end of its production cycle, i.e., for solar cell production its energy conversion efficiency η_{cell} . Therefore, the problem definition and model building is rather easy given enough predictive information of each solar cell is acquired during the process chain.

However, a well-fit model alone is not helpful in understanding the cell efficiencies and their time trends, as the learned relations are rather hidden in a so-called “black box” surrounding the

algorithm. This issue will be addressed here by applying additive feature impacts, sample-wise assigning each input feature a value that drives the model away from the mean. As these features, and therefore their impacts, can be correlated, the source of variation can be distributed among many features. We therefore propose a hierarchical feature agglomeration, grouping the impacts for a fair comparison (described in Section 2.3).

This work is distinct from the previous works as the approach does not need numerical device simulations and fully utilizes the data of our manufacturing execution system (MES).^[1–5] Furthermore, we improve on our previous study,^[6] as we utilize more data (sample- and feature-wise) and introduce the aforementioned mitigation of the issue of highly correlated features by a hierarchical clustering approach in Section 2.3. Also, by investigating the time trends of solar cell efficiency and its impacts, we explicitly emphasize the

differentiation between features which are relevant for short-term changes in the efficiency and others which influence the efficiency on a general level and do not show temporal changes.


2. Approach

2.1. Data Acquisition and Model

We use the data of over 500 000 Q.ANTUM cells based on p-type Czochralski silicon (Cz-Si) wafers of a 1 week ramp-up acquired fully utilizing our Tra.Q single-wafer-tracking system within one of our production lines.^[7] These data consist of inline measurements (such as wafer base resistivity, emitter sheet resistance, and antireflective coating thickness), path- and tool-related information (e.g., diffusion boat and tube numbers), process data (such as recipes) as well as the final I - V characteristics. As we want to explain the temporal evolution of cell efficiencies, all time-related information such as logging times or virtual IDs are removed, resulting in a total of 329 single features per cell. If this not being done, the commonly greedy machine learning models would certainly also or predominantly include these features to model the time trends of cell efficiencies, not providing assistance in explaining these.

Our model, an ensemble of decision trees with feature subsampling,^[8] is then trained on a random set of 80% of the data

S. Wasmer, K. Hübener, B. Klöter^[†]
Division R&D Wafer/Cells
Hanwha Q CELLS GmbH
Sonnenallee 17-21, Bitterfeld-Wolfen 06766, Germany
E-mail: s.wasmer@q-cells.com

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/solr.202100477>.

^[†]Present address: WAVELABS Solar Metrology Systems GmbH, Spinnereistr. 7, Leipzig 04179, Germany

DOI: 10.1002/solr.202100477

using as target variable the standardized (zero mean, unit variance) solar cell conversion efficiency $\eta_{\text{cell, std}}$ and validated on the remaining 20% unseen data. The random splitting into train and unseen validation data ensures overfitting can be detected, as model fitting is stopped when train and validation errors diverge and the number of trees is set accordingly. We use the term “validation data set” here as it is used to find the best model hyperparameters, and is therefore not a true test set. The standardization follows the recommendation of earlier works for statistical and commercial sensitivity reasons.^[1] Another possibility to split the data is to sort according to time, train on the first 80%, and then validate on the remaining 20% of samples. This may be adequate to validate and optimize the model for its capability to predict ahead, but has two drawbacks to later use the model to explain the whole timeframe. First, the model may be biased to the first 80% of samples and train samples would have to be taken for model explanations, possibly leading to different interpretations for the remaining 20% of samples. Second, if, for example, a poorly performing diffusion boat (manifested by a boat number in the data) would be used only in the last 20% of samples, this would have never been seen by the model and therefore these cells could not be explained by the model. This shows that model fitting is always a trade-off between prediction accuracy and interpretability.

2.2. Additive Feature Impacts

As we do not focus on improving the model accuracy further and further, but rather on understanding what the model has learned, we use additive feature impacts based on Shapley values originally derived for game theory.^[9] These explain a model prediction by assigning for each sample and input feature i a distinct value driving the model away from the mean

$$y_{\text{pred}} = y_0 + \sum_i \Delta y_i \quad (1)$$

where y_{pred} is the model prediction, y_0 is the expected value or the mean of the model, and the Δy_i are the impacts depending on the input feature values. They basically average out all other influences and therefore provide the impact of a single feature alone, uncovering nonlinear relations as well as interactions. Therefore, these are much more generalized than, e.g., the coefficients of a linear regression.^[6] In the case of standardized targets such as being used here, y_0 defaults to zero. The feature impacts Δy_i explain each sample locally. To determine the global impact of each feature, we use their absolute value here, averaged over all validation samples. Also, the global impact on cell efficiency variance in a variance-based sensitivity analysis could be used,^[3] which usually results in more skewed distributions of the global impacts, but the order remains similar.

2.3. Hierarchical Clustering

We find in our analysis that the aforementioned features and therefore their impacts can be highly correlated, possibly distributing sources of variation among many input features. A typical example are emitter sheet resistances measurements being conducted inline and averaged each on three lanes over the wafer,

resulting in three highly similar data points per sample. To alleviate this, we propose to group features according to their similarity with respect to their impact using hierarchical clustering,^[10] iteratively merging the most similar features or clusters, respectively. This is beneficial compared with, e.g., a principal component analysis which results in arbitrary linear combinations of impacts or features, as it is clearly visible which features are joined. We then only analyze the impacts of the formed feature clusters, which here contain between one and up to 13 features. For the hierarchical clustering, we use a distance metric based on the absolute value of the cosine similarity S between the vectors (length equal to the number of samples in the validation data set) containing the impacts of two features Δy_i and Δy_j

$$D(\Delta y_i, \Delta y_j) = 1 - |S(\Delta y_i, \Delta y_j)| \quad (2)$$

which is confined to $[0, 1]$ and is a slightly altered version of the cosine distance by using the absolute value of the cosine similarity. Contrary to the Euclidean distance, this ensures that also negatively correlated features (also resembling a dependence) will be clustered. Furthermore, it nicely normalizes the distances to the range between 0 and 1 and which, in turn, generalizes the cutoff distance in the following hierarchical clustering (see also Figure 1).

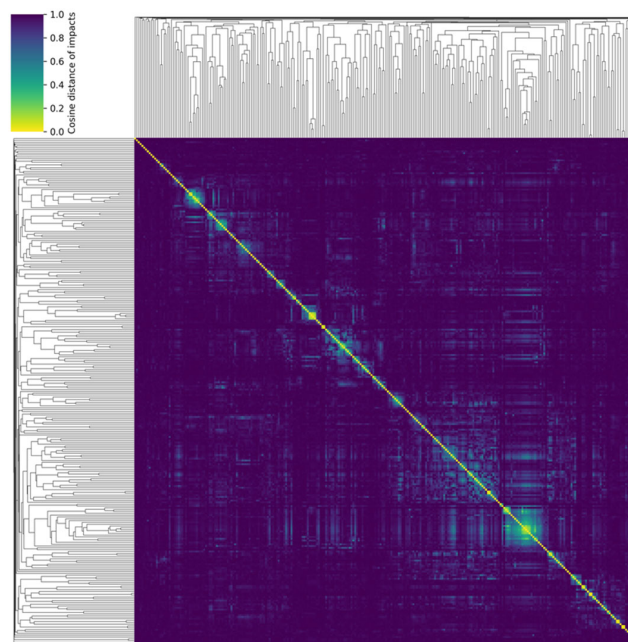


Figure 1. Distance matrix of the feature impacts and the resulting dendrogram by hierarchical clustering. Each row and column depicts one of the 329 features and the color depicts the distance (as defined by Equation (2)) between each combination. The tree-like structures on the left and top illustrate the agglomeration of features showing at which distances (ranging from 0 on the bottom to 1 on the top for the upper dendrogram) the clusters or features are joined. We choose a cutoff distance of 0.6 below which joining is being conducted. The accumulation of low distances around the bisecting line shows that indeed similar impacts are clustered by the algorithm. Features with vanishing mean absolute impact were removed beforehand.

The huge advantage of using the features' impacts instead of their raw values is that all impacts exhibit the same scale or units and that they are continuous in the case of a regression problem as dealt with here. The raw input features' values may also be discrete, causing issues in a reliable correlation analysis. A further advantage of this clustering approach compared with, e.g., removing highly correlated features before model fitting is that we still utilize all the available information and do not have to define a kind of arbitrary threshold of correlation to remove redundant features.

The clustering result of the 329 features examined here using average linkage and a maximum distance of clusters to be formed of 0.6 can be seen in Figure 1, where the features of low pairwise distance nicely accumulate around the bisecting line.

3. Results

The temporal evolution of the standardized cell efficiencies of the holdout validation data set can be seen in **Figure 2a,b**. In summary, the model fares quite well on this unseen validation data (as defined in Section 2.1.) with a Pearson coefficient of correlation of 0.84 and a mean absolute error of 0.33 between the standardized true and predicted values. The high prediction accuracy for the hourly moving averages (correlation coefficient of 0.99 and mean absolute error of only 0.025) is especially remarkable as 1) all time-related quantities are removed beforehand, 2) as we deal with a large timeframe, usually harder to learn than on data from a daily basis, and 3) as there is only little material-related influence (compare, e.g., previous studies^[3,11] for the huge material-related impacts in multicrystalline silicon solar cells).^[3,11] The background of the last point is that to achieve high goodness of fits for multicrystalline silicon solar cells it is already sufficient to include inline measurements such as base resistivity and dislocation density from photoluminescence imaging to already explain big fractions of the absolutely much higher cell efficiency variance. As can be seen from the grayscale colorbar of the dots illustrating the model predictions in the same range as the raw

actual or true data, even the most of the really poorly performing cells are explained by the model (e.g., the ones in the cluster in the beginning and in the efficiency drop after 105 h). Note that the time trend seen in Figure 2b actually is a proof of our stable manufacturing, as the maximum hourly deviations of cell efficiencies is below one unit of the total standard deviation.

Figure 3a shows the Pareto chart of the 20 most important features and clusters regarding their mean absolute impact that is averaging the absolute impact over all validation samples. We highlight the necessity of grouping features for a fair comparison of the impacts, as the chart is dominated by clusters. In a manual review, the grouped features of the most important clusters usually can be assigned to one single process and turn out to be reasonable from a human domain knowledge point of view. In **Figure 3b**, the temporal evolutions of the most important impacts on $\eta_{\text{cell,std}}$ regarding time trends are shown. The seven features and clusters shown in **Figure 3b** are chosen by picking the ones of **Figure 3a** with the highest amplitudes. Beneficially, this approach helps in revealing features such as feature 77, the recipe at the diffusion process, that have a rather timely singular impact on the efficiency. This shows that the approach presented here can also be used to evaluate experiments during production (in this case the test of different recipes at the diffusion process between 20 and 65 h after start with the best being applied at around 50 h after start), allowing to identify room for process enhancement.

Furthermore, it can be clearly seen that the subtle ups and downs of **Figure 2b** can often be assigned to only a few quantities or processes, and especially we can hold responsible a wet bench process for the general trend of cell efficiencies. Interestingly, the top most "Cluster 166" (redundant boat position information) of **Figure 3a** does not show up prominently here, meaning that its high impact is distributed rather homogeneously over time. We also find that a feature relating the wait time between two processes had a negative influence on the cell efficiency during the time period after around 70 h with a low cycle time of the production line, helping us to determine the critical inter process wait times.

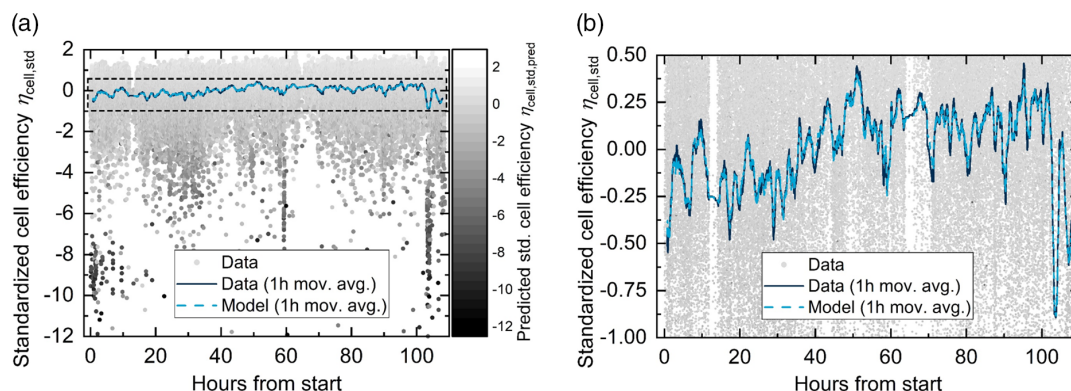


Figure 2. a) Time-resolved development of the standardized cell efficiency during the ramp-up for the holdout validation data set (dots) as well as their 1 h moving averages and those of the model predictions (lines). The grayscale colormap of the dots illustrates the model predictions in the same range as the raw actual or true data showing that even most of the really poorly performing cells are explained by the model (e.g., the ones in the cluster in the beginning and in the efficiency drop after 105 h). b) Zoomed in plot showing the excellent model accuracy on an hourly basis.

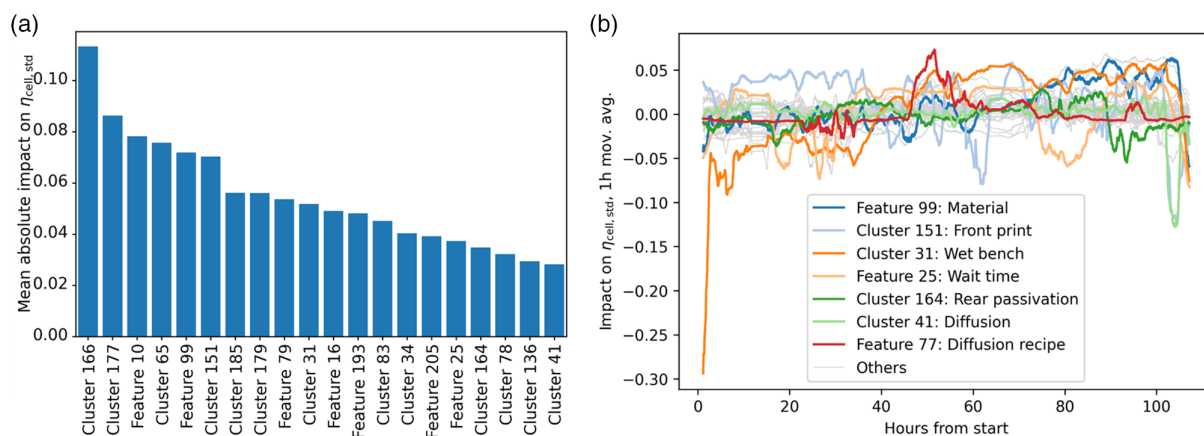


Figure 3. a) Mean absolute impact of the 20 most important impacts. b) Time-resolved plot of the most important impacts (regarding time trends) on the standardized cell efficiency.

4. Discussion

Compared with other works, we focus here not on creating the most accurate machine learning model for our solar cell production, but rather on the understanding of the global hourly time trends of solar cell efficiency seen in a 1 week data set. For the used hourly moving averages, the achieved sample-wise Pearson coefficient of correlation of 0.84 and a mean absolute error of 0.33 between the standardized true and predicted values turned out to be absolutely sufficient to model the hourly trends very accurately (correlation coefficient of 0.99 and mean absolute error of only 0.025). This underlines our impression that for these kind of questions to the model and the data set, it is not necessary to switch the model or tune the hyperparameters further and further, as improvements in the sample-wise prediction accuracy are usually only in the percentage region. For this reason, we refrain from stating the chosen hyperparameters, as these usually are specific to a certain data set and might not work for other data sets, analogously to detailed recipes commonly not being published in research literature of solar cell production. The main prerequisite in applying our approach to solar cell production is to have path- and tool-specific (e.g., diffusion boat and tube numbers), process (such as recipes), and inline measurement (such as wafer base resistivity, emitter sheet resistance, and antireflective coating thicknesses) data related to each single wafer (apart from the I - V characteristics of the finished cell), e.g., by a single-wafer tracking in a MES, which needs to join all these different data sources. Concerning the machine learning aspects, as the data are already structured (contrary to, e.g., images), no exorbitant requirements are posed to the used computer hardware and the results presented here for the tabular data consisting of 500 000 rows and ≈ 300 columns can be achieved on modern laptops with at least 8 GB of RAM. The central advancement by this research can be seen in the proposed solution for, what we encounter in our analyses, one of the most detrimental problems in interpretable machine learning: correlated features. These might severely affect every additive feature impact method or, e.g., regression coefficients. Our approach of grouping features according to their similarity with respect to their impacts

using hierarchical clustering turns out to be necessary as the Pareto bar chart is dominated by clusters. It should be generally transferable to other areas and data sets with tabular data.

5. Conclusion

In summary, we showed the valuable benefit of using interpretable machine learning algorithms for explaining the efficiencies of mass-produced Q.ANTUM solar cells based on p-type Cz-Si wafers including their time trends. As for the model explanation, care has to be taken regarding correlated features, which we addressed with a hierarchical clustering approach that it is applicable to other data sets and machine learning tasks affected by correlated features. We determined impacts which are relevant for short-term changes in the efficiency and others which influence the efficiency on a general level and do not show temporal changes. The shown methods help to optimize a running production in a continuous improvement process or to quickly troubleshoot, e.g., during a ramp-up of production of new cell technologies such as ones using passivating contacts on n-type wafers. They are also applicable to understand off-spec cells featuring, for instance, hot spots or optical issues. Another possible application of the time trends of additive feature impacts could be in predictive maintenance, where increasingly negative impacts of single processes over time could be quickly identified and mitigated.

Acknowledgements

B.K. contributed to this work while being at Q CELLS. The authors would like to thank the R&D and production staff at Hanwha Q CELLS for data acquisition.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

Research data are not shared.

Keywords

machine learning, manufacturing and processing, PERC, silicon solar cells

Received: June 30, 2021

Revised: October 1, 2021

Published online: October 14, 2021

-
- [1] R. Evans, M. Boreland, *IEEE J. Photovoltaics* **2018**, *8*, 38.
[2] P. P. Altermatt, Y. Yang, Y. Sheng, D. Chen, Y. Chen, Z. Feng, P. J. Verlinden, *AIP Conf. Proc.* **2018**, *1999*, 110001.
[3] S. Wasmer, J. Greulich, H. Höffler, N. Wöhrle, M. Demant, F. Fertig, S. Rein, *IEEE J. Photovoltaics* **2017**, *7*, 118.
[4] H. Wagner-Mohnsen, P. P. Altermatt, *IEEE J. Photovoltaics* **2020**, *10*, 1441.
[5] Y. Buratti, C. Eijkens, Z. Hameiri, *ACS Appl. Energy Mater.* **2020**, *3*, 10317, 10322.
[6] S. Wasmer, B. Klöter, in *Proc. 36th European Photovoltaic Solar Energy Conf. and Exhibition*, Vol. 272, WIP Renewable Energies, Marseille, France **2019**, p. 274.
[7] S. Wanka, D. Rychtarik, J. Müller, S. Geissler, P. Kappe, M. Spallek, U. V. Bauer, C. Ludwig, P. Wawer, in *Proc. 37th IEEE Photovoltaic Specialists Conf.*, IEEE, Seattle, USA **2011**.
[8] L. Breiman, J. H. Friedman, R. Olshen, C. J. Stone, *Classification and Regression Trees*, Brooks/Cole Publishing, Monterey, CA **1984**.
[9] L. S. Shapley, *Contributions to the Theory of Games II*, Vol. 28, Princeton University Press, Princeton, NJ **1953**, pp. 307–317.
[10] D. Müllner, *Modern hierarchical, agglomerative clustering algorithms*, <https://arxiv.org/abs/1109.2378v1> (**2011**) (accessed: June 2021).
[11] M. Demant, S. Rein, J. Haunschild, T. Strauch, H. Höffler, J. Broisch, S. Wasmer, K. Sunder, O. Anspach, T. Brox, *Prog. Photovoltaics Res. Appl.* **2016**, *24*, 1533.