Comparing Time Series Classification And Forecasting To Automatically Detect Distributed Generation

Aleksandr Petrusev, Rebecca Bauer, Rémy Rigo-Mariani, Vincent Debusschere, Patrick Reignier, Nouredine Hadjsaid

Comparing Time Series Classification And Forecasting To Automatically Detect Distributed Generation

Aleksandr Petrusev[1,2], Rebecca Bauer[1,3], Rémy Rigo-Mariani[1], Vincent Debusschere[1], Patrick Reignier[2], Nouredine Hadjsaid[1]
[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, G2Elab, 38000, Grenoble, France
[2] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000, Grenoble, France
[3] Karlsruhe Institute of Technology, Institute for Automation and Applied Informatics, Karlsruhe, Germany
aleksandr.petrusev@grenoble-inp.fr; rebecca.bauer@student.kit.edu

***Abstract*— This paper introduces tools for the automatic detection of "hidden" behind-the-meter solar generation in case where there is no monitoring or connection agreement contract with the system operator. The objective is to reach the highest precision while discriminating the nodes with and without solar generation. The proposed methods are based on exogeneous information (smart meter and temperature data) and artificial intelligence techniques consisting of neural networks as well as analytical classification algorithms. A wide range of models differing in size, architecture and number of parameters has been investigated, and the best performing ones are presented in the article. The first method involves time series classification (TSC), and the second involves time series forecasting (TSF). Open-access consumption data were used for the training of the neural networks. The implemented solutions were tested across all the nodes of the simulated electrical grid and the sensitivity of the tools was analyzed with regard to the level of PV penetration. One of the proposed tools is able to detect up to 100% of new PV installation, depending on the exogenous conditions.**

***Index Terms*—Artificial intelligence, classification, behind the meter solar production, time series, neural network.**

I. Introduction

Distributed renewable energy, especially photovoltaic (PV), has grown rapidly over the past two decades. Information about PV generation is crucial for distribution system operations such as status estimation, reconfiguration and voltage management. However, some behind-the-meter solar installations may not be subject to monitoring or connection agreement with the system operator. If not identified, that "hidden" generation, may incur additional uncertainty into the net charge (by reducing the net load compared to the expected one during daytime) and particularly makes it more difficult to securely operate the distribution grid. With the constant growth of installed PV capacities, this problem becomes more and more important. Therefore, the development of a tool that automatically detects nodes with PV production in a distribution grid based on smart meter data can be an essential asset for operators with a lack of observability. There have been several approaches aiming at detecting PV installations in distribution grids. As discussed below, these often involve domain knowledge in form of detailed mathematical modelling and a mix of methods.

An approach to identify customers with PV power generation using net energy consumption data from smart meters is proposed in [1]. In order to reduce the amount of data needed to accurately identify solar prosumers to a single data point, the authors propose a method of dimensionality reduction, which outperforms k-means, combined with a classification method. This method is based on agglomerative clustering and self-organizing maps (artificial neural network-based clustering technique). The disadvantage of this method is that the clustering of customers into two groups (with and without PV) is implemented without historical data about consumption of these customers in previous years. Thus, the algorithm can identify customers as "with PV" if their total consumption or peak consumption is below the average value, but it is also possible even without PV.

A computer algorithm that automatically detects PV panels using very high-resolution color satellite imagery (0.3 meters per pixel) has also been tested [2]. A Random Forest Classification machine learning technique to detect the presence of PV is proposed. The main problem of the algorithm lies in its principle - it needs high-resolution satellite imagery. Moreover, it needs labeled training data.

A distributed photovoltaic systems capacity estimation approach is presented in [3]. Using a support vector machines (SVM), the algorithm determines whether a customer has a PV or not. Several features describing the discrepancy of net load curves between customers with PV and those without are extracted, based on weather status driven characteristics of PV output power – e.g. ratio of total electricity consumption, concave shape index, concavity degree and load ramping rate. The disadvantage of the method is that it needs the output power data from PV of known customers.

Another approach for solar prosumer identification is change-point detection [4], which detects abnormal energy consumption behaviors including unauthorized PV installations. Change-points in customer load may be caused by other abnormalities. Therefore, the existence of the unauthorized PV installation is further verified through a statistical inference known as permutation test with the

Spearman's rank correlation coefficient. However, this approach is unable to detect PV installations until after the rolling window length is completed (2–14 days lengths were evaluated). Moreover, the cloud cover index is needed.

Finally, authors in [5] present a method for detecting and disaggregating behind-the-meter solar generation using weather data, advanced metering infrastructure, substation monitoring and generation monitoring for a few PV systems nearby the circuit. This method, like the previous one, needs various types of data such as solar radiation data, data from other PV stations, and detailed weather data or even satellite imagery.

In this paper, the objective is, on the contrary, to not require any detailed modelling of the grid and production as well as rely strictly on smart metering and temperature data. The motivation lies in the capacity of DSOs to use available data from users without facing privacy issues not dealt with previously and to limit as much as possible the requirements for getting the results just needed for the operation of the grid.

Machine learning techniques as neural networks (NNs) are used in a very large variety of contexts in the energy field [6], especially forecasting and disaggregation. To the best of our knowledge, detection of PV installations has not been extensively covered with in mind the practical compromise of data limitation, simplicity of implementation and usable results. Two different approaches are explored in this paper, denoted Method A and Method B. Method A is based TSC, and Method B leans on the concept of TSF which is applied for classification task. Both involve NNs. While approach A explores convolutional neural networks (CNNs) with a wide range of architectures and settings, Method B is more transparent and combines a conventional Multi-Layer Perceptron (MLP) together with an analytical classification algorithm. The two approaches with similar, albeit not equal assumptions, are compared, regarding their efficiency to detect PV production.

The paper is organized as follows: Section II describes the simulation setup and presents both methods A and B. The obtained results and sensitivity analysis are discussed in Section III before conclusions are drawn in Section IV.

II. METHODS

A. Experimental setup

To simulate the hourly net consumption profiles of the grid for two consecutive years (called year *n-1* and year *n*), open-access consumption data of hundreds of households of London were used [7]. The consumption profiles were aggregated into 14 groups for each node (a dozen of households per node).

To model PV generation in some nodes, the web application NREL's PVWatts [8] was used to estimate solar radiation from a specific geographic position (typical meteorological year, TMY). Additionally, the DarkSky API [9] was used to obtain temperature data for the same geographic position and datetimes as the net consumption. The PV generation profiles, obtained using the above data were integrated to seven nodes during the year *n* and scaled in accordance with the consumption profiles of these nodes and the objectives of the experiments. The PV installed capacity in any given node is expressed with regards to the peak load value of this node $P_{nom}^{PV} / \max(P^{load})$. The dataset creation for Method A varies slightly, as described in section B.

Therefore, there are 14 nodes for the simulation over two years, of which seven have PV added during the second year. The goal is to develop an approach that can detect these seven randomly chosen nodes, based on consumption and temperature data for no more than two years.

B. Method A (TSC-based)

The first approach relies on CNNs and an ensemble model based on the Multi-channel deep CNN (MCDCNN) [10] built within the *PyTorch* package. A graphic representation of a CNN and a convolution can be found in Figure 1 for illustration. The overall implemented process is shown in Figure 2.

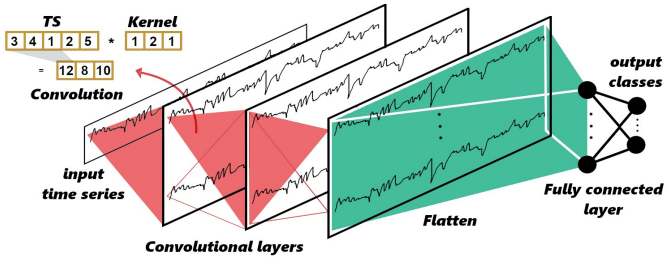

Figure 1. Convolutional neural network architecture.

The only input data is the simulated net consumption of all nodes. Inputs are time series (TS) of 24h, outputs are two classes that identify whether there is a PV installation (1) or not (0). Optionally, the classified days can be applied a majority vote on. All models are aimed to be as simple as possible in terms of size, i.e. the trainable parameters, i.e. the number of weights and biases (e.g. convolution kernel size, NN weights, and max-pooling kernel size).

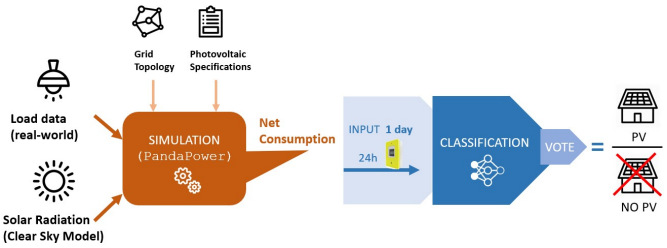

Figure 2. Operating principle of Method A.

The models are detailed in Figure 2: There is one CNN and a MCDCNN that is an ensemble of CNN-branches applied to each node individually and concatenated afterwards.

The data used is the London data set from [7], simulated once with one set of PV nodes and a second time with an "inverted" set of nodes regarding PV attribution for both training and testing (two years each). Hence each node is simulated once with PV and once without. The simulation is conducted with various levels of PV capacity: 8.5% (normal), 25% (slightly higher), 50% (high), 75% (very high) (see Section II). Input data is shuffled, but not scaled (shows

slightly better results). The training set is from April to September as the time series are most discriminable in these months due to higher PV generation.

TABLE I. MODEL DETAILS

Models	*CNN*	*MCDCNN*
Input	24h TS	24h TS (1 branch/node)
# CLs	2	2
# FCLs	1	2
Regularization	MP, BN	MP, BN
AFs inside	Sigmoid	ReLU
AF output	Sigmoid	Sigmoid
# Parameters	561	699

MP = Max-Pooling, BN = Batch Normalization, TS = Time Series.

Training is conducted with the Adam optimizer, the BCEWithLogitsLoss (binary cross entropy) loss function (calculates cost or error), a learning rate of 0.001, a batch size of 200, and 500 epochs (number of training iterations with whole data set). Those choices were found to be the most relevant during the exploration stage. The initialization is based on the PyTorch default, Kaiming for CLs and Xavier for FCLs [11]. As the data set is balanced, accuracy is chosen as an evaluation metric.

The existing method for comparison (benchmark) is given by the CNN of the repository of the TSC review [12] that contains many state-of-the-art models (*Keras* package) and that works with the UCR/UEA archive [13]. It is used for the general assessment of TSC algorithms due to comparability.

C. *Method B (TSF based)*

The second method is designed to be less considered as a "black box" but more transparent while still presenting a high precision. The objective is to detect whether new PVs have been installed in the nodes of a distribution grid during a considered period (e.g. during the last year or month).

The developed tool consists of a NN coupled with an analytical classification algorithm that is applied separately for each node. The operating principle is shown in Figure 3. The first part of the data (at the bottom) represents the year *n-1* ("training set") and the second (at the top) the year *n*, for which the recently installed PV shall be detected.

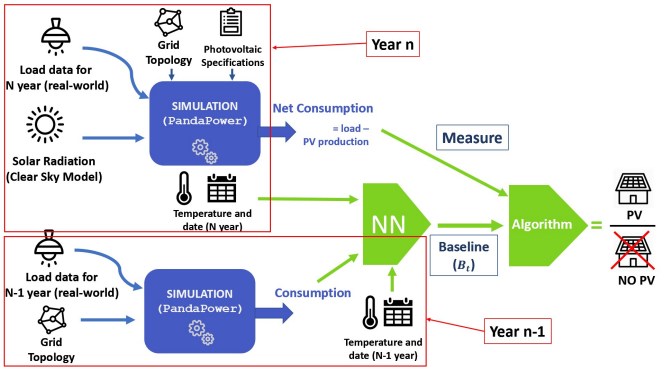

Figure 3. Principle of operation of method B.

Trained by consumption data from the year *n-1*, the NN produces the expected consumption for a given day of the year *n* at an hourly resolution, assuming that no new PV has been added. This forecast is called *baseline,* which is then fed to the classification algorithm.

The net energy consumption measured by the meters (which provides the difference between load and PV generation) for the considered period of the year *n* is also fed to the classification algorithm and is called *measurement*.

Finally, the analytical classification method compares the baseline and measurement values during daytime and nighttime to detect PV.

The first layer of the NN (Figure 4) contains 31 neurons, that is, 31 features, which were chosen after sensitivity studies showing the strongest effect on consumption:

- 24 features to determine at what hour of the day the simulation will be performed – hi, where i = 1. ... 24;
- 4 others indicate the season – Sh (winter), Sp (spring), Se (summer), Sa (autumn);
- The last 3 are temperature (t°), weekend (W) and holiday (F).

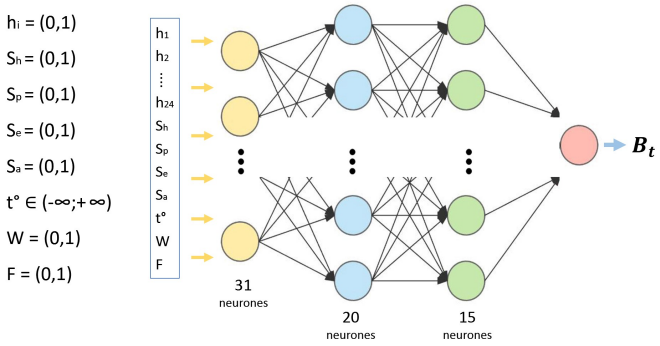

Figure 4. Neural network and its features as used in method B.

The NN contains two hidden layers, of 20 and 15 neurons each. The output layer consists of a single neuron, which calculates the consumption for the chosen hour. This architecture showed the highest precision as the result of sensitivity studies.

The training is conducted with the Adam optimizer and the mean absolute error loss function, a learning rate of 0.03 and 1000 epochs are set. The time of training is tens of seconds for each bus.

The analytical classification algorithm is presented in Figure 5. The basic idea is that, in absence of PV generation, the difference of baseline and measurement values is approximately the same for hours of the day and of the night. Hence, if that difference is greater during hours of the day, then the algorithm detects that new PVs have been added in the considered period, since PV generation can occur only during the day.

At first the algorithm calculates E_t (in %), the difference between the baseline (B_t) and the measurement (P_t), for each hour *t* of the period T.

Similarly, E_t^S is computed as the difference between the baseline and the measurement, but only for the hours of the day (between 9 a.m. and 4 p.m. of each day), i.e. a total period T^S of 2920 hours per year (or less if we consider less than a year, i.e. T < 8760 hours). The period between 9 a.m. and 4 p.m. is chosen because during these hours there is a significant generation of PV whatever the season.

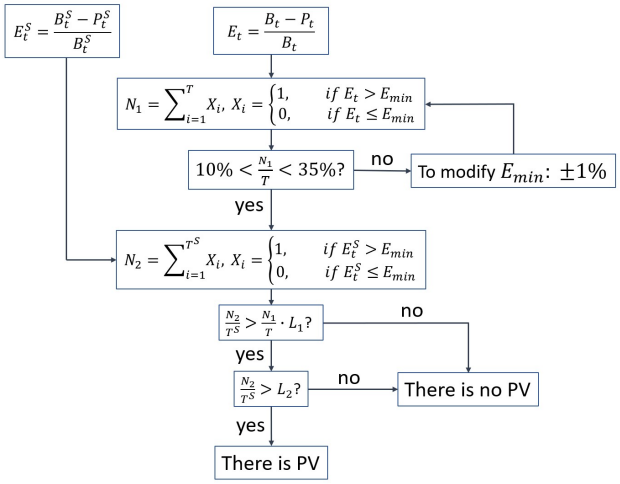

Figure 5. Principle of operation of the algorithm at the core of method B.

Then, the algorithm enumerates the number of hours N_1 when E_t is greater than the threshold E_{min} (10% by default). It therefore counts the number of hours for which the baseline is at least 10% greater than the measurement.

If N_1 lays outside the interval of [10%; 35%] of the total number of hours over the considered period, the algorithm adjusts the threshold E_{min} to avoid any overestimation or underestimation from the NN and afterwards repeats the process, counting N_1 again. This reduces the estimation error and makes the algorithm immune to changes in the number of inhabitants. The values of 10% and 35% have been obtained empirically through sensitivity studies. E_{min} is the only threshold that is to be adapted.

Once E_{min} is set, the algorithm calculates N_2 - the number of hours for which the baseline is greater than the measurement during the hours of sunshine by at least the value of the previously obtained E_{min} threshold.

Then the algorithm checks if N_2/T^S is greater than N_1/T by at least L_1% (140% by default). Thus, it checks whether the situation where the baseline is greater than the measurement on E_{min} % is more frequent during the hours of sunshine than any other time period. If there is no new installed PV, N_2/T^S and N_1/T will be roughly the same. If there is a new PV generation, then N_2/T^S will be significantly larger than N_1/T.

Finally, the algorithm checks that N_2/T^S is larger than L_2%. That is, at least L_2% of baseline hours are larger than the measurement by E_{min}%. In case both last checks are successful, the algorithm concludes that the node has a local PV generation and hence that a new PV was installed at this node.

III. Results and Comparison

A. Results for Method A

The test accuracies for all models as well as results from benchmark models are depicted in Figure 6. The accuracy was calculated according to the following expression:

$$Accuracy = \frac{number\ of\ correct\ detection\ results}{number\ of\ all\ results}, \qquad Accuracy \in [0;1]$$

For the London dataset [7] with 8.5% of PV integration, the scores are hardly above 0.5 (i.e. the random score). Hence, to see if the network is able to discriminate with higher PV production rates, the percentage was increased to 25%, 50% and 75%, which is very high and not that realistic. The results show that the CNNs are the most suitable models with an accuracy of 0.74. The MCDCNN does not perform well. The results can be explained by the CNNs having different consumption curves and are hence able to generalize more than the MCDCN. Thus, it can adapt better to unseen data. Also, the CNN uses sigmoid as an activation function instead of PReLU. A quick check on the hidden layers verifies that the neurons do not saturate, i.e. the network is functioning.

There are two benchmark models, a CNN with 614 parameters and a ResNet with 504,258 parameters. Both have low scores for the realistic 9% PV integration London dataset, and work much better on higher PV percentages, although they do not surpass the model proposed here. Note that, interestingly, the CNN has a lot less parameters than the ResNet, showing that more parameters do not automatically yield better performance.

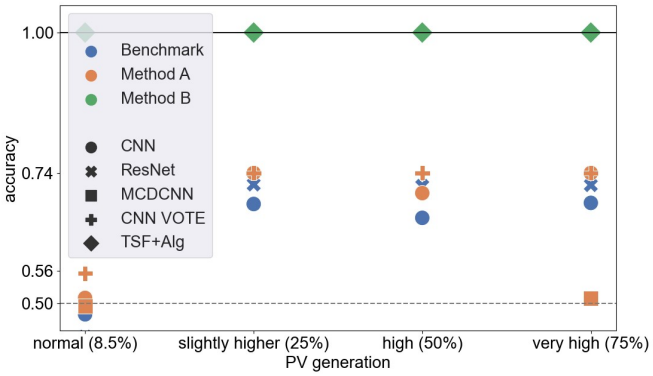

Figure 6. Accuracy test scores from April to September for three models of Method A (orange), its benchmark (blue) and Method B (green) on the London data, simulated with varying degrees of PV generation (8.5%-75%).

Examining why the models do not perform better, the reason can be found in the combination of model design and data. Figure 7 shows an excerpt from the unscaled data, with low and very high PV production for a few days in May 2012. It becomes clear that PV does have some impact: there now appear negative values of the net load for very high PV production and the curves seem stretched vertically. However, the general pattern of smaller variations is not altered. Hence, it still contains many of the information as the net consumption without any PV. The model choice then comes into play. CNNs are designed to extract smaller patterns

(depending on the filter size). Our basic assumption was that PV changes the curves' small patterns *enough*, i.e. such that CNNs will detect them. It now becomes clear that this assumption does not hold. Also, CNNs detect similar curves largely independent of their absolute height. However, those absolute differences are necessary for the classification (i.e. PV or not) in the studied problem. This is the reason why CNNs are not able to see the added PV, i.e. differentiate between PV and none.

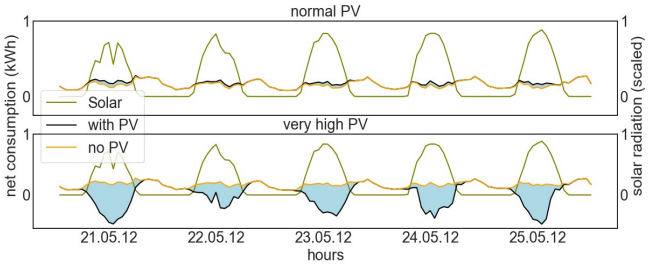

Figure 7. Net consumption profiles (unscaled) with (yellow) and without (black) PV and their difference (blue) for normal (upper) and very high (lower) PV plus solar radiation (green) for five days in May 2012. TODO

B. *Results for Method B*

The implemented Method B is trained on and works for each node individually. The 14 nodes were prepared, with PV connected to seven of them, randomly selected.

The accuracy of the method depends on which months were selected for the analysis, mainly because they differ in the level of solar radiation. To confirm this theory, two cases are considered - using consumption and temperature data for the six most sunny months (from April to September) and for the six least sunny months (from October to March).

The dependence of the average accuracy of the tool with respect to the period under consideration and to $P_{nom}^{PV} / \max(P^{load})$ for the whole network is presented in Figure 8 for the period from April to September and in Figure 9 for the period from October to March.

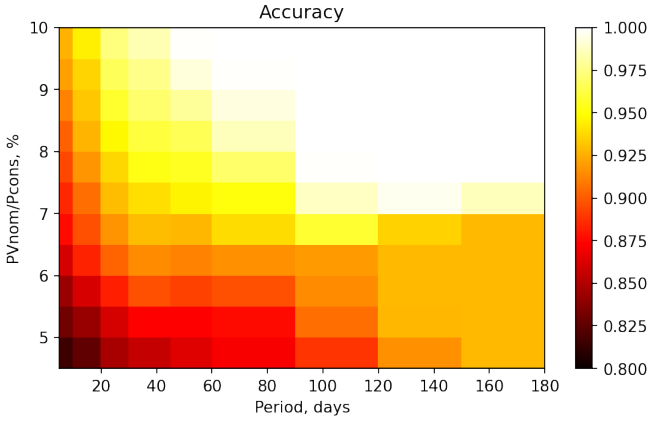

Figure 8. Average accuracy dependence from April to September with Method B.

In the following results and corresponding figures, the "Average" performance refers to the average accuracy calculated for all possible periods of x days over 6 months with on a rolling basis. (e.g. for "period = 60 days", the average accuracy of 122 possible 60-day periods between April and September was calculated).

The average accuracy from April to September is then between 0.8 ($P_{nom}^{PV} / \max(P^{load}) = 4.5$ %, over a period of five days) and 1 ($P_{nom}^{PV} / \max(P^{load}) > 7.0$ %, for a period of more than two months). The results show that the average accuracy for the sunniest months is higher for longer periods as a longer period reduces the impact of cloudy days, when PV generates less energy. It is also obvious that a greater ratio $P_{nom}^{PV} / \max(P^{load})$ facilitates the detection of PV, so the average accuracy is also higher.

The results are different for period from October to Mars. The average accuracy for these months is between 0.87 ($P_{nom}^{PV} / \max(P^{load}) = 10.0$ %, over a period of five months) and 0.5 ($P_{nom}^{PV} / \max(P^{load}) = 4.5$ %, for a period of six months). For ratios $P_{nom}^{PV} / \max(P^{load})$ lower than 8.5 %, the average accuracy is better for shorter periods, because on average during these months the PV systems do not generate enough power for detection, but there are still few days with high solar radiation level where detection is possible.

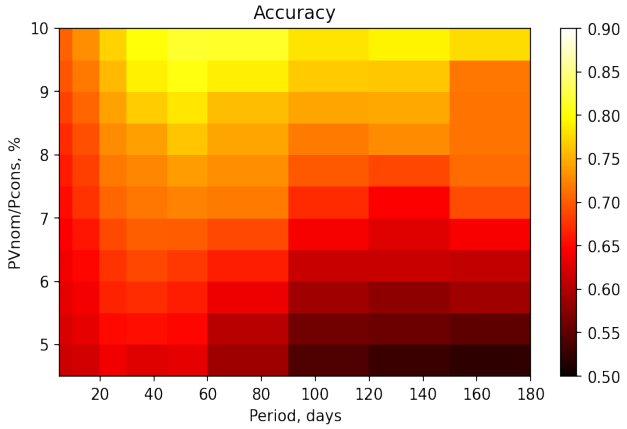

Figure 9. Average accuracy dependence from October to March.with Method B

Thus, it can be concluded that it takes an average of the sunniest three months of a year to get the highest detection accuracy.

TABLE II. DISTRIBUTION OF THE INSTALLED PV POWER AND THE CORRESPONDING ENERGY PRODUCTION

Bus	2	3	4	8	9	11	12
$\frac{P_{nom}^{PV}}{\max(P^{load})}, \%$	6,2	6,2	6,2	6,2	6,2	6,2	6,2
$\frac{W_{nom}^{PV}}{W^{load}}, \%$	3,3	3,2	3,3	3,1	3,1	3,1	3,7

Considering the periods of the three sunniest months, 100 % accuracy of the method on the test system can be achieved with values of 6.2 % for the installed PV capacity in

the node with respect to the maximum load ($P_{nom}^{PV} / \max(P^{load})$), and 3.1 % for the generated PV energy compared to the energy consumption of the same node (W_{nom}^{PV} / W^{load})(Table II).

It should be mentioned that the sensitivity of the tool depends on consumption profiles (for two years). Thus, a sudden change in the consumer behavior, such as an increased consumption during sunshine, may affect the performances. However, if the number of inhabitants changes, this should not have a significant impact on the results, because the algorithm can compensate for this by adjusting the threshold.

IV. Conclusion

The paper prosed two methodologies for the detection of hidden PV generation in a distribution grid (methods A and B). It was found that forecasting tools, as proposed in Method B, works better than direct classifications algorithms (as proposed with Method A). It is mainly explained by the fact that neural networks are a lot better at learning small patterns than at discriminating similar curves that have been vertically distorted by PV. Thereby the PV detection approach (Method B), that uses only smart meter and temperature data, can be chosen as the best performing method in this case.

The overall approach of using neural networks for such classification performs well on the considered problem, as the network can be trained offline within minutes and can then analyze any time period within seconds.

From the comparison of both approaches it can be concluded that, for energy time series classification, it is necessary to use some domain knowledge. With the assumptions made here, the tool is well usable in practice, especially since past consumption data and temperature data are often available.

Future works will consist in testing the tools on larger grids in simulation. A next step will then be to not only detect PV installations, but also to disaggregate its values from the net load and hence approximate the amount of PV generation. Regarding the data, further thoughts can be given to noise and anomaly detection as the simulated data has been very clean in the presented study. Validating the presented techniques on real data could be as well rewarding. Regarding the (C)NNs, future work can experiment on larger grids with a larger and more diverse range of consumption profiles. It can be stated that Method B represents a good basis for PV detection in distribution grids, offering many opportunities to be expanded further upon in future work.

Acknowledgment

This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003). Thank goes also to Benedikt Heidrich from Karlsruhe Institute of Technology, Institute for Automation and Applied Informatics, for helpful discussions.

References

[1] Donaldson Daniel L, Jayaweera Dilan. “Effective solar prosumer identification using net smart meter data”. *Int J Electric Power Energy Syst* 2020;118:105823. https:// doi.org/10.1016/j.ijepes.2020.105823. ISSN 0142-0615.

[2] Malof JM, Bradbury K, Collins LM, Newell RG. “Automatic detection of solar photovoltaic arrays in high resolution aerial imagery”. *Appl Energy* 2016;183:229–40.

[3] Wang F, Li K, Wang X, Jiang L, Ren J, Mi Z, et al.. “A distributed PV system capacity estimation approach based on support vector machine with customer net load curve features”. *Energies*; 11(7):1750, 2018.

[4] Zhang X, Grijalva S.. “A data-driven approach for detection and estimation of residential PV installations”. *IEEE Trans Smart Grid*;7(5):2477–85, 2016.

[5] Michaelangelo Tabone, Sila Kiliccote, and Emre Can Kara. “Disaggregating solar generation behind individual meters in real time”. *In Proceedings of the 5th Conference on Systems for Built Environments*. ACM, 43–52, 2018.

[6] Ali S, Choi B.J. “State-of-the-Art Artificial Intelligence Techniques for Distributed Smart Grids: A Review”. *Electronics*, 9, 1030, 2020.

[7] “Smart meters in London”. URL: https://www.kaggle.com/jeanmidev/smart-meters-in-london

[8] PVWatts Calculator. URL: https://pvwatts.nrel.gov/index.php

[9] Dark Sky API. URL: https://darksky.net/dev

[10] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, J. Leon Zhao. “Exploiting Multi-Channels Deep Convolutional Neural Networks for Multivariate Time Series Classification”, *Web-age information management*, pp 298–310, 2014.

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan et al.. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, *Advances in Neural Information Processing Systems 32*, pp 8024-8035, 2019.

[12] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller. “Deep Learning for time series classification: a review”, *Data Mining and Knowledge Discovery*, pp 917-963, 2019.

[13] Hoang Anh Dau, Anthony Bagnall, Kaveh, Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. “The UCR Time Series Archive”, IEEE/CAA *Journal of Automatica Sincia,* Vol. 6, No. 6., 2019