Load disaggregation at the distribution level using artificial neural networks

William Templier
IM²AG, Université Grenoble-Alpes
Grenoble, France
william.templier@etu.univ-grenoble-alpes.fr

Supervised by: Vincent Debusschere and Rémy Rigo-Mariani

I understand what plagiarism entails and I declare that this report is my own, original work.
Name, date and signature:
Templier, William - 06/11/2021

Abstract

The increasing number of solar panels can incur instability and variability in the electricity distribution grid, as solar energy produced at the individual-level affects the local stability of electric quantities. For the distribution system operators, this downstream production is not monitored by default but impacts the power consumption at mixed point of connection. In that case, the only available information is the *net load*, which is the electricity consumed downstream minus the local solar energy production. In order to improve the grid planning and operation, disaggregating both components from the net load can be relevant. The aim of this internship was to explore methods based on artificial neural networks to perform the task of load disaggregation. A single-layer neural network was trained and evaluated on three different datasets with increasing complexity. Different data reduction scenarios to evaluate the sufficient sample size were also tested. On all the datasets, the *season* scenario (training on one month per season) obtained the best overall performances, with an averaged RMSE for both components of 0.01, 0.05, and 0.08. Finally, the field of probabilistic forecasting was briefly explored in order to provide prediction intervals with the disaggregated components. For this exploratory phase, further works is required but preliminary results are promising.

1 Context & problem

The energy transition needed to limit the impact of climate change will in part rely on the increasing number of renewable energy sources in the electricity grid, and more specifically photovoltaics (PVs). Nevertheless, these means of energy production are mostly decentralized, in contrast to conventional heavy duty fossil fuel power plants and a great majority of these new small scale assets are connected to middle and low voltage distribution grids. In addition, they are owned by private actors and are not necessarily visible/controllable by the distribution system operators (DSO). PVs owners can typically turn into "prosumers" and their net load (*i.e.*, from the DSO perspective) varies along the PV production profiles. They may even inject surplus of local generation into the grid in case of low consumption and significant solar capacities installed. This brings uncertainty and variability into the legacy grid, which incurs new operation and planning challenges for the DSOs.

One issue for the DSOs is that the power generated by PVs can be "masked", and cannot be directly measured. If no connection agreement exists between the prosumers and the DSOs, the PVs are hidden ("behind-the-meter"). Within the *Artificial Intelligence for Smart Grids* (AI for SG) PhD subject, the task to automatically detect PVs production (but not the level of production) was tackled by Aleksandr Petrusev (Ph.D candidate) and Rebecca Bauer (graduate student) with artificial neural networks (ANNs) using historical data analysis [Petrusev *et al.*, 2021].

An important task for the DSO is the net load forecasting for operational purposes. In particular, the DSO is responsible for the power delivery at any point of its grid in time, while ensuring that the system operates within technical limits (*e.g.*, voltage/current constraints). This forecasting task is traditionally performed with tools that consider historical consumption profiles only, and thus may loose some relevance in case of significant solar penetration.

The DSO has information about the net load of, for example, a given profile i:

$$P_{i,t}^{N} = P_{i,t}^{load} - P_{i,t}^{pv} \tag{1}$$

where P stands for *power*, $P_{i,t}^{N}$ being the net load at time t, $P_{i,t}^{load}$ the actual consumption (pure load) of the profile downstream, and $P_{i,t}^{pv}$ its solar energy production.

In order to make net load forecasts that are more robust to intermittent energy production, one idea is to disaggregate both components of the net load $P_{i,t}^{N}$. This would give the DSO more accurate information about the prosumers' profiles and allow them to independently predict the production and consumption profiles. The goal of this internship is to evaluate if ANN models could tackle the task of load disaggregation.

2 Related work

From the literature around net load disaggregation, two main classes of methods can be considered to address the problem: *model-based* or *data-driven* approaches. The former approach uses parametric models based on knowledge about the geometry of the PV and combine weather and solar irradiance models to estimate solar power generation and subtract it from the net load ([Tabone *et al.*, 2018; Bu *et al.*, 2020]). The latter is an umbrella term for most commonly statistical, machine learning, or hybrid methods. As an example of a hybrid method, [Bu *et al.*, 2020] devised a game-theoretic approach that uses pure load and PV generation of fully observable customers to disaggregate both components from customer profile where only the net load is known.

If we narrow down to our area of interest, namely machine learning-based methods (and more specifically, ANNs), the literature on PV/load disaggregation *per se* is still sparse, as pointed out by [Saeedi *et al.*, 2021]. The latter study is the only one we could find where the disaggregation problem is directly addressed - in contrast to study using ANNs to forecast PV generation. Our literature review found that ANNs are ubiquitous in load disaggregation at the consumer-level: *i.e.*, from the measured load, disaggregate the respective contributions of each home appliance (*e.g.*, dishwasher, refrigerator, and such). In contrast, as already pointed out, load disaggregation at the distribution-level using ANNs, net load profiles and exogenous data is still in its infancy.

3 The task

Instead of building upon existing literature on the subject of load disaggregation using ANNs, my tutors and I decided to start from scratch: from the simplest model in my toolkit, what kind of results would it yield? The idea was to build a first reference model that would serve as existing tool to build upon for the PhD student (A. Petrusev) in future works. The chosen model is a single-layer neural network that is trained to disaggregate the pure load ($\hat{P}^{load}_{i,t}$) and the PV generation ($\hat{P}^{pv}_{i,t}$). A model D was designed, with predictors P_N, the net load, G, the solar irradiance T, the temperature, and DT, which represents date-time features (season and hour), see Figure 1. Thus, the model can be expressed as follows:

$$\hat{P}^{pv}_{i,t}, \hat{P}^{load}_{i,t} = D(P^N_{i,t}, G_t, T_t, DT_t) \tag{2}$$

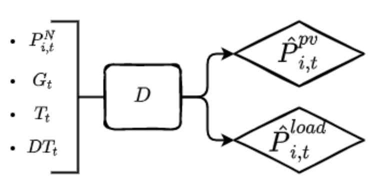

Figure 1: MLP model.

For the performance metric, the *root-mean-square error* (RMSE), was used as it is common in the literature. It is computed as follows for the solar power component (likewise for P^{load}), over a horizon of T samples (for profile i):

$$RMSE_{\hat{P}^{pv}_i} = \sqrt{\frac{1}{T}\sum_{t\in T}(P^{pv}_{i,t} - \hat{P}^{pv}_{i,t})^2} \tag{3}$$

and then we normalize using the maximum value: $\frac{RMSE_{\hat{P}^{pv}_i}}{max(P^{pv}_i)}$.

The first step was to build a dataset for a profile. For simplicity, only one prosumer for the consumption profile on the grid is considered. The pure load profiles are from the *SmartMeter Energy Consumption Data in London Households*[1] during the year 2012. PV generation and exogenous data (typically temperature T and irradiance G) are also required, as they represent the most impacting parameters on the solar production. These were found on the *Photovoltaic Geographical Information System* (PVGIS)[2] database.

With these data, a first dataset (*case 1*) was built. To train and evaluate the model, a 10-fold cross-validation, which randomly splits the training and validation sets at different indices and then averages the results over the ten splits, was used. Through a grid search over a hyper-parameter space[3], the network architecture yielding the best results was selected, *i.e.*, the lowest RMSE. The code snippet below demonstrates how the model is instantiated with the selected hyper-parameters, trained on the training set using the 10-fold evaluation case, and finally, disaggregates $\hat{P}_{\substack{pv\\i,t}}$ and $\hat{P}^{load}_{i,t}$ using the holdback set.

```
model = D(
    # 1 layer, 31 units
    hidden_layer_sizes=31,
    learning_rate_init=0.001,
    activation='relu',
    max_iter=800, # epochs
    batch_size=32,
    solver='adam' # optimizer
)

model.train(
    X_train,
    y_train,
    kf=KFold(n_splits=10)
)
pred_pv, pred_load = model(X_hb, y_hb)
```

Initially, we had two scenarios: use samples over the whole year to train the model, with (*full-year*) or without the samples between 22:00 and 05:00 (*day-only*). Indeed, solar energy is not produced when the sun is down ($P^N = P^{load}$). In this case, the model is not forced to learn a nonexistent relationship between P^N and P^{pv} during these night hours. Then, we explored to what extent the size of the training set can be reduced. Six scenarios were added: *6-month* (every other month), *season* (one month per season), *2-month*, *1-month*, *2-week*, and *1-week*. The idea behind dataset reduction is that it is not practically feasible for DSOs to require a full-year of samples to deploy its ANN-based load disaggregation solution on their distribution grid. Knowing a reason-

[1] https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households

[2] https://re.jrc.ec.europa.eu/pvg_tools/en/#MR

[3] namely, the number of `hidden layers` and `units per layer`; the `learning rate`; the `optimizer`; number of `epochs`; `batch size`; and the `activation` function.

able sample size, DSOs can monitor P^{load} and P^{pv} from prosumers in an intrusive manner (ideally for the shortest time as possible), then having sufficient data, the disaggregation can be performed without any behind-the-meter sampling.

For scenarios in which we reduce the training/validation set, it is possible to build a holdback set to test the model on previously unseen data. For instance in the *season* scenario, the model trains on four months, and disaggregate the remaining eight months as holdback set. For the *one-month* scenario, we computed an aggregated metric of taking each month individually (train) against the others (holdback). A similar approach was used for the *one-week* scenario (1 *vs* 51). For the *two-week* scenario, we repeatedly took a week and another 6 months later (2 *vs* 50). The Scikit-learn[4] machine learning API (version 0.24.1) was used.

Upon our results from *Case 1* (discussed in the following section), displaying very good performances, it was decided to investigate more realistic datasets. Indeed, using complete, consistent, and sanitized data from the PVGIS database simplified the problem, as the production relies on an analytical model for the PV generator (indiscriminately from potential shading effect or equipment default). In a second step, actual measurements from the Grenoble INP-Ense3 building equipped with PV panels were considered. A new dataset was built, keeping the same pure load profile, but changing the PV, temperature and irradiance features. Using the same architecture and hyper-parameters as in *Case 1*, we trained and evaluated the performance with this new case (*Case 2*) exactly as in the first one.

Nevertheless, prosumers do not necessarily have a weather station next to their solar panels and even if it was the case, these data (irradiance, temperature) would not be available to the DSOs. Thus, a realistic scenario consists in accessing to weather information from regional agencies, to obtain temperature and irradiance data for Grenoble (from PVGIS), while the power measurement comes from actual solar installations (Ense[3] building in our case). Once again, with this new case (*Case 3*), the same model was trained and its performance evaluated.

3.1 Results

As briefly mentioned above, the disaggregation task with *Case 1* was rather easy, as shown by the results in Figure 2 below.

Two elements stand out: first, the RMSEs are low and we can reduce the dataset set considerably, until the *1-month* scenario. Interestingly, the *season* scenario yields very good results (1.16 % ($\hat{P}^{pv}$), 0.36 % ($\hat{P}^{load}$)), as if having one month for each season is enough to capture the seasonality effect on both consumption and solar power generation. Even if the *6-month* scenario trains on more samples, performance on the holdback are not better. Despite not having a holdback set for the two first scenarios, we see no improvements from *full-year* to *day-only* despite our hypothesis. We would have imputed this to the fewer sample size, but results in reduction scenarios invalidate this hypothesis.

[4]https://scikit-learn.org/stable/

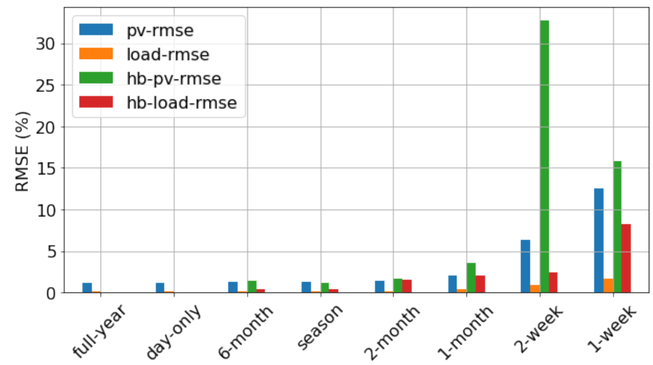

Figure 2: RMSEs for *Case 1*. The blue and orange bars represent the 10-fold test case; the green and red one, the holdback test case.

Results for *Case 2* are presented in Figure 3. Despite the RMSEs increasing, we can observe a very similar trend compare to *Case 1*. Once again, data reduction is not detrimental to performance; this time, the *6-month* scenario still yields the overall lowest RMSEs (4.5 % ($\hat{P}^{pv}$), 0.77 % ($\hat{P}^{load}$)).

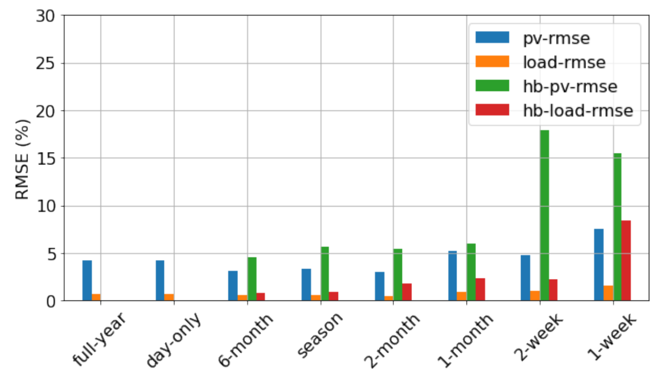

Figure 3: RMSEs for *Case 2*.

Finally, RMSEs for all scenarios have increased in *Case 3*, but less than between the first two cases (Figure 4). The general trend is once more the same, and the *6-month* scenario still holds the best performance (6.4 % ($\hat{P}^{pv}$), 1.09 % ($\hat{P}^{load}$)).

Figure 5 plots the disaggregated components' curves along their respective ground truth for a week in the holdback dataset. We notice a good match between the curves. Figure 6 presents the same curves for a week in December but with the *2-weeks* scenario.

3.2 Discussion

In their study, [Saeedi *et al.*, 2021] trained machine learning models on two datasets: a real one and a simulated one. They also devised different test-cases such as 90 %(train)-10 %(test), a one-month-out, and a one-season-out. Their best model was a Random Forest Regressor yielding a RMSE of 4 % on the real set and 2 % on the simulated one, while their multi-layer perceptron (MLP) model yielded respectively 8 %

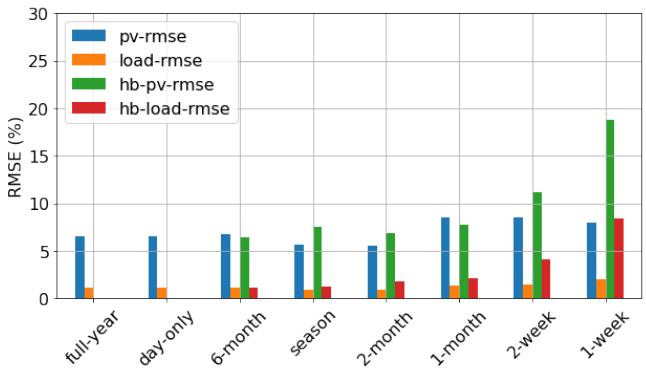

Figure 4: RMSEs for *Case 3*.

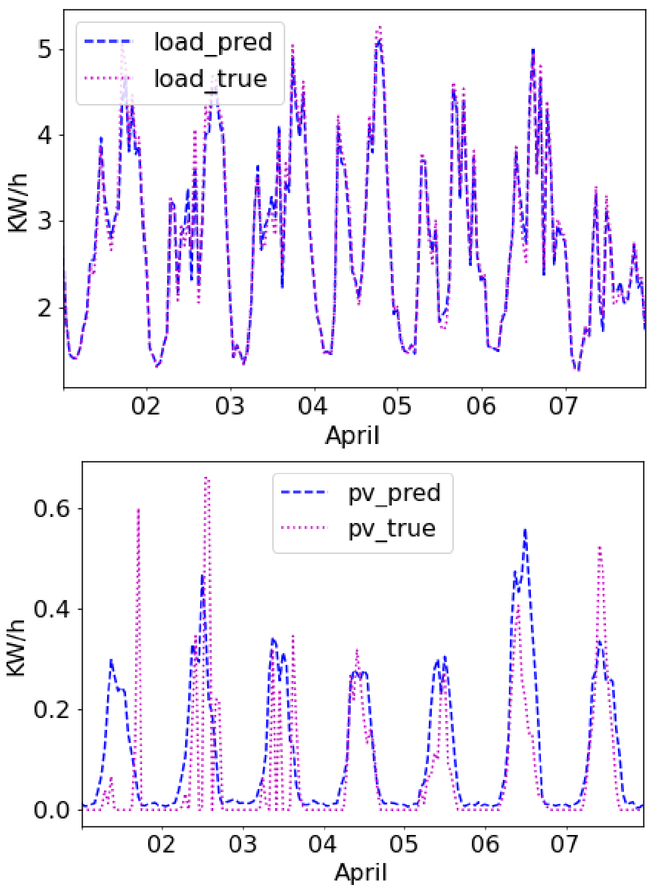

Figure 5: Disaggregated components on the holdback set for the *season* scenario (*Case 3*).

and 4 %. Unfortunately, they provide no link or access to their real dataset, so it is impossible for us to compare our results (and models) with theirs. As seen with our three cases, the dataset is a crucial factor regarding model performances.

The only scenario we have in common with [Saeedi *et al.*, 2021] is the 90 %(train)-10 %(test) (*i.e.*, our 10-fold) over the full year. For their simulated dataset (which is to a certain extent comparable to our *Case 1*) they obtained a RMSE of

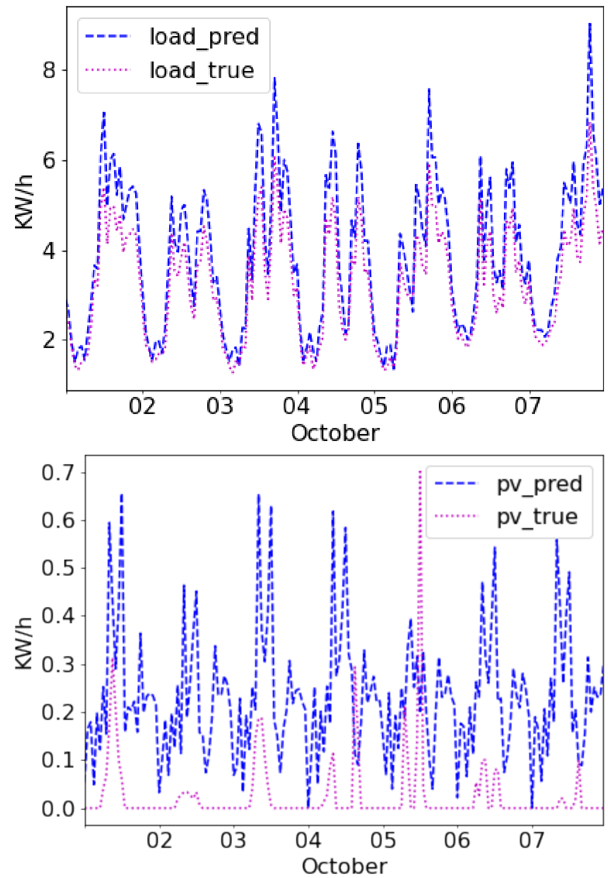

Figure 6: Disaggregated components on the holdback set for the *2-week* scenario (*Case 3*).

4 % with an MLP. In contrast, our similar model yielded a $\approx$ 0.68 % average mean. Even if we cannot conclude that our results a superior, it is interesting to note that, unlike theirs, our dataset was not subject to an extensive data cleaning and feature extraction process: the numerical features were normalized and the timestamp encoded.

4 Going further

As *Case 3* is considered the most realistic (spatially decoupled PV generation and exogenous data), we wanted to explore how its performance could be improved. A proposition made was to relax the assumption to disaggregate both components by finding respective values as close as possible to their ground truths. The idea was to produce a prediction interval wherein the real value would lie (*e.g.*, within a specific prediction interval of, for instance, 90 %).

Probabilistic forecasting was then considered. Instead of yielding a single value per timestep (*deterministic/point forecasting*), probabilistic forecasting models estimate a probability density function (PDF) rather than a single value, which allows for risk management [van der Meer *et al.*, 2018]. Many methods can be used to produce probabilistic forecasts, and ANNs are among them. For example, [Sharma *et al.*, 2021] used MLPs and LSTMs (long short-time memory) networks to perform solar energy forecasting. Nevertheless, to the extent of our research, no papers on PV/load disaggregation using probabilistic forecasting were published.

Gluon Time Series (GluonTS), is a library for deep-learning-based time series modeling developed by *Amazon Web Services*[Alexandrov *et al.*, 2019] for probabilistic forecasting. The API provides implemented models ready to be trained, like feed-forward neural networks or LSTMs. The general idea is to train the model on a dataset minus a fixed-length prediction window. Besides model-specific hyper-parameters to tune, we can also choose a *context-length* parameter to forecast the prediction window using the previous N timesteps.

As pointed out by [van der Meer *et al.*, 2018], probabilistic forecasts need to be assessed with specific performance metrics, since those formulated for deterministic point forecasts (*e.g.*, RMSE) can lead to invalid conclusions. From the literature, we retain two criteria: **reliability** and **sharpness**. The former validates the statistical consistency between the distributional forecasts and the observations. It is typically assessed by the **coverage rate** of the prediction interval (PI) using an indicator $I_t^{(i)}$:

$$I_t^{(i)} = \begin{cases} \hat{y}_t^i \in [\hat{L}_t^i, \hat{U}_t^i] \rightarrow \text{hit} \\ \hat{y}_t^i \notin [\hat{L}_t^i, \hat{U}_t^i] \rightarrow \text{miss} \end{cases} \tag{4}$$

where $\hat{L}_t^i$ and $\hat{U}_t^i$ are the lower and upper bounds (typically the 5^{th} and 95^{th} quantiles for a 90 % PI) for the series i at time t.

Given a sequence of $\{I_t^{(i)}\}_{t=1}^T$, where T is the set of timesteps included in the forecast range, we can compute the **prediction interval coverage probability** (PICP):

$$PICP = \frac{1}{N}\sum_{i,t} I_t^{(i)} \tag{5}$$

Sharpness measures the concentration of the predictive distributions, *i.e.*, the width of the PI. It is evaluated using the **weighted quantile loss** (wQL):

$$wQL_\tau(Y, \hat{Y}) = 2\frac{\sum_{i,t} \mathbb{P}_\tau(y_t^{(i)}, \hat{y}_t^{(i)})}{\sum_{i,t} |y_t^{(i)}|} \tag{6}$$

where $\tau \in [0, 1]$ and the quantile loss per time index is:

$$\mathbb{P}_\tau(y_t^{(i)}, \hat{y}_t^{(i)}) = \begin{cases} \tau(y_t^{(i)} - \hat{y}_t^{(i)}) & \text{if } y_t^{(i)} > \hat{y}_t^{(i)} \\ (1-\tau)(y_t^{(i)} - \hat{y}_t^{(i)}) & \text{otherwise} \end{cases} \tag{7}$$

We first tried to test a simplified architecture where only the solar generation is disaggregated, mostly for technical reasons, as the API requires familiarization and we have not yet managed to make models with two outputs. For our exploration, we trained on eleven months excluding September and forecasted one week in the latter month. The evaluation module of GluonTS returns a set of both point and probabilistic metrics, and we shall only present here the RMSE computed from the median forecast and the PICP and wQL for the 90 % prediction interval.

The first results using a one-layer neural network model were quite disappointing regarding sharpness: the PI were symmetrical along the x-axis, basically stating with a 90 % confidence that the PV generation can take any value $\in [-y, y]$ KW/h which is both impossible (negative generation) and uninformative. As GluonTS offers different deep learning models, such as LSTMs, we decided to test the latter. LSTMs are part of the Recurrent Neural Network (RNN) family, a type of network able to learn temporal dependencies in data and establish a correlation between the previous data points and the current data point in the training sequence [Sharma *et al.*, 2021]. We thought this could be particularly relevant in our case, as PV generation at time t is highly correlated with the generation at $t-1$. Unlike [Sharma *et al.*, 2021] who observed similar results with an MLP or a LSTM, the latter model yielded far more satisfying results in our case. This can be imputed to either implementation difference or more probably, difference in the dataset.

Regarding point-forecast metrics our LSTM-model resulted in a 9.8 % RMSE. As for probabilistic forecast metrics, the PICP was 97 % and wQL 0.16. Unfortunately, we were not able to build a holdback set with the remaining eight months, as the API requires a dataset with continuous timestamps. Figure 7 shows the probabilistic forecast for solar generation over a week in September.

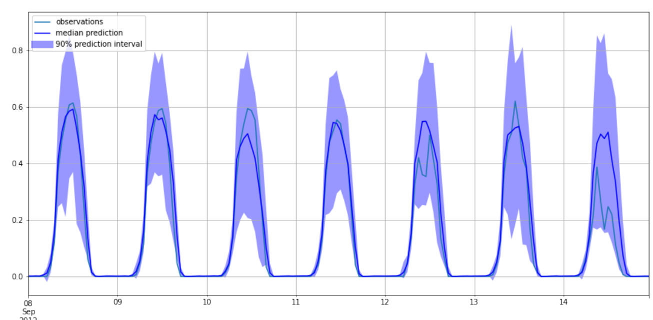

Figure 7: Prediction intervals and median point forecast over a week on the *season* scenario.

The GluonTS API is oriented towards time series forecasting and thus cannot be, to a certain extent, flexibly steered in the desired direction. As it is also recent, documentation and tutorials are limited and thus familiarization and adaption is time-consuming. Nevertheless, our short detour into probabilistic forecasting was enough to show that it is promising and further research should be made in that direction.

5 Conclusion

As the scientific literature around the usage of ANNs for net load disaggregation is still sparse, the main contribution of our research is to show that a fairly simple neural network architecture (*i.e.*, singe-layer neural network) with minimal preprocessing on real-world PV generation data and spatially decoupled exogenous data (*Case 3*) can yield relatively low RMSEs for both components. Our research also explores data reduction to probe how much data a DSO would need to obtain from a future prosumer in order to perform a reliable net load disaggregation. We found that having 6 months or one month per season was the best compromise between low error on the holdback set and reduced dataset.

Our research also briefly explored probabilistic forecasting as providing prediction intervals can be of great use in risk management and dealing with the uncertainty and variability inherent to solar power generation. We found that in this case, a simple neural network was not a complex enough model to yield informative prediction intervals. Turning to a more complex model, namely LSTMs, designed to deal with temporal sequences, we were able to obtain informative prediction intervals with good performance metrics. Nevertheless, we believe this path needs further research and ideally, an API tailored to our needs, inspired by GluonTS and based on Scikit-learn or Pytorch, could be developed for probabilistic load disaggregation.

Acknowledgements

First and foremost, I would like to thank Rémy Rigo-Mariani for his constant help and scientific support during this internship - even though not officially tied to the project. During this pandemic time, his physical presence at the lab was greatly appreciated. Of course, *mille mercis* to Vincent Debusschere for accepting and supervising my internship - his academic and scientific feedback were essential in framing the project. Finally, I am very grateful to Nouredine Hadj-Saïd for making this internship possible in his lab, it was (and still is) a fantastic learning opportunity as I was able to directly transfer freshly learnt content from my MoSIG courses to a practical context.

References

[Alexandrov *et al.*, 2019] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Sundar Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Turkmen, and Yuyang Wang. Gluonts: Probabilistic time series models in python. *CoRR*, abs/1906.05264, 2019.

[Bu *et al.*, 2020] Fankun Bu, Kaveh Dehghanpour, Yuxuan Yuan, Zhaoyu Wang, and Yingchen Zhang. A data-driven game-theoretic approach for behind-the-meter pv generation disaggregation. *IEEE Transactions on Power Systems*, 35(4):3133–3144, 2020.

[Petrusev *et al.*, 2021] Aleksandr Petrusev, Rebecca Bauer, Rémy Rigo-Mariani, Vincent Debusschere, Patrick Reignier, and Nouredine Hadjsaid. Automatic detection of distributed solar generation based on exogenous information. Manuscript submitted for publication, 2021.

[Saeedi *et al.*, 2021] Ramyar Saeedi, Sajan Kaduvettykunnal Sadanandan, Anurag Srivastava, Kevin Davies, and Assefaw Gebremedhin. An adaptive machine learning framework for behind-the-meter load/pv disaggregation. *IEEE Transactions on Industrial Informatics*, 2021.

[Sharma *et al.*, 2021] Vinayak Sharma, Jorge Angel Gonzalez Ordiano, Ralf Mikut, and Umit Cali. Probabilistic solar power forecasting: Long short-term memory network vs simpler approaches. *CoRR*, abs/2101.08236, 2021.

[Tabone *et al.*, 2018] 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*, page 43–52, New York, NY, USA, 2018. Association for Computing Machinery.

[van der Meer *et al.*, 2018] D.W. van der Meer, J. Widén, and J. Munkhammar. Review on probabilistic forecasting of photovoltaic power production and electricity consumption. *Renewable and Sustainable Energy Reviews*, 81, 2018.