Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

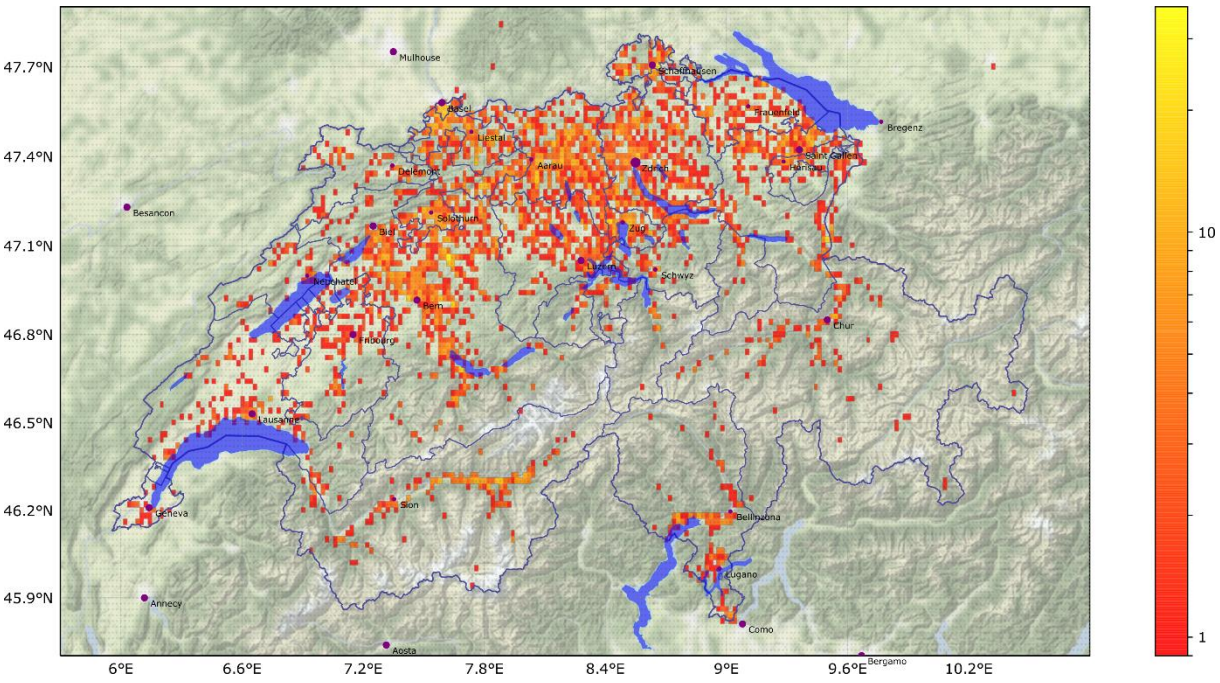
Department of the Environment,
Transport, Energy and Communication DETEC

Swiss Federal Office of Energy SFOE
Energy Research

Final report

SODA

Solar data analytics for production forecasting and anomaly detection



Date: 11.08.2020

Place: Bern

Publisher:
Swiss Federal Office of Energy SFOE
Research Programme PV and CSP
CH-3003 Bern
www.bfe.admin.ch
energieforschung@bfe.admin.ch

Agent:
CSEM
CH-2000 Neuchâtel
www.csem.ch

BKW AG
CH-3000 Bern
www.bkw.ch

Authors:
Pierre-Jean Alet, CSEM, pierre-jean.alet@csem.ch
Rafael Carrillo, CSEM, rafael.carrillo@csem.ch

SFOE programme manager: Stefan Oberholzer, stefan.oberholzer@bfe.admin.ch
SFOE contract number: SI/501803-01

Summary

The SODA project originates from the intuition that photovoltaic (PV) production data can indirectly provide weather maps from which future production can be forecast. However, gaps and noise in production data are unavoidable. Based on the outcomes of a first proof-of-concept project by BKW and CSEM, SODA aimed at demonstrating the applicability of big-data analytics to forecast the power generation of distributed PV systems over Switzerland. Its scope included both linear and non-linear methods, the latter being based on recurrent neural networks.

The proposed algorithms have been tested on entire years and on uniquely large datasets: more than 300 real PV systems spread over Switzerland, and 1000 synthetic ones that reproduce the statistical distribution of installed PV in the country in terms of size, orientation and location. Graph-based methods proved very powerful to reconstruct missing or faulty data. The NRMSE of this reconstruction is well below 20% for gaps of up to 4 h on the real dataset. As a result, the full forecasting pipeline proved very robust against faulty data.

All the developed methods outperform state-of-the art techniques which combine numerical weather prediction with machine learning (e.g., support vector regression) at least up to three hours ahead. Some of the most promising, non-linear methods such as graph convolutional neural networks, even outperform these state-of-the-art techniques for forecasts up to six hours ahead.

Zusammenfassung

Das SODA-Projekt entspringt der Intuition, dass Photovoltaik (PV)-Produktionsdaten indirekt die Wetterbedingungen abbilden und damit die zukünftige Produktion vorhersagen können. Lücken und Rauschen in den Produktionsdaten sind jedoch unvermeidlich. Basierend auf den Ergebnissen eines ersten "Proof-of-Concept"-Projekts der BKW und dem CSEM soll SODA die Anwendbarkeit der Grossdatenanalyse zur Vorhersage der Produktion von über das Schweizer Territorium verteilten PV-Anlagen demonstrieren. Zu diesem Zweck betrachtete es sowohl lineare als auch nicht-lineare Methoden, die auf rekurrenten neuronalen Netzen basieren.

Die vorgeschlagenen Algorithmen wurden an ganzen Jahren und an einzigartig grossen Datensätzen getestet: mehr als 300 reale PV-Anlagen, die über die Schweiz verteilt sind, und 1000 synthetische Systeme, deren Grösse, Ausrichtung und Standortverteilung die Statistiken des Landes reproduzieren. Graphenbasierte Methoden erwiesen sich als sehr leistungsfähig, um fehlende oder fehlerhafte Daten zu rekonstruieren. Der normalisierte mittlere quadratische Wurzelfehler (NRMSE) dieser Rekonstruktion liegt deutlich unter 20% für Lücken von bis zu vier Stunden, angewandt auf den realen Datensatz. Infolgedessen ist das gesamte Vorhersagesystem sehr robust gegenüber fehlerhaften Daten.

Alle entwickelten Methoden übertreffen für Vorhersagehorizonte von mindestens drei Stunden den Stand der Technik, die numerische Wettervorhersage mit maschinellem Lernen (z.B. Support-Vektor-Regression) kombinieren. Einige der vielversprechendsten, nichtlinearen Methoden, wie z.B. neuronale Netze mit Graphenfaltung, übertreffen sogar diese hochmodernen Techniken für Vorhersagen mit einem Vorlauf von bis zu sechs Stunden.

Résumé

L’origine du projet SODA est l’intuition que les données de production photovoltaïques (PV) permettent indirectement de cartographier les conditions météo et de là de prédire la production

future. Cependant des trous et du bruit dans ces données sont inévitables. Sur la base d'une preuve de concept réalisée par les BKW et le CSEM, SODA visait à démontrer l'applicabilité du traitement des données massives pour prévoir la production de systèmes PV distribués sur le territoire suisse. Pour cela il a envisagé des méthodes linéaires aussi bien que des méthodes non linéaires basées sur des réseaux de neurones récurrents.

Les algorithmes proposés ont été testés sur des années entières et sur des jeux de données exceptionnellement grands : plus de 300 systèmes PV réels distribués sur la Suisse et 1000 systèmes synthétiques dont la distribution de taille, orientation et emplacement reproduit la statistique dans le pays. Les méthodes à base de graphes se sont montrées très puissantes pour la reconstruction des données manquantes ou erronées. L'erreur quadratique moyenne normalisée de cette reconstruction est en-dessous de 20% pour des trous jusqu'à quatre heures de durée appliqués au jeu de données réel. Grâce à cela, le système de prévision complet est très robuste face aux données imparfaites.

Toutes les méthodes développées dans le projet ont atteint des performances au-delà de l'état de l'art, qui combine des prévisions météo numériques avec de l'apprentissage automatique (par exemple, régression à vecteur support), pour des horizons de prédiction d'au moins trois heures. Certaines des méthodes non-linéaires les plus prometteuses, par exemple les réseaux de neurones à convolution de graphes, battent même ces techniques de référence pour des prévisions jusqu'à six heures.

Seite absichtlich frei

Contents

List of abbreviations

AR	Auto-regressive
CNN	Convolutional neural network
CSEM	Centre suisse d’électronique et de microtechnique
FNN	Feedforward neural network
GCRNN	Graph convolutional recurrent neural network
GMM	Gaussian mixture model
GNN	Graph neural network
LASSO	Least absolute shrinkage and selection operator
LSTM	Long short-term memory
MAE	Mean absolute error
ML	Machine learning
NRMSE	Normalised root-mean-square error
NWP	Numerical weather prediction
PV	Photovoltaics
RMSE	Root-mean-square error
SDS	Solare Datensysteme
SFOE	Swiss Federal Office of Energy
ST-AR	Spatio-temporal auto-regressive

1 Introduction

Photovoltaic (PV) production will be central to Switzerland's energy transition as the largest contributor to the replacement of thermal power generation. Its distributed nature and its dependence on weather creates challenges for grid operations. However, its development coincides with an increasing digitalisation of the power sector. Monitoring equipment, PV inverters and smart meters can all communicate production data. However, the instrumentation of small, distributed PV systems cannot operate at the same standard as that of large power plants, and does not benefit from dedicated communication infrastructure. As a result, gaps and noise in production data are unavoidable.

The SODA project originates from the intuition that PV production data can indirectly provide weather maps from which future production can be forecast. It built on a first proof-of-concept project by BKW and CSEM. Based on the outcome of that project, SODA aimed at demonstrating the applicability of big-data analytics to forecast the power generation of distributed photovoltaic systems over the country. Its scope included both linear and non-linear methods, the latter being based on recurrent neural networks. A major scientific challenge was to make the forecasting algorithms robust against unavoidable faults or gaps in data and to quantify their performance in a reliable way.

The project ran from November 2018 to June 2020. This final report puts the research in perspective, presents the different classes of investigated algorithms and the resulting framework, and results of their systematic evaluation.

2 Context

2.1 Motivation of the project

Relevant data for PV production forecasting is already increasingly being generated through e.g., early smart meters and monitoring of PV systems such as the Solar-Log line of products offered by BKW's subsidiary Solare Datensysteme GmbH (SDS). Thousands of such systems are in operation in Switzerland, which provides a much higher density of sensors than can be provided by weather stations. In addition, support for smart metering and smart command and control systems is part of the new law implementing Switzerland's energy strategy 2050 [1]. However, these sources of data are currently separate and underused.

In 2017, CSEM demonstrated in collaboration with BKW a promising approach which uses monitoring PV data from SDS's Solar-Log products to deliver short- to mid-term (1 h to 24 h) forecasts of PV production by individual PV systems. Indeed, it yielded a single-site normalised RMSE below 30% for 6 h-ahead forecasts, which is already better than the state of the art. However, the results were only validated on a subset made of 29 systems spread over an area of about 200 km x 200 km in South-West Germany. These systems were selected because they are relatively close to each other and provide continuous data over the same period of more than 100 days. Scalability and robustness of these algorithms remained an open question, as was their combination with grid measurements to derive net consumption estimations and forecasts.

2.2 Background / State of the art

Production forecasting is a critical technology for enabling large-scale penetration of PV generation into the power grid [2]. PV power production is characterized by significant variability since it depends on meteorological conditions. Thus, most PV forecasting approaches are based on numerical weather

predictions (NWP) that in general have a limited spatial and temporal resolution, which poses a challenge for accurate production forecasting. Another challenge is data quality i.e., one needs clean and uninterrupted data to learn accurate prediction models. However, most real-life datasets are corrupted by noise and gaps in the measurements.

For the forecasting of single PV systems, the state of the art is the combination of multiple machine learning approaches with numerical weather forecasts as inputs [3]. These approaches yield a root-mean-square error (RMSE) of about 45% to 50% of the daytime average power of the system for day-ahead forecasts. Forecasts for grid operation purposes have focused on a regional level, at which they benefit from a strong smoothing effect to yield an RMSE from 21% (1 h ahead) to 31% (24 h ahead) of the daytime average power [4]. They estimate the current and near-future levels of regional PV power production using data from sampled systems, static information on all installed PV systems, numerical weather forecasts, and machine learning algorithms.

Many forecasting schemes have been proposed to improve forecasting accuracy. Since non-stationary characteristics of solar irradiance mainly come from cloud movement and its stochastic blocking of sunlight [5], the works in [6], [7] used cloud motion vector schemes using sky imagers. However, sky imagers are too costly to be deployed with all PV sites, and they are also only capable of predicting in a very short-term horizon i.e., less than an hour. The works in [8], [9] use satellite images for hourly PV output forecasting. However, wide-area satellite images are not capable of capturing site-specific information, thus they do not provide meaningful information for site-specific PV forecasting. The works in [10], [11] use forecasted cloud information to improve forecasting accuracy. However, these schemes face the same limitations as weather forecasting.

Most recently, several studies have used multi-site spatio-temporal historical data for PV forecasting without requiring exogenous cloud data [12]–[21]. For example, simple and fast multi-site forecasting techniques using autoregressive (AR) are used in [12], [13], and the work in [14] further develops the AR model into the local vector AR model considering local weather changes. However, nonlinear methods outperform the linear techniques for forecasting horizons of more than an hour [22], and recent studies have used machine learning techniques to predict multi-site PV output for hourly horizons. For example, feedforward neural network (FNN) [15] and long short-term memory (LSTM) networks [16], [23], [24] were proposed for multi-site spatio-temporal PV forecasting. LSTM networks perform well in time-series forecasting by processing sequence information using internal memory, thus the forecasting accuracy has been improved. In addition, spatial information can be used to trace the indirect cloud formation/movement, and recent studies exploit convolutional neural network (CNN) [18]–[20] to capture the complex spatial dependencies. As passing clouds certainly influence neighbouring PV sites sequentially, they can capture cloud cover and cloud movement by considering spatial relations with improved forecasting performance.

Regarding data quality, validation and cleaning of data from grid sensors is a prerequisite for any data-driven solution, which may be affected by measurement or transmission errors. Published solutions for PV rely on the knowledge of characteristics of the systems [25] which themselves may be faulty or missing. The work of [21] proposed a processing pipeline that takes into account data cleaning and treatment of missing data. They propose to fill the missing gaps by finding a representative PV signal from the same geographical area and replacing the missing segment by the representative signal.

2.3 Goals

The project aims at demonstrating the applicability of big-data analytics on the production data of distributed photovoltaic systems to provide power generation forecasts and to identify technical issues on these systems.

The main research questions that this project addressed were:

- How to make big-data PV forecasting and nowcasting algorithms robust against noisy data?
- How to make big-data PV forecasting and nowcasting algorithms robust against randomly missing data?
- What is the optimal way of sampling neighbours to be considered as inputs for the now/forecasting algorithms?
- How do regression algorithms and recurrent neural networks compare in terms of accuracy and scalability?

The results of the project are a series of machine learning algorithms combined in a data processing chain. Raw measurements and reference data (e.g., location) feed this chain to generate spatially, temporally (to 15-min) resolved PV production forecasts.

3 Approach and methodology

The main aim of the project is to investigate data-driven methods to forecast PV production over a large area with fine temporal resolution using only past production data. To achieve this goal we resort to graph-based machine learning (ML) and signal processing methods [26] to model the spatio-temporal correlations of the production data. One of the main challenges in forecasting is data quality i.e., we need clean and uninterrupted data to develop/learn accurate prediction models. However, most of real-life datasets are corrupted by noise and gaps in the measurements. Thus, the development of robust solutions to address noisy and incomplete data is of paramount importance.

Figure 1 depicts a block diagram of the concept developed within SODA. The gap reconstruction and pre-processing module fills the gaps from missing data as well, denoises it, and normalizes and de-trends it. The output of this module is in the correct form for the learning or the forecast modules. The learning module learns the graph models used for forecasting from historical data. Finally, the forecast module provides a forecast of the PV production over the entire area based on the learnt models and the cleaned data.

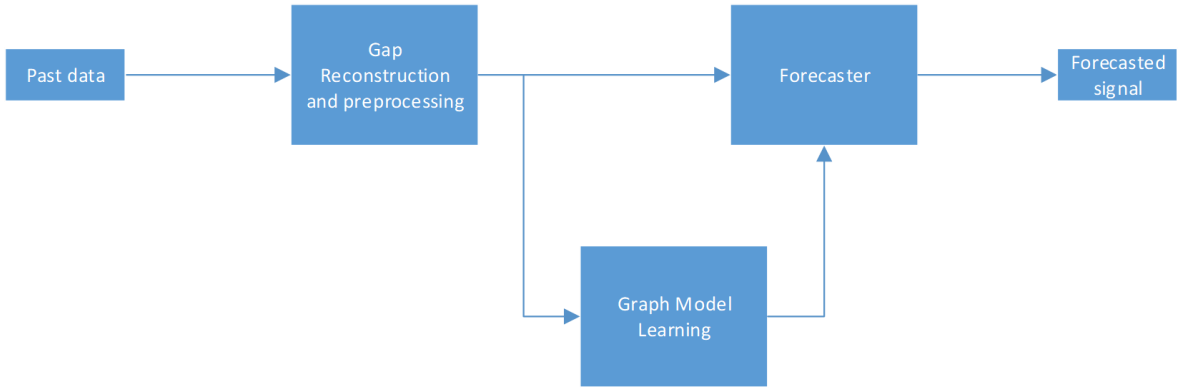


Figure 1. Block diagram of the proposed approach for robust PV production forecasting.

In this section, we detail the approach and methodology used to answer the research questions posed in this project. First, we describe the development of a database based on real and synthetic datasets to validate the different proposed approaches in a controlled manner. Second, we present a graph-

based algorithm to reconstruct missing data. Finally, we detail the graph-based forecasting methods developed in the project.

3.1 Databases

Since the project focuses on core algorithmic questions, we worked with offline PV production data extracted from the dataset of BKW's subsidiary Solare Datensysteme GmbH (SDS), dubbed "real database" in the rest of the report, as well as a synthetic database generated within the project in order to validate the different hypothesis in a controlled manner.

3.1.1 Real production database

The "real" dataset consists in PV production data across Switzerland recorded by Solar-Log devices between 2014 and 2017. As of 2017 about 10'000 plants were included in the dataset. Prior to their transfer, these data were anonymized: any field containing personal information were excluded and spatial coordinates were rounded to 0.01 degrees. The latter approximation provides a good balance between privacy (most "pixels" defined by this resolution will either contain no plants – as in mountain areas – or several) and accuracy.

The dataset was stored in an influxDB database, and python routines for data wrangling were developed. Since a considerable part of the metadata (e.g., plant size or location) is not reliable and the production data contains gaps (missing or null values) for periods up to weeks, the need of a synthetic database with clean and continuous data arose.

3.1.2 Synthetic production database

We needed a synthetic database which resembles the real database in terms of location, size, orientation and angle of the PV plants. To achieve this goal, we generated a dataset that matches the statistical distribution of these properties. The modelling chain is based on the PV-LIB python library [27] that computes the power production of a PV plant based on system parameters and weather data such as irradiance, wind speed and air temperature. This database was generated in four steps.

The first step was to generate locations for the PV plants using a Gaussian mixture model (GMM) with 80 components fitted on the locations of the nodes (PV plants) in the real database. From this GMM we sampled the locations of the synthetic database. The second step was to infer the distribution of the different characteristics of the panels (e.g., size, orientation pitch angle) to have models to sample from these parameters (see Figure 2 for a comparison between the distributions of real database and

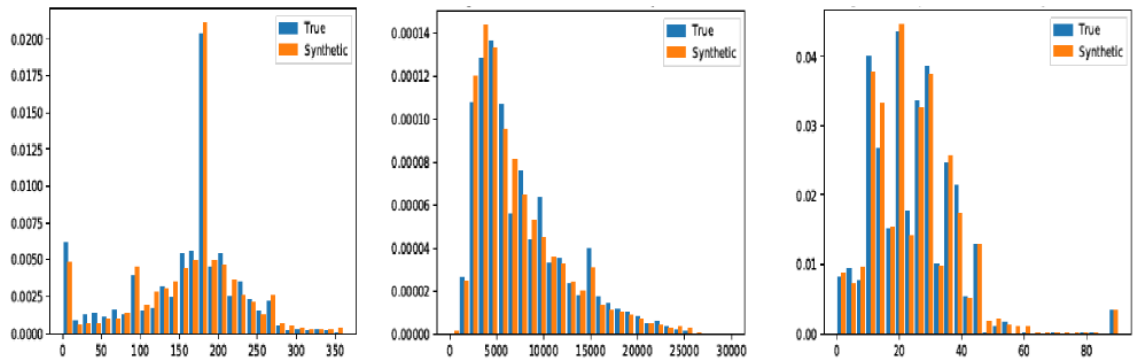


Figure 2. Histogram of the true and synthetic distribution of the main parameters of the PV systems. From left to right: orientation, size and angle.

synthetic database parameters). The third step to generate the synthetic power production was to input weather data for each generated location. To do so, we acquired a historical weather dataset from ARMINES (Mines ParisTech) with the following characteristics:

- continuous data from 2016 to 2018 with 15 minutes temporal resolution;
- spatial resolution of 1 km;
- with altitude compensation (temperature) and horizon shading (irradiance);
- global, diffuse and direct horizon irradiance, air temperature, wind speed and wind direction.

Finally, using PV-LIB we synthesized the production for all sampled locations from 2016 to 2018. We generated 10,000 nodes distributed across Switzerland. Figure 3 shows a comparison of the generated location and the real locations. Figure 4 shows the production of two sites in the real and synthetic database within the same location.

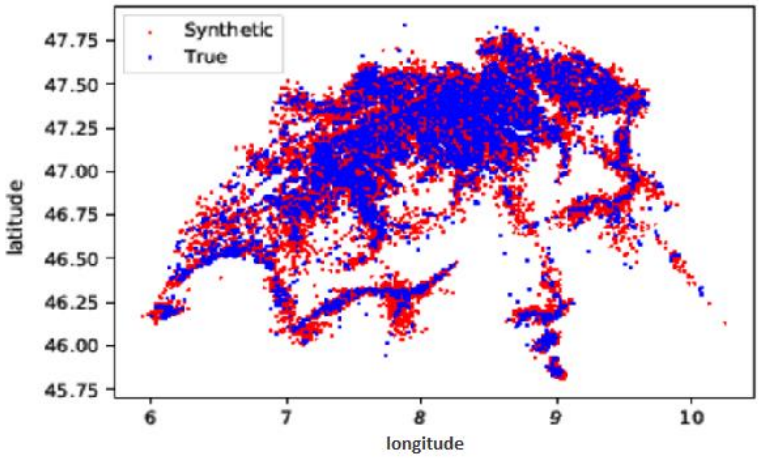


Figure 3. Locations of the PV plants in the real (blue) and synthetic (red) databases.

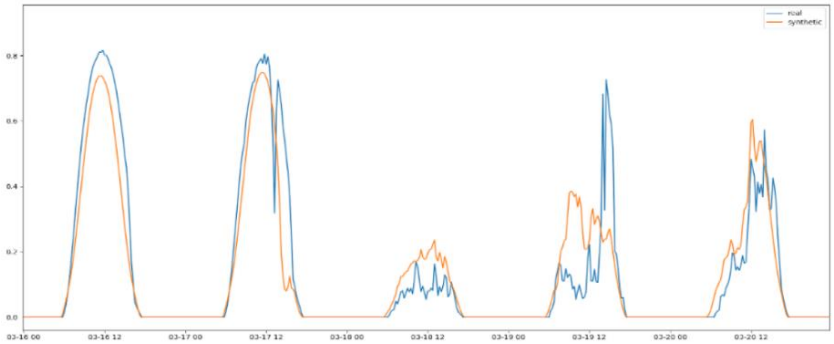


Figure 4. PV power production (normalized) for two plants within the same location in the real database (blue) and the synthetic database (orange).

3.2 Graph signal processing preliminaries

Let us begin with some basic definitions on graphs. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ denote an *undirected weighted graph* where: $\mathcal{V}$ denotes the set of vertices or *nodes*, $|\mathcal{V}| = N$, $\mathcal{E}$ denotes the set of *edges* or links, A denotes the weighted *adjacency matrix* (symmetric $N \times N$), where $A_{ij} > 0$ if nodes i and j are connected ($i \neq j$). Let $L \in \mathbb{R}^{N \times N}$ be the *graph Laplacian* matrix associated to the graph $\mathcal{G}$, defined as $L = D - A$, where D is the degree matrix of the graph, defined as $D_{ii} = \sum_j A_{ij}$ and D_{ij}=0 outside the main diagonal. The graph Laplacian can be interpreted as a difference operator for signals defined on the graph and its eigenvectors (and corresponding eigenvalues) define many properties of the graph.

We define a *graph signal* $x : V \rightarrow \mathbb{R}^N$, defined on the vertices of graph, such that x_i represents the signal's value at node i. Let $U = [u_0, u_1, u_2, \ldots u_{N-1}]$ be the matrix containing the eigenvectors of L as columns with associated eigenvalues $\{\lambda_0, \lambda_1, \lambda_2, \ldots \lambda_{n-1}\}$. The *graph Fourier transform* of the signal x is defined as $\hat{x} = U^T x$ and its domain (frequency) is defined by Λ, the diagonal matrix containing the eigenvalues. Based on this definition, the *spectral graph convolution* with a graph filter h can be defined as: $x * h = U h(\Lambda)\, U^T x$, where $h(\Lambda)$ denotes the diagonal matrix containing the graph Fourier transform of filter h (see [26] for more details on graph signal processing methods). Two drawbacks of spectral convolution are that it is not localized in space and it has a high computational load. For applications where fast and localized graph convolution is needed, polynomial approximations have been proposed such as those based on the Chebyshev polynomials [28].

Graph neural networks (GNN) are neural networks that leverage the graph structure of the input signals. Several types of GNNs are being investigated in the literature, such as recurrent graph neural networks, convolutional graph neural networks, graph autoencoders and spatio-temporal graph neural networks [29]. Within the project we have investigated several GNN architectures for short term forecasting of multi-site PV production. They will be described in section 3.4.

3.3 Graph-based data reconstruction: filling the gaps

Our proposed approach to fill the gaps in PV power production time series is inspired by the work in [30]. The method is based on building a graph model to capture the spatial dependencies among the PV systems and exploit the spatio-temporal relations to reconstruct the missing parts of the data. The main assumption of our method is that the normalized production data is smooth in the temporal as well as in the spatial domains. We construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $|\mathcal{V}| = N$ is the number of PV systems and the adjacency matrix A is created by placing edges for the 10 nearest neighbors of each plant and the weights are computed as a function of the distance between plants using a Gaussian kernel. Figure 5 shows an example of such a graph.

Each node has an associated time-series signal representing the power production over some time interval i.e., a batch of data, sampled at M consecutive time instances at a regular sampling time. We can model this data batch as the matrix $X \in \mathbb{R}^{N \times M}$, where each row represents the time-series data associated to a given node. All the time series are normalized by the peak production such that the maximum in each row (node) is one.

Let $L \in \mathbb{R}^{N \times N}$ be the graph Laplacian matrix associated to the graph $\mathcal{G}$ and $G \in \mathbb{R}^{M \times (M-1)}$ denotes the temporal difference operator such that every row of $Z = XG$ contains the time difference signal for each node. Let z_i denotes the i-th column of Z, thus we can define the graph-signal smoothness function as:

$$f(z) = \sum_{i=1}^{M-1} z_i^T L z_i = tr[Z^T L Z].$$

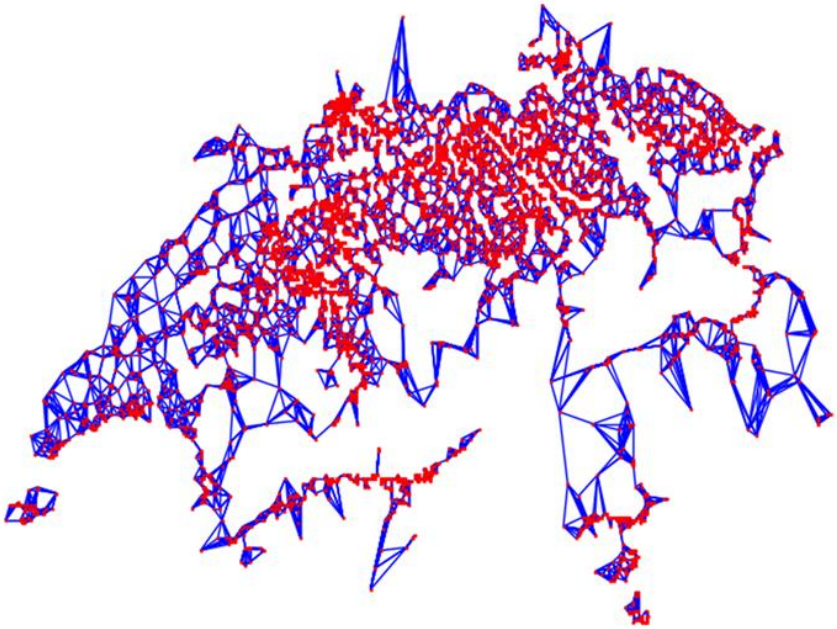


Figure 5. Example of a graph constructed using the locations (coordinates) of the monitored PV plants. The edges are selected using the 10 nearest neighbouring plants.

The corrupted signal with gaps can be modeled as:

$$Y = S \circ X_o + S \circ W,$$

where $S \in \{0,1\}^{N \times M}$ is a binary sampling operator that models the gaps in the signal, X_o denotes the original complete clean signal, W is a realization of white additive Gaussian noise and $\circ$ denotes the Hadamard product operator.

To reconstruct the signal X_o from the measured signal Y, we solve the following convex problem:

$$\min_X tr[(XG)^T LXG] \ \ subject\ to\ \|S \circ X - Y\|_F \leq \epsilon,$$

where $\|\cdot\|_F$ denotes the Frobenius (or L2) norm for matrices. The problem finds the smoothest graph time-series signal that is consistent with the available data. The constant ϵ defines the radius of the data fidelity constraint that can be related to the measurement noise level i.e., the closer ϵ is to zero the more accurate we assume our measurements are. In case $\epsilon = 0$, we assume our measurements are noise free. Within the project we developed an efficient algorithm to solve the convex problem. It is implemented in python and is capable of scaling to large networks of tens of thousands of nodes.

3.4 Forecast methods

The developed forecast methods are based on the assumption that the current power production in a system can be modelled as a function of the past production data of a subset of nodes over a predefined time interval. The rationale behind this model is that events measured in the past production of some nodes e.g., clouds or storms passing by, are informative to predict the production in other nodes. In the project we developed one linear and three non-linear forecasting methods which are described in the four next subsections.

3.4.1 Spatio-temporal auto-regressive forecast model

The first forecasting method developed is a linear spatio-temporal auto-regressive (ST-AR) method that models the prediction of the production at a particular site as a linear combination of past production from all sites in the network. Formally, the production at site $x \in \mathcal{N}$, where $\mathcal{N}$ denotes the set of nodes (sites), at time t follows the relation:

$$P_t^x = \beta_0 + \sum_{y \in \mathcal{N}} \sum_{i=1}^{L} \beta_i^y P_{t-i}^y + \epsilon_x$$

where P_q^z denotes the production in site z at time q, L is the past history horizon (in discrete samples) i.e., the order of the model or the number of past samples we keep in model, ϵ_x is the error term, and β_i^y is the model coefficient for lag i and site y. We assume that all time series are sampled at the same rate and at the same time.

Since the power production data is not stationary and has a strong dependency on time, both daily and seasonally, a proper normalization and de-trending of the data becomes a key step to use a linear model. The normalization method we chose is based on a data-driven computation of the clear sky production based on historical data.

For each node, annual sunrise and sunset profiles were computed as well as the clear sky production profile i.e., the maximal production profile. Figure 6 shows an example of the aforementioned profiles. Each clear sky production profile is adapted to the day's sun time based on the annual sunrise and sunset profiles. After the daily clear sky production profile is computed, the production data is normalized by dividing the measured data by the computed clear sky profile. To address the problem of normalization during night times, when the production is zero, we fill the values with the mean value of the production of the previous day. Figure 7 shows an example of a normalized signal. As a result of the normalization step, the daily and seasonal trends are removed from the time series such that the linear models can learn the shadowing effects due to weather, mountains or nearby constructions.

The production is modelled as a linear combination of the L past production values for all nodes in $\mathcal{N}$ i.e., all nodes available. To select the most informative coefficients, we proposed to use the group LASSO (Least Absolute Shrinkage and Selection Operator) estimator [31].

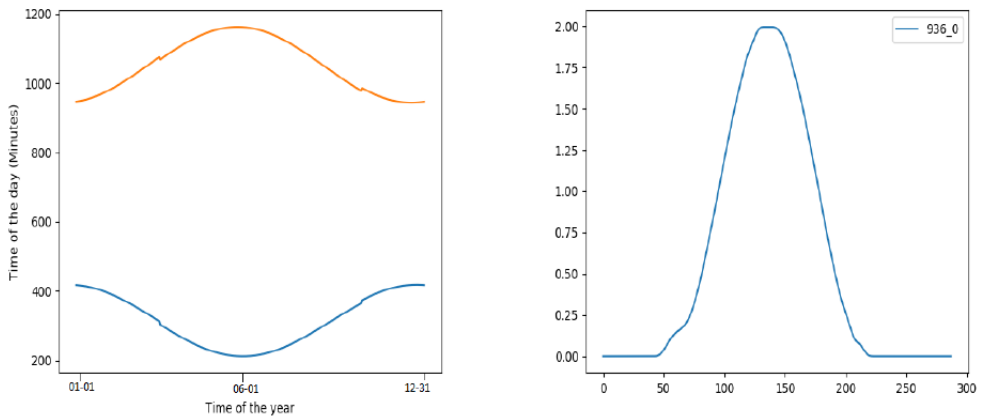


Figure 6. Example of computed profiles for one node. Left: yearly sunrise (blue) and sunset (orange) profiles. Left: Clear sky production profile.

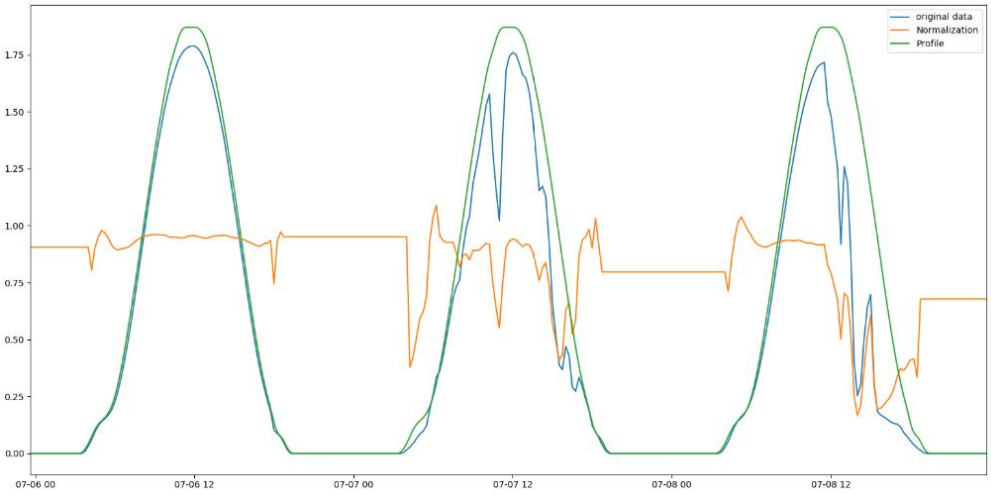


Figure 7. Normalization example. Original signal (blue), clear sky profile (green) and normalized signal (orange).

The group LASSO adds a regularization term to the classical least squares problem to promote group sparse solutions i.e., it will drive entire groups of coefficients belonging to the same node to zero. The group LASSO estimator solves the following convex problem to learn the model coefficients β_i^y from T measurements:

$$\hat{\beta} = \arg\min_{\beta} \|P^x - X\beta\|_2^2 + \lambda \cdot \sum_{y \in \mathcal{N}} \|\beta^y\|_2 ,$$

where λ is a regularization parameter that controls the sparsity of the solution i.e., the number of nodes selected in the model, and β^y denotes the subvector of coefficients associated to node y. In this vector form, the vector $\beta \in \mathbb{R}^{LN+1}$, where N is the number of sites (or nodes) in $\mathcal{N}$, is formed by grouping all β_i^y that belongs to the same node y i.e.,

$$\beta = \left[\beta_0, \beta_1^{x_1}, \ldots, \beta_L^{x_1}, \beta_1^{x_2}, \ldots, \beta_L^{x_2}, \ldots, \beta_1^{x_N}, \ldots, \beta_L^{x_N}\right]^T,$$

the measurement vector is defined as

$$P^x = [P_1^x, P_2^x, \ldots, P_T^x]^T$$

and the regressor or design matrix X is defined as

$$X = \begin{bmatrix} 1 & P_0^{x_1} & \cdots & P_{1-L}^{x_1} & \cdots & P_0^{x_N} & \cdots & P_{1-L}^{x_N} \\ 1 & P_1^{x_1} & \cdots & P_{2-L}^{x_1} & \cdots & P_1^{x_N} & \cdots & P_{2-L}^{x_N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & P_T^{x_1} & \cdots & P_{T-L}^{x_1} & \cdots & P_T^{x_N} & \cdots & P_{T-L}^{x_N} \end{bmatrix}.$$

To learn the model coefficients for all nodes, we need to solve the group LASSO problem for all nodes in $\mathcal{N}$. By doing so, the problem can be interpreted as a graph learning problem (see [32]) where we learn the topology of a directed graph that minimizes the prediction error. Figure 8 shows an example of a set of edges obtained for a central node in Switzerland.

Once the graph model is learnt, we use the model coefficients to forecast the production for all nodes in $\mathcal{N}$ (or for a selected group of nodes) for a horizon of H time steps ahead. To do so, we first predict

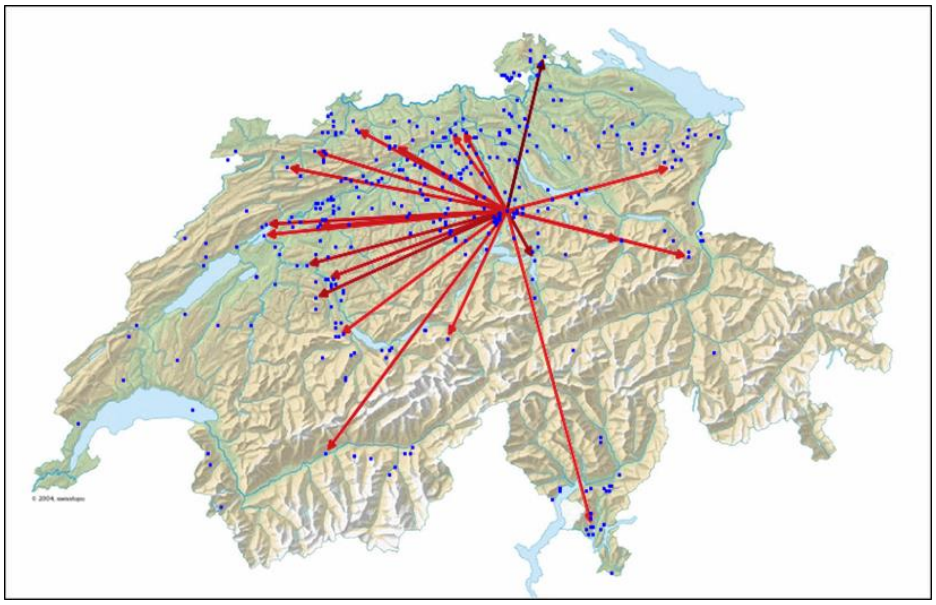


Figure 8. Example of a set of edges obtained by the group LASSO approach for a node in central Switzerland.

the production for one time step ahead as:

$$\hat{P}^x_{t+1} = \beta_0 + \sum_{y \in \mathcal{X}} \sum_{i=0}^{L-1} \beta_i^y P^y_{t-1}, \forall x \in \mathcal{N}.$$

Predictions for h time steps ahead, $h \in [2, \dots, H]$, use the past measured data as well as the predictions at previous times. The predictions are computed with the following relation:

$$\hat{P}^x_{t+h} = \beta_0 + \sum_{y \in \mathcal{X}} \sum_{i=1}^{h-1} \beta_i^y \hat{P}^y_{t+h-i} + \sum_{y \in \mathcal{X}} \sum_{i=h}^{L-1} \beta_i^y P^y_{t+h-i}, \forall x \in \mathcal{N}.$$

Since the model coefficients were learnt using normalized data, the same normalization is applied to the input data to produce normalized predictions. After the predictions are computed a de-normalization step is needed i.e., multiplication by the daily profiles for each node. In our setting we use 3 hours of past data samples i.e., $L = 12$, to forecast the production over a horizon of 6 hours ahead with a temporal resolution of 15 minutes i.e., $H = 24$.

We use the ST-AR model in a sliding window approach: we use two months of historical data for training the model to perform the forecasts for the next two weeks i.e., the model is updated every two weeks.

3.4.2 Graph convolutional recurrent neural networks

The second forecast method investigated is based on a graph convolutional recurrent neural network (GCRNN) encoder-decoder architecture. The type of RNN used is a long-short term memory (LSTM) network. LSTM have been designed to handle both short- and long-term dependencies in sequential data, with a gating mechanism that protects the cell state and update it only if needed [33]. In our setting, the encoder is an LSTM network that acts as a Kalman filter to estimate the state of the system, given a sequence of past observations $y_{t_0,t_i} = (y(t_0), \dots, y(t_i))$. The decoder is another LSTM

that takes as initial state the state produced by the encoder and that predicts the power for the chosen horizon period, see Figure 9. In our setting we use 6 hours of past data samples to forecast the production over a horizon of 6 hours ahead (with a temporal resolution of 15 minutes).

In classical LSTMs, the cell state c(t) and the output h(t) are updated recursively from an input sequence $x_{t_0,t_i} = (x(t_0), \dots, x(t_i))$ following the equations:

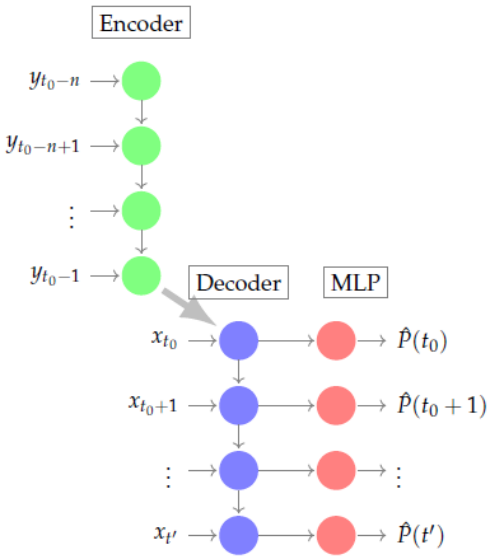


Figure 9. Encoder-decoder architecture. A multi-layer perceptron (MLP) layer is used at the output of the decoder to transform the LSTM outputs into the desired power production.

$$
\begin{aligned}
f(t) &= \sigma(W_f \cdot [h(t-1), x(t)] + b_f) \\
i(t) &= \sigma(W_i \cdot [h(t-1), x(t)] + b_i) \\
c(t) &= f(t) * c(t-1) + i(t) * \tanh(W_c \cdot [h(t-1), x(t)] + b_c) \\
h(t) &= \sigma(W_o \cdot [h(t-1), x(t)] + b_o) * \tanh(c(t))
\end{aligned}
$$

where σ is the sigmoid function and the weights W_f, W_i, W_c, W_o and biases b_f, b_i, b_c, b_o are learned in the course of training with stochastic gradient descent to minimize the reconstruction loss function. In the above series of equations, $*$ denotes the Hadamard product. At time t_0, the state $c(t)$ and output $h(t)$are either initialized to zero (for the encoder) or to the last state of the encoding sequence (for the decoder). Notice that $c(t), h(t) \in \mathbb{R}^{F \times N}$, where F is the number of features, thus carrying the full matrix multiplication in the node space would necessitate a huge amount of memory for large grids with a large number of nodes. In order to propagate the signal from nodes to nodes more efficiently, we modified the classical LSTM and replaced the dot product operations with graph convolutional layers with Chebyshev polynomial approximations as in [28]. The convolutional layers have learnable parameters that are included in the training loss function. In order to propagate the information using graph convolutional layers, we defined a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where the adjacency matrix A is created by using the K nearest neighbors principle and the weights are learned also using backpropagation and stochastic gradient descent.

3.4.3 Graph spatio-temporal attention networks

The second architecture studied in the project is based on graph attention networks [34] to jointly find spatial and temporal correlations of the PV production data. We will define the problem of the future PV production forecasting by taking the observed past L data points (or lags) and predicting H steps

ahead. Considering that the resolution of the PV data is 15 minutes, we will take one day of past PV data thus the number of past data points is $L = 96$ and the forecast horizon of six hours ahead i.e., $H = 24$.

In this setting, we assume that this graph is fully connected. To extract features, we define an overlapping window mechanism, where the input data is split into fifteen overlapping windows. Thus, each window has length of 12 temporal data points and the next window is overlapping with the previous one by 50 percent (Figure 10).

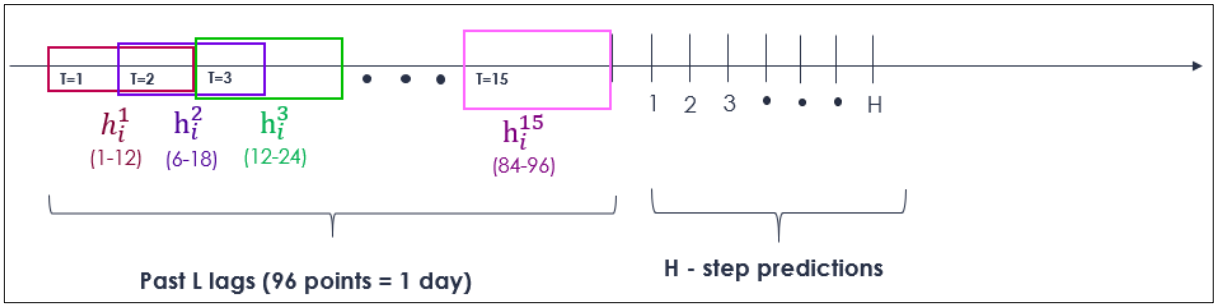


Figure 10. Sliding window approach to divide the observed temporal horizon (past L samples).

First, spatial correlations are modelled by a spatial attention mechanism. Then the extracted features are passed to the temporal attention network. In the following we describe the spatial and temporal mechanisms.

We form feature vectors $h_i^k \in \mathbb{R}^F$, where F is the number of features, for the node i at the time window k. In this spatio-temporal setting we first compute the spatial features for each node and time window as:

$$h'_{i,k} = \sigma(\sum_{j \in N_i} \alpha_{ij}^k \, W_{new} h_j^k) \, ,$$

where the spatial correlation between nodes i and j is embedded as the matrix element α_{ij}. W_{new} is a weight matrix. The spatial attention matrix α is calculated as the output of the following attention mechanism:

$$\alpha_{ij}^k = softmax(LeakyRelu\big(W_{s1} h_i^k + W_{s2} h_j^k\big),$$

where W_{s1} and W_{s2} are learnable weight matrices. The softmax function is used to normalize the coefficients and LeakyReLu denotes the leaky linear rectifier activation function.

Next, the spatial features $h_{i,k}$ are fed to the temporal attention network that determines which time intervals are most relevant for the prediction. Thus, the output of the spatio-temporal graph attention network is:

$$y_{it} = \sum_{k=1}^{T} \gamma_{tk} \, Q \; \sigma(\sum_{j \in N_i} \alpha_{ij}^k \, W_{new} h_j^k) \, .$$

The matrix γ_{tk} is the temporal relation between time windows t and k computed as the output of the following attention mechanism:

$$\gamma_{tk} = softmax(LeakyRelu\big(W_{temp1} h'_{it} + W_{temp1} h'_{ik}\big),$$

where W_{temp1} and W_{temp2} are learnable weight matrices. The matrix Q is a single-layer weight. To predict H steps ahead, we added a multi-layer perceptron layer (MLP) on top of this:

$$\hat{y}_{fin} = W_{out}^H * Relu\big(W_{mlp} y + b_{mlp}\big),$$

where $\hat{y}_{fin}$ is final output of our forecaster i.e., the sequence of H time steps forecasts for all nodes and y is the matrix of spatio-temporal features y_{it}. W_{mlp}, W_{out}^{H} and b_{mlp} are learnable weights in the MLP layer. We dubbed this architecture as graph attention spatio-temporal-geo network (ST-geo).

We also explored variants of the ST-geo architecture. The first variant, dubbed ST-geo-knn, changes slightly the architecture of the spatio-temporal attention network by restricting the support of the spatial attention matrices α^k to only include the K nearest neighbours for each node. In practice we set $K = 0.3N$ i.e., around 30% of the number of nodes in the network. The next explored architectures invert the order of the temporal and spatial attention mechanisms i.e., it computes first the temporal features and then the spatial features. The two architectures are dubbed temporal-spatial (TS) network and TS-geo-knn network. The first one only uses past production data as input features while the second one adds geographical information to the input features and restricts the support of the spatial attention matrices to only K nearest neighbours of each node. The temporal-spatial architectures scale better in comparison to the spatio-temporal, since the input attention matrices have a squared dependence on the number of time windows $(N\,T^2)$ as opposed to the squared dependency on the number of nodes $(T\,N^2)$ that the spatio-temporal have.

3.4.4 Graph convolutional transformer

The last architecture is based on the transformer architecture and exploits the multihead attention mechanism from transformers [34]. The proposed architecture has a similar encoder-decoder structure as the GCRNN encoder-decoder, described in section 3.4.2, and takes the same inputs and outputs for the encoding and decoding sequences. However, internal operations differ as there are no cell states keeping memory as for LSTM and the model uses the dot product attention from transformers.

Let us denote by $y_{t_0,t_i} = (y(t_0), \dots, y(t_i))$ the encoder input sequence, and by $x_{t_{i+1},t_f} = (x(t_{i+1}), \dots, x(t_f))$ the decoder input sequence. The encoder architecture is depicted in Figure 11.

In the encoder, the following operations take place:

- The input sequence y_{t_0,t_i} is duplicated three times, to form the query, key and value sequences, y_Q, y_K and y_V.
- A 1D convolution is applied to the sequences y_Q, y_K and y_V along the time axis. The weight convolutions are distinct for queries, keys and values (three independent 1D-convolution). The purpose of the 1D convolution is to extract meaningful causal information from each node time signal. The convolution weights are the same for all nodes.
- A dot-product attention with graph-convolutional layers is applied to the outputs of the 1D convolutional layers. Graph convolutions are applied after the temporal (1D) convolutions to mix node information to have both temporal and spatial information mixed. The adjacency matrix for the graph convolutions are learned for all heads independently. The weights of the attention mechanism are node independent.
- As a final step of the encoder, the different heads are concatenated and passed to a linear layer to produce the output sequence y'.

The decoder operations are similar t the encoder and are depicted on Figure 12. The main changes are that no graph convolution operation takes place for queries, keys and values. Moreover, time (1D) convolutions are only made for queries and keys. The query is the signal x_{t_{i+1},t_f}, whereas keys and values are the duplicated encoder outputs. Finally, the input signal is also multiplied by the output of the attention heads.

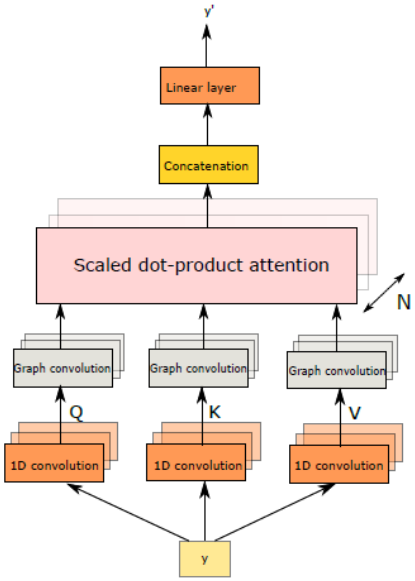


Figure 12. Encoder architecture.

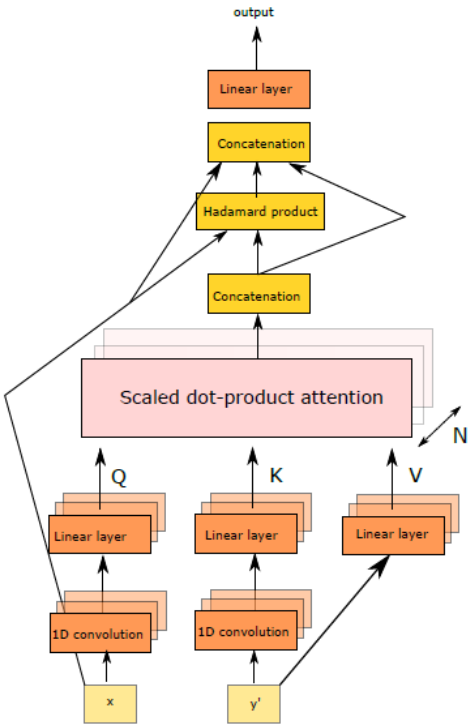


Figure 11. Decoder architecture.

In our setting we use 6 hours of past data samples to forecast the production in a horizon of 6 hours ahead (with temporal resolution of 15 minutes).

4 Results

This section illustrates the performance of the robust framework proposed for multi-site PV power production forecasting. To evaluate our methods, we selected a clean sample of 303 nodes from the real dataset, without gaps longer than 30 minutes, and, for a greater-scale evaluation, a sample of 1,000 nodes from the synthetic dataset. All sites have uninterrupted data for 2016 and 2017 with a temporal resolution of 15 minutes. Figure 13 and Figure 14 show the spatial distribution of the synthetic and real datasets, respectively. Colours indicate the peak production of each plant.

The non-linear models (GCRNN, ST attention networks and GCTransformer) were trained with observed data for 2016 and evaluated over the full year of 2017. In the case of the linear model (ST-AR or lasso), we divided the test year (2017) into 24 test periods (batches) of two weeks each. The prediction models were trained with data from the two months prior to each test period. The performance metric for both reconstruction accuracy and forecasting accuracy is the normalized root-mean-square error (NRMSE) defined at site x and forecast horizon i as:

$$NRMSE(x,i) = \frac{1}{\mathrm{P}_{\max}(x)} \sqrt{\frac{1}{T}\sum_{t=0}^{T} |\hat{P}(x,t+i) - P(x,t+i)|^2 \, 1_{P(x,t+i)>0}}$$

where $P(x,t+i)$ and $\hat{P}(x,t+i)$ are the ground truth power and predicted power, respectively, of site x at time $t+i$, $\mathrm{P}_{\max}(x)$ is the maximum power of site x over the evaluation interval and T is the number of time steps in the evaluation interval (1 year). The NMRSE is only computed for night periods, thus the indicator function $1_{P(x,t+i)>0}$ is included in the metric computation. The indicator function is defined as 1 if $P(x,t+i) > 0$ or 0 otherwise. For completeness, we also show the normalized mean absolute error (NMAE) to measure forecasting accuracy. The NMAE is more robust to outliers by not over-penalizing large deviations. The NMAE at site x and forecast horizon step i is defined as:

$$NMAE(x,i) = \frac{\sum_{t=0}^{T} |\hat{P}(x,t+i) - P(x,t+i)|}{\sum_{t=0}^{T} P(x,t+i)}.$$

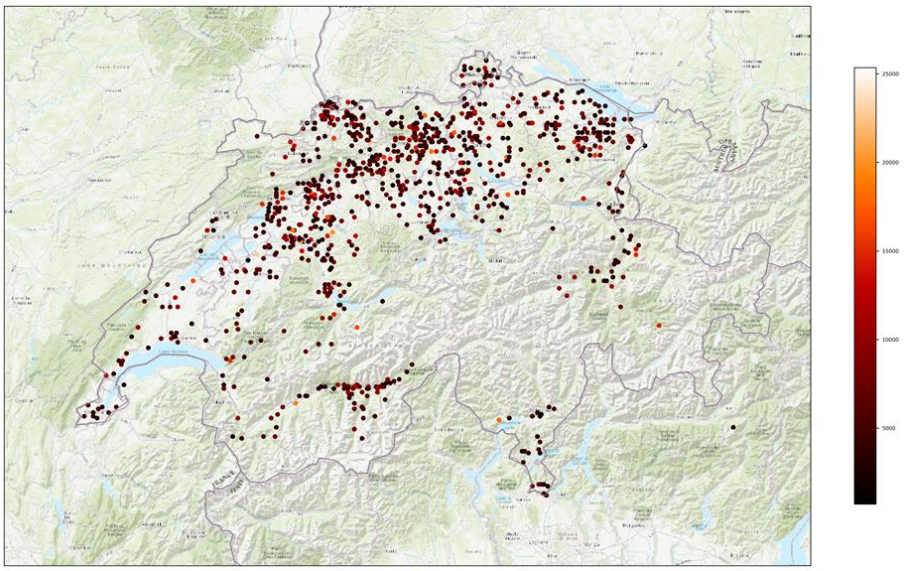


Figure 13. Spatial distribution of the synthetic dataset. Colours indicate the peak production at each site.

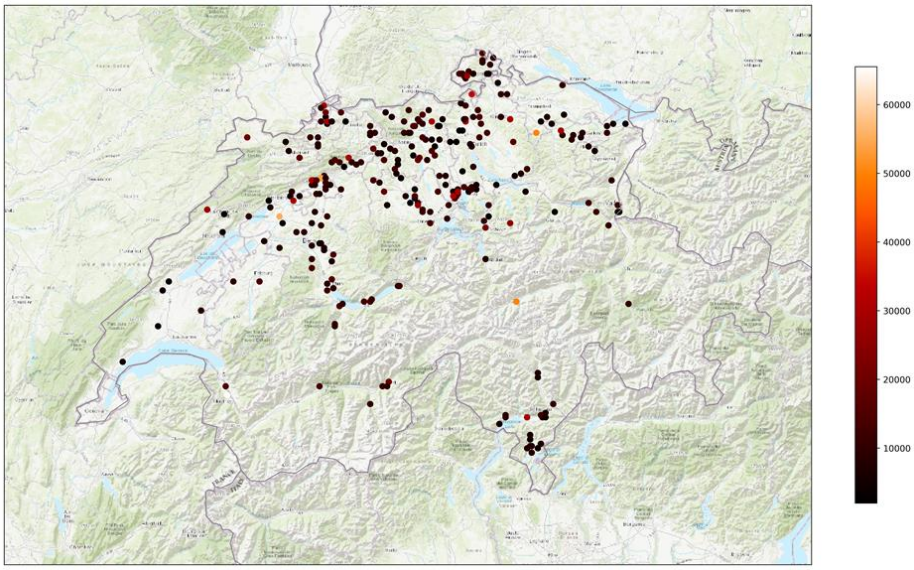


Figure 14. Spatial distribution of the real dataset. Colours indicate the peak production at each site.

4.1 Graph-based data reconstruction algorithm

We start by evaluating the performance of the graph reconstruction method. We evaluate the performance of the graph-based reconstruction method by simulating gaps in the time-series. The simulated gaps are drawn from a statistical model such that the expected length of the gaps per day can be varied. The spatial graphs for each data set (real and synthetic) were constructed using the 20 nearest neighbours and using a Gaussian kernel to compute the weights in the adjacency matrix. We varied the expected length of gaps from 2 to 16 hours (per day) to evaluate the performance of the algorithm over a period of one year in batches of two weeks. Figure 15 shows the results from the tests. We compare our reconstruction algorithm to a simple linear interpolation to illustrate the effectiveness of the proposed method when large gaps are present in the data. The results show that

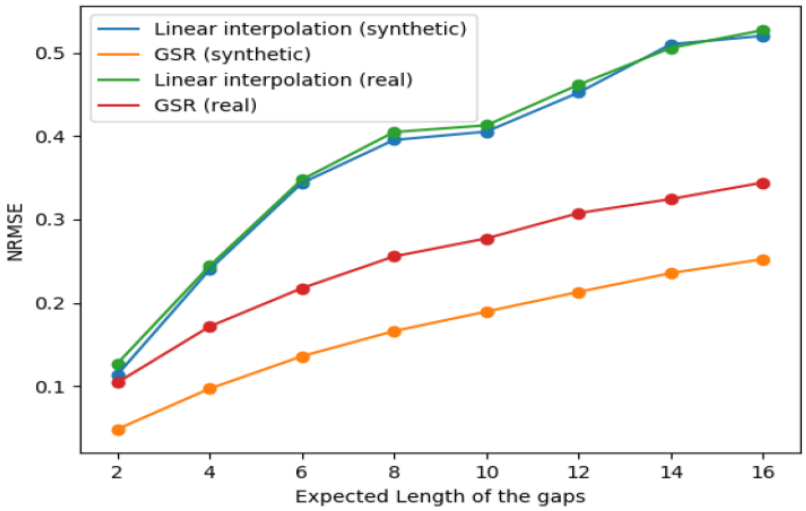


Figure 15. NRMSE for signal reconstruction against expected length of gaps (in hours) per day in the data.

reconstruction error is lower for the synthetic dataset than for the real dataset. In the synthetic dataset the proposed method achieves errors below 20% for gaps with expected lengths up to 8 hours. In the real dataset, the proposed method yields errors below 20% for gaps with expected lengths up to 4 hours.

Figure 16 shows a visualization of a reconstructed signal and the original signal for three nodes in a window of six days. The signals come from the real dataset with simulated gaps with an expected length of 8 hours per day.

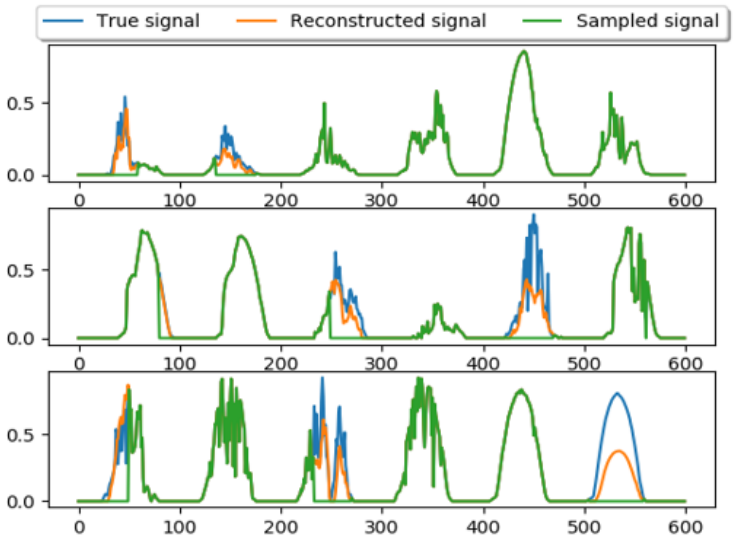


Figure 16. Reconstruction example of a corrupted signal with gaps of expected length of 8 hours per day. Visualization for three nodes in a window of six days.

4.2 Forecasting results using uninterrupted data

We now evaluate the performance of the proposed methods using clean and uninterrupted data (i.e. without missing values). The forecast horizon is set to 6 hours ahead i.e., $H = 24$ samples ahead with

a sampling time of 15 minutes. Figure 17 and Figure 18 show the NMAE and the NMRSE evolution, respectively, for the synthetic dataset in the prediction horizon (in discrete time steps). Figure 19 and Figure 20 show the NMRSE and NMAE, respectively, for the real dataset. For each prediction step the median value (solid line) and the interquartile distance (shadow bounds) are shown.

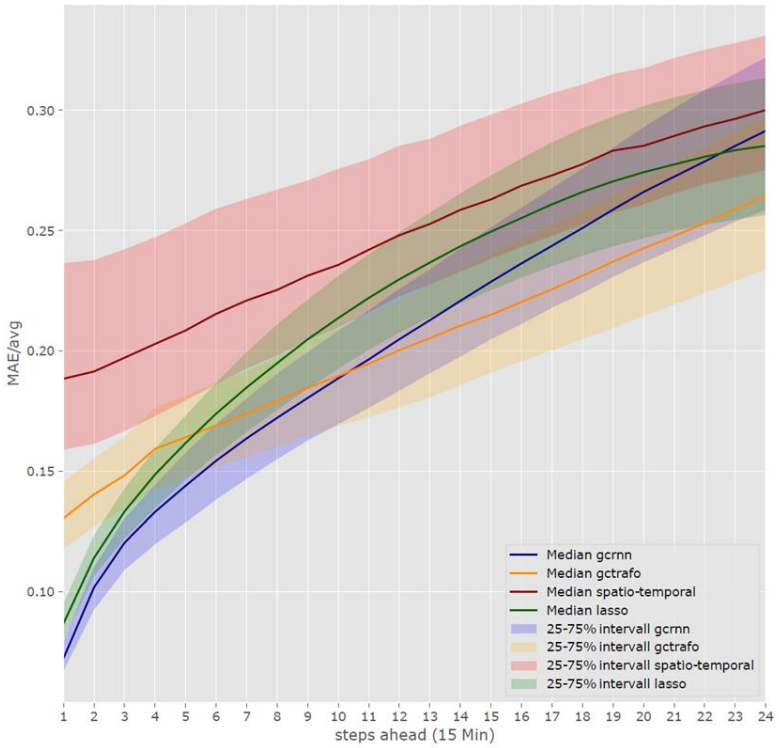


Figure 18. Forecast NMAE for the synthetic dataset. The forecast horizon is 6 hours ahead in steps of 15 minutes.

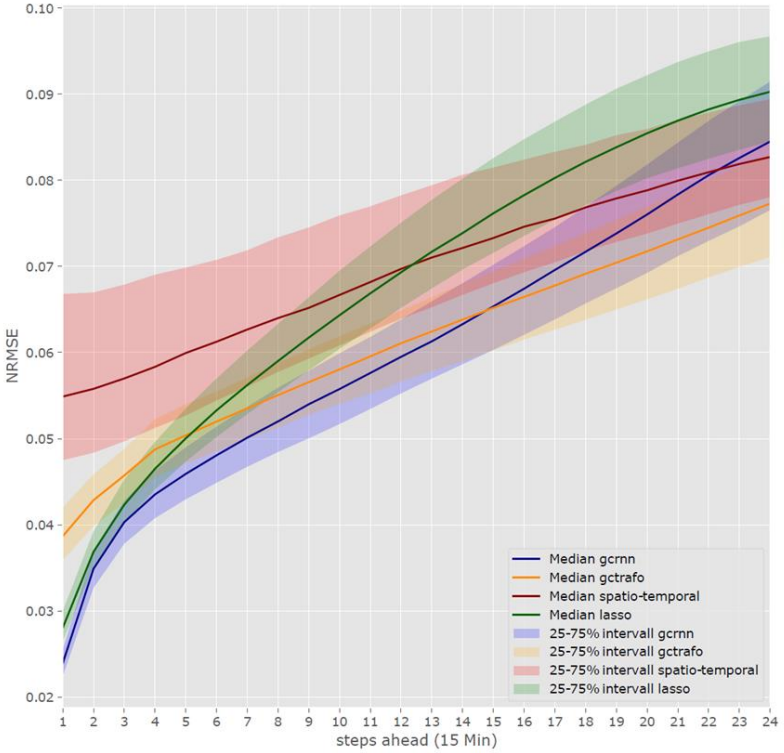


Figure 17. Forecast NMRSE for the synthetic dataset. The forecast horizon is 6 hours in steps of 15 minutes.

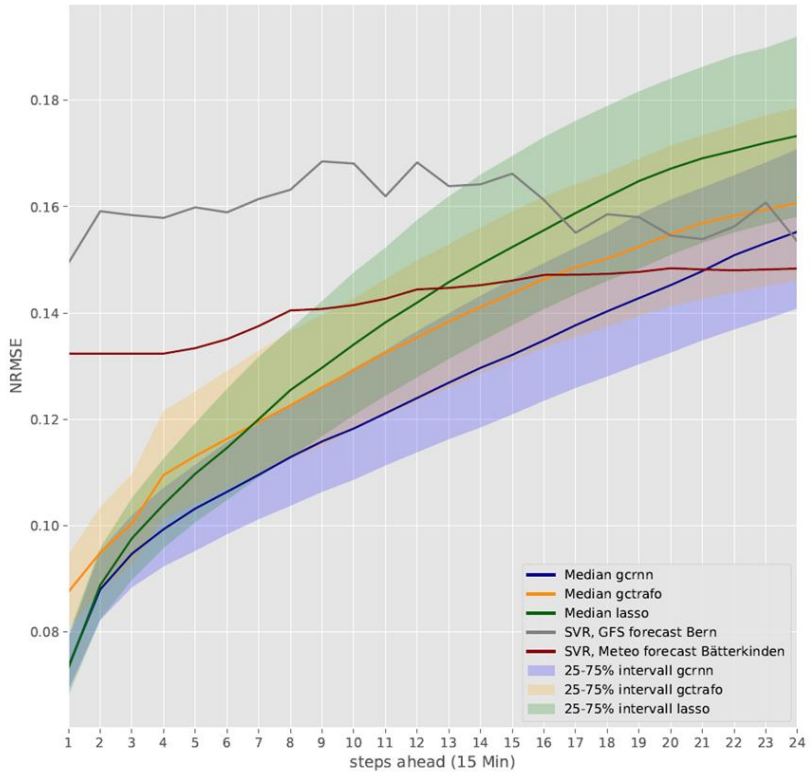


Figure 20. Forecast NMRSE for the real dataset. The forecast horizon is 6 hours in steps of 15 minutes.

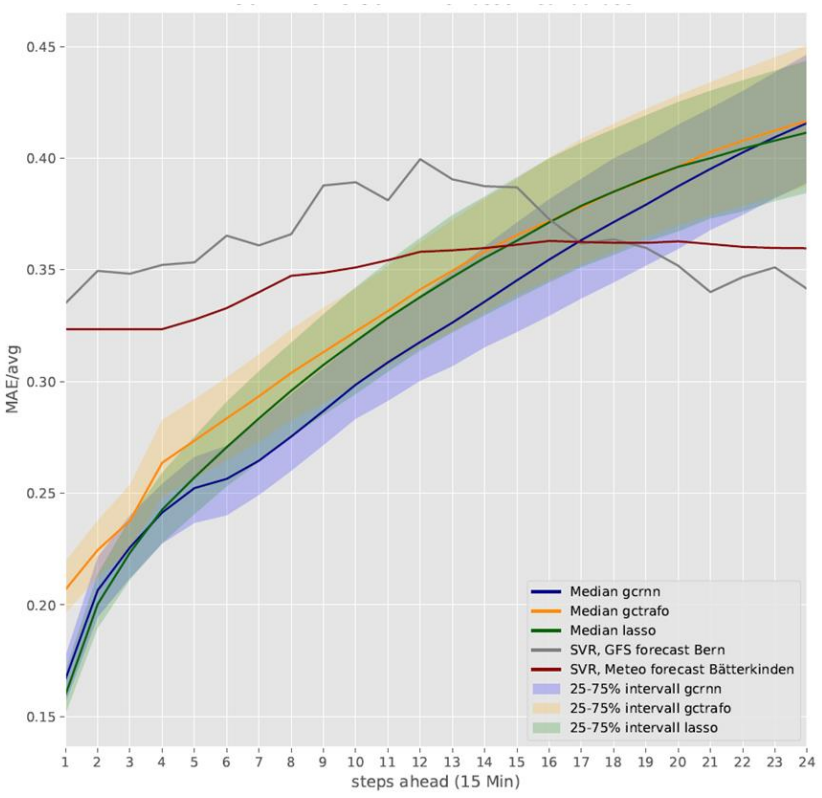


Figure 19. Forecast NMRSE for the real dataset. The forecast horizon is 6 hours in steps of 15 minutes.

For the real dataset, we have also included the forecasts for single sites using NWP as benchmarks. The forecasts are computed for Bern and Bätterkinden. The benchmark forecasts are computed using support vector regressors (SVR) with NWP as inputs (global irradiance and temperature, see [35] for further details). The forecasts for Bern were computed using historical NWP from the global forecast system (GFS) that have a temporal resolution of 3 hours, thus only two points are available for the 6 hours ahead horizon. The forecasts for Bätterkinden were computed using historical NWP from Meteotest[1] with a temporal resolution of 1 hour.

The maximum errors were obtained for the last time step of the prediction horizon i.e., 6 hours ahead prediction, as expected.

In the case of the synthetic dataset, the median NRMSE is below 10% for all horizon steps. The linear Lasso method and GCRNN perform better than all other methods for horizons up to 1 hour ahead, achieving NRMSE below 5%. For horizons longer than 3 hours ahead, the Lasso method has the largest NRMSE over all methods, while the ST attention method achieves the largest NMAE for the entire horizon. Considering the entire horizon, the GCTransformer method achieves the lowest average NRMSE and NMAE.

In the case of the real dataset, all methods achieve an NRMSE below 19% which is similar or better than state of the art methods for the same horizon [2]. All the proposed methods outperform the SVR methods that use NWP for horizons up to 3 hours ahead. Moreover, the GCRNN method outperforms the methods that use NWP for horizons up to 5 hours ahead (NMRSE) or 4 hours ahead (NMAE), and outperforms all the other proposed methods.

Table 1 reports the computational times required by the proposed methods, both for training and evaluation, for the entire year. The results show that most methods scale well up to 1000 nodes. The only methods that were not included in the comparison were the spatio-temporal attention methods that do not scale well with the number of nodes. In fact, the graph attention methods do not scale well with the number of nodes in their current form, only tests with up to 100 nodes were made. The linear lasso method does not scale well with the number of nodes mainly because only CPUs were used for both training and evaluation. The GCRNN and GCTransformer have architectures that are more computationally efficient with respect to the number of nodes and these methods can take advantage of GPUs and parallel computation.

Method	Training		Evaluation	
	300 nodes	1000 nodes	300 nodes	1000 nodes
Lasso (CPU)	5h (2017)	48h (2017)	2h (2017)	3h (2017)
Gcrnn (1 GPU)	12h (2016)	24h (2016)	30min (2017)	1h (2017)
Gctrafo (1 GPU)	12h (2016)	36h (2016)	30min (2017)	1h (2017)
st-att (CPU)	-	-	-	-
ts-att (CPU)	-	82h (2016)	-	1.3h (2017)

Table 1. Computational times required for training and evaluation for all the proposed methods

[1] https://meteotest.ch/en/

4.3 Forecasting results using incomplete data

Finally, we evaluated the complete forecast framework from incomplete data i.e., we receive incomplete data, we fill the gaps using the graph-based reconstruction method and then we use the cleaned data for forecasting. We evaluated the proposed approach using two different scenarios: 1) gaps with 4 hours duration on average and 2) gaps with 8 hours duration on average. Figure 21 and Figure 22 show the forecasting errors of the test comparing against the results obtained using clean data (i.e., no gaps) for the real and synthetic datasets, respectively. The robustness test was only made using the ST-AR method.

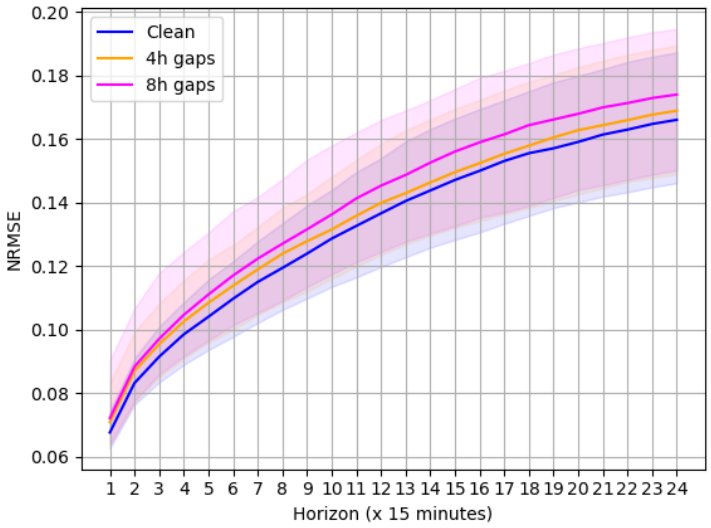


Figure 22. Forecast results from incomplete data. NRMSE for the real dataset.

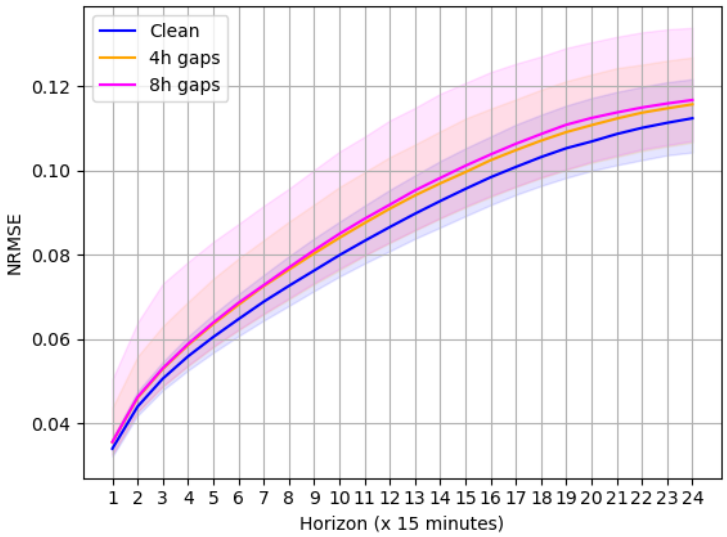


Figure 21. Forecast results from incomplete data. NRMSE for the synthetic dataset.

The results show the robustness of the proposed approach with the median NRMSE for both 4- and 8-hours gaps staying slightly above the median error from uninterrupted data. However, the interquartile distance of the NMRSE is increased for both datasets and scenarios. Moreover, in the case of the real dataset, the interquartile distance of the errors is closer to the one obtained using uninterrupted data.

5 Conclusions and outlook

The SODA project investigated linear and non-linear methods to forecast the local production of photovoltaics based on imperfect production measurements. Since the investigated methods outperform methods based on numerical weather forecasts on time horizons between 0 h and 6 h, the project confirmed the intuition that PV systems can effectively be used as distributed weather stations for forecasting purposes. The project also validated the initial hypothesis that graph-based methods are well-suited to the task of PV production forecasting by exploiting the spatio-temporal correlation between different systems. This hypothesis was not only validated by the quantitative performance of the algorithms but also by their interpretability: most methods yield explicit relationships between the PV systems which make physical sense.

Graph-based methods proved very powerful to reconstruct missing or faulty data. The NRMSE of this reconstruction is well below 20% for gaps of up to 4 h on the real dataset. Not only is the uncertainty on the reconstruction lower with graph-based methods than with conventional, linear interpolation, its increase with an increasing duration of gaps is also slower.

As a result, the full forecasting pipeline proved very robust against faulty data: the median NRMSE across all the systems with the ST-AR method is 17% for a 6 h horizon over one year and increases by less than one percentage points with gaps in data of up to 8 h.

All the developed methods outperform state-of-the art techniques which combine numerical weather prediction with machine learning (e.g., support vector regression) at least up to three hours ahead. Some of the most promising, non-linear methods such as graph convolutional neural networks, even outperform these state-of-the-art techniques for forecasts up to six hours ahead.

Finally, as compared to the state of the art, the project has significantly increased the confidence in the investigated algorithms since they have been tested on entire years and on uniquely large datasets: more than 300 real PV systems spread over Switzerland, and 1000 synthetic ones that reproduce the statistical distribution of installed PV in the country in terms of size, orientation and location.

While this scale proves the practical applicability of these methods, using such algorithms for energy management or grid operations across Switzerland will require a scale-up by at least an order of magnitude. Substantial research and development efforts are expected to ensure the computational load remains manageable. In parallel, the next steps shall include the integration of a broader range of sensors and input data, and moving from an offline system operating on historical data to an online one running on live data so that these technologies can become part of smart grid operations.

6 References

[1] *Loi fédérale sur la transformation et l'extension des réseaux électriques*. 2017.

[2] P.-J. Alet *et al.*, 'Forecasting and Observability: Critical Technologies for System Operations with High PV Penetration', Munich, Jun. 2016.

[3] M. Pierro *et al.*, 'Multi-Model Ensemble for day ahead prediction of photovoltaic power generation', *Solar Energy*, vol. 134, pp. 132–146, Sep. 2016, doi: 10.1016/j.solener.2016.04.040.

[4] C. Cornaro *et al.*, 'A New Approach for Regional Photovoltaic Power Estimation and Forecast', in *33rd European Photovoltaic Solar Energy Conference and Exhibition*, Nov. 2017, pp. 2485–2491, doi: 10.4229/EUPVSEC20172017-6BV.3.16.

[5] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de-Pison, and F. Antonanzas-Torres, 'Review of photovoltaic power forecasting', *Solar Energy*, vol. 136, pp. 78–111, Oct. 2016, doi: 10.1016/j.solener.2016.06.069.

[6] C. W. Chow *et al.*, 'Intra-hour forecastiouing with a total sky imager at the UC San Diego solar energy testbed', *Solar Energy*, vol. 85, no. 11, pp. 2881–2893, Nov. 2011, doi: 10.1016/j.solener.2011.08.025.

[7] R. Marquez and C. F. M. Coimbra, 'Intra-hour DNI forecasting based on cloud tracking image analysis', *Solar Energy*, vol. 91, pp. 327–336, May 2013, doi: 10.1016/j.solener.2012.09.018.

[8] H. S. Jang, K. Y. Bae, H.-S. Park, and D. K. Sung, 'Solar Power Prediction Based on Satellite Images and Support Vector Machine', *IEEE Transactions on Sustainable Energy*, vol. 7, no. 3, pp. 1255–1263, Jul. 2016, doi: 10.1109/TSTE.2016.2535466.

[9] R. Perez, S. Kivalov, J. Schlemmer, K. Hemker, D. Renné, and T. E. Hoff, 'Validation of short and medium term operational solar radiation forecasts in the US', *Solar Energy*, vol. 84, no. 12, pp. 2161–2172, Dec. 2010, doi: 10.1016/j.solener.2010.08.014.

[10] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, 'Predicting solar generation from weather forecasts using machine learning', in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct. 2011, pp. 528–533, doi: 10.1109/SmartGridComm.2011.6102379.

[11] K. Y. Bae, H. S. Jang, and D. K. Sung, 'Hourly Solar Irradiance Prediction Based on Support Vector Machine and Its Error Analysis', *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 935–945, Mar. 2017, doi: 10.1109/TPWRS.2016.2569608.

[12] C. Yang, A. A. Thatte, and L. Xie, 'Multitime-Scale Data-Driven Spatio-Temporal Forecast of Photovoltaic Generation', *IEEE Transactions on Sustainable Energy*, vol. 6, no. 1, pp. 104–112, Jan. 2015, doi: 10.1109/TSTE.2014.2359974.

[13] X. G. Agoua, R. Girard, and G. Kariniotakis, 'Short-Term Spatio-Temporal Forecasting of Photovoltaic Power Production', *IEEE Transactions on Sustainable Energy*, vol. 9, no. 2, pp. 538–546, Apr. 2018, doi: 10.1109/TSTE.2017.2747765.

[14] J. Xu, S. Yoo, J. Heiser, and P. Kalb, 'Sensor network based solar forecasting using a local vector autoregressive ridge framework', in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, Pisa, Italy, Apr. 2016, pp. 2113–2118, doi: 10.1145/2851613.2853124.

[15] Y. Kashyap, A. Bansal, and A. K. Sao, 'Spatial Approach of Artificial Neural Network for Solar Radiation Forecasting: Modeling Issues', *Journal of Solar Energy*, 2015. https://www.hindawi.com/journals/jse/2015/410684/ (accessed May 14, 2020).

[16] A. Ghaderi, B. M. Sanandaji, and F. Ghaderi, 'Deep Forecast: Deep Learning-based Spatio-Temporal Forecasting', *arXiv:1707.08110 [cs]*, Jul. 2017, Accessed: May 14, 2020. [Online]. Available: http://arxiv.org/abs/1707.08110.

[17] J.-I. Lee, I.-W. Lee, and S.-H. Kim, 'Multi-site photovoltaic power generation forecasts based on deep-learning algorithm', in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2017, pp. 1118–1120, doi: 10.1109/ICTC.2017.8190872.

[18] Q. Zhu, J. Chen, L. Zhu, X. Duan, and Y. Liu, 'Wind Speed Prediction with Spatio–Temporal Correlation: A Deep Learning Approach', *Energies*, vol. 11, no. 4, Art. no. 4, Apr. 2018, doi: 10.3390/en11040705.

[19] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang, and G. Liu, ‘Convolutional Graph Autoencoder: A Generative Deep Neural Network for Probabilistic Spatio-Temporal Solar Irradiance Forecasting’, *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 571–583, Apr. 2020, doi: 10.1109/TSTE.2019.2897688.
[20] J. Jeong and H. Kim, ‘Multi-Site Photovoltaic Forecasting Exploiting Space-Time Convolutional Neural Network’, *Energies*, vol. 12, no. 23, Art. no. 23, Jan. 2019, doi: 10.3390/en12234490.
[21] J. Á. González Ordiano, S. Waczowicz, M. Reischl, R. Mikut, and V. Hagenmeyer, ‘Photovoltaic power forecasting using simple data-driven models without weather data’, *Comput Sci Res Dev*, vol. 32, no. 1, pp. 237–246, Mar. 2017, doi: 10.1007/s00450-016-0316-5.
[22] P. Lauret, C. Voyant, T. Soubdhan, M. David, and P. Poggi, ‘A benchmarking of machine learning techniques for solar radiation forecasting in an insular context’, *Solar Energy*, vol. 112, pp. 446–457, Feb. 2015, doi: 10.1016/j.solener.2014.12.014.
[23] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, ‘Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks’, in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor, MI, USA, Jun. 2018, pp. 95–104, doi: 10.1145/3209978.3210006.
[24] W. Lee, K. Kim, J. Park, J. Kim, and Y. Kim, ‘Forecasting Solar Power Using Long-Short Term Memory and Convolutional Neural Networks’, *IEEE Access*, vol. 6, pp. 73068–73080, 2018, doi: 10.1109/ACCESS.2018.2883330.
[25] S. Killinger, N. Engerer, and B. Müller, ‘QCPV: A quality control algorithm for distributed photovoltaic array power output’, *Solar Energy*, vol. 143, no. Supplement C, pp. 120–131, Feb. 2017, doi: 10.1016/j.solener.2016.12.053.
[26] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, ‘The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains’, *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.
[27] J. S. Stein, W. F. Holmgren, J. Forbess, and C. W. Hansen, ‘PVLIB: Open source photovoltaic performance modeling functions for Matlab and Python’, in *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, Jun. 2016, pp. 3425–3430, doi: 10.1109/PVSC.2016.7750303.
[28] M. Defferrard, X. Bresson, and P. Vandergheynst, ‘Convolutional neural networks on graphs with fast localized spectral filtering’, in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, Spain, Dec. 2016, pp. 3844–3852, Accessed: Aug. 06, 2020. [Online].
[29] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, ‘A Comprehensive Survey on Graph Neural Networks’, *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020, doi: 10.1109/TNNLS.2020.2978386.
[30] K. Qiu, X. Mao, X. Shen, X. Wang, T. Li, and Y. Gu, ‘Time-Varying Graph Signal Reconstruction’, *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 870–883, Sep. 2017, doi: 10.1109/JSTSP.2017.2726969.
[31] M. Yuan and Y. Lin, ‘Model selection and estimation in regression with grouped variables’, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006, doi: 10.1111/j.1467-9868.2005.00532.x.
[32] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, ‘Learning graphs from data: A signal representation perspective’, *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019, doi: 10.1109/MSP.2018.2887284.
[33] S. Hochreiter and J. Schmidhuber, ‘Long Short-Term Memory’, *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[34] A. Vaswani *et al.*, ‘Attention is all you need’, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, California, USA, Dec. 2017, pp. 6000–6010, Accessed: Aug. 07, 2020. [Online].
[35] M. Boegli and Y. Stauffer, ‘SVR based PV models for MPC based energy flow management’, *Energy Procedia*, vol. 122, pp. 133–138, Sep. 2017, doi: 10.1016/j.egypro.2017.07.317.