Forecasting Net Load in France
The EDF Data Challenge

Eloi Campagne[1] Yvenn Amara-Ouali[2] Yannig Goude[3] Argyris Kalogeratos[4]

[1,4] *Centre Borelli, Université Paris-Saclay, ENS Paris-Saclay*
[1,2,3] *OSIRIS, EDF R&D*
{eloi.campagne, argyris.kalogeratos}@ens-paris-saclay.fr
{yvenn.amara-ouali, yannig.goude}@edf.fr

Résumé. Cet article présente un Data Challenge axé sur l'amélioration de la prévision de la demande nette d'électricité à court terme dans le contexte d'un réseau électrique décentralisé. L'intégration croissante des sources d'énergie renouvelables et les incertitudes liées à leurs fluctuations soulignent la nécessité d'une prévision précise de la demande. Les participants du challenge ont pour objectif de prévoir la consommation nette, c'est à dire la consommation totale moins la production renouvelable – solaire et éolienne –, en France, en utilisant des données disponibles à différentes résolutions géographiques (régionales et nationales), ainsi que des données de prix de l'électricité. Ce papier présente d'abord le contexte de la prévision de demande nette en électricité, puis présente les solutions apportées en amont du challenge.

Mots-clés. Data Challenge, Apprentissage automatique, Modélisation statistique, Production d'énergie renouvelable.

Abstract. This article presents a Data Challenge focused on improving short-term electricity net demand forecasting in the context of a decentralized power grid. The complexities arising from the increasing integration of renewable energy sources, and the uncertainties associated with their fluctuations, underline the need for accurate demand forecasting. Challenge participants aim to forecast net consumption, i.e. total consumption minus renewable production – solar and wind –, in France, using data available at different geographical resolutions (regional and national), as well as electricity price data. This paper first presents the context of the problem of net electricity demand forecasting, and then presents some of the interesting solutions produced for the challenge.

Keywords. Data Challenge, Machine Learning, Statistical Modelling, Renewable Energy Production.

1 Context and motivations

The effective operation of the electrical system relies on maintaining a balance between electricity supply and demand. Since electricity cannot be stored, its production needs to be constantly adjusted to match consumption. Providing accurate forecasts for short-term electricity demand is therefore crucial for all participants in the energy market. The shift towards

a decentralized electricity network introduces new uncertainties, which pose additional challenges for demand forecasting. The increasing contribution of renewable energy sources like solar and wind power, brings fluctuations and intermittency to the electricity market. These fluctuations and intermittency occur at various spatial scales due to the presence of wind farms and photovoltaic power plants. The Covid crisis, along with the current economic downturn, add further complexity to forecasting due to the non-stationarity in consumption patterns (Alasali et al., 2021). The availability of new geolocalized data and individual electricity consumption data can be exploited by models that are able to take advantage of additional information and help in minimizing forecast uncertainty (Obst et al., 2021, de Vilmarest and Goude, 2021). Furthermore, recent advancements in adaptive forecasting algorithms have demonstrated improvements in forecasting quality, particularly for aggregate load forecasting (Brégère and Huard, 2022, Antoniadis et al., 2022). However, most existing research overlooks the valuable information available at different spatial or relational scales. The aim of the EDF's FNL Challenge is therefore to focus on methods that can take into account geolocalized data to forecast net demand over France: in particular, the aim is to model renewable electricity production as accurately as possible.

With the announcement of the challenge and the call for participation, the organisers provided a starter kit containing pre-filled code notebooks designed to help participants familiarise themselves with the context of the problem. The starter kit includes a document that motivates the challenge, provides background information and describes the technical aspects of the forecasting problem. It also includes the datasets of interest, code for running a typical pipeline to handle the task (i.e. data loading and visualisation), and a number of basic models that provide a performance benchmark for the problem, with ready-to-use optimisation modules. Section 2 describes all of these elements, followed by Section 3 which presents the solutions produced in advance of the challenge.

2 Net Load Forecasting in France

2.1 Datasets

Two consumption datasets are available for the challenge: a regional dataset $\mathcal{D}_r$ and a national dataset $\mathcal{D}_n$. Each of them includes meteorological information (temperature, wind, cloud cover, etc.) and calendar information (time of day, type of day, holiday, etc.), enabling net demand to be forecast. A price dataset $\mathcal{D}_p$ is also available, with information on spot prices. For the challenge, the initial training period $\mathcal{T}^{\mathbf{tr}}_{\text{init}}$ runs from 2016-06-01 00:00 to 2021-06-01 23:30. The Covid containment periods (from 2020-03 to 2020-05-10, then from 2020-10-30 to 2020-12-15, and finally from 2021-04-03 to 2021-05-03) have been invalidated. The initial test period $\mathcal{T}^{\mathbf{te}}_{\text{init}}$ runs from 2021-06-02 00:00 to 2022-06-01 00:00:00. As the test period is hectic, we have retained only the last week of each month and the entire month of May 2022 in the test dataset. The remaining weeks have been added to the training dataset, we denote those weeks by $\Delta\mathcal{T}^{\mathbf{tr}}$. So, the final training period $\mathcal{T}^{\mathbf{tr}}_{\text{fin}}$ and test period $\mathcal{T}^{\mathbf{te}}_{\text{fin}}$ can be expressed as $\mathcal{T}^{\mathbf{tr}}_{\text{fin}} = \mathcal{T}^{\mathbf{tr}}_{\text{init}} \cup \Delta\mathcal{T}^{\mathbf{tr}}$ and $\mathcal{T}^{\mathbf{te}}_{\text{fin}} = \mathcal{T}^{\mathbf{te}}_{\text{init}} \setminus \Delta\mathcal{T}^{\mathbf{tr}}$ (Figure 1).

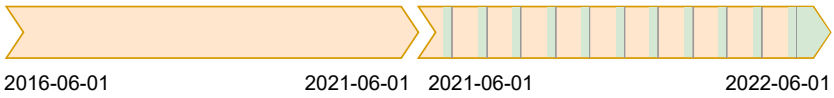


Figure 1: Training and test periods. Orange corresponds to $\mathcal{T}_{\text{fin}}^{\mathbf{tr}}$ and green to $\mathcal{T}_{\text{fin}}^{\mathbf{te}}$.

Regional dataset. In this dataset, 12 administrative regions of France are considered: Hauts de France, Normandie, Ile de France, Grand Est, Bretagne, Pays de la Loire, Centre Val de Loire, Bourgogne, Franche Comte, Nouvelle Aquitaine, Auvergne Rhone Alpes, Occitanie, and Provence Alpes Cote d'Azur. Note that Corse is not part of the study. In each region, there is a number of weather stations, between 1 and 5, and the meteorological variables are aggregated with a weighted average over these stations. Finally, a linear interpolation is used to obtain half-hourly data.

National dataset. The data in $\mathcal{D}_n$ are of the same type as those in $\mathcal{D}_r$, and differ only in their geographical resolution.

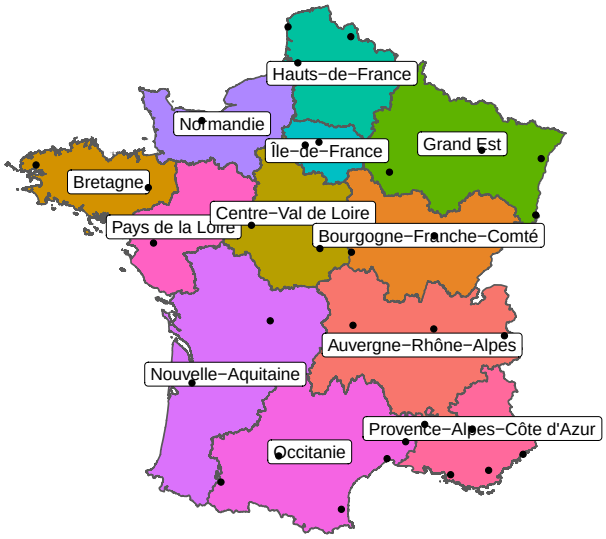


Figure 2: Map of the French mainland with its 12 administrative regions, and 32 weather stations appearing as black dots.

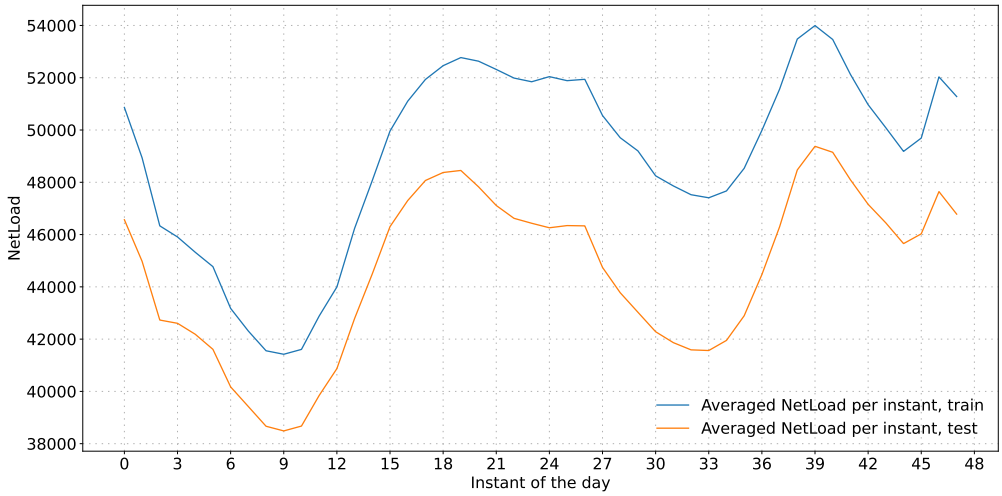


Figure 3: Averaged Net Load over time, for $\mathcal{T}_{\text{fin}}^{\mathbf{tr}}$ and $\mathcal{T}_{\text{fin}}^{\mathbf{te}}$.

Price dataset. The price data in $\mathcal{D}_p$ correspond to the spot price of electricity in France. The data have been coupled with consumption and production data, since they are highly correlated: indeed, with very low or even zero marginal costs, renewable energy types are the first to be called upon. If renewable energy is sufficient to cover all demand, the price of spot electricity is close to 0€ per MWh.

2.2 Evaluation

The challenge is to forecast the net load in France. In fact, three components need to be taken into account in the assessment: total load, solar and wind production. Therefore, we can measure the performance of the models with the following measure:

$$\ell(y, \hat{y}) = \ell_{\mathbf{load}}(y, \hat{y}) + \ell_{\mathbf{solar}}(y, \hat{y}) + \ell_{\mathbf{wind}}(y, \hat{y}). \tag{1}$$

However, models may seek to optimize the loss directly in relation to the net load (without seeking to predict the various components), and so the measure can just be $\ell(y, \hat{y}) = \ell_{\mathbf{netload}}(y, \hat{y})$ – typically, MAPE or RMSE can be taken as a loss.

2.3 Models

In this section, we present the models developed as part of the challenge.Then, in order to make the most out of these models, participants are introduced to expert aggregation.

2.3.1 Baseline Models

Persistence Model. Persistence models are among the simplest forecasting models. The forecast at time t is given by the value of the signal observed at $t - \Delta t$. where Δt is the only parameter of the model. In this case, we choose a 1-day persistence.

Gradient Boosting Models. Gradient Boosting refers to a class of machine learning models that combine sequentially weak learners, typically decision trees, to create a strong final predictive model. The process involves iteratively fitting new models to the residual errors of the previous models, and this way to gradually improve the overall prediction. At each iteration, the new model is trained to minimize a loss function by adjusting its parameters in the direction that reduces the gradient of the loss. We use XGBoost and CatBoost (Chen and Guestrin, 2016, Prokhorenkova et al., 2019) as reference implementations for the Challenge.

Generalized Additive Models. Generalized Additive Models (GAMs) is a class of semi-parametric regression models that was first developed in (Hastie and Tibshirani, 1986) and (Hastie, 2017) and are now widely used in electricity consumption forecasting. Indeed, GAMs are interesting in practice, since their additive aspect makes them highly explainable, but this also means that the choice of variables must be meticulous. Consider a prediction model aiming to predict for each time t a variable of interest y_t using $(x_j)_{j=1\dots d}$ explanatory variables such that $y_t = X_t\beta_0 + \sum_{j=1}^{d} f_j(x_{t,j}) + \varepsilon_t$, where β_0 is the intercept, $X_t = [x_{t,1}, \dots, x_{t,d}]$ and (ε_t) is i.i.d. random noise. Here we consider that each non-linear effect f_j is decomposed on a spline basis $(B_{j,k})$ with coefficient $\mathbf{B_j} \in \mathbb{R}^{m_j}$ where m_j is the chosen spline basis dimension, such that $f_j(x) = \sum_{k=1}^{m_j} \beta_{j,k} B_{j,k}(x)$. These coefficients are then estimated by minimizing the ridge-regression criterion ensuring the smoothness of the functions f_j by controlling the second derivatives (Wood et al., 2016). Three GAM models were developed for the challenge, a model forecasting net load, a model forecasting load and total renewable production, and a model forecasting load, wind production and solar production. These models are respectively referred to as GAM-1, GAM-2 and GAM-3.

2.3.2 Aggregation of Experts

Several models have been developed in the literature, each with its own distinctive features, which may also complement each other. Expert aggregation is an ensemble technique that allows to benefit from the advantages of each model: we can combine them using robust online aggregation of experts, as developed in (Cesa-Bianchi and Lugosi, 2006). For each instant in the prediction, a weight is assigned to each expert according to its previous forecasts: the better the past forecasts, the greater the weight at time t. Let $x_{j,t}$ be the j^{th} expert at time t and $p_{j,t}$ its corresponding weight, then the expression of the predicted load at time t is given by $\widehat{y}_t = \sum_{j=1}^{K} p_{j,t} x_{j,t}$, where K is the number of experts in the mixture. One way to compute the weights is to use polynomially weighted averages with multiple learning rate (ML-Poly), an algorithm developed in (Gaillard et al., 2014). A key advantage lies in the upper bound of the algorithm's average error:

$$\textbf{Average error of the algorithm} \lesssim \textbf{Average error of the best combination of experts} + \sqrt{\frac{\textbf{Number of experts}}{\textbf{Number of days}}} \tag{2}$$

Over time, the algorithm's average performance will converge to match the performance of the best experts.

3 Results

3.1 Numerical Results

Numerical results at the national level are shown in Table 1.

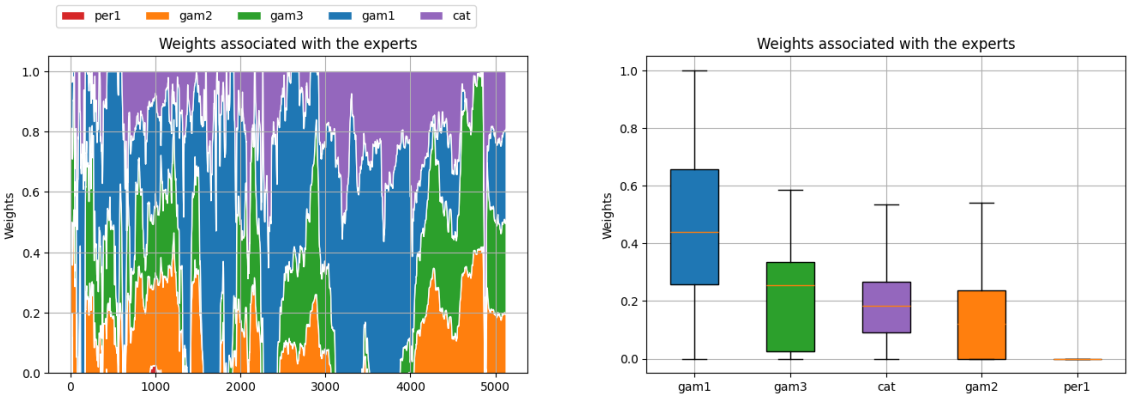
Model	RMSE (MW)	MAPE (%)
GAM-1	*1849*	*3.31*
GAM-2	1990	3.63
GAM-3	1967	3.56
CAT	3341	5.99
PER-1	5239	8.93
Mixture	**1511**	**2.66**

Table 1: Numerical performance in MAPE (%) and RMSE (MW) for $\mathcal{T}_{\text{fin}}^{\textbf{te}}$ at national level.

The hyperparameters of the baseline models above were optimized using the CMA-ES algorithm (Hansen et al., 2003).

3.2 Aggregation of Experts

Figure 4 shows the weights associated with each baseline expert.



(a) Weights associated to the experts.

(b) Contibution of the experts.

Figure 4: Aggregation of the baseline models for $\mathcal{T}_{\text{fin}}^{\mathbf{te}}$.

4 Perspectives

These models do not take into account the spatial aspect of the data: data can be represented using graphs. We believe it is interesting to make use of the deep relationships that exist between the regions for forecasting as their features are strongly correlated, see Figure 5, and therefore to develop graph neural networks (GNNs) models for the Challenge which could be then be used as novel experts in an aggregation.

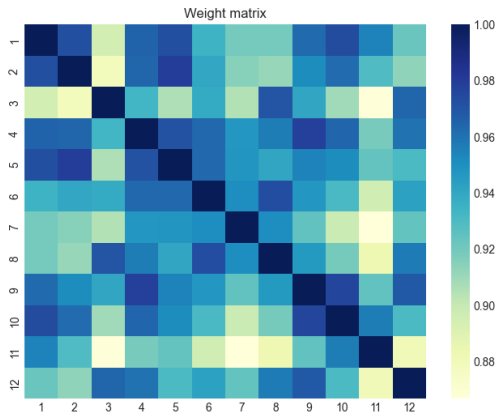


Figure 5: Correlation between the regions after a dimension reduction of the feature vector.

In graph theory, objects are represented by nodes, and the relationships between them are represented by edges. A graph $\mathcal{G}$ is a couple $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ a set of edges, i.e. $\mathcal{E} = \{(e_{ij}) = v_i v_j \mid v_i, v_j \in \mathcal{V}\}$. These graphs can be represented using adjacency matrices defined as $\mathbf{A} = (\mathbf{A}_{ij}) \in \mathbb{R}^{N \times N}$ such that $\mathbf{A}_{ij} = 1$ if and only if $e_{ij} \in \mathcal{E}$. Instead of a binary-weighted adjacency matrix, a more flexible weight matrix $\mathbf{W}$ with real-valued weights can be utilized. In the context of the Challenge, regions and the hidden links between them

can respectively be seen as nodes and edges. Both regression and classification tasks can be performed on graphs at different levels: node-level, edge-level and graph-level, see Figure 6.

- The **node-level** focuses on individual nodes within a graph. It involves analyzing the properties or attributes of each node. For example, in the context of electricity forecasting, a node-level task could be to predict the consumption for each region.
- The **edge-level** pertains to the analysis of the edges or connections between nodes in a graph. It involves examining the relationships, weights, or properties associated with each edge. For example, in the context of electricity forecasting, an edge-level task could be to quantify the relationships between the regions.
- The **graph-level** refers to the analysis of the entire graph structure as a whole. It involves examining global properties, overall connectivity, or emergent behaviors of the graph. For example, in the context of electricity forecasing, a graph-level task could be to predict the national consumption.

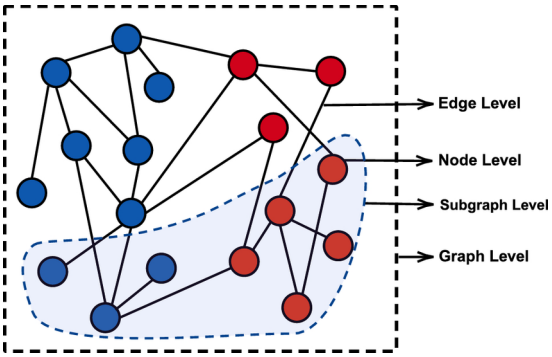


Figure 6: Various hierarchies of tasks in a graph representing edge, node, subgraph, and graph (Waikhom and Patgiri, 2022).

In the context of GNNs, message passing refers to the process of exchanging information between nodes, edges, and the global level of a graph (see Figure 7). Message passing is a fundamental operation in GNNs that enables nodes to gather and aggregate information from their neighbors, incorporate it into their own representations, and propagate it throughout the graph. Hence, a GNN corresponds to a set of layers that use the message-passing mechanism. Node representations are therefore updated as the graph is iterated through (in other words, at each layer traversed, representations are updated).

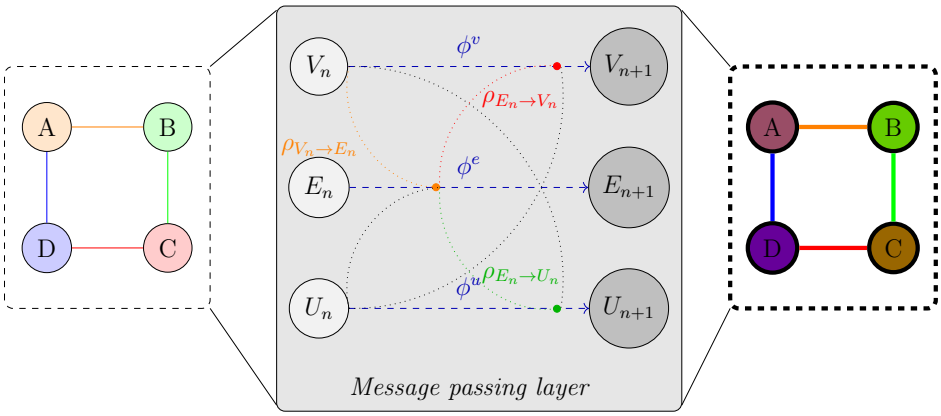


Figure 7: Example of a message passing layer in a GNN. V_n, E_n and U_n respectively refer to node, edge, and global level at stage n. ϕ are update functions and ρ are propagation functions.

A message passing layer therefore enables a node to update its embedding by taking into account information from its neighbors: thus, by propagation, d message passing layers enable a node to take into account information from its d^{th}-order neighbors. Using the notations of Figure 7, the message passing mechanism for each level can be written as follows:

- $V_{n+1} = \phi^v \left(V_n \ ; \ \rho_{E_n \to V_n}, \rho_{U_n \to V_n}\right)$
- $E_{n+1} = \phi^e \left(E_n \ ; \ \rho_{V_n \to E_n}, \rho_{U_n \to E_n}\right)$
- $U_{n+1} = \phi^u \left(U_n \ ; \ \rho_{V_n \to U_n}, \rho_{E_n \to U_n}\right)$

In a standard neural network, such as a feedforward neural network, the update process is typically a local operation. In contrast, message passing in GNNs is a more dynamic and relational process allowing nodes to capture both local and global graph structures and dependencies. This relational approach enables GNNs to capture complex graph patterns and dependencies that cannot be easily captured by standard neural networks, making them suitable for tasks involving graph-structured data. The message passing mechanism can also be seen as a special case of graph convolution. Convolutional operations, originally developed for regular grid-structured data such as images, enable the extraction of meaningful features by considering the local neighborhood of each element (LeCun et al., 1995). By extending convolution to graphs, we can capture and analyze the structural patterns and relationships present in the graph data, see (Daigavane et al., 2021) for a visual understanding. Graph convolution extracts localized features, where information from neighboring nodes is aggregated to compute features for each node allowing to capture the local connectivity and dependencies between nodes. It also helps to leverage the inherent structure and connectivity of the graph data, uncover hidden patterns (e.g. the relationship between the consumptions of two different regions), and make informed predictions (e.g. the consumption of a given region) based on the relationships between nodes.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, and for each node $v \in \mathcal{V}$, we associate a feature vector X_v which corresponds to the explanatory variables of the corresponding node. Each node $v \in \mathcal{V}$ has an associated label y_v and we want to learn a representation h_v such that, for a GNN f, we have $f(h_v) = y_v$. To do this, we iteratively update the representations of a node by aggregating the representations of its neighbors in the graph. The representation of a node v at iteration k, denoted $h_v^{(k)}$ can be expressed as

$$\begin{aligned} h_v^{(k)} &= \mathrm{UPDATE}^{(k)} \left(h_v^{(k-1)}; \mathrm{AGGREGATE}^{(k)} \left(h_v^{(k-1)}; \left\{ h_u^{(k-1)} \mid u \in \mathcal{N}_v \right\} \right) \right), \\ h_v^{(0)} &= X_v, \end{aligned} \tag{3}$$

where UPDATE usually involves combining the prior representations with the current one and a linear mapping. AGGREGATE is usually a combination of a pooling function (max, sum,...) with an activation function (ReLU, tanh,...). Two approaches are being studied for the Challenge: a basic graph convolutional network (GCN) model (Kipf and Welling, 2017) and the SAGE model (Hamilton et al., 2018). These models allow a gain of around 0.2% in the aggregation.

References

Alasali, F., Nusair, K., Alhmoud, L., and Zarour, E. (2021). Impact of the covid-19 pandemic on electricity demand and load forecasting. *Sustainability*, 13(3):1435.

Antoniadis, A., Gaucher, S., and Goude, Y. (2022). Hierarchical transfer learning with applications for electricity load forecasting.

Brégère, M. and Huard, M. (2022). Online hierarchical forecasting for power consumption data. *International Journal of Forecasting*, 38(1):339–351.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM.

Daigavane, A., Ravindran, B., and Aggarwal, G. (2021). Understanding convolutions on graphs. *Distill*, 6(9):e32.

de Vilmarest, J. and Goude, Y. (2021). State-space models win the ieee dataport competition on post-covid day-ahead electricity load forecasting.

Gaillard, P., Stoltz, G., and Van Erven, T. (2014). A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196. PMLR.

Hamilton, W. L., Ying, R., and Leskovec, J. (2018). Inductive representation learning on large graphs.

Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models (with discussion), statistical science.

Hastie, T. J. (2017). Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Obst, D., de Vilmarest, J., and Goude, Y. (2021). Adaptive methods for short-term electricity load forecasting during covid-19 lockdown in france. *IEEE Transactions on Power Systems*, 36(5):4754–4763.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2019). Catboost: unbiased boosting with categorical features.

Waikhom, L. and Patgiri, R. (2022). A survey of graph neural networks in various learning paradigms: methods, applications, and challenges. *Artificial Intelligence Review*.

Wood, S. N., Pya, N., and Säfken, B. (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association*, 111(516):1548–1563.