Forecasting Electricity Prices: An Optimize Then Predict-Based Approach

Léonard Tschora, Erwan Pierre, Marc Plantevit, Céline Robardet

Forecasting Electricity Prices: an Optimize then Predict-based approach

Léonard Tschora[1,2], Erwan Pierre[2], Marc Plantevit[3], and Céline Robardet[1]

[1] Univ Lyon, INSA Lyon, LIRIS, UMR5205, F-69621 Villeurbanne
[2] BCM Energy, F69003 Lyon, France
[3] EPITA Research Laboratory (LRE), FR-94276, Le Kremlin-Bicêtre, France

Abstract. We are interested in electricity price forecasting at the European scale. The electricity market is ruled by price regulation mechanisms that make it possible to adjust production to demand, as electricity is difficult to store. These mechanisms ensure the highest price for producers, the lowest price for consumers and a zero energy balance by setting day-ahead prices, i.e. prices for the next 24 hours. Most studies have focused on learning increasingly sophisticated models to predict the next day's 24 hourly prices for a given zone. However, the zones are interdependent and this last point has hitherto been largely underestimated. In the following, we show that estimating the energy cross-border transfer by solving an optimization problem and integrating it as input of a model improves the performance of the price forecasting for several zones together.

Keywords: Electricity Price Forecasting · Optimization-based data augmentation · Machine learning

1 Introduction

Energy challenges are even more important as our societies have become extremely dependent on it. However, the production of energy, and in particular electricity, is linked to many intricate factors, based on different estimates such as weather forecasts (influencing both production and consumption) or production capacities for various means. Added to this complexity is a tariff regulation mechanism [13] used to balance production and consumption, as electricity is hard to store. This algorithm maximizes social welfare defined as the sum of consumer surplus, supplier surplus and congestion rents from cross-border exchanges. It ensures the highest price for producers, the lowest price for suppliers and a constant energy balance by setting day-ahead prices, i.e., 24-hourly prices for the next day.

Being able to forecast day-ahead energy prices is crucial to control energy production and for a successful energy transition. Thus, many works [5,16,9,15] have sought to produce the most accurate price prediction models possible. In [14], we have shown that approaches based on machine learning models are superior to benchmark auto-regressive models. They provide much more accurate predictions and are fast enough to be used operationally. We also strove for predicting the prices of different zones jointly. Although we did not obtain a

significant improvement in the forecasts, the analysis of the contributions of the variables highlighted the importance of integrating data from foreign countries for the price forecast. For example, we have shown that Swiss prices contribute significantly to increasing the accuracy of French, Belgian and German price forecasts. We concluded that we had not used enough information to correctly model the European network, in particular that we had not sufficiently taken into account transfer capacities and cross-border energy flows in our models.

We propose to overcome these limitations by putting forward different ways to integrate cross-border flows into predictive models. Cross-border flows are constrained by the Available Transfer Capacity (ATC) between two countries that share a border. However, this maximum capacity is not fully used continuously and knowing the flows between countries would undoubtedly improve the prediction models. For this, we propose to take advantage of domain knowledge to estimate cross-border flows by a combinatorial optimization model.

The proposed approach is reversed from the predict-then-optimize approaches [3,11] used to solve many decision-making problems by combining machine learning and combinatorial optimization. In this framework, some parameters of a combinatorial optimization problem are estimated from other features based on historical data. Our approach use a combinatorial optimization model to estimate features that are then used to train a machine learning model.

In this paper, we introduce the problem of electricity price forecasting on the European market (Section 2). Our research hypothesis is that we can improve the model prediction by enriching the input data thanks to domain knowledge. Especially, we introduce the problem of estimating the cross-border flows (Section 3). We design two distinct combinatorial optimization problems and their combination. Then, we use the results of these optimization problems in a multi-zone forecasting model that predicts prices for 35 distinct zones of the European market (Section 4). The experimental evaluation (Section 5) confirms that the cross-border flows estimation makes it possible to improve the model performance. We then conclude with a broader discussion and a forward look (Section 6).

2 Electricity price forecasting problem on the European market

Unlike other commodities (e.g., cereals, oil), electricity cannot be efficiently stored. To prevent failures on the electricity network, balancing algorithms are used. On the European market, the EUPHEMIA [13] algorithm fixes hourly prices by matching demand, production and exchanges across Europe in a way to maximize the social welfare while taking into account the market and network constraints: **(1)** The energy balance must be zero for all zones at all times. **(2)** The flow of energy between two zones must not exceed the maximum transfer capacity between these two zones. **(3)** Where possible, the energy flow between two areas is maximized to generate more profit from congestion rents. This algorithm runs daily at noon and determines the day-ahead prices, matched demand and supply and energy flows of the 46 European zones (see Figure 1). The electricity price

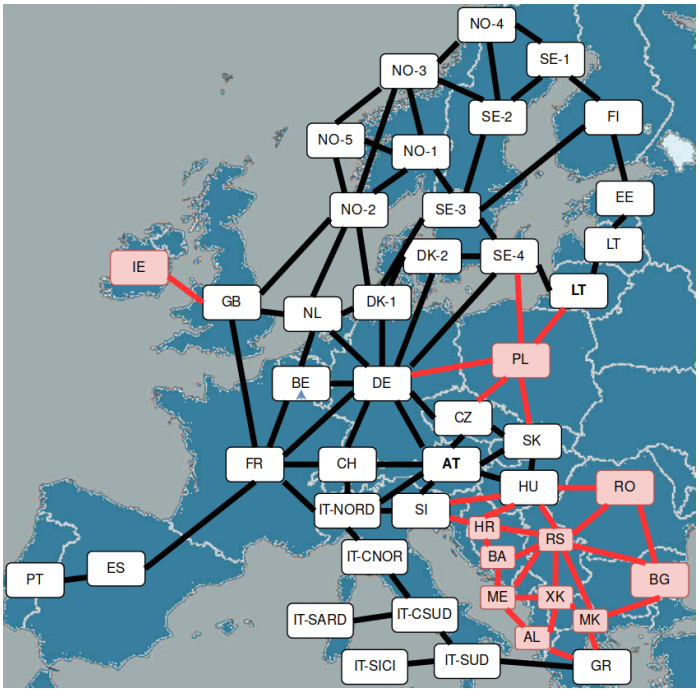

Fig. 1. European electricity market map: Some countries are divided into several zones (e.g., Italy, Norway). Prices are established for each zone. Energy can flow between connected zones. Areas or connections colored in red are excluded from our dataset due to lack of data.

forecasting problem (EPF) consists in predicting the prices over 24 hours before their settlement. Electricity prices are constrained by fundamentals variables: consumption, generation, transfer capacities. More precisely, pricing algorithms use a forecast of those variables for the next day.

To solve the EPF problem, we represent the European market on day d using a graph. Each zone is represented by a node z for which day-ahead prices $\mathbf{D}_z \in \mathbb{R}^{24}$ must be predicted. For some problems, the required amount of energy to be produced $\mathbf{E}_z \in \mathbb{R}^{24}$ also has to be predicted. Connected zones on the market are linked in the graph by edges (z, z'), associated with day-ahead flows $\mathbf{F}_{z,z'} \in \mathbb{R}^{24}$. The features used by the pricing algorithm are **(1)** Consumption forecast for the next day $\mathbf{C}_z \in \mathbb{R}^{24}$ **(2)** Renewable generation forecast for the next day $\mathbf{R}_z \in \mathbb{R}^{24}$, **(3)** Programmable generation forecast for the next day $\mathbf{G}_z \in \mathbb{R}^{24}$, **(4)** Maximal generation capacity for the next day $\mathbf{V}_z \in \mathbb{R}^{24}$, **(5)** Current Prices $\mathbf{P}_z \in \mathbb{R}^{24}$, **(6)** Available Transfer Capacities for the next day $\mathbf{A}_{z,z'} \in \mathbb{R}^{24}$ which is the maximum amount of energy that can be sent from z to z'. Since renewable energy production is subject to external factors that are not controllable (wind speed, solar radiation, etc...), we distinguished the two types of source by $\mathbf{R}_z$ and $\mathbf{G}_z$.

Hence, $\mathbf{C}, \mathbf{R}, \mathbf{G}, \mathbf{P}, \mathbf{V}$ and $\mathbf{A}$ are known at prediction time, while $\mathbf{D}$, $\mathbf{E}$ and $\mathbf{F}$ are unknown. In what follows, we propose to take advantage of knowledge from the field of electricity pricing to estimate the flows $\mathbf{F}$ between zones by combinatorial optimization, before using those results to forecast the day-ahead prices $\mathbf{D}$.

3 Estimate cross-border flows by combinatorial optimization

The EUPHEMIA algorithm sets electricity prices on the European market by satisfying the constraints listed in Section 2. These constraints lead to sophisticated and counter-intuitive flows between zones, some zones playing the role of transit zones to make possible energy exchanges between two other zones. To better model these dynamics, we use domain knowledge to approximate day-ahead flows $\mathbf{F}$ and use them as predictive variables into EPF models. In this section, we describe four different methods for predicting $\mathbf{F}$.

3.1 A formalization by linear programming

The most natural way to formulate the flow optimization problem is to write the EUPHEMIA algorithm as a linear programming problem. In doing so, the network constraints are explicitly enforced, and the Day-Ahead Flow $\mathbf{F}$ and required energy generation $\mathbf{E}$ are computed:

$$\mathbf{Flin} = \underset{\mathbf{F}_{z,z'} \text{ and } \mathbf{E}_z}{\arg\max} \sum_{z,z'} \mathbf{F}_{z,z'}(\mathbf{P}_{z'} - \mathbf{P}_z)$$

$$\textit{under const.} \begin{cases} \mathbf{C}_z - \mathbf{R}_z - \mathbf{E}_z + \sum_{z'} \mathbf{F}_{z,z'} - \sum_{z'} \mathbf{F}_{z',z} = 0 & \forall z \\ \mathbf{E}_z \leq \mathbf{V}_z & \forall z \\ \mathbf{F}_{z,z'} \leq \mathbf{A}_{z,z'} & \forall z, z' \end{cases}$$

Flow-related profit is maximized under three constraints. The first constraint ensures a zero energy balance, the second stipulates that the planned production must not exceed its maximum capacity, and the third imposes that the flows do not exceed the capacities of the lines. The cost aims to maximize congestion rents by maximizing potentially valuable flows: flows from a zone with lower prices to a zone with higher prices. In this set-up, we consider the consumption $\mathbf{C}_z$ and renewable generation forecasts $\mathbf{R}_z$ as fixed. The required generation $\mathbf{E}_z$ is determined to match $\mathbf{C}_z - \mathbf{R}_z$.

3.2 Formalizing the problem by a least-squares loss

The formulation of the problem by linear programming has a major drawback. We allow the generation of zone to expand to its maximum capacity $\mathbf{E}_z \leq \mathbf{V}_z$ without penalty to the cost. In practice, switching power plants on or off has a

Problem	**CC**	**MAE** (MWh	**SMAPE** (%)
Flin	0.153	944.63	120.32
Flsq	**0.389**	**418.05**	**105.93**

Table 1. CC, MAE and SMAPE metrics between **Flin** and **Flsq** optimized flows and actual day-ahead flow values **F** on the train dataset.

cost that is not linear with respect to the generated volume. We thus propose to rewrite the problem by transforming the energy balance constraint into a cost to be minimized.

$$\mathbf{Flsq} = \underset{\mathbf{F}_{z,z'}}{\arg\min} \sum_{z} \left(\sum_{z'} \mathbf{F}_{z,z'} - \sum_{z'} \mathbf{F}_{z',z} + \mathbf{C}_z - (\mathbf{R}_z + \mathbf{G}_z) \right)^2$$
$$\textit{under constraint} \;\; 0 \leq \mathbf{F}_{z,z'} \leq \mathbf{A}_{z,z'} \; \forall_{z,z'}$$

The squared loss ensures that unbalanced zones are heavily penalized. Thus, we do not have to penalize the objective by the price difference and, we can also remove the determination of $\mathbf{E}_z$ from the problem and we use the programmable generation forecast $\mathbf{G}_z$ instead.

3.3 Combining the two formalizations

To study the estimation quality of these two models, we solved the two optimization problems for each hour of our train dataset (see Section 4.1) using scipy[4]. As flow values are known a posteriori, we can evaluate the quality of the estimation on the train set using standard measures (see description in Section 5). The metrics obtained are reported in Table 1. It is obvious that **Flsq** outperforms **Flin** on the dataset. However, by analyzing the estimations with a lower granularity, we observe that the performances vary according to graph edges. For example, the flow on the edge between Norway-5 and Norway-1 is well handled by problem **Flsq** as shown in Figure 2 (left) while the flow on edge between France and Germany is better handled by problem **Flin** (see Figure 2 right). To take advantage of these two models, we sought to identify the market conditions allowing to differentiate these two scenarios. For this, we first define the loss difference between the results of the two problems as

$$\mathbf{L}^{(t)}(z, z') = |\mathbf{F}^{(t)}_{z,z'} - \mathbf{Flsq}^{(t)}_{z,z'}| - |\mathbf{F}^{(t)}_{z,z'} - \mathbf{Flin}^{(t)}_{z,z'}|$$

where $t = (d, h)$ is one of the N possible time-steps. We analyze the relationship between $\mathbf{L}^{(t)}(z, z')$ and the characteristics of the market

$$x \in (\mathbf{C}_z, \mathbf{C}_{z'}, \mathbf{R}_z, \mathbf{R}_{z'}, \mathbf{P}_z, \mathbf{P}_{z'}).$$

[4] https://scipy.org/

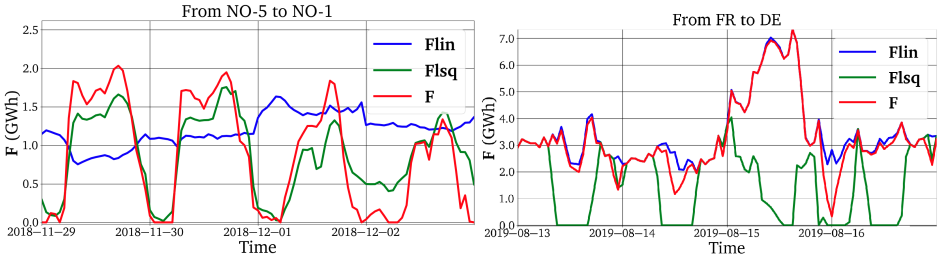

Fig. 2. Optimized flows **Flin** (blue), **Flsq** (green) and the actual Day-Ahead flows **F** (red) between the Norway-5 and Norway-1 zones (left) and the France and Germany zones (right). On the left, we observe that **Flsq** comes close enough to the Day-Ahead flow **F**, while **Flin** does not. On the right, we observe the opposite.

We break down x into 100 quantiles x_q and compute the average loss for each (x, q):

$$\mathbf{L}(z, z', x, q) = \frac{1}{N} \sum_{t \in T(x,q)} \mathbf{L}^{(t)}(z, z')$$

with $T(x, q) = \{t \mid x^{(t)} \in [x_q, x_{q+1}]\}$. Market conditions where $\mathbf{L}(z, z', x, q) > 0$ correspond to situations where it is preferable to use **Flin** instead of **Flsq**. We name the results of this combination **Fcmb**. To generate **Fcmb** on the test dataset, we keep the same market conditions (z, z', x, q) as found on the train dataset. This prevents data leaks related to the use of posterior data for a prediction.

3.4 One-sided flows

In the above formalization, we enable bilateral flows between two zones, i.e. $\mathbf{F}_{z,z'} > 0$ and $\mathbf{F}_{z',z} > 0$ can both occur, which matches the logic of EUPHEMIA. However, in practice most connections never have two-sided flows. To further improve our flow modeling, we identify one-sided connections and apply one-sideness in our flow estimations. For each link (z, z'), we count the number of times on the train dataset when the flow is one-sided i.e. when we have $\mathbf{F}_{z,z'} \geq 0$ and $\mathbf{F}_{z',z} = 0$. If this occurs more than 75% of the time, we consider the edge (z, z') as always one-sided. For this, we keep the most important predicted flow from which we subtract the least important flow. We set the latter to 0. In this way, the energy balance in the two zones remains the same. We apply this transformation to **Fcmb** and call the result **Fos**.

4 Electricity price forecasting models

4.1 The dataset

In this section, we tackle the EPF problem on the European market. The data is available free of charge[5] and we collected 35 out of 46 zones, linked by 63 connections. For each zone z, the attributes are $\mathbf{X}_z = (\mathbf{C}_z, \mathbf{R}_z, \mathbf{G}_z, \mathbf{P}_z) \in \mathbb{R}^{96}$. Hence, each day is described by 35×96 predictive features and the targets to be predicted are the 24 hourly prices for each zone. We exclude the Swiss and Great-Britain prices from the prediction task. Although they are part of the network, their prices are determined prior to the closing of EUPHEMIA and we prefer to use them as predictive variables. We predict the 24 prices of the remaining 33 zones every day: $\mathbf{Y} \in \mathbb{R}^{792}$. Our dataset spans from 01/01/2016 to 31/12/2021. We use the last two years (2020, 2021) as test set. Two years is a good duration because the prices show a strong seasonality. The year 2019 is kept as a validation set for hyper-parameter search.

In addition to the 35×96 predictive features cited above, we consider the Available Transfer Capacities for each connection **A** or instead one of the flow estimates **Flin**, **Flsq**, **Fcmb**, or **Fos** for each link, leading to 126×24 additional predictive variables. Each line of our dataset corresponds to a day and has 6384 values.

4.2 The machine learning models

We use Deep Neural Network and Convolutional Neural Network to predict the electricity prices. Deep Neural Networks (**DNN**) [6,7,8,12] are the most commonly used models in EPF. Its training samples are vectors $s \in \mathbb{R}^{6384}$. Convolutional Neural Networks (**CNN**) have also seen a growing interest in EPF over the past years [7,4,1]. We compute the convolutions along time and each sample is a vector $s \in \mathbb{R}^{(35+126)\times 24}$. Finally, we propose to use a Graph Neural Network (**GNN**), which is new for the EPF domain. **GNNs** make it possible to exploit data structured as graphs as described in Section 2. We train our **GNN** for the node prediction problem by stacking graph convolution layers that update the node embeddings. This is followed by linear layers that map node embeddings to their predicted values. We use tensorflow and pytorch-geometric libraries [6]. Each model (**DNN**, **CNN**, **GNN**) is trained on 5 different versions of our dataset according to the method use to estimate **F**: **A**, **Flin**, **Flsq**, **Fcmb**, **Fos**. To be fair in our experiments, we set a time limit for the hyper-parameter search. More precisely, we let our program explore the hyper-parameter grid for 24 hours for each model with $\mathbf{F} = \mathbf{A}$ on a 20cpus computer and use the same configuration for all variants of **F**. This introduces a slight bias as the resulting best configuration is chosen for its performance on the **A** dataset. After finding the optimal configuration, we calculate forecasts on the test dataset using

[5] https://transparency.entsoe.eu

[6] https://www.tensorflow.org/, https://pytorch-geometric.readthedocs.io

Problem	CC	MAE (MWh)	SMAPE (%)
A	0.14	917.59	111.43
Flin	0.116	876.51	111.12
Flsq	**0.380**	388.95	105.35
Fcmb	0.367	396.46	102.52
Fos	0.375	**314.5**	**81.19**

Table 2. Metrics for flow estimation on the test dataset for the different methods. The **Flsq** method outperforms the **Flin** methods. The **Fcmb** method does not improves the metrics, while the **Fos** method improves performances.

recalibration. It consists in re-training the model using the most recent data before making forecasts. Once a test set sample is predicted, we can integrate its predictions into the training dataset and retrain the model. We recalibrate our models every 30 days.

5 Experiments

We compare the values between the predicted $\hat{Y}_{z,h}$ and the real $Y_{z,h}$ target variables for the different zones z and hours h. We use standard measures as $MAE(Y, \hat{Y})$ (the average of the absolute difference between the values over the target variables), $SMAPE(Y, \hat{Y})$ (the symmetric mean absolute percentage error over the target variables), and CC (the average correlation coefficient over the target variables). To check the statistical significance of the results, we use the Diebold & Mariano (DM) test [2] that compares two models M_1 and M_2. The null hypothesis H_0 is that $Loss(M_1) > Loss(M_2)$, i.e.the first model is less efficient than the second. We can reject H_0 and conclude that M_1 outperforms M_2 if the resulting P-value is lower than a fixed threshold of 0.05. We use $SMAPE$ as $Loss$ to better account for the different price scales. To make the experiments reproducible, the source code and the data are made available[7].

5.1 Results

Flow estimate The results of the flow estimation problems on the test set are first presented in Table 2. For comparison, we also calculated the error between the network constraints **A** and the actual flows. We make the same observation as for the train set: **Flin** barely improves the quality of the flows while **Flsq** dramatically reduces the error. Then, their combination **Fcmb** does not shows notable metric improvement while setting up one-sided flows **Fos** does. We perform DM tests that confirms that the flow estimate quality increases with the complexity of the estimation method i.e. **Fos** outperforms every method, **Fcmb** outperforms every method except **Fos** and **Flsq** is better than **Flin**.

[7] https://github.com/Leonardbcm/OPALE.git

Price forecast The results of the EPF problem on the test period are presented in Table 3. The left part display the metrics, while the right part of details the P-values of the DM tests. On each line, we first compare the model on the line with the same model using other flow estimates (first 5 columns), then we compare it to other models using the same flow estimate (last 3 columns). We can for instance confirm that the **DNN** model using the network constraints **A** is significantly more efficient than the **CNN** using **A** (first line).

The **CNN** models are less competitive. They obtain the worst metrics and the DM test confirms that they are significantly less efficient than other models using the same flows (penultimate column). The **GNN** models are the most adequate models for this problem. Their metrics are better and the DM test statistically confirms that they outperform other models using the same flows (last column). The **DNN** models thus stand in between. We now analyse the performance variations with respect to the flow estimation method. We compare results obtained using the network constraints **A** and those using estimation methods **F** (4th column). We notice that, except for **Flin**, estimating the flows significantly improves performances for all models. Moreover, the **Flin** method is significantly less efficient than every other (5th column). However, which flow estimation method is better for all models remains unclear. The best flow estimate for the **DNN** model is the **Flsq** (3rd row), **Fcmb** (9th row) for the **CNN**, while for the **GNN**, it is impossible to statistically decide between **Fos** and **Flsq**, despite metric differences.

Detailing the DM tests by zone, we observe that replacing **A** by **Flsq**, **Fcmb** or **Fos** leads to overall improvements, even though local decrease can occur (FR, HU, CZ, SK, SI, NO-5, DE, AT). Using **Flin** improves performances less often than other methods and can degrade forecasts on multiple neighboring areas (Italy for the CNN). **Fcmb** shows the biggest improvements and the lowest decrease for all models. Using **Fcmb** in a EPF model seems to be a reasonable default choice. Lastly, almost all zones profit from using **F** for the **DNN**.

5.2 SHAP Values

It is possible to further analyze our models and determine the impact of the different groups of features on the predictions. To that end, we consider the SHAP value approach [10], a feature attribution method that assigns to each feature a value that reflects its contribution in the prediction process. We denote the contribution of a column c to the target o on day d as $\Phi_c^{d,o}$. A column $c = (f, h, z)$ refers to the feature f at hour h for zone z or pair of zones (z, z') if f is an edge attribute. Hence, the contribution tensor $\Phi \in \mathbb{R}^{731\times792\times6385}$ is made of 3.7 billion values. For computational issues, we only compute 500 SHAP values on the first 30 days of the test dataset. We normalize the results so that the sum of each contribution equals 1 for each target of a given day to obtain $\bar{\Phi}_c^{d,o}$, and the sum of the contributions for each feature f is denoted $\bar{\Phi}_f$.

We compute $\bar{\Phi}_f$ for each $f \in (\mathbf{C}, \mathbf{G}, \mathbf{R}, \mathbf{P}, \mathbf{F})$ and display them in Table 4. First, we observe that the **GNN**'s top contributing features are the prices that explain 30% of the forecasts, against approximately 20% for the other models.

Models	CC	MAE (€/MWh)	SMAPE (%)	A	Flin	Flsq	Fcmb	Fos	DNN	CNN	GNN
DNN_A	0.893	13.97	29.76	-	1.0	1.0	1.0	1.0	-	**0.008**	1.0
DNN_Flin	0.903	13.44	28.51	**0.0**	-	1.0	1.0	0.998	-	**0.0**	1.0
DNN_Flsq	0.904	12.96	28.26	**0.0**	**0.0**	-	**0.001**	**0.0**	-	**0.0**	1.0
DNN_Fcmb	0.906	13.07	28.38	**0.0**	**0.0**	0.999	-	**0.001**	-	**0.0**	1.0
DNN_Fos	0.909	13.2	28.84	**0.0**	**0.002**	1.0	0.999	-	-	**0.0**	1.0
CNN_A	0.866	14.41	32.17	-	**0.035**	0.992	1.0	0.958	0.992	-	1.0
CNN_Flin	0.865	14.54	32.23	0.965	-	1.0	1.0	0.998	1.0	-	1.0
CNN_Flsq	0.875	14.19	32.01	**0.008**	**0.0**	-	0.996	0.175	1.0	-	1.0
CNN_Fcmb	0.867	14.04	31.81	**0.0**	**0.0**	**0.004**	-	**0.002**	1.0	-	1.0
CNN_Fos	0.872	14.26	31.87	**0.042**	**0.002**	0.825	0.998	-	1.0	-	1.0
GNN_A	0.925	10.23	24.59	-	0.819	0.981	0.928	1.0	**0.0**	**0.0**	-
GNN_Flin	0.926	10.22	24.6	0.181	-	0.957	0.884	0.996	**0.0**	**0.0**	-
GNN_Flsq	0.925	10.17	24.6	**0.019**	**0.043**	-	0.31	0.942	**0.0**	**0.0**	-
GNN_Fcmb	0.926	10.18	**24.46**	0.072	0.116	0.69	-	0.937	**0.0**	**0.0**	-
GNN_Fos	**0.926**	**10.14**	24.52	**0.0**	**0.004**	0.058	0.063	-	**0.0**	**0.0**	-

Table 3. (Left) Metrics on the test period. (Right) DM test's P-values. For each trained model (line), the P-value is computed against the same model with other flows (first 5 columns) and against other models with the same flows (last 3 columns). The null hypothesis states that the column model outperforms the row model. With a threshold of 0.05, the bold values indicate that the row model outperform the column model.

Model	DNN					CNN					GNN				
	FA	**Flin**	**Flsq**	**Fcmb**	**Fos**	**FA**	**Flin**	**Flsq**	**Fcmb**	**Fos**	**FA**	**Flin**	**Flsq**	**Fcmb**	**Fos**
C	19.4	19.2	19.3	19.3	18.8	19.3	19.2	18.9	18.8	18.6	16.7	17.0	16.7	16.7	17.0
G	20.4	20.5	20.0	20.0	19.5	20.1	20.6	10.7	20.5	20.0	18.1	18.0	18.0	18.0	18.3
R	21.5	21.6	20.7	20.7	20.1	22.1	21.5	21.0	20.9	22.3	19.4	19.1	19.5	19.6	19.5
P	20.5	20.5	20.7	20.5	20.3	20.1	20.5	21.3	21.7	20.4	31.6	31.5	31.0	30.9	30.6
F	18.5	18.2	19.3	19.4	21.2	18.4	18.1	18.1	18.2	18.7	14.2	14.4	14.8	14.8	14.5

Table 4. Average contribution (%) for the predictions grouped by feature. For the **DNN** and **CNN** models, we observe that the average contribution of the flows **F** increases as we use more sophisticated estimation methods.

The **GNN** also uses **F** the less (14% against 18-20%). Next, we observe that the **DNN** model favors the use of **F** at the expense of **C**, **G** and **R** as we use more sophisticated flow estimate (**Flsq**, **Fcmb**, **Fos**). In contrast, the average contribution of **F** in the **CNN** and **GNN** does not show a clear pattern. To detail these observations, we display in Figure 3 the differences of contribution between **A** and the used estimate **F**. Green squares on coordinate (i, j) indicate that the contribution of **F** is more important than the contribution of **A** for predicting the zone i for model j. We observe that the **Flin** contribution differences are mostly negative i.e. models rely less on **Flin** than on **A** for forecasting prices. Next, we see that the **DNN** increases the contribution of **F** for almost all zones. Finally, the **CNN** always lowers the contribution of Spain (ES), Portugal (PT), and Italy (CNOR, CSUD and SARD). These zones are characterized by having few (1 or 2) connections. Latvia (LT) has a similar behavior for the **GNN** model.

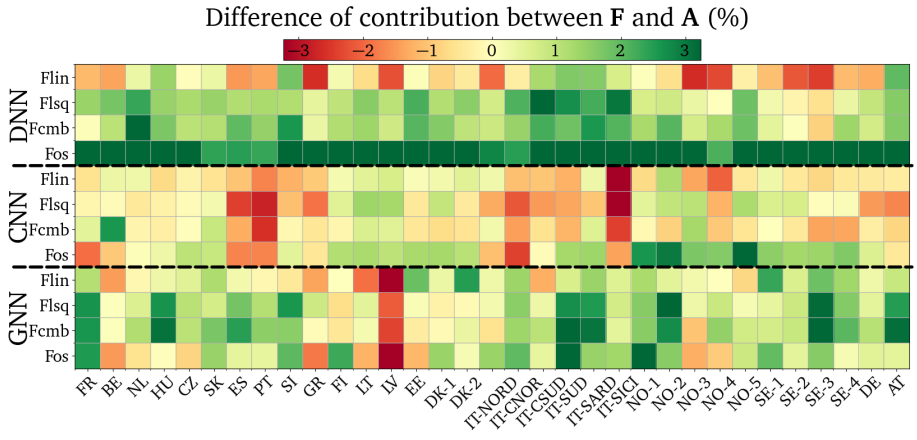

Fig. 3. Difference in contribution made by the flow estimates **F** compared to the available transfer capacity **A** for the different models and zones. The green (resp. red) squares indicate that **F** contributes more (resp. less) than **A**.

5.3 Discussion

The joint analysis of the model's performances and SHAP values of **Flin** shows us significant degradation of the forecasts and less contribution for the forecast than **A**. This leads us to conclude that the **Flin** method is not a good flow estimation method. Apart from **Flin**, other flow estimation methods are all beneficial for the EPF task, without being able to select the best one overall. The **DNN** is the less sophisticated model and cannot model the network. However, observing both a significant performance improvement and an increase of the average contribution of **F** over **A** for almost all zones, we infer that the **DNN** model takes benefit from using flow estimation methods. Next, the **CNN** use a matrix of arbitrary-arranged input features and a convolution kernel and dilation rate inconsistent with the European network. Consequently, zones and flows are not associated. Hence, **CNN** is the less tailored model for EPF. Lastly, the GNN model uses the graph representation of the network, with connections modeled as edges. This ability lowers the contribution of **A** or **F** in the forecast: node embeddings are already updated using their neighbors even with no flows. This is even more the case for isolated zones as their relationships with other zones are simpler. Another consequence is that **GNN** is the best model for EPF at the European scale.

6 Conclusion

In this paper, we introduce the problem of day-ahead electricity price forecasting considering many zones together and their interdependence due to price regulation mechanisms. While many works have focused on the construction of increasingly

sophisticated models for specific regions of the European market, we propose new ways of estimating features based on domain knowledge, and this upstream of learning. We show that an *optimize then predict* strategy makes it possible to improve the learned models by fully considering cross-border energy flows estimated by several optimization problems. A SHAP-value analysis confirms that the estimated flows contribute more to the prediction than the Available Transfer Capacities, especially when the model is simple (DNN). For more sophisticated models such GNN, flows better influence predictions at the center of the European market while being less important for the zones at the periphery of the market.

Two main directions can be considered as future work. First, we could replace the generation forecast used in our models by a start-up/shut-down cost model for power plants. It would better capture dynamics between generation and day-ahead prices. Going further, we could also model part of the EUPHEMIA algorithm. Then, our work brings forward the question of mixing optimization problems and Machine Learning. Integrating the optimization problem as a layer in our Neural Network to achieve a *Optimize and Predict* framework would directly link the task loss (day-ahead price forecast) to the sub-task (flow estimation).

Acknowledgment. This research has partially been funded by the ANRT (French National Association for Research and Technology).

References

[1] H.-Y. Cheng, P.-H. Kuo, Y. Shen, and C.-J. Huang. Deep convolutional neural network model for short-term electricity price forecasting, 2020.

[2] F. Diebold and R. Mariano. Comparing predictive accuracy. *Journal of Business and Economic Statistics*, 20:134–44, 02 1992.

[3] O. El Balghiti, A. N. Elmachtoub, P. Grigas, and A. Tewari. Generalization bounds in the predict-then-optimize framework. In *Neurips*, volume 32, 2019.

[4] Z. Khan, S. Fareed, M. Anwar, A. Naeem, H. Gul, A. Arif, and N. Javaid. Short term electricity price forecasting through convolutional neural network (cnn). In *WAINA*, 02 2020.

[5] A. Krizhevsky, L. Sutskever, and G. E. Hinto. Imagenet classification with deep CNNs. Technical report, University of Toronto, 2012.

[6] J. Lago, G. Marcjasz, B. De Schutter, and R. Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021.

[7] J. Lago, F. D. Ridder, and B. D. Schutter. Forecasting day-ahead electricity prices deep learning approaches and empirical comparison of traditional algorithms. Technical report, Delft University of Technology, 2018.

[8] J. Lago, F. D. Ridder, P. Vrancx, and B. D. Schutter. Forecasting day-ahead electricity prices in europe: The importance of considering market integration. *Applied Energy*, 211:890–903, feb 2018.

[9] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Graph convolutional recurrent neural network: Data-driven traffic forecasting. *CoRR*, abs/1707.01926, 2017.

[10] S. Lundberg and S. Lee. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874, 2017.

[11] J. Mandi, V. Bucarey, M. Mulamba, and T. Guns. Predict and optimize: Through the lens of learning to rank. *CoRR*, abs/2112.03609, 2021.

[12] H. Mosbah and M. El-Hawary. Hourly electricity price forecasting for the next month using multilayer neural network. *Canadian Jour. of Elect. and Comp. Engin.*, 39, 09 2015.

[13] PCR. Euphemia public description. Technical report, Price Coupling of Region, 2016.

[14] L. Tschora, E. Pierre, M. Plantevit, and C. Robardet. Electricity price forecasting on the day-ahead market using machine learning. *Applied Energy*, 313:118752, 2022.

[15] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014.

[16] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao. Exploiting multi-channels deep CNNs for multivariate time series classification. Technical report, Univ. of China Hefei, 2015.