



HAL
open science

Forecasting Electric Vehicle Charging Station Occupancy: Smarter Mobility Data Challenge

Yvenn Amara-Ouali, Yannig Goude, Nathan Doumèche, Pascal Veyret, Alexis Thomas, Daniel Hebenstreit, Thomas Wedenig, Arthur Satouf, Aymeric Jan, Yannick Deleuze, et al.

► To cite this version:

Yvenn Amara-Ouali, Yannig Goude, Nathan Doumèche, Pascal Veyret, Alexis Thomas, et al.. Forecasting Electric Vehicle Charging Station Occupancy: Smarter Mobility Data Challenge. 2023. hal-04119408

HAL Id: hal-04119408

<https://hal.science/hal-04119408>

Preprint submitted on 7 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

Forecasting Electric Vehicle Charging Station Occupancy: Smarter Mobility Data Challenge

Yvenn Amara-Ouali
EDF R&D

Yannig Goude
EDF R&D

Nathan Doumèche
EDF R&D

Pascal Veyret
EDF R&D

Alexis Thomas
Ecole des Mines de Paris

Daniel Hebenstreit
Graz University of Technology

Thomas Wedenig
Graz University of Technology

Arthur Satouf
CY Tech

Aymeric Jan
SLB, AI Lab

Yannick Deleuze
Veolia S&TE

Paul Berhaut
Air Liquide

Sébastien Treguer
INRIA

Tiphaine Phe-Neau
Renault Digital

Abstract

The transport sector is a major contributor to greenhouse gas emissions in Europe. Shifting to electric vehicles (EVs) powered by a low-carbon energy mix would reduce carbon emissions. However, to support the development of electric mobility, a better understanding of EV charging behaviours and more accurate forecasting models are needed. To fill that gap, the Smarter Mobility Data Challenge has focused on the development of forecasting models to predict EV charging station occupancy. This challenge involved analysing a dataset of 91 charging stations across four geographical areas over seven months in 2020-2021. The forecasts were evaluated at three levels of aggregation (individual stations, areas and global) to capture the inherent hierarchical structure of the data. The results highlight the potential of hierarchical forecasting approaches to accurately predict EV charging station occupancy, providing valuable insights for energy providers and EV users alike. This open dataset addresses many real-world challenges associated with time series, such as missing values, non-stationarity and spatio-temporal correlations. Access to the dataset, code and benchmarks are available at <https://gitlab.com/smarter-mobility-data-challenge/tutorials> to foster future research.

1 Introduction

Electric mobility. The transport sector is currently one of the main contributors to greenhouse gas emissions in Europe [1]. The development of electric vehicles (EVs) combined with a low-carbon energy mix would make it possible to considerably reduce these emissions. Fortunately, the EV initiative is one of the few programmes listed by the International Energy Agency (IEA) that aligns with the IEA's net-zero emissions goals. In 2021, China led global EV sales with 3.3 million units, tripling its 2020 sales, followed by Europe with 2.3 million units, up from 1.4 million in 2020. The U.S. market share of electric vehicles doubled to 4.5%, with 630,000 units sold. Meanwhile, electric vehicle sales in emerging markets more than doubled [2]. Electric mobility development entails new needs for energy providers and consumers [3]. Companies and researchers are proposing innovative solutions including pricing strategies and smart charging [4–8]. However, their implementation

requires a precise understanding of charging behaviours and better EV charging models are necessary in order to better understand the impact of EVs on the grid [9, 10]. In particular, forecasting the occupancy of a charging station can be a critical need for utilities to optimise their production units according to charging demand. On the user side, knowing when and where a charging station will be available is critical, but large-scale datasets on EVs are rare [11, 12]. This is why the dataset provided in this challenge is valuable in itself. This challenge aims at testing statistical and machine learning forecasting models to forecast the states of a set of charging stations in the Paris area at different geographical resolutions.

Hierarchical forecasting The problem of the challenge presents an intrinsic structure, called hierarchy, since it consists in forecasting quantities at increasing scales (stations, areas, and global). Hierarchical time series forecasting has been studied for various other applications where the data is directly hierarchically organised or where there is a latent hierarchical representation. For example, in retail, goods are often classified into categories (such as food or clothing) and inventory management can be done at different geographical (national, regional, shop) or temporal (week, month, season) scales. Moreover, electricity systems often have an explicit (electricity network) or implicit (customer types, tariff options...) hierarchy. Recent work shows that exploiting this structure can improve forecasting performance at different levels of hierarchy. For instance, [13] focuses on tourism demand, [14] on macroeconomic forecasting, and [15–17] on electricity consumption data.

Overview. The paper is structured as follows. Section 2 describes the forecasting problem at hand and the baseline models. Section 3 presents the methods proposed by the three winning teams. Finally, Section 4 summarizes the findings and discusses our results. The full dataset, as well as the benchmark consisting of the baseline models, the winning solutions, and the aggregations, are available at <https://gitlab.com/smarter-mobility-data-challenge/tutorials> and distributed under the Open Database License (ODbL).

2 EV charging dataset and target

Dataset description The dataset is based on the real-time charging station occupancy information of the Belib network, which is available on the Paris Data platform (ODbL) [18]. The Belib network consists of 91 charging stations in Paris, each offering 3 plugs. Because Paris Data does not store the data it daily produces, the EDF R&D team has set up a pipeline to collect this data every 15 minutes from July 2020 on the platform’s dedicated API. The data is then stored in a data lake based on Hadoop technologies (HDFS, PySpark, Hive, and Zeppelin). For the purpose of the challenge, the data has been divided into a training set and a test set. The training set contains D_{train} points ranging from 2020-07-03 00:00 to 2021-02-18 23:45 CET. The test set contains D_{test} points ranging from 2021-02-19 00:00 to 2021-03-10 23:45 CET. The test set has been divided into two subsets: a public set for validation purposes and a private set used to quantify the performance while minimising the risk of overestimating the performance of potential overfitting methods. To create the public and the private sets, the dataset was split into three time periods. The first is assigned to the public set, and the third to the private set. Then, we randomly draw points from the second period to assign 20% of them to the public set and the rest to the private set. Another challenge related to the availability of the data is the occurrence of missing values after 2020-10-22, as illustrated by Figure 1.

At any given time, a plug takes one of four states. Either it is plugged into a car and provides electricity, corresponding to state c (charging). Either it is connected to a car that is already fully charged, corresponding to state p (passive). Either the plug is free, corresponding to state a (available). The special state o (other) regroups the cases where the plug does not work. We denote by $y_{t,k} = (a_{t,k}, c_{t,k}, p_{t,k}, o_{t,k}) \in \{0, 1, 2, 3\}^4$ the vector representing the state of station $k \in \{1, \dots, 91\}$ at time t , where $a_{t,k}$ is the number of available plugs, $c_{t,k}$ the number of charging plugs, $p_{t,k}$ the number of passive plugs, and $o_{t,k}$ the number of other plugs, at station k and time t . Of course, $a_{t,k} + c_{t,k} + p_{t,k} + o_{t,k} = 3$. The additional variables associated with station k are

- temporal data: *date*, *tod*, *dow*, and *trend*,
- spatial data: *latitude*, *longitude*, and *area* (south, north, east, and west) of the station.

The *dow* and *tod* features return respectively the day of week (1 for Monday to 7 for Sunday) and the time of day, corresponding to the 15 minute step position in the day (0 for 00:00:00 to 95 for

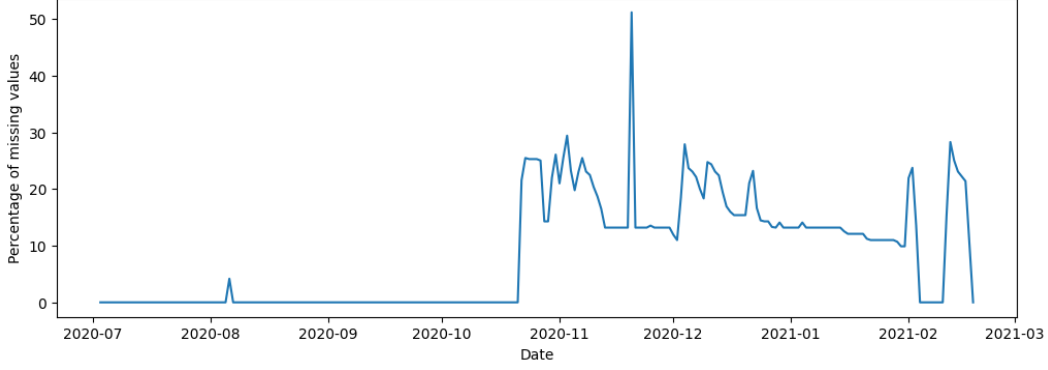


Figure 1: Percentage of missing values per day



Figure 2: The 91 stations (yellow dots on the left) and the 4 areas of Paris (colored on the right)

23:45:00). The *trend* feature returns the numerical conversion of the time index. The data is then aggregated into 4 areas of about 20 stations each, as shown in Figure 2.

Evaluation The goal is to forecast the states of the different plugs at 3 hierarchical levels: individual stations (denoted by $y_{t,i}$, for $i \in \{1 \dots 91\}$), areas ($y_{t,south}$, $y_{t,north}$, $y_{t,east}$, and $y_{t,west}$), and global ($y_{t,global}$), where $y_{t,zone} = \sum_{i \in \text{zone}} y_{t,i}$ is the sum of the plugs per state in a zone (south, north, east, west, or global). Therefore, let $z_t = (y_{t,1}, \dots, y_{t,91}, y_{t,south}, y_{t,north}, y_{t,east}, y_{t,west}, y_{t,global})$ be the aggregated matrix containing the statutes of all stations at the different hierarchical levels at time t . The goal is therefore to provide the best estimator \hat{z} of z . Performance is evaluated by the the following loss function, which encodes each hierarchical level as a penalty.

$$L(z_t, \hat{z}_t) = \ell_{\text{station}}(z_t, \hat{z}_t) + \ell_{\text{area}}(z_t, \hat{z}_t) + \ell_{\text{global}}(z_t, \hat{z}_t), \quad (1)$$

where

$$\begin{aligned} \ell_{\text{station}}(z_t, \hat{z}_t) &= \sum_{k=1}^{91} \|y_{t,k} - \hat{y}_{t,k}\|_1, \\ \ell_{\text{area}}(z_t, \hat{z}_t) &= \sum_{\text{zone} \in \{\text{south}, \text{north}, \text{east}, \text{west}\}} \|y_{t,\text{zone}} - \hat{y}_{t,\text{zone}}\|_1, \\ \ell_{\text{global}}(z_t, \hat{z}_t) &= \|y_{t,\text{global}} - \hat{y}_{t,\text{global}}\|_1, \end{aligned}$$

and where $\|x\|_1 = \sum_{k=1}^p |x_k|$ is the usual ℓ^1 norm on \mathbb{R}^p . Note that $L(z_t, \hat{z}_t) = \|z_t - \hat{z}_t\|_1$.

Baseline models An initial baseline of two simple models is established before evaluating the performance of more complex solutions. The first naive estimator of z_t consists in taking the median of each coefficient of z_t over D_{train} for a given value of the temporal data. This amounts to computing the median per type of day and quarter-hour, i.e.,

$$\hat{z}_t = \text{median}_{t' \in \text{Cal}_t} \{z_{t'}\},$$

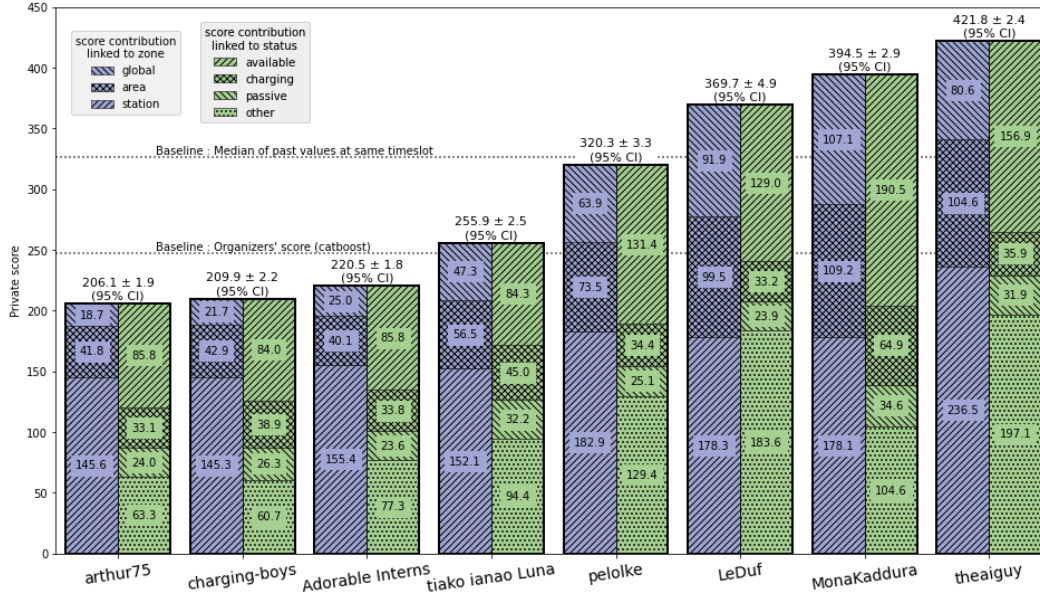


Figure 3: Ranking of the competitors

where

$$Cal_t = \{t' \in D_{train}, \text{dow}(t') = \text{dow}(t)\} \cap \{t' \in D_{train}, \text{tod}(t') = \text{tod}(t)\}.$$

Notice that the set Cal_t corresponds to the timestamps of the same day of the week and the same hour of the day. The second baseline model is the parametric model called CatBoost. It is a tree-based gradient boosting algorithm that specializes in regression for categorical data. It is implemented using the python library CatBoost [19] and has demonstrated excellent performance for a great variety of regression tasks [20–22]. The performance of these models is shown by the dotted lines in Figure 3.

3 Winning Solutions

This section describes the methods used by the three winning teams. The ranking of the competitors is shown in Figure 3, where the confidence intervals are constructed by bootstrapping. A subsection is dedicated to each of the winning teams, as their approaches are quite different and informative for the analysis of the dataset. In the last subsection, their strengths are combined thanks to aggregation methods.

3.1 Arthur Satouf (team Arthur75)

Handling missing values As shown in Figure 1, the dataset presents a lot of missing data. Common techniques were considered to impute these [23], including computing the mean by station, forward and backward filling, simple moving average, weighted moving average, and exponential moving weighted average (EMW) [24]. These techniques are evaluated by measuring the mean absolute error (MAE) on a validation subset of the training set. As a result, the EMW is the most effective technique, and it is thus implemented for both forward and backward filling approaches. Specifically, we use the last 8 known values to forward fill the first 8 missing values. The same procedure is applied to backward filling.

Model selection The benchmark consists of usual time series models [25–27], such as SARIMAX, LSTM, XGBoost, random forest, and CatBoost. The evaluation metric used is the MAE, and the time series cross-validation technique is applied to evaluate the performance of the models [28, 29]. The CatBoost algorithm is ultimately chosen for its fast optimization relying on parallelization and its ability to handle categorical data without preprocessing. As explained in Section 2, the states of any station k satisfy at any time t the equation $a_{t,k} + c_{t,k} + p_{t,k} + o_{t,k} = 3$, which is enforced in the CatBoost estimator as follows.

Table 1: Example of a data conversion to a string

Given station at a given time	Available	Charging	Passive	Other	Target
14h15-16/08/2021	1	2	0	0	1200
14h30-16/08/2021	0	1	1	1	0111
14h45-16/08/2021	0	0	3	0	0030

- At the station level, the problem is transformed from a multi-task regression problem to a classification problem. This is achieved by concatenating the values of each task as a string, resulting in 20 unique classes. In this approach, the sum of the four vectors always equals three, given that there are three plugs. After predicting a given target, the target is decomposed into four values. Table 1 provides an example.
- At the area level, CatBoost was also used as a regression problem, as shown in Figure 4. However, each area had its own model, and each area used a combination of CatBoost regressor and Regressor-Chain [30]. Regressor-Chain involves building a unique model for each task and using the result of each task as an input for the next prediction model. The output of each model, along with the previous output, is then used as input for the next task. This approach helps to keep the sum of plug equal to the right number and takes into account the correlation between tasks, making the prediction more robust.
- At the global level, the approach is similar to the one applied to the area level, with only 4 models as there are no longer areas.

A time series cross validation is used once again to tune the hyperparameters and to validate the models. It relies on the mean absolute percentage error [31] at the area and the global levels, and on the F-measure [32] at the station level. In total, 21 CatBoost models are used to forecast the private datasets.

Area => CatBoostRegressor + RegressorChain

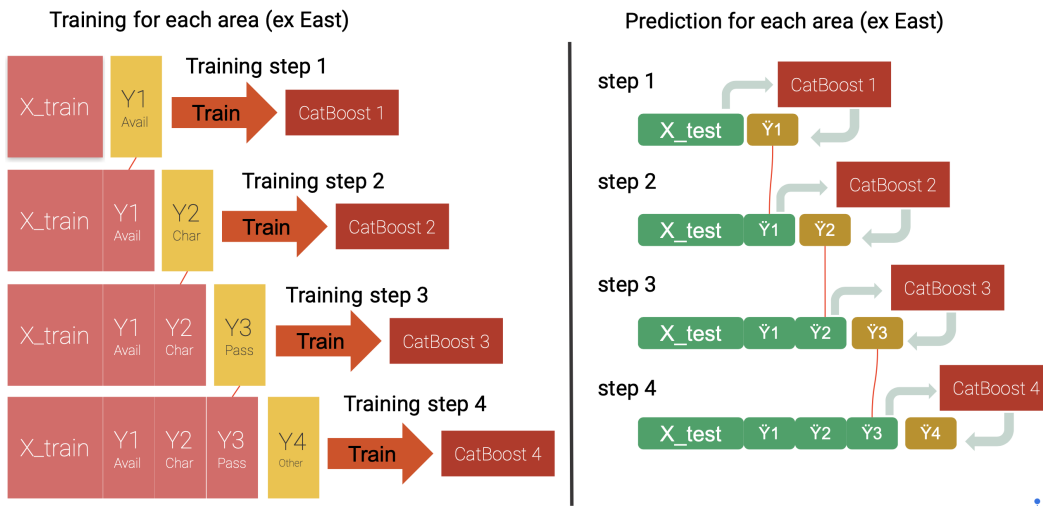


Figure 4: Using Regressor Chain with CatBoost-Regressor to train and forecast "area" data set.

3.2 Thomas Wedenig and Daniel Hebenstreit (team Charging-Boys)

Data exploration, preprocessing and postprocessing Our exploratory experiments does not show any signs of a trend within the time series. Regarding stationarity, we run the Augmented Dickey–Fuller test [33] on the daily averages of the target values for each station and find inconclusive results. Therefore, we cannot assume stationarity for all target-station pairs, which is why we employ differencing in the construction of our ARIMA model. As usual in statistical frameworks, we assume

that the noise interferes with the high frequencies of the signal. To denoise, we preprocess the time series by computing a rolling window average with a window size of 2.5 hours [34].

During our data exploration, we encounter a significant change in the behavior of the individual stations in the end of October 2020, just before the COVID-19 regulations were enforced in Paris. We also assume that several stations were turned off after this event, as labels were missing over large time intervals. Thus, we experiment with different methods of missing value imputation, but find that simply dropping the timestamps with missing values performs best. We also add custom features, namely a column indicating whether the current date is a French holiday, as well as sine and cosine transforms of *tod*, *dow*, the month, and the position of the day in the year. To ensure that our regression models return integer outputs that sum to 3 for each station and timestamp (since stations have exactly 3 plugs), we round and rescale these predictions in a post-processing step.

Tree-based regression model Using `skforecast`, we train an autoregressive XGBoost model [26] with 100 estimators. We train it on all of the 91 stations individually, each having 4 targets, resulting in 364 models. Each model receives the last 20 target values, as well as the sine/cosine transformed time information as input, and predicts the next target value. We also discard all features that are constant per station (e.g., station name, longitude, and latitude). The final regression model achieves a public leaderboard score of 177.67.

Tree-based classification model To effectively enforce structure in the predictions (i.e., that they sum to 3), we transform the regression problem discussed above into a classification problem. For a given station and timestamp, consider the set of possible target values $\mathcal{C} = \{\mathbf{x} \in \{0, 1, 2, 3\}^4 \text{ s.t. } \sum_{i=1}^4 x_i = 3\}$. We treat each element $c \in \mathcal{C}$ as a separate class and only predict class indices $\in \mathcal{I} = \{0, \dots, 19\}$ (since $|\mathcal{C}| = 20$). While \mathcal{I} loses the ordinal information present in \mathcal{C} , this approach empirically shows competitive performance. When training a single XGBoost classifier with 300 estimators for all stations, we achieve a public leaderboard score of 178.9. We also experiment with autoregressive classification (i.e., including predictions of previous timestamps), but find no improvement in the validation error.

ARIMA model We fit a non-seasonal autoregressive integrated moving average (ARIMA) model [35] for each target-station combination. To predict the value of a given target, we only consider the last $p = 2$ past values of the same target (in the preprocessed time series) and do not use any exogenous variables for prediction (e.g., time information). We apply first-order differencing to the time series ($d = 1$) and design the moving average part of the model to be of first-order ($q = 1$). We observe that the forecasts using these models have very low variance, i.e., each model outputs an approximately constant time series. These predictions achieve a competitive score on the public leaderboard (third place).

Ensemble method The final model is an ensemble of the tree-based regression model, the tree-based classification model, and the ARIMA model. For a single target, we compute the weighted average of the individual model predictions (per timestamp). The ensemble weights are chosen to be roughly proportional to the public leaderboard score ($w_{\text{reg}} = 0.35$, $w_{\text{class}} = 0.25$, $w_{\text{ARIMA}} = 0.4$). Since the predictions of the tree-based models have high variance, we can interpret mixing in the ARIMA model’s predictions as a regularizer, which decreases the variance of the final model. As the tree-based models also use time information for their predictions, we use the entirety of the available features.

3.3 Nathan Doumèche and Alexis Thomas (team Adorable Interns)

Data analysis Several challenges arise from the data, as shown in Figure 1. An interesting phenomenon is the emergence of a change in the data distribution on 2020-10-22, characterized by the appearance of missing data. A reasonable explanation is that the detection of missing values is due to an update in the software that communicates with the stations. The update would have taken place on 2020-10-22, allowing the software to detect new situations in which stations were malfunctioning. This hypothesis is supported by the fact that the stations with missing values are those that were stuck in states corresponding to the absence of a car, i.e., either the state *a* or the state *o* (see Figure 5). In fact, 88% of the stations that were stuck in either *a* or *o* for the entire week before 2020-10-22 had

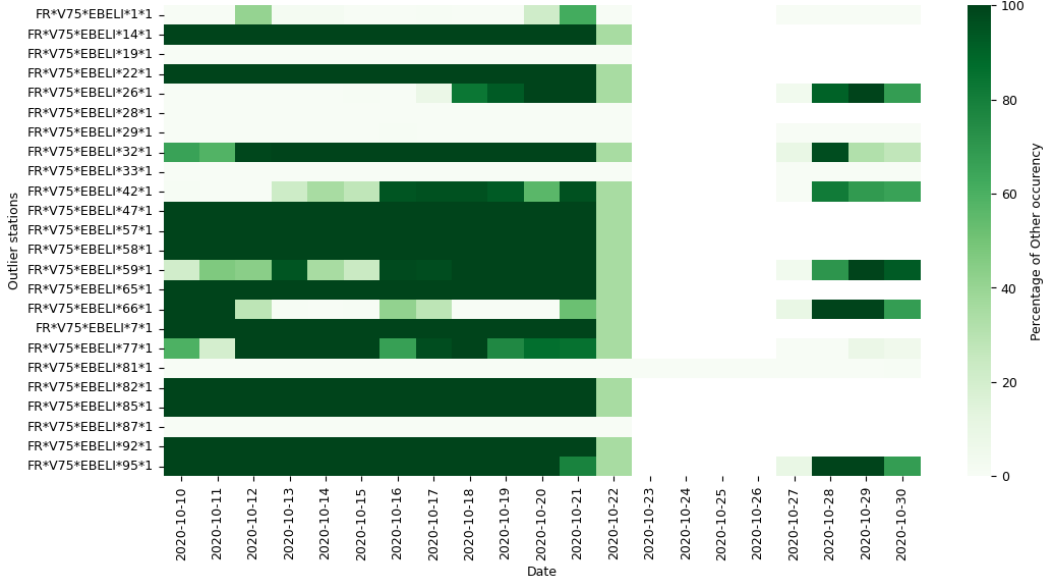


Figure 5: Percentage of state o occurrences per outlier per day around 2020-10-22

missing values on 2020-10-22. Perhaps the users avoided the malfunctioning stations, or perhaps the users tried to connect to the station, but the plug was unresponsive, so the users went undetected. An important implication of this hypothesis is that the data before the change should not be invalidated, since the behaviour of the well-functioning stations did not change. Another challenge of the dataset was its shortness. In fact, we expect a yearly seasonal effect due to holidays [36] that cannot be distinguished from a potential trend because there is less than one year of data. All these observations suggest giving more weight to the most recent data.

Empirical loss As usual in the supervised learning setting, we need to choose a model \mathcal{F} to construct the estimator $\hat{z}_t \in \mathcal{F}$. To estimate the entire D_{test} period at once, we cannot rely on online models such as autoregressive models or hidden-state neural networks (RNN, LSTM, transformers...), although they perform well for time series forecasting [37], and in particular for EV charging station occupancy forecasts [38, 39]. Once a model \mathcal{F} is chosen, we define an empirical loss L on the training data. Then, a learning procedure, such as a gradient descent, fits the estimator \hat{z} that minimizes L , with the hope that \hat{z} will minimize the expectation of the test loss (1) [40, 41]. Given a training set $T_{train} \subseteq D_{train}$, we consider two empirical losses. The first one is $L_{equal}(\hat{z}) = \sum_{t \in T_{train}} \|z_t - \hat{z}_t\|_1$. It gives equal weight to all data points. The second one is $L_{exp}(\hat{z}) = \sum_{t \in T_{train}} \exp((t - t_{max})/\tau) \|z_t - \hat{z}_t\|_1$, where $\tau = 30$ days and $t_{max} = 2021-02-19 00:00:00$. This time-adjusted loss function is common for non-stationary processes [42] because it gives more weight to the most recent observations. This makes it possible to give more credit to the data after the change in the data distribution and to capture the latest effect of the trend, while using as much data as possible.

Benchmark phase To compare the performance of the models, we define a training period T_{train} , which covers the first 95% of D_{train} , and a validation period T_{val} , which covers the last 5%. Models are trained on T_{train} to minimize L_{equal} or L_{exp} , and then their performance is evaluated on T_{val} by $L_{val}(\hat{z}) = \sum_{t \in T_{val}} \|z_t - \hat{z}_t\|_1$. The *Mean* model estimates $\hat{y}_{t,k}$, $\hat{A}_{t,k}$ and \hat{G}_t by their mean over the training period for each value of (tod, dow) . Idem for the *Median* model. They are robust to missing values since the malfunctioning of a station k only affects $\hat{y}_{t,k}$. We compare them with the CatBoost model presented in Section 2. Let $C(d, i)$ be the CatBoost model of depth d trained with i iterations using L_{equal} , and $C_{exp}(d, i)$ the same model trained using L_{exp} . In this setting, we train twelve CatBoost models: one for each pair of state (a, c, p, o) and hierarchical level. After hyperparameter tuning, we found $C(4, 150)$ and $C_{exp}(5, 200)$ to be the best models in terms of tradeoff between performance and number of parameters, knowing that early stopping and a small number of parameters prevent overfitting [see, e.g., 43]. All of these models take advantage of the fact that malfunctioning stations tend to stay in specific states.

Table 2: Evaluation of the performance of the Adorable Interns’ models in both phases

	Mean	Median	$C(4, 150)$	$C_{exp}(5, 200)$
Benchmark Phase	316	309	292	261
Validation Phase	323	303	233	189

Validation phase The contest organizers allowed participants to test their models on a subset T_{val} of D_{test} . In this phase, we trained our best models on the entire D_{data} period and sent them to be tested with the test loss (1). Table 2 shows that the hierarchy of the models is preserved. The submitted model was therefore $C_{exp}(5, 200)$. Note that this model is also interesting because its small number of parameters ensures robustness and scalability. In addition, tree-based models are quite interpretable, which is paramount for operational use [44].

3.4 Aggregation of the teams’ forecasts

Naive aggregations of uncorrelated estimators are known to have good asymptotic [45] and online [46] properties. In practice, they often achieve better performance than the individual estimators [see, e.g., 47, 48]. Once again, this is the case in this challenge, as evidenced by Table 3, which shows the performance of the top 3 teams compared to two aggregation techniques. The uniform aggregation –denoted by *Uniform agg.*– corresponds to the mean of each team’s prediction, while the weighted aggregation –denoted by *Weighted agg.*– is computed by gradient descent using the MLpol algorithm [49] to minimise the error on the training set. Notice how the weighted aggregation outperforms the other forecasts for the total loss, as well as for all the subdivisions of the loss by hierarchical level and by state.

Table 3: Score by target of the top 3 teams and aggregations

	Available	Charging	Passive	Other	Station	Area	Global	Total
Arthur75	86	33	24	63	145	42	19	206
Charging Boys	84	39	26	61	145	43	22	210
Adorable Interns	86	34	24	77	155	40	25	220
Uniform agg.	83	33	22	63	142	40	20	202
Weighted agg.	82	33	22	59	137	40	19	196

4 Summary of findings and discussion

This paper presents a dataset in the context of hierarchical time series forecasting of EV charging station occupancy, providing valuable insights for energy providers and EV users alike. The results discussed in this paper focus on two key aspects of the Smarter Mobility Data Challenge: data preprocessing and training the hierarchical forecasting model. As with any real-world data set, specific techniques are required to deal with missing data and outliers (see, e.g., Section 3.1). Data preprocessing was a crucial step, and the addition of relevant exogenous features, such as the national holidays calendar, significantly improved the results, in contrast to the use of a more complex model that could not perform well without prior augmentation of the data samples. All three winning solutions described in this paper were robust enough to maintain a high private test score, showing good generalization of the models. The choice of the empirical cost function that drives the training process produced the best results when more recent data points were given greater weight (see, e.g., Section 3.3). Aggregating the forecasts of the three winning teams even yielded a better global score, with a notable improvement at the station level. The hierarchical models presented are promising and could help improve the overall EV charging network. This open dataset is interesting because it encompasses many real-world problems related to time series matters, such as missing values, non-stationarities, and spatiotemporal correlations. In addition, we believe that sharing the benchmark models derived from this challenge will be useful for making comparisons in future research.

Limitations The deployment of electric vehicles (EVs) is progressing at a remarkable pace [50], making any dataset merely a snapshot of a swiftly evolving world [see also 51]. To enhance forecasting

accuracy, additional features could be incorporated into a dataset. Numerous covariates, such as mobility and traffic information, meteorological data, and vehicle characteristics, could be included. In a forthcoming release of the dataset, we intend to incorporate traffic data and meteorological data.

Ethical concerns To the best of our knowledge, our work does not pose any risk of security threats or human rights violations. Knowing when and where someone plugs in their EV could lead to a risk of surveillance. However, this dataset does not contain any personal information about the user of the plug or their car, so there is no risk of consent or privacy.

Acknowledgments

We thank Cédric Villani, Jean-Michel Poggi, and Marc Schoenauer for being part of the jury and for their insightful comments on the algorithms and on the paper. We thank Jerome Naciri, and Jean-Yves Moise for their help in organizing the challenge. Moreover, we thank all the contestants for their original solutions. Finally, the authors thankfully acknowledge the *Manifeste IA* network of French industrial companies and the TAILOR European project on trustworthy AI for founding this challenge.

References

- [1] International Energy Agency. Transport and environment report, 2022. URL <https://www.eea.europa.eu/publications/transport-and-environment-report-2022/transport-and-environment-report>. Accessed on March 2nd, 2023.
- [2] International Energy Agency. Electric vehicles, 2022. URL <https://www.iea.org/reports/electric-vehicles>. Accessed on March 2nd, 2023.
- [3] Réseau de Transport d'Électricité. Futurs énergétiques 2050 : les scénarios de mix de production à l'étude permettant d'atteindre la neutralité carbone à l'horizon, 2022. URL <https://www.rte-france.com/analyses-tendances-et-prospectives/bilan-previsionnel-2050-futurs-energetiques#Lesdocuments>. Accessed on March 2nd, 2023.
- [4] D. Dallinger and M. Wietschel. Grid integration of intermittent renewable energy sources using price-responsive plug-in electric vehicles. *Renewable and Sustainable Energy Reviews*, 16(5): 3370–3382, 2012.
- [5] Q. Wang, X. Liu, J. Du, and F. Kong. Smart charging for electric vehicles: A survey from the algorithmic perspective. *IEEE Communications Surveys & Tutorials*, 18(2):1500–1517, 2016.
- [6] M. Alizadeh, H.-T. Wai, M. Chowdhury, A. Goldsmith, A. Scaglione, and T. Javidi. Optimal pricing to manage electric vehicles in coupled power and transportation networks. *IEEE Transactions on Control of Network Systems*, 4(4):863–875, 2017.
- [7] Z. Moghaddam, I. Ahmad, D. Habibi, and Q.V. Phung. Smart charging strategy for electric vehicle charging stations. *IEEE Transactions on Transportation Electrification*, 4(1):76–88, 2018.
- [8] C. Crozier, T. Morstyn, and M. McCulloch. The opportunity for smart charging to mitigate the impact of electric vehicles on transmission and distribution systems. *Applied Energy*, 268: 114973, 2020.
- [9] O. Kaya, K.D. Alemdar, A. Atalay, M.Y. Çodur, and A. Tortum. Electric car sharing stations site selection from the perspective of sustainability: A gis-based multi-criteria decision making approach. *Sustainable Energy Technologies and Assessments*, 52:102026, 2022.
- [10] A. Ciociola, D. Giordano, L. Vassio, and M. Mellia. Data driven scalability and profitability analysis in free floating electric car sharing systems. *Information Sciences*, 621:545–561, 2023.
- [11] L. Calearo, M. Marinelli, and C. Ziras. A review of data sources for electric vehicle integration studies. *Renewable and Sustainable Energy Reviews*, 151:111518, 2021.

- [12] Y. Amara-Ouali, Y. Goude, P. Massart, J.-M. Poggi, and H. Yan. A review of electric vehicle load open data and models. Energies, 14:2233, 2021.
- [13] R.J. Hyndman, R.A. Ahmed, G. Athanasopoulos, and H.L. Shang. Optimal combination forecasts for hierarchical time series. Computational statistics & data analysis, 55(9):2579–2589, 2011.
- [14] G. Athanasopoulos, P. Gamakumara, A. Panagiotelis, R.J. Hyndman, and M. Affan. Hierarchical Forecasting, pages 689–719. Springer International Publishing, Cham, 2020.
- [15] T. Hong, J. Xie, and J.D. Black. Global energy forecasting competition 2017: Hierarchical probabilistic load forecasting. International Journal of Forecasting, 2019.
- [16] M. Brégère and M. Huard. Online hierarchical forecasting for power consumption data. International Journal of Forecasting, 38:339–351, 2022.
- [17] S.B. Taieb, J.W. Taylor, and R.J. Hyndman. Hierarchical probabilistic forecasting of electricity demand with smart meter data. Journal of the American Statistical Association, 116:27–43, 2020.
- [18] City of Paris. Paris data, 2023. URL <https://opendata.paris.fr/pages/home/>. Accessed on March 2nd, 2023.
- [19] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018.
- [20] E.A. Daoud. Comparison between xgboost, lightgbm and catboost using a home credit dataset. International Journal of Computer and Information Engineering, 13(1):6 – 10, 2019.
- [21] G. Huang, L. Wu, X. Ma, W. Zhang, J. Fan, X. Yu, W. Zeng, and H. Zhou. Evaluation of catboost method for prediction of reference evapotranspiration in humid regions. Journal of Hydrology, 574:1029–1041, 2019.
- [22] J.T. Hancock and T.M. Khoshgoftaar. Catboost for big data: an interdisciplinary review. Journal of Big Data, 7(94), 2020.
- [23] I. Pratama, A.E. Permanasari, I. Ardiyanto, and R. Indrayani. A review of missing values handling methods on time-series data. In 2016 international conference on information technology systems and innovation (ICITSI), pages 1–6. IEEE, 2016.
- [24] J. Stuart Hunter. The exponentially weighted moving average. Journal of Quality Technology, 18(4):203–210, 1986.
- [25] N.K. Ahmed, A.F. Atiya, N.E. Gayar, and H. El-Shishiny. An empirical comparison of machine learning models for time series forecasting. Econometric reviews, 29(5-6):594–621, 2010.
- [26] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794, 2016.
- [27] M.H.D.M. Ribeiro and L. dos Santos Coelho. Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. Applied soft computing, 86: 105837, 2020.
- [28] J.-P. Kreiss and E. Paparoditis. Bootstrap methods for dependent data: A review. Journal of the Korean Statistical Society, 40(4):357–378, 2011.
- [29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

- [30] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. volume 85, pages 254–269, 2009.
- [31] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi. Mean absolute percentage error for regression models. Neurocomputing, 192:38–48, 2016.
- [32] T.Y. Chen, F.-C. Kuo, and R. Merkel. On the statistical properties of the f-measure. In Fourth International Conference on Quality Software, 2004. QSIC 2004. Proceedings., pages 146–153. IEEE, 2004.
- [33] D.A. Dickey and W.A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. Journal of the American statistical association, 74(366a):427–431, 1979.
- [34] R.J. Hyndman and G. Athanasopoulos. Forecasting: principles and practice. OTexts, 2018.
- [35] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung. Time series analysis: forecasting and control. John Wiley & Sons, 2015.
- [36] Q. Xing, Z. Chen, Z. Zhang, X. Huang, Z. Leng, K. Sun, Y. Chen, and H. Wang. Charging demand forecasting model for electric vehicles based on online ride-hailing trip data. IEEE Access, 7:137390–137409, 2019.
- [37] L. Bryan and Z. Stefan. Time-series forecasting with deep learning: A survey. Philosophical Transactions of the Royal Society A, 379:20200209, 2021.
- [38] T.-Y. Ma and S. Faye. Multistep electric vehicle charging station occupancy prediction using hybrid lstm neural networks. Energy, 244:123217, 2022.
- [39] F. Mohammad, D.-K. Kang, M.A. Ahmed, and Y.-C. Kim. Energy demand load forecasting for electric vehicle charging stations network based on convlstm and biconvlstm architectures. IEEE Access, pages 1–1, 2023.
- [40] V. Vapnik. Principles of risk minimization for learning theory. In J. Moody, S. Hanson, and R.P. Lippmann, editors, Advances in Neural Information Processing Systems, volume 4. Morgan-Kaufmann, 1991.
- [41] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York, New York, 2nd edition, 2017.
- [42] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar. Learning in nonstationary environments: A survey. IEEE Computational Intelligence Magazine, 10(4):12–25, 2015.
- [43] X. Ying. An overview of overfitting and its solutions. Journal of Physics: Conference Series, 1168, 2019.
- [44] S.B. Jabeur, C. Gharib, S. Mefteh-Wali, and W.B. Arfi. Catboost model and artificial intelligence techniques for corporate failure prediction. Technological Forecasting and Social Change, 166:120658, 2021.
- [45] A.B. Tsybakov. Optimal rates of aggregation. In Bernhard Schölkopf and Manfred K. Warmuth, editors, Learning Theory and Kernel Machines, pages 303–313, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [46] N. Cesa-Bianchi and G. Lugosi. Prediction, learning, and games. Cambridge University Press, Cambridge, 2006.
- [47] C.S. Bojer and J.P. Meldgaard. Kaggle forecasting competitions: An overlooked learning opportunity. International Journal of Forecasting, 37(2):587–603, 2021.
- [48] T. McAndrew, Nutcha N. Wattanachit, G.C. Gibson, and N.G. Reich. Aggregating predictions from experts: A review of statistical methods, experiments, and applications. WIREs Computational Statistics, 13(2):e1514, 2021.

- [49] P. Gaillard, G. Stoltz, and T. van Erven. A second-order bound with excess losses. In Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, Proceedings of The 27th Conference on Learning Theory, volume 35 of Proceedings of Machine Learning Research, pages 176–196, Barcelona, Spain, 2014. PMLR.
- [50] S.P. Sathiyam, C.B. Pratap, A.A. Stonier, G. Peter, A. Sherine, K. Pragmaash, and V. Ganji. Comprehensive assessment of electric vehicle development, deployment, and policy initiatives to reduce ghg emissions: Opportunities and challenges. IEEE Access, 10:53614–53639, 2022.
- [51] C. Hecht, J. Figgenger, and D.U. Sauer. Predicting electric vehicle charging station availability using ensemble machine learning. Energies, 14(23), 2021.

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [\[Yes\]](#) See Section 1.
 - (b) Did you describe the limitations of your work? [\[Yes\]](#) See Section 4, paragraph **Limitations**.
 - (c) Did you discuss any potential negative societal impacts of your work? [\[Yes\]](#) See Section 4, paragraph **Ethical Concerns**.
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [\[Yes\]](#)
2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? [\[N/A\]](#)
 - (b) Did you include complete proofs of all theoretical results? [\[N/A\]](#)
3. If you ran experiments (e.g. for benchmarks)...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [\[Yes\]](#) See the end of the Abstract, or the end of the introduction to get the link to the GitLab account with the code, data, and instructions (as a Readme).
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [\[Yes\]](#) The global description of the algorithms is detailed throughout the paper. The specific choice of parameters is detailed in the code.
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [\[Yes\]](#) See Figure 3.
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [\[Yes\]](#) The data set is relatively small (about 180 Mb) and the algorithms of the benchmark are not computationally intensive. The experiments can be re-run on CPU.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? [\[Yes\]](#) See Section 2, paragraph **Dataset description**.
 - (b) Did you mention the license of the assets? [\[Yes\]](#) See Section 2, paragraph **Dataset description**.
 - (c) Did you include any new assets either in the supplemental material or as a URL? [\[N/A\]](#)
 - (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [\[Yes\]](#) See Section 4, paragraph **Ethical concerns**
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [\[Yes\]](#) See Section 4, paragraph **Ethical concerns**.
5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [\[N/A\]](#)
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [\[N/A\]](#)
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [\[N/A\]](#)