Forecast of electrical consumption in Industrial Kitchens

Vasco Andrade - Instituto Superior Técnico - University of Lisbon, Portugal
vasco.andrade@tecnico.ulisboa.pt

***Abstract*—Forecasting methods have been increasingly used in the power sector, for various purposes. Anticipating network disturbances increases the security of the power system by allowing preventive measures to be taken, to keep the system running. The forecast of active power consumption, combined with load flexibility management, allows for greater energy efficiency, especially at the building's level. Although this topic has already been researched in the literature with regard to the forecast of aggregate loads, it has not yet been substantially addressed in relation to the forecast of individual loads of electrical appliances.**

The main contribution of this thesis is the prediction of active power at the appliance level in industrial kitchens, both in the case of day-ahead and hour-ahead predictions. With this purpose, methodologies based on decision trees, gradient boosting and neural networks were applied. Post-processing algorithms were also proposed to reduce prediction errors.

The results obtained were satisfactory, mainly because this topic still presents very few advancements in the literature. A proposed post-processing technique also showed good performance, with an error reduction of up to 29% in some appliances, when compared to the reference techniques, and up to 50% when compared with the results without post-processing.

***Index Terms*—Load Forecast; Appliance; Weighted Prediction; Machine Learning; Day-ahead; Hour-ahead**

I. Introduction

With the increasing energy demand in the world and the objectives assumed by different governments to reduce carbon emissions, energy efficiency has become an intensively discussed topic. In the European Union (EU) only, energy production and use is responsible for 80% of all greenhouse gas (GHG) emissions, where buildings account for 40% of energy consumption and 36% of carbon emissions [1], indicating that it is possible to increase energy saving potential in this situation.

In the UK, the catering industry only is estimated to be responsible for an excess in consumption of 21 600 million kWh per year [2]. UK's industry represents about 20% of the total energy consumption in the UK and generates 32% of the UK's heat-related carbon emissions, mostly from fossil fuels [3]. Forecasting of power consumption is one of the most focused and studied areas in the past decades. This changed the energy systems in general and how power consumption is managed particularly. In industrial kitchens (IK) load forecasting becomes an important task. The activity of this type of industry requires the use of various types of equipment with high energy consumption rates for meal preparation, such as cooking, refrigeration, lighting, and hot water supplying [5], with refrigeration often representing the largest category of energy usage [2]. Energy efficiency in this type of building has not been intensively researched yet, especially when combined with forecasting techniques, and it presents an opportunity to try and develop power forecast approaches for this type of industry. Having accurate forecasts combined with load flexibilities and renewable energy systems can result in a considerable increase in the energy efficiency of these buildings. Researchers in [6] formulated seven possible reduction scenarios, and 5 of them also produced financial payback in one year or less. The most favorable scenario, with the replacement of fryers and reduction of freezing demand, resulted in an energy saving of 58%. In this paper, Machine Learning (ML) and Deep Learning (DL) algorithms are implemented to make day-ahead and hour-ahead predictions of the average power consumption of appliances in IK, to effectively understand resource consumption. Integration of the appliances' flexibilities with forecasting is also provided in this work. This study is integrated into the project *nexIK: Exploring the Human-Water-Energy Nexus in Industrial Kitchens* (https://nexik.tecnico.ulisboa.pt/).

II. State of the Art

In this section, it is presented an overview of the forecasting methods in the literature and some related work on areas of interest to the topic of IKs, particularly related to forecasting in commercial buildings or residential load forecasts. There are several research works in the field of forecasting, which usually follow at least one of the following approaches, that have been increasing its complexity: statistical, ML, and DL. However, the increasing complexity of the algorithms do not necessarily mean an improvement on the results obtained.

In the statistical approach, the methods used incorporate tools such as regressions and moving averages, resulting in the auto-regressive moving average (ARMA), auto-regressive integrated moving average (ARIMA), and seasonal auto-regressive integrated moving average (SARIMA). A major downside of the ARMA model is that it cannot handle non-stationary data. The integration element in the ARIMA is important to remove the non-stationarity [7], while the seasonal element in SARIMA improves the ARIMA model by giving a better interpretation of the temporal influence in the data.

The field of artificial intelligence (AI) presents both ML and DL approaches. Some of the most researched ML approaches are the k-Nearest neighbours (k-NN) regression, support vector machines (SVM), extreme gradient boost (XGBoost) and neural network (NN) based [8]. To make day-ahead predictions on the load profile of refrigerators, authors in [9] use clustering techniques alongside models Fuzzy-Rule based and SVM. Here, researchers use historical data relative to several months, allowing them to obtain and generate the daily profiles of 20000 refrigerators, with the Fuzzy-logic model outperforming the SVM.

Examples of DL methods are convolutional neural networks (CNN) and long short-term memory (LSTM). Operations such as convolution and memory units make these algorithms reliable and currently the state of the art in AI approaches. In [4] researchers test the LSTM model for short-term forecasting in residential loads, and by comparison with some benchmarks as k-nearest neighbours (KNN) regression and extreme learning machine (ELM), LSTM shows a better performance in terms of mean absolute percentage error. LSTM presents a mean absolute percentage error (MAPE) of 8.58% while k-NN presents a MAPE of 15.3%. The authors conclude that the LSTM method is more suited than other algorithms especially when the consistency in the pattern of the load decreases. DL methods are sometimes used in hybrid methods to try and achieve better performances. Authors in [10] develop a model combining a CNN with an LSTM for time-series predictions, resulting in an increased prediction quality when compared with the standard LSTM, especially for multivariate datasets.

Researchers in [8] implemented different types of models for load forecast in smart buildings, from statistical-based ARIMA and SARIMA to ML-based as random forests (RF) and XGBoost, and DL based such as LSTM. The results show a better performance of the XGBoost in short-term forecasts, not only in terms of accuracy but also in execution time. Another model based on gradient boosting decision trees (GBDT) is categorical boosting (CatBoost), which is specifically designed to handle high-dimensional data with categorical features [11]. It offers several features that make it convenient and efficient to work with such data and provides built-in feature importance and permutation importance calculations, allowing to assess the significance of the different features in the model. In the context of load forecasting, authors propose a two-layer CatBoost model [12]. The first layer involves factor correlation analysis, while the second layer consists of the load forecasting model, which uses CatBoost. This model presented superior performance when compared to XGBoost.

III. Proposed Methodology

A. Pre-processing of data

Data preprocessing involves transforming raw, real-world data into a consistent and understandable format [13]. In this study, the input data has a very low sampling period, which poses a challenge for forecasting since it requires predicting a large number of samples, making it computationally intensive. To address this, the data is resampled to higher periods, such as thirty minutes for day-ahead forecasts and five minutes for hour-ahead forecasts. This reduces the number of samples to forecast, with 48 samples for day-ahead predictions and 12 samples for hour-ahead predictions. The resampling is done by taking the mean value of the original data within each resampling period.

Smoothing filters such as median filters were applied to reduce noise caused by the changes in sampling rate and to smooth the signal. Median filters have excellent noise-reduction capabilities and give less blurring than linear filters [14]. In this study, a median filter with a window size of 7 is used. A simple filter is also applied to replace negative load values with zero values.

Authors in [15] use the method of sliding windows to reconstruct the data into a multidimensional matrix containing features and targets, in order to smooth the non-linearity and non-stationary behavior of load time series. For this work, the features considered are the historical values of power and the targets the load values to be predicted, but separated into two different matrices. Using data re-sampled, it results in a feature dataset comprising sequences of the historical W_{hist} load values, while the targets comprise sequences of the next W_{pred} values. To avoid train targets being included in the test set as features, a buffer zone in data was created between the end of the training set and the beginning of the test set, with the size of one historic window.

B. Forecasting models

Based on the results achieved in the previously mentioned studies and to analyze ML and DL algorithms, the models used in this work are XGBoost, CatBoost, and LSTM. A detailed explanation of these models is given below.

1) XGBoost: Extreme Gradient Boosting is an ensemble model that uses gradient boosting algorithms combined with decision trees to achieve high accuracy and efficiency in a variety of tasks, such as classification and regression, and is particularly effective for high-dimensional datasets, with many features. In this algorithm, decision trees (also called weak learners) are created in sequential form, perform a prediction and sequentially add them to the model [16]. By iteratively adding decision trees to the model, the final result is a strong, upgraded predictor. The mathematical formulation for XGBoost is detailed in equation 1.

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{1}$$

In the previous equation, K is the number of trees, f is the functional space of F, being F the set of possible decision trees. In XGBoost, each new learner is trained to minimize the residual error of the previous model. This is done by optimizing an objective function, which measures the difference between the predicted and actual values of the target variable [16]. One key feature of XGBoost to optimize the objective function is the use of regularizations to prevent overfitting. This helps to ensure that the model generalizes well to new data. The equation of the objective function with regularization terms is presented in 2 [17]. where the first term is the loss function and the second is the regularization parameter.

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

2) CatBoost: Categorical boosting has several similarities with XGBoost, but also some significant differences, especially in the way both handle categorical variables. XGBoost requires that categorical variables be pre-processed into numerical features before training, while CatBoost can handle categorical features directly without the need for pre-processing [11]. To handle these features, CatBoost introduces two additional terms: the categorical feature index j, and the categorical feature value v. The split gain is calculated for each categorical feature, and the feature with the highest gain is selected for the split [18].

3) LSTM: Long Short-Term Memory is one of the most used models in DL applications, including load forecasting. LSTM networks were introduced to overcome the problem of vanishing gradient in recurrent neural networks (RNN) by incorporating a special memory cell that can selectively remember or forget information over time. The memory cell is composed of three gates [19]: input, forget, and output gates.

The input to the LSTM is a sequence of vectors, such as a time series, and the output is also a sequence of vectors, the predicted values. The LSTM layers can be stacked to increase the depth of the network and improve its ability to capture complex dependencies in the input sequence, but it also increases the processing time of the network. A mathematical formulation of the nodes present in a cell is analytically described in equations (3) - (7) [10].

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf}X_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{4}$$

$$c_t = f_t c_{t-1} + i_t \, tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \tag{5}$$

$$o_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{6}$$

$$h_t = o_t \, tanh(c_t) \tag{7}$$

where c_t, i_t, f_t, o_t are respectively the state cell, input gate, forget gate, and output gate at the time instant t. Also, the final output from the LSTM unit is represented as h_t. x_t is the current sample and the activation function σ is the sigmoid function. This algorithm can be computationally expensive, especially for long input sequences.

C. Model optimization

1) Hyperparameter optimization: ML and DL algorithms rely on hyperparameters to perform the forecasts, that are not learned during the training process but defined beforehand, which can significantly affect the performance and thus the values of these hyperparameters must be selected carefully [20]. For decision tree-based algorithms, the parameters in analysis are learning rate, maximum depth of each tree, column sample by tree, subsample, and regularization parameters, while for the LSTM model, the parameters in analysis are the neurons in each layer, number of hidden layers, activation function and dropout in each neuron.

In this work, a Bayesian optimization is used to perform the tuning of the models. This type of optimization involves constructing a probabilistic model of the objective function, which is updated as more evaluations are performed. The model is then used to select the next set of hyperparameters to be evaluated, based on the expected improvement in performance. To perform this optimization, it is used the *Optuna* framework in *Python*. It provides a high-level interface for defining and running the search experiments, as well as a wide range of search algorithms for finding the best set of hyperparameters [21].

Apart from the parameters tuned using the Optuna library, for the GBDT-based algorithms the number of trees to be built was defined as $n_{estimators}$ = 3000. For the LSTM model, two deeply connected layers were also added, with the last layer presenting a number of neurons corresponding to the number of targets of the analysis. Also, the number of epochs for the LSTM to run was defined as 50. Parameters such as *early stopping rounds* = 50 and checkpoint callbacks were also used in order to reduce the training time of the models. The hyperparameter optimization of the algorithms was computed separately from the main algorithm, mainly because it is a very computationally expensive and time-consuming task [22].

2) Window size analysis: The historical window size is crucial to feed the model data from which it can extract the patterns of load consumption. Authors in [23] perform analysis ranging from 1 to 30 days of historic window size and identify the optimal length as 5 days, for day-ahead forecasts. Generally, the accuracy of the forecast model improves as the size of the training window increases, i.e. it is better to have sizeable historical data for training the prediction model so that it covers all possible patterns spanning time series [24]. However, a significant drawback of having large training data is, if the behavior of the underlying model changes, the trained model may not track the changes and result in erroneous readings. The values chosen to analyze the historic window size influence in the day-ahead analysis go from one to three days, while for the hour-ahead analysis ranges from two to twenty-four hours.

3) Training size: To evaluate the model performance for different training lengths, cross-validation based techniques were applied. Cross-validation is a widely used forecast evaluation technique to avoid potential overfitting issues [25]. A commonly used approach is based on an expanding window cross-validation. This technique performs cross-validation by fixing the length of the test set and expanding the length of the training set in each iteration [26]. Since the order of data points is crucial for the forecasts, the training subsets can only correspond to observations that occurred prior to the observations that form the test set. This way, this technique was used with the training set expanding in 4 iterations.

D. Post-processing of data

The output of the forecast models consists of 2D matrices, due to the use of the sliding window method, with each line corresponding to the forecast of the W_{pred} samples ahead of the current timestamp. This means that there is more than one prediction for each timestamp on the test set (except for the first and last timesteps). With this, it is possible to update the predicted values at each timestamp, based on techniques that consider the different values available for each timestamp. This was done considering different methods, as shown below:

- Most recent prediction
- Persistence-based prediction
- Average value of predictions
- Weighted predictions

Using the most recent prediction for a timestamp is straight forward assuming that the prediction for a point in time t is given by the last value predicted for time instant t. In other words, is the prediction made at the time instant $t-1$, which should present the prediction with the most up-to-date information.

The optimization using a persistence model takes as the main assumption that for each prediction windowm the results of the forecasting model present a trend in the results. It assumes that the first prediction for the time instant t presents a similar error in relation to the first prediction for the time instant $t+W_{pred}$. This method consists in analyzing the first W_{pred} timestamps, and finding which prediction presents the lowest error for each t. For $W_{pred}+t$, it is assumed that the most accurate prediction is at the same relative distance in time as it was for the first W_{pred} points analyzed.

The algorithm to compute the average predicted value for each timestamp takes into account how the sequences of predicted values are presented. The predicted values shape presents the same dimensions as the target values for the test set. Then, it is necessary to compute the average value on the diagonals of the matrix of predictions to obtain the average value for each timestep.

The weighted prediction method is based on the Roth-Erev algorithm, presented in equation (8), that considers two features: the previous coefficient value and the error made in the last prediction [27]. The algorithm works using a weight factor W that works as an inertia constant, with a value between 0 and 1. It is regulated based on the dependence of each appliance on the past values to accurately predict the future load. The closer the weight value is to 1, the larger the dependency on the past values.

$$C_{t+1}^{n} = C_{t}^{n} \times W - |(R_t - P_t^n)| \times (1-W) \tag{8}$$

The flowchart representative of this methodology is presented in Figure 1. In (8), coefficients C_{t+1}^{n} and C_{t}^{n} are one dimensional vectors, with C_{t}^{n} representing the mean errors between each prediction position n for the the historical values until t and test values. This way, C_{t+1}^{n} is also a 1-D vector, where its positions n are determined individually with the historic and actual values of error. Each position represents the prediction number of the respective time instant: if position n yields the lowest error for t, it means that the n_{th} prediction made for t is the most correct one. Then, this index is saved and used to set the n_{th} prediction as the most accurate value for $t+1$. This implies that the final prediction for time instant t can never present an error lower than the lowest error prediction for the same instant. The error values for $|(R_t - P_t^n)|$ are then added to the historic component. This process is carried out until the end of the prediction dataset is reached.

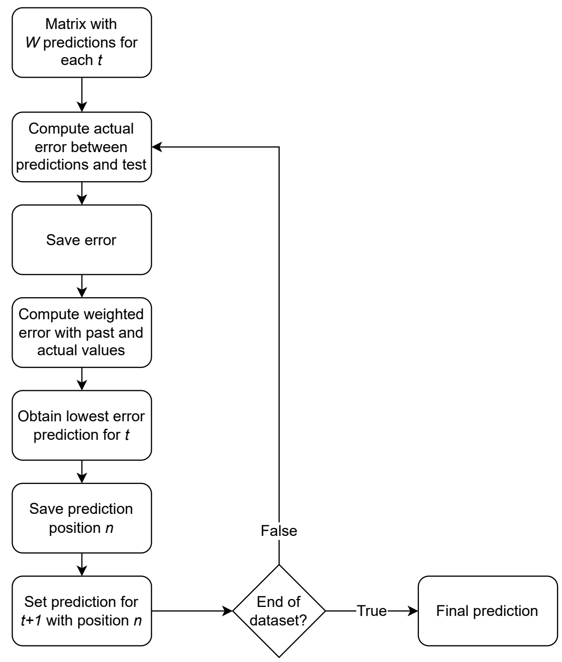

Fig. 1. Flowchart for weighted predictions

E. Validation

The predictions were evaluated in different stages: at the output of the forecast models, to provide an overview of the forecast quality, during the model optimization stage to obtain the best hyperparameters, historic window size, and training length, and finally, after the post-processing techniques, to provide a more conclusive result about the methodology. The metrics were computed with the average error for all prediction windows,

Three different metrics were used. The simplest metric consists in computing the mean absolute error (MAE) of each prediction. Another metric that is often used in load forecasting is the root mean square error (RMSE) [28], which emphasizes the peaks in the error. A normalized version of RMSE with relation to the maximum value on the respective dataset is also presented as the normalized root mean square error (NRMSE). The corresponding expressions of the metrics are presented in equations (9) - (11).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y(t) - \hat{y}(t)| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y(t) - \hat{y}(t))^2} \tag{10}$$

$$NRMSE = \frac{1}{y_{max}}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_(t) - \hat{y}(t))^2} \tag{11}$$

In the previous equations, $y(t)$ stands for the truth value of power at the instant t, $\hat{y}(t)$ is the predicted value for t, y_{max} is the maximum value on the dataset (determined after the pre-processing of data), and N represents the number of values.

IV. RESULTS

A. Data description

In this work, the datasets used are from two different kitchens in Funchal, Portugal. These kitchens have several appliances with different characteristics. The electricity consumption was measured using the e-Gauge at a sampling rate of 1/5 Hz. The period of measurement for kitchen 1 ranges from the sixth of February of 2019 (6^{th} of February 2019 00:00:00) to the fourth of March at 08:47 (4^{th} of March 2019 08:47:17). Since the last six hours of these datasets consist in zero values only, the period considered goes only to the 03:00:00 of the 4^{th} of March. For Kitchen 2, the period of measurement starts on the eleventh of March 2019 at 11:20 (11^{th} of March 2019 11:20:10) and ends on the third of April of 2019 at 08:45 (3^{rd} of April 2019 08:45:59). The datasets contain information about electric measurements such as voltage, current, reactive power, active power, and apparent power (in each line, for three-phase appliances).

B. Window size analysis results

For this analysis, the models' hyperparameters were kept constant and unchanged, while the training size was set as the first 19 days for K1 datasets and the first 18 days for K2. These values represent the maximum training size to fulfill the requirements set for the test set length. Tables I and II present the results obtained for the day-ahead and hour-ahead scenarios respectively.

TABLE I
HISTORIC WINDOW SIZE ANALYSIS - DAY-AHEAD (NORMALIZED ERROR)

Kitchen	Window size	XGBoost	CatBoost	LSTM
Kitchen 1	24h	16.05%	15.45%	**25.94%**
	48h	**15.52%**	**15.27%**	26.26%
	72h	16.07%	15.56%	27.88%
Kitchen 2	24h	**13.95%**	**13.83%**	**20.67%**
	48h	14.98%	14.93%	20.80%
	72h	15.03%	14.99%	21.27%

TABLE II
HISTORIC WINDOW SIZE ANALYSIS - HOUR-AHEAD (NORMALIZED ERROR)

Kitchen	Window size	XGBoost	CatBoost	LSTM
Kitchen 1	2h	10.15%	9.80%	13.75%
	4h	10.10%	9.77%	14.35%
	6h	10.19%	9.82%	14.06%
	8h	10.32%	9.91%	12.56%
	12h	10.15%	9.74%	**11.42%**
	24h	**9.65%**	**9.19%**	14.97%
Kitchen 2	2h	4.79%	4.64%	5.26%
	4h	4.69%	4.54%	**4.90%**
	6h	4.71%	4.6%	5.10%
	8h	4.77%	4.66%	5.75%
	12h	4.83%	4.71%	5.51%
	24h	**4.07%**	**3.99%**	6.44%

There are two different horizon lengths that better suit the forecast models, the 48-hour horizon for K1 and the 24-hour horizon for K2. For both kitchens LSTM model is showing poor performance and the GBDT models are performing with a similar accuracy. This way the horizon windows are set as 48 hours for K1 and 24 hours for K2.

As for the day-ahead scenario, hour-ahead results for the GBDT models present similar results throughout the different horizons tested, with the 24-hour horizon window size having the lowest error. LSTM model presents different optimal window sizes than the other models but with larger errors for both kitchens and for every window size. Then, the horizon window size was set as 24 hours for K1 and K2, in the hour-ahead scenario.

C. Best training length

The tests made in this analysis were performed maintaining the forecast models' hyperparameters unchanged and the historic window size as determined in the previous section. The splitting of the datasets resulted in training subsets of approximately 4 days and 18 hours for K1 and 4 days and 12 hours for K2. In Tables III and IV it is presented the average NRMSE of the appliances for the day-ahead and hour-ahead scenarios respectively.

TABLE III
DAY-AHEAD ANALYSIS - BEST TRAINING LENGTH

Kitchen	Training Subsets	XGBoost	CatBoost	LSTM
Kitchen 1	1	20.05%	19.32%	23.97%
	2	16.02%	16.10%	**23.41%**
	3	18.13%	17.69%	25.82%
	4	**15.52%**	**15.27%**	25.71%
Kitchen 2	1	**12.41%**	**11.91%**	**18.52%**
	2	13.76%	13.86%	21.22%
	3	14.06%	13.18%	20.70%
	4	13.95%	13.83%	20.84%

TABLE IV
HOUR-AHEAD ANALYSIS - BEST TRAINING LENGTH

Kitchen	Training Subsets	XGBoost	CatBoost	LSTM
Kitchen 1	1	12.56%	12.54%	13.53%
	2	9.75%	9.34%	**13.34%**
	3	10.02%	9.56%	15.43%
	4	**9.65%**	**9.19%**	15.85%
Kitchen 2	1	**4.02%**	4.10%	**5.40%**
	2	4.49%	4.52%	6.04%
	3	4.52%	4.49%	5.70%
	4	4.07%	**3.99%**	5.70%

In the day-ahead analysis, K1 benefits from the increase in the number of subsets used to perform the training of models. Despite the LSTM model presenting better results for the lesser number of subsets used, there is a considerable difference in performance relative to XGBoost and CatBoost. This way, the training length for K1 - day-ahead scenario - was set with the 4 folds, approximately 19 days. In K2 results the smallest training length presents consistent and better results for every model. Since the two kitchens are independent from each other: have different appliances and working periods, the training length for K2 - day-ahead scenario - was set with the first subset, around 4 days.

Hour-ahead analysis results for K1 present a similar behavior as for day-ahead, with the longer training length presenting better results for XGBoost and CatBoost. For K2 there is also a big difference in the results, with the best results of XGBoost and CatBoost being with 1 and 4 folds respectively. Since the lowest error in this analysis was obtained for 4 subsets and the difference in the results of XGBoost is not very significant, the training length for both kitchens was set as the same, with the 4 subsets used, approximately 19 days.

D. Day-ahead results

1) Kitchen 1: This section presents the results of the forecast models for the appliances in K1. Table V presents the average RMSE value between every prediction and target window, for the different methods and appliances. It is possible to conclude that the GBDT algorithms present similar results, with neither model presenting consistent and better results. Thus, it is impossible to identify whether XGBoost outperforms CatBoost or vice-versa clearly. The biggest improvement of XGBoost on accuracy was on the Dual Fryer, with a reduction of 10.6% in error in relation to CatBoost, while this model presented the biggest improvement in performance in the Freezer, with a reduction of 10.2% of the error in relation to XGBoost. On the other hand, the LSTM model presents a poor performance for the majority of the appliances, when compared with the two previous methods.

TABLE V
DAY-AHEAD RMSE [W] - KITCHEN 1

Device	**XGBoost**	**CatBoost**	**LSTM**
Blast Chiller	193.4	186.3	207.5
Convection Oven 1	246.6	248.5	281.0
Convection Oven 2	288.4	283.0	302.4
Dish Washer	160.3	162.9	203.0
Drinks Refrigerator	62.1	61.4	65.6
Dual Fryer	31.3	35.0	44.0
Fish Refrigerator	26.2	25.7	52.5
Freezer	54.1	48.6	203.6
Garde Manger 1	17.3	17.4	54.4
Garde Manger 2	11.0	10.7	26.6
Ice Machine	78.3	76.4	89.7
Infrared Shelf	192.1	202.4	216.1
Meat Refrigerator	26.3	25.3	52.3
Mise en Place	42.1	42.4	134.9
Salamander 1	932.9	908.4	919.2
Salamander 2	605.1	645.4	672.0
Vegetables Refrigerator	19.4	19.5	41.4

2) Kitchen 2: In Table VI it is presented the RMSE values of the different models, for each appliance. As for the results of K1, the forecast results in K2 are similar for the GBDT models, with only the Hothouse registering a considerable reduction of 17.8% of RMSE with XGBoost, in relation to CatBoost. The LSTM model continues to present a poor performance compared to the other two models. Despite presenting similar values of error for some appliances as the Blast Chiller or the Convection Oven, it presents errors two to four times bigger for other appliances, as is the case with Refrigerators.

TABLE VI
DAY-AHEAD RMSE [W] - KITCHEN 2

Device	**XGBoost**	**CatBoost**	**LSTM**
Blast Chiller	26.2	28.2	29.6
Convection Oven	110.4	112.7	118.5
Exhaust	59.0	59.4	86.6
Heating Lamp 1	1.1	1.0	1.0
Heating Lamp 4	0.2	0.6	0.3
Hothouse	152.2	185.1	255.4
Induction Plate	124.0	119.2	120.7
Microwave	0.5	0.6	0.0
Refrigerator 1	25.8	26.1	33.7
Refrigerator 2	20.1	18.7	41.1
Refrigerator 3	26.8	26.6	41.9
Refrigerator 4	15.0	14.1	46.2
Sous Vide	145.2	142.5	161.6

One factor that considerably influences these results is related to the nature of this type of forecast. In IK some appliances have very short operating times when compared with the length of a day, which makes detecting consumption patterns in short periods much harder since the forecast models will only detect the overall daily changes in the load. It is the case for Garde Mangers and some Refrigerators that require sampling periods deeply under 30 minutes. Also, having such a small dataset to perform day-ahead analysis also constrains the ability to perform this type of forecast. Models based on neural networks require large amounts of data to properly calculate the model parameters. This is mainly the reason why the LSTM-based model has a very poor performance in this analysis.

E. Hour-ahead results

1) Kitchen 1: The results of the forecast models for K1 appliances are presented in Table VII. The main difference in relation to the day-ahead analysis is that the LSTM model presents better performance in relation to the day-ahead analysis. Although the GBDT models indicate a higher accuracy for most of the appliances, the fact is that in most cases, the error difference in relation to the LSTM model is not drastic as it was for the day-ahead analysis. Apart from that, both GBDT models still present similar error values, as for the day-ahead analysis.

2) Kitchen 2: As for K1, the results of the different forecast models for K2 appliances, namely the average RMSE for each prediction window and respective standard deviation, are presented in Table VIII. Here XGBoost clearly outperforms the remaining models. It presents the best performance in the majority of the appliances, despite presenting similar or higher standard deviation values in relation to the remaining models. The biggest decrease in the error in relation to the CatBoost model is obtained for Sous-Vide, where the XGBoost model presents an improvement of 18.9%, while in relation to the LSTM model, it presents a decrease of 28.7% for the Blast Chiller.

Summarizing the results of the two kitchens for the hour-ahead analysis, GBDT models still present better results than the LSTM model, especially XGBoost. Despite having appliances in which LSTM presents better results, the differences

TABLE VII
HOUR-AHEAD RMSE [W] - KITCHEN 1

Device	XGBoost	CatBoost	LSTM
Blast Chiller	116.1	118.9	148.0
Convection Oven 1	309.3	311.6	350.0
Convection Oven 2	309.6	313.8	326.6
Dish Washer	253.5	259.6	272.8
Drinks Refrigerator	128.0	127.3	143.6
Dual Fryer	65.9	69.0	76.7
Fish Refrigerator	44.4	44.7	49.8
Freezer	44.4	42.8	340.1
Garde Manger 1	6.5	7.0	5.7
Garde Manger 2	0.9	0.9	0.7
Ice Machine	77.9	77.5	90.5
Infrared Shelf	108.5	104.9	120.9
Meat Refrigerator	18.5	18.2	19.1
Mise en Place	98.3	97.5	105.1
Salamander 1	307.8	299.4	338.3
Salamander 2	215.9	230.6	249.2
Vegetables Refrigerator	41.7	42.1	43.7

TABLE VIII
HOUR-AHEAD RMSE [W] - KITCHEN 2

Device	XGBoost	CatBoost	LSTM
Blast Chiller	33.5	36.8	47.0
Convection Oven	221.6	232.0	272.9
Exhaust	18.9	20.7	23.6
Heating Lamp 1	0.7	0.9	0.6
Heating Lamp 4	7.4	8.8	7.6
Hothouse	145.6	146.7	153.6
Induction Plate	76.7	79.9	88.1
Microwave	0.5	0.5	0.4
Refrigerator 1	24.6	27.0	30.4
Refrigerator 2	1.8	2.1	1.4
Refrigerator 3	6.7	4.8	7.6
Refrigerator 4	7.7	9.5	8.5
Sous Vide	92.5	114.1	95.4

in relation to the other models are not very significant, while the opposite is not always verified. There are appliances that achieve much better results with the GBDT models, such as the Convection Oven 1, Freezer and Salamander 1 in K1, or the Blast Chiller and the Convection Oven in K2.

F. Training times

In ML and DL models, hyperparameter optimization can significantly impact the training time required. For GBDT algorithms the main hyperparameters to affect the time required for model training are learning rate, maximum depth, and number of trees. Also, data re-sampled for the day-ahead analysis results in a smaller dataset than for hour-ahead, due to the respective re-sampling periods chosen. The length of the training set and historic window size also affect the training time of the models, because data with a smaller historic window will present fewer features, requiring less time to train. Since different appliances use different sets of hyperparameters, Table IX presents the average training time for each model, for day-ahead and hour-ahead analysis respectively.

The main conclusion is that the XGBoost model averages a lower training time in every situation, except for the day-ahead analysis in K2, where LSTM is the fastest model. Between the models that presented better results in the previous analysis, XGBoost clearly outperforms CatBoost in terms of training time, presenting an advantage, even for the situations where XGBoost had slightly lower accuracy than CatBoost.

TABLE IX
AVERAGE TRAINING TIMES [S]

Analysis	Kitchen	XGBoost	CatBoost	LSTM
Day-ahead	Kitchen 1	**16.2**	106.4	35.9
	Kitchen 2	8.8	18.1	**6.2**
Hour-ahead	Kitchen 1	**64.8**	68.7	254.2
	Kitchen 2	**61.0**	79.7	360.5

From the several analysis made and from the discussion on the results obtained, it is possible to conclude that XGBoost presents the more adequate choice to perform the forecasts in this work. Despite not presenting the best accuracy for all scenarios, it presents the best trade-off between prediction accuracy and model training time. This way, the post-processing methodologies were analyzed for the results obtained with XGBoost.

G. Post-processing results

First, the predictions were adjusted to identify and update values lower than 0. Then, the main goal is to determine which is the best method to update the prediction windows, for each time instant.

1) Day-ahead analysis: The results obtained for K1 and K2 appliances are presented in Tables X and XI respectively, indicating a clear improvement in the performance of the methodology, when compared with the error at the output of the forecast models. For both scenarios the weighted prediction method presents the best results, outperforming also the remaining post-processing techniques used. For K1 results, it presents a reduction of 35.8% in relation to the forecast results and 5.5% in relation to the last prediction. For K2 results it presents an improvement of 24.3% in relation to the forecast results and 9.9% in relation to the last prediction method.

TABLE X
RMSE [W] OF POST-PROCESSING METHODS FOR K1 - DAY-AHEAD ANALYSIS

Device	Forecast Results	Last Prediction	Persistence Prediction	Average Prediction	Weighted Prediction
Blast Chiller	193.4	144.1	148.5	190.5	**129.2**
Convection Oven 1	246.3	164.3	181.1	244.9	**158.4**
Convection Oven 2	288.4	**191.4**	224.7	283.8	192.0
Dish Washer	160.3	**112.5**	122.6	157.4	113.0
Drinks Refrigerator	62.1	48.3	54.8	61.1	**41.7**
Dual Fryer	31.3	24.0	27.5	30.6	**22.5**
Fish Refrigerator	26.2	16.1	19.1	25.5	**15.2**
Freezer	54.1	34.6	36.5	53.2	**31.8**
Garde Manger 1	17.3	15.2	15.8	17.0	**13.8**
Garde Manger 2	11.0	8.9	10.6	10.9	**8.2**
Ice Machine	78.3	51.6	56.5	76.3	**44.1**
Infrared Shelf	192.1	120.0	132.0	187.4	**98.7**
Meat Refrigerator	26.3	19.5	22.6	25.8	**18.0**
Mise en Place	42.1	30.4	33.9	41.2	**27.4**
Salamander 1	932.9	577.0	606.5	919.5	**536.6**
Salamander 2	605.1	453.5	472.1	600.5	**451.4**
Vegetables Refrigerator	19.4	16.4	18.2	19.0	**14.7**
Total Error	2986.8	2027.7	2183.3	2944.8	**1916.6**

TABLE XI
RMSE [W] OF POST-PROCESSING METHODS FOR K2 - DAY-AHEAD ANALYSIS

Device	Forecast Results	Last Prediction	Persistence Prediction	Average Prediction	Weighted Prediction
Blast Chiller	26.2	19.7	23.1	26.2	**18.8**
Convection Oven	110.4	103.8	102.7	104.9	**100.6**
Exhaust	59.0	48.2	46.9	57.5	**40.6**
Heating Lamp 1	**1.1**	**1.1**	**1.1**	**1.1**	**1.1**
Heating Lamp 4	0.2	**0.1**	**0.1**	0.2	0.2
Hothouse	152.2	130.1	127.7	148.3	**111.6**
Induction Plate	124.0	109.1	117.7	121.1	**100.8**
Microwave	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
Refrigerator 1	25.8	28.7	24.0	24.4	**21.2**
Refrigerator 2	20.1	18.2	17.5	19.2	**12.9**
Refrigerator 3	26.8	27.5	26.0	26.6	**24.0**
Refrigerator 4	15.0	14.6	12.8	13.0	**7.9**
Sous Vide	145.2	**91.6**	98.8	140.9	95.5
Total Error	706.4	593.3	592.9	683.9	**534.7**

Figure 2 presents the result of the post-processing methodologies for the Exhaust of K2. This figure presents the first point from each prediction window of the respective methods. Here it is possible to visualize the influence of the weighted prediction algorithm, especially when the device is switched ON.

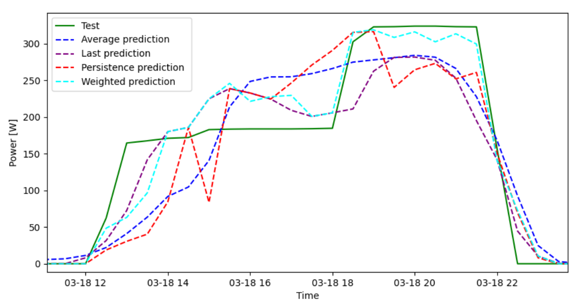

Fig. 2. Resuls for day-ahead analysis - Exhaust

2) Hour-ahead analysis: The results obtained with the application of the post-processing techniques for K1 and K2 appliances are presented in Tables XII and XIII respectively. Here, several appliances present equal errors in the last prediction and persistence-based prediction techniques. This happens when, for the first window of predictions in the test set, the most accurate value for each time instant is the most recent prediction made. In this analysis, the weighted prediction was not as effective as for the day-ahead analysis. For K1 results, the last prediction method presented a reduction of 50.2% in the error in relation to the forecast results and a reduction of 1.9% in relation to the weighted predictions. Regarding K2 results, the weighted predictions present an improvement of 50.3% in the error of the forecast results and an improvement of 1% in relation to the last prediction method.

Figure 3 presents the result of the post-processing methodologies for the Convection Oven 2 of K1. The results present a slight improvement on the results obtained with the weighted prediction method, in relation to the remaining post-processing methods, with the persistence-based predictions and the most recent predictions overlapped, as the results from Table XII indicate.

TABLE XII
RMSE [W] OF POST-PROCESSING METHODS FOR K1 - HOUR-AHEAD ANALYSIS

Device	Forecast Results	Last Prediction	Persistence Prediction	Average Prediction	Weighted Prediction
Blast Chiller	116.1	**42.8**	101.9	111.4	43.5
Convection Oven 1	309.3	**160.7**	225.6	299.8	163.6
Convection Oven 2	309.6	171.1	171.1	300.0	**168.4**
Dish Washer	253.5	174.6	174.6	247.5	**172.4**
Drinks Refrigerator	128.0	**65.8**	**65.8**	124.2	70.2
Dual Fryer	65.9	43.6	44.5	63.5	**42.6**
Fish Refrigerator	44.4	33.4	33.4	43.1	**30.8**
Freezer	44.4	31.5	41.1	41.5	**28.8**
Garde Manger 1	6.5	**6.3**	6.4	**6.3**	6.4
Garde Manger 2	0.9	**0.8**	**0.8**	0.9	**0.8**
Ice Machine	77.9	32.3	53.8	73.3	**31.3**
Infrared Shelf	108.5	**33.3**	83.4	102.4	34.7
Meat Refrigerator	18.5	**13.4**	**13.4**	18.1	**13.4**
Mise en Place	98.3	**82.6**	93.4	94.7	85.5
Salamander 1	307.8	**92.9**	**92.9**	294.3	104.4
Salamander 2	215.9	**51.0**	**51.0**	208.4	60.9
Vegetables Refrigerator	41.7	30.2	39.5	40.5	**28.8**
Total Error	2143.0	**1066.3**	1292.6	2070.0	1086.7

TABLE XIII
RMSE [W] OF POST-PROCESSING METHODS FOR K2 - HOUR-AHEAD ANALYSIS

Device	Forecast Results	Last Prediction	Persistence Prediction	Average Prediction	Weighted Prediction
Blast Chiller	33.5	19.6	19.6	31.2	**18.0**
Convection Oven	221.6	100.5	103.2	210.9	**97.9**
Exhaust	18.9	7.5	7.5	17.9	**6.9**
Heating Lamp 1	0.7	**0.6**	0.7	0.7	**0.6**
Heating Lamp 4	7.4	**3.8**	**3.8**	7.0	3.9
Hothouse	145.6	88.8	88.8	142.4	**86.8**
Induction Plate	76.7	**32.1**	**32.1**	72.8	32.4
Microwave	0.5	**0.2**	**0.2**	0.5	0.3
Refrigerator 1	24.6	**17.5**	21.3	24.2	18.5
Refrigerator 2	1.8	**1.6**	**1.6**	1.8	**1.6**
Refrigerator 3	6.7	**6.3**	**6.3**	6.7	6.4
Refrigerator 4	7.7	5.9	5.9	7.6	**5.6**
Sous Vide	92.5	**35.9**	35.9	88.2	38.4
Total Error	638.0	320.4	327.1	611.8	**317.4**

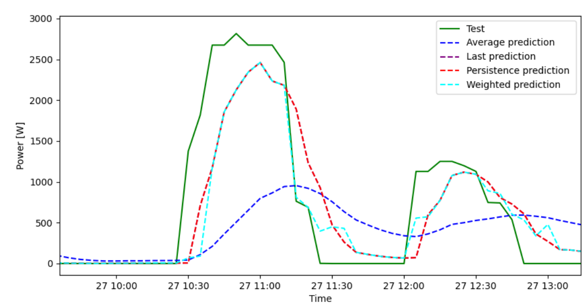

Fig. 3. Results for hour-ahead analysis - Convection Oven 2

V. Integration with Flexibilities

In the *nexIK* project flexibility was defined as the capacity of electric devices to adjust their energy consumption to support the electric grid and keep the used comfort [29]. Three parameters are considered when classifying the flexibility of a particular appliance: load shift, load reduction, and user comfort. Based on this and the appliances commonly found in IK, it was developed the classification presented in Table XIV

A. Methodology

The main goal of this methodology is to provide a framework for a combination of forecasts and flexibilities, with the

TABLE XIV
CLASSIFICATION OF IK APPLIANCES ACCORDING TO THEIR FLEXIBILITY POTENTIAL (SOURCE: [29])

Appliance	Not Flexible	Flexibility	
		Time	Power
Dishwasher		X	
Freezer		X	X
Refrigerator		X	X
Microwave	X		
Stove	X		
Oven	X		
HVAC / AC		X	X
Lighting			X

results of this analysis focused on the accuracy of the forecasts of the number of seconds in the ON state of each appliance. This way it is possible to perceive future flexibility in time to perform load-shifting operations.

The main feature of this methodology consists in separately forecasting the number of seconds in which an appliance is ON and the power consumption in the ON state, allowing to understand the potential load flexibility in time and how much power is available. This method was developed for appliances that present a clear ON/OFF behavior, as is the case with Refrigerators, Salamanders, and Garde Mangers, for example. Forecasting the duration of the ON states allows to apply variations to the different loads, by providing the user knowledge about the periods with greater power consumption. From inspection of the datasets, the appliances chosen for this analysis were the Refrigerators, Freezer, Garde Mangers, and Mise en Place from kitchen 1, and the Hothouse, Sous-Vide, and Refrigerators from kitchen 2. The Dish Washer of kitchen 1 was not included since this methodology requires that the appliances present an ON/OFF behavior, which is not the case with the data obtained for this appliance.

To obtain a clear view of the ON states of the appliances, the sampling rate was changed to one minute. This way it is possible to capture the activation patterns for all devices mentioned. However, this represents a problem in terms of execution time for the day-ahead analysis, where a sampling rate of one minute implies the forecast of 3600 samples. This way, the integration with flexibilities was limited to the hour-ahead scenario, with a change in the historical window size from one day to one hour, in order to reduce the number of samples in each historic window used to train the model.

B. Results

This section presents the results obtained for the forecasting of seconds in the ON state of the appliances mentioned in the previous section. The forecasts were performed using the same training lengths, with the historical and prediction windows presenting a size of 60 samples each. The forecast model used was XGBoost due to the good results presented for the previous analysis.

With a very small sampling period, the training time of the models must be taken into account. To apply the proposed post-processing techniques, the training time has to be lower than the sampling period, to allow the prediction update at each timestep. This way, the training data was considerably shortened, to 4 days in kitchen 1 and 3 days in kitchen 2.

TABLE XV
RMSE [S] OF POST-PROCESSING METHODS

Kitchen	Device	Forecast Results	Last Prediction	Weighted Prediction	Training Time [s]
Kitchen 1	Drinks Refrigerator	22.5	10.3	**8.0**	20.6
	Fish Refrigerator	10.5	**5.9**	6.1	19.7
	Freezer	10.7	6.1	**5.4**	35.9
	Garde Manger 1	4.7	**2.8**	3.1	24.6
	Garde Manger 2	13.2	**6.8**	7.3	33.9
	Meat Refrigerator	14.8	10.0	**9.7**	21.0
	Mise en Place	23.2	8.9	**8.8**	55.8
	Vegetables Refrigerator	4.0	**2.4**	2.8	9.8
Kitchen 2	Hothouse	6.0	**2.6**	4.3	25.1
	Refrigerator 1	18.4	7.0	**6.1**	37.6
	Refrigerator 2	17.1	**6.2**	9.1	27.6
	Refrigerator 3	20.5	**7.5**	21.1	21.1
	Refrigerator 4	17.9	**6.6**	**6.6**	55.6
	Sous Vide	3.6	**1.0**	**1.0**	28.0

These results represent the average error between the test windows and the predicted windows of the duration of the ON state. Regarding the accuracy of post-processing methodologies, the results vary among the different appliances, but with the last prediction method slightly outperforming the weighted prediction method. Regarding the training time, every appliance in this analysis presented a model training time lower than one minute. However, for the Mise en Place and Refrigerator 4 this duration is close to the limit, indicating that depending on the appliance, it might be necessary to modify the settings of the forecast, by reducing the complexity of the model or by changing the hyperparameters to values that increase the training speed of the model.

The post-processing techniques however might not be suitable to be applied in this situation, due to the sampling period used in this methodology and the training time of the forecast models. For these techniques to be applied, the training time has to be lower than the sampling period, to allow the prediction update at each timestep.

VI. CONCLUSION

Providing accurate load forecasts is useful for decreasing economic costs, increasing security and management of the network, and for the integration with different elements of the electric grid, such as renewable energy sources. Here, apart from the economic advantages, quality forecasts present a big step for the energy transition.

Section IV presents the results obtained for the case study, which included the analysis of two kitchens with 17 and 13 appliances each. With the purpose of increasing the performance of the forecast models to the maximum, different optimization procedures were adopted, such as determining the best historical window size of the features, the best training length, and the tuning of each model hyperparameters. Here it was verified that the day-ahead and hour-ahead analysis presented different optimal settings for each kitchen, namely in the day-ahead scenario where K1 and K2 had historic window sizes of two days and one day respectively.

Also, the training time analysis detailed in presented crucial results to evaluate the reliability of the forecast models.

Together with the forecast results, it was concluded that despite the different models presenting advantages and disadvantages, XGBoost presented the overall best performance/efficiency ratio.

Regarding the post-processing methodologies, the weighted prediction method showed high effectiveness in the day-ahead analysis, presenting an overall error reduction in K1 up to 5.5% in relation to the last prediction method and 35.8% in relation to the results without post-processing. In this analysis, some appliances presented considerable reductions in the error in relation to the last prediction, as is the case of the Blast Chiller (10.3%) and Infrared Shelf (17.8%). In K2 the weighted prediction method reduced the error up to 9.9% in relation to the last prediction method and 24.3% in relation to the results without post-processing, with the Refrigerator 1 presenting an error reduction of 29.3% in relation to the last prediction method. For the hour-ahead scenario, the weighted prediction method did not improve the results, with the last prediction technique presenting barely the same results. Still, both techniques presented an overall error reduction of up to 50% in relation to the results without post-processing.

REFERENCES

[1] M. Economidou, V. Todeschi, P. Bertoldi, D. D'Agostino, P. Zangheri, and L. Castellazzi, 'Review of 50 years of EU energy efficiency policies for buildings', *Energy and Buildings*, vol. 225, p. 110322, Oct. 2020, doi: 10.1016/j.enbuild.2020.110322.

[2] S. Mudie, E. A. Essah, A. Grandison, and R. Felgate, 'Electricity use in the commercial kitchen', *Int. J. Low-Carbon Tech.*, p. ctt068, Sep. 2013, doi: 10.1093/ijlct/ctt068.

[3] J. I. Chowdhury, Y. Hu, I. Haltas, N. Balta-Ozkan, G. Jr. Matthew, and L. Varga, 'Reducing industrial energy demand in the UK: A review of energy efficiency technologies and energy saving potential in selected sectors', *Renewable and Sustainable Energy Reviews*, vol. 94, pp. 1153–1178, Oct. 2018, doi: 10.1016/j.rser.2018.06.040.

[4] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, 'Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network', *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019, doi: 10.1109/TSG.2017.2753802.

[5] Nonaka, T., Shimmura, T., Fujii, N., Mizuyama, H. (2015). Energy Consumption in the Food Service Industry: A Conceptual Model of Energy Management Considering Service Properties. In: Umeda, S., Nakano, M., Mizuyama, H., Hibino, H., Kiritsis, D., von Cieminski, G. (eds) Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth. APMS 2015. IFIP Advances in Information and Communication Technology, vol 460. Springer, Cham. https://doi.org/10.1007/978-3-319-22759-7_69

[6] S. Mudie and M. Vadhati, 'Low energy catering strategy: insights from a novel carbon-energy calculator', *Energy Procedia*, vol. 123, pp. 212–219, Sep. 2017, doi: 10.1016/j.egypro.2017.07.244.

[7] P. M. R. Bento, J. A. N. Pombo, M. R. A. Calado, and S. J. P. S. Mariano, 'Stacking Ensemble Methodology Using Deep Learning and ARIMA Models for Short-Term Load Forecasting', *Energies*, vol. 14, no. 21, p. 7378, Nov. 2021, doi: 10.3390/en14217378.

[8] S. Hadri, Y. Naitmalek, M. Najib, M. Bakhouya, Y. Fakhri, and M. Elaroussi, 'A Comparative Study of Predictive Approaches for Load Forecasting in Smart Buildings', *Procedia Computer Science*, vol. 160, pp. 173–180, 2019, doi: 10.1016/j.procs.2019.09.458.

[9] P. Faria, A. Jozi, and Z. Vale, 'Forecasting Refrigerators Consumption to Support their Aggregated Participation in Demand Response', in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, Madrid, Spain: IEEE, Jun. 2020, pp. 1–6. doi: 10.1109/EEEIC/ICPSEurope49358.2020.9160527.

[10] A. Agga, A. Abbou, M. Labbadi, and Y. El Houm, 'Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models', *Renewable Energy*, vol. 177, pp. 101–112, Nov. 2021, doi: 10.1016/j.renene.2021.05.095.

[11] A. V. Dorogush, V. Ershov, and A. Gulin, 'CatBoost: gradient boosting with categorical features support'. arXiv, Oct. 24, 2018. Accessed: May 03, 2023. [Online]. Available: http://arxiv.org/abs/1810.11363

[12] W. Xiang, P. Xu, J. Fang, Q. Zhao, Z. Gu, and Q. Zhang, 'Multi-dimensional data-based medium- and long-term power-load forecasting using double-layer CatBoost', *Energy Reports*, vol. 8, pp. 8511–8522, Nov. 2022, doi: 10.1016/j.egyr.2022.06.063.

[13] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, 'Big data preprocessing: methods and prospects', *Big Data Anal*, vol. 1, no. 1, p. 9, Dec. 2016, doi: 10.1186/s41044-016-0014-0

[14] K.-J. Park and S.-Y. Son, 'A Novel Load Image Profile-Based Electricity Load Clustering Methodology', *IEEE Access*, vol. 7, pp. 59048–59058, 2019, doi: 10.1109/ACCESS.2019.2914216

[15] Q. Jiang, Y. Cheng, H. Le, C. Li, and P. X. Liu, 'A Stacking Learning Model Based on Multiple Similar Days for Short-Term Load Forecasting', *Mathematics*, vol. 10, no. 14, p. 2446, Jul. 2022, doi: 10.3390/math10142446.

[16] 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*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[17] X. Yao, X. Fu, and C. Zong, 'Short-Term Load Forecasting Method Based on Feature Preference Strategy and LightGBM-XGboost', *IEEE Access*, vol. 10, pp. 75257–75268, 2022, doi: 10.1109/ACCESS.2022.3192011.

[18] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, 'CatBoost: unbiased boosting with categorical features'.

[19] P. Li, K. Zhou, X. Lu, and S. Yang, 'A hybrid deep learning model for short-term PV power forecasting', *Applied Energy*, vol. 259, p. 114216, Feb. 2020, doi: 10.1016/j.apenergy.2019.114216.

[20] R. Khalid and N. Javaid, 'A survey on hyperparameters optimization algorithms of forecasting models in smart grid', *Sustainable Cities and Society*, vol. 61, p. 102275, Oct. 2020, doi: 10.1016/j.scs.2020.102275.

[21] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19). Association for Computing Machinery, New York, NY, USA, 2623–2631. https://doi.org/10.1145/3292500.3330701

[22] M. Massaoudi, S. S. Refaat, I. Chihi, M. Trabelsi, F. S. Oueslati, and H. Abu-Rub, 'A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting',*Energy*, vol. 214, p. 118874, Jan. 2021, doi: 10.1016/j.energy.2020.118874.

[23] J. Foster, X. Liu, and S. McLoone, 'Adaptive sliding window load forecasting', in *2017 28th Irish Signals and Systems Conference (ISSC)*, Killarney, Ireland: IEEE, Jun. 2017, pp. 1–6. doi: 10.1109/ISSC.2017.7983638.

[24] I. A. Khan, A. Akber, and Y. Xu, 'Sliding Window Regression based Short-Term Load Forecasting of a Multi-Area Power System', in 2019 *IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, Edmonton, AB, Canada: IEEE, May 2019, pp. 1–5. doi: 10.1109/CCECE.2019.8861915.

[25] Sylvain Arlot, Alain Celisse "A survey of cross-validation procedures for model selection," Statistics Surveys, Statist. Surv. 4(none), 40-79, (2010)

[26] R. J. Hyndman and G. Athanasopoulos, Forecasting: principles and practice. OTexts, 2018.

[27] Z. X. Jing, H. W. Ngan, Y. P. Wang, Y. Zhang and J. H. Wang, "Study on the convergence property of re learning model in electricity market simulation," 8th International Conference on Advances in Power System Control, Operation and Management (APSCOM 2009), Hong Kong, China, 2009, pp. 1-5, doi: 10.1049/cp.2009.1813.

[28] Nti, I.K., Teimeh, M., Nyarko-Boateng, O. et al. Electricity load forecasting: a systematic review. Journal of Electrical Systems and Inf Technol 7, 13 (2020). https://doi.org/10.1186/s43067-020-00021-8

[29] Oliveira, Ana, Ribeiro, Miriam, Rodrigues, Ricardo, Morais, Gonçalo, Morais, Hugo, and Pereira, Lucas, 'On The Role Of Industrial Kitchens In Sustainable Energy Systems: The nexIK Vision' In *CIRED 2023 - The 27th International Conference and Exhibition on Electricity Distribution* Jun 2023