Electrical Load Profile Forecasting

basque center for applied mathematics

Report

Leonel Fernando Nabaza Ruibal

August 26, 2025

Abstract

This report addresses the problem of predicting user consumption patterns rather than raw demand values, a perspective that has received relatively little attention in the load forecasting literature. The study first applies clustering methods to group daily consumption profiles into representative types, creating user categories that capture behavioral trends. The main objective is then to predict the type of consumption profile a user will exhibit, framing the problem as a classification task rather than a regression one. To this end, a manual implementation of a tree bagging algorithm was developed in MATLAB®, extended to random forests through feature randomization. The performance of the models was analyzed across different dataset configurations, with attention to the robustness of predictions. Beyond the technical evaluation, the report also explores broader applications of load forecasting—highlighting how predicting consumption patterns can inform demand-side management, tariff design, anomaly detection, and smart grid operations—thus situating the work within the emerging role of classification-based approaches in energy forecasting.

1 Introduction

With the rise of energetic demand around the world it is essential for both governments and companies to accurately determine the future patterns of consumption of the users, particularly in the electric sector. Many Machine Learning approaches to the prediction of electrical consumption are based on *regression*, which consists of trying to guess the numerical amount of energy that will be consumed [3, 16]. This is also known as *load forecasting*.

Nevertheless there is another field whose implications might be more impactful: predicting the shape -the behaviour- of a user's electrical consumption [17]. This is known as *profile load forecasting*. In this realm, aspects such as the number of consumption spikes in a day, the times in which they will occur, and the periodicity of the consumer's behaviour are dealt with [5]. In Machine Learning terms, this is studied as a *classification problem*.

This classification-based method offers several advantages over traditional regression-based forecasting. First, by identifying typical consumption patterns, it allows for user segmentation and clustering, which can be essential for personalized energy-saving recommendations, demand-side management, and tariff design [17, 10]. Second, it is less sensitive to random fluctuations in exact consumption values, focusing instead on broader behavioural trends, which tend to be more stable and informative over time. Third, the output of profile classification can feed into downstream tasks such as anomaly detection, resource allocation, and infrastructure planning [18].

Moreover, profile load forecasting has been shown to improve forecasting robustness in scenarios with limited historical data, such as for new consumers or consumers with solar installations [12].

The formalization of the problem is the following: given a set of points $\{(\mathbf{X}^{(i)}, Y^{(i)})\}_{i=1}^{n}$, where $\mathbf{X}^{(i)} \in \mathbb{R}^m$ and corresponds an observation that we have (e.g. the hourly consumption that we had the previous day and the previous week, the temperature, the weather, etc.) and $Y^{(i)} \in \{C_1, C_2, \ldots, C_k\}$, a set of consumption profiles (e.g. one profile establishes that there will two consumption peaks in the morning and the evening, another that there will be only one peak in the afternoon, another that the consumption will be almost null, etc.) we strive to find a model

$$\begin{array}{rccl} \hat{f}\colon & \mathbb{R}^m & \to & \{C_1, C_2, \ldots, C_k\} \\ & \mathbf{x} & \mapsto & \hat{f}(\mathbf{x}) \end{array}$$

such that it minimizes some error measurement, for example the accuracy[1].

For this research project, we implemented a model based on *bagged trees*, which applies bootstrap aggregation to a set of *decision trees* in order to learn and predict consumption patterns. An important aspect to highlight is that each prediction produced by the model carries an associated level of certainty. In other words, predictions are not just point estimates but come with a quantifiable degree of confidence. For instance, in a disease prediction scenario, it would be preferable for the model to output a set of likely diagnoses with high confidence (e.g., $\hat{f}(\mathbf{x}) = \{\text{flu, pulmonary infection, asthma}\}$) rather than a single, uncertain prediction. Therefore, beyond the use of bagged trees, it is crucial to incorporate methods such as *conformal prediction* to ensure reliable and interpretable results. In our case, we also implemented a method to make conformal predictions, based on [1].

[1]The ratio of correctly predicted cases with respect to all tested cases.

For both the training and testing of the developed model, a dataset of electrical power consumption on 114 apartments was used. Both its preparation and preprocessing will be explained in further sections. Once the data was prepared it was randomly divided into a training, calibration and testing datasets. The training dataset was used for fitting our model, the calibration set for computing parameters of interest for the conformal prediction, and the test set for assessing the accuracy of the model. Several configurations of the dataset were used, each involving different kinds of data.

2 Dataset

The used dataset consists of the collection of power consumptions from 114 apartments located in Amherst, Massachusetts [2]. There is data from 2014, 2015 and 2016, which consists of power measurements of each apartment at a different timestamp, and additional weather information. Both the information from 2014 and 2015 have a granularity of 15 minutes, while for 2016 is 1 minute. Moreover, both data from 2016 and 2015 start from first day of the year, while 2014 is missing more than the first half of the year. Finally, in 2014 there are missing data from several apartments and a high number of 'Not a number' (NaN) values and there are no signs that the numbering of the apartments is consistent during the years. That is why we decided only to use the data from 2016 which is the most complete dataset.

3 Data processing

For the preparation of the data, an aggregated table of these measurements, found in a Github repository from the Basque Center for Applied Mathematics (BCAM), was used. Via a MATLAB® script (`Corrected_Dataset_Preparation_2016.m`) the following transformations were applied to the table:

1. Remove those rows that absolutely no data from any apartment.
2. Convert timestamps to a date and hour format.
3. Group measurements by hour and average the consumption values for each apartment.
4. Create new columns in order to provide more information: day of the week, year, day of the year.

And the object `2016grouped_struct_of_vectors.mat` was obtained. Then, for the purpose of later having an easier manipulation, a struct of vectors was created. This struct consists of six vectors of same dimensions: apartment (the number of the apartment), time slot (hour of the measurement), day of the year, day of the week, year and consumption. The ith element of these vectors corresponds to the consumption of a certain apartment, at a particular hour, day of the week and year. The script for this part is in `v2_2016_Corrected_Final_Dataset_Creation.m` file and it outputted the object `data_all.mat`.

Once the struct was created, a new form of the dataset was created: a matrix where each row represents the hourly consumption during the day (24 consumptions) of each apartment and day of the year. If there was some missing value for some hour, the data for that day and apartment was discarded since the amount of complete data was large enough. Script `Creation_Dataset_For_Classification.m` was used and it created the object `Daily_consumptions.mat`.

This concludes the preparation of the data. Both the struct of vectors and last mentioned matrix were later used in the next part of the project, to further create new dataset.

4 Procedure

4.1 Consumption Datasets

The last matrix that we mentioned in the previous section contains, for each row, the hourly power consumption of an apartment during a day. In other words, each row can be considered as vector $\mathbf{v} = (v_1, \ldots, v_{24}) \in \mathbb{R}^{24}$. So, in order to create a set of load profiles -a set of typical consumption patterns- we grouped the vectors that were close among them. For this purpose, we applied a clustering algorithm to create these profiles. We applied $k-$means to the data (via the script in `Consumption_Classification.m`) and used the corresponding Euclidean norm adapted to the number of dimensions

$$||\mathbf{v}|| = \sqrt{\sum_{i=1}^{24} v_i^2}.$$

In a nutshell the k-means algorithm groups data into k clusters by minimizing the distance between each point and the centre of its assigned cluster. It works by assigning each vector to the nearest centre, recalculating the centres as the mean of the assigned vectors, and repeating this process until the assignments no longer change or converge.

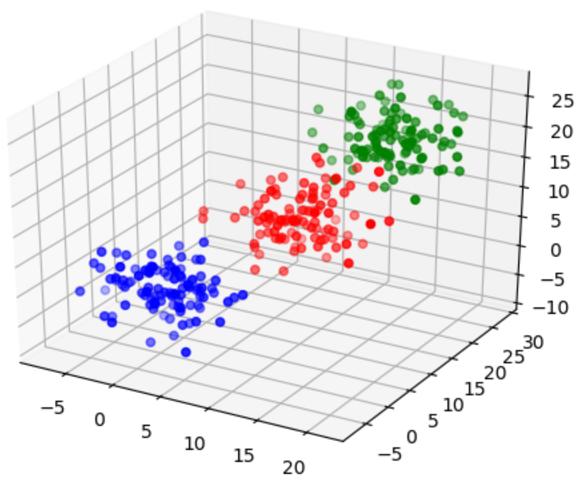
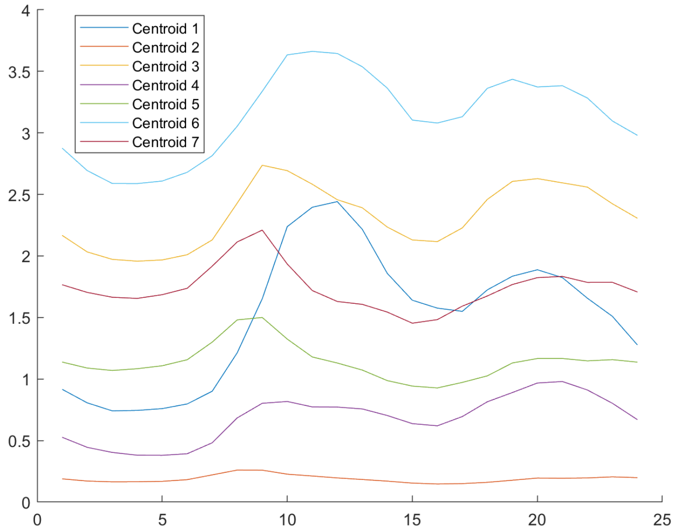
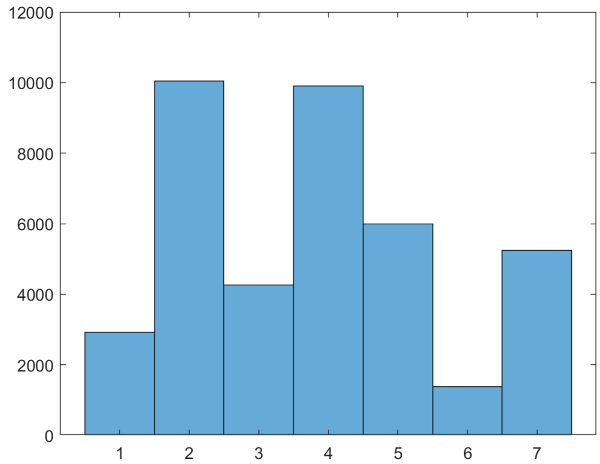


Figure 1: Example of data classified via $k-$means

For our dataset of daily consumptions we decided to create 7 load profiles, which can be seen in the following figure, along with their distributions:



(a) Consumption Profiles



(b) Distribution of consumption profiles

Figure 2: Load profiles and their distributions over the samples

For the used dataset we can observe that the load profiles present similar patterns, since the main difference between the centroids is the level (the amount) of power consumption along the day. This result can be mainly attributed to the fact that the used data was not normalized, leading to a distribution dependent on the level of consumption rather than distinguishing different 'shapes'. Nevertheless, this clustering is still useful to identify the profile (high consumer, low consumer, etc.) of a user.

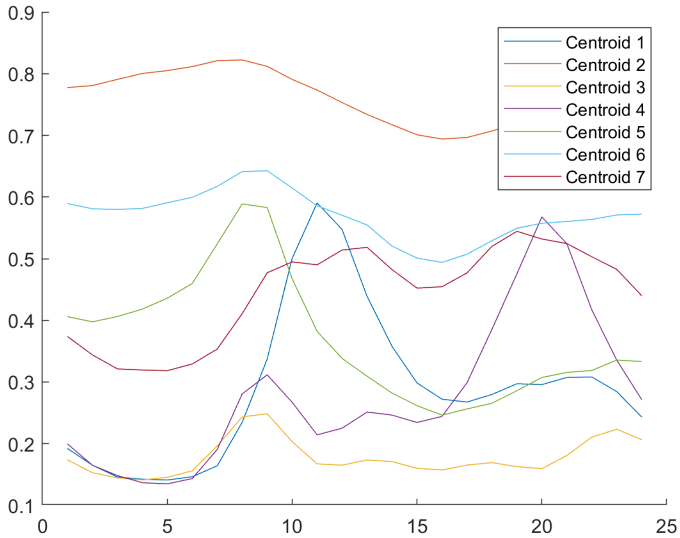
Having finished the classification, the results were saved in a matrix, whose first 24 columns are identical to those in `Daily_Consumptions.mat`, and a new column with the corresponding cluster of each pattern, in `Vector_and_consumption_type.mat`. Then, another two additional datasets were created, which will be used to trying to predict the future load profile: one dataset which contains the consumption of the previous day and the cluster of the following one, and another which contains the consumption of the two previous days and the cluster of the following one. The respective objects are `Dataset_Consumption_Prediction_24hours.mat` and `Dataset_Consumption_Prediction_48hours.mat`.

Additionally, since we are not only interested in the level of the consumer but in it the patter of consumption too, we performed the same clustering but having each daily consumption vector normalized in the following way:

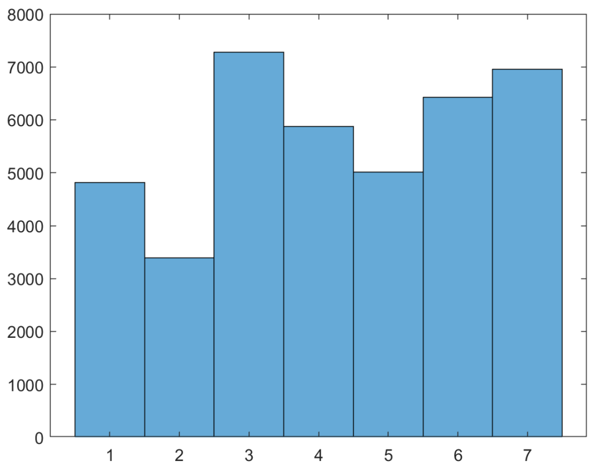
$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\max_i v_i} \tag{1}$$

Note that is not necessary to take the modulus in this case, since all the components are non-negative.

Having the same number of clusters as before, the obtained results are the following:



(a) Consumption Profiles Normalized



(b) Distribution of normalized consumption profiles

Figure 3: Normalized load profiles and their distributions over the samples

In this case we can appreciate that, as expected, we obtained a set of profiles corresponding to different distributions (patterns) of consumption during the day. We proceeded to create a similar new dataset with the patterns and their cluster, in object `Normalized_Vector_and_consumption_type.mat`.

Furthermore, three additional datasets were created and will be used to trying to predict the future load profile: one which contains the consumption of the previous day and the cluster of the following one, and another which contains the consumption of the two previous days and the cluster of the following one and one which contains the consumption of the previous day, the same of the previous week and the cluster of the following day. Their respective objects names are `Dataset_Consumption_Prediction_24hours_Normalized.mat`, `Dataset_Consumption_Prediction_48hours_Normalized.mat` and `Dataset_Consumption_Prediction_48hours_Week&Day_Before_Normalized.mat`. The corresponding script is `Consumption_Classification.m`. It is important to note that modifications, specially from the third dataset will be created in the following parts for further experimentation.

4.2 Tree Bag

For this project we implemented a personalized tree-based bagging model. As to provide context Tree bagging consists of training multiple decision trees on different subsets of the training data, each generated through random sampling with replacement. During prediction, the outputs of all trees are combined: for classification is generally done by majority vote. This ensemble method reduces variance and improves model stability. Two scripts were designed, one for fitting and one for predicting, named `FitTreeBagging.m` and `PredTreeBagging.m`. For the prediction function we used majority voting, since in the case of this particular function we are interested in providing only one output.

Manually comparing the accuracy of these scripts with MATLAB®'s own Tree Bagger, similar results were obtained.

Nevertheless, following 1, the fact of providing only one prediction can carry a high degree of uncertainty. To tackle this problem we implemented *conformal prediction*. As mentioned in [1], conformal prediction is a straightforward way to generate prediction sets for any model. It allows us to create statistically friendly prediction sets for our models. In other words, for a given pattern $\mathbf{X}_{\text{test}}$ we can provide a set of predictions $\mathcal{C}(\mathbf{X}_{\text{test}})$ (possible classes) with the guaranty that the correct output will be inside this set with a confidence $1-\alpha$ (being α the error we allow), $\mathbb{P}(Y_{\text{test}} \in \mathcal{C}(\mathbf{X}_{\text{test}})) \geq 1-\alpha$. For making use of the functions related to this procedure it is important to divide the dataset into training data for fitting the bagged trees, calibration data for computing an internal parameter for conformal prediction, and testing data.

The used scripts for this system are the fit function for the bagged trees, `Conformal_Prediction_Fit.m` and `Conformal_Prediction_Predict.m`. The output of the prediction function will be -for each introduced pattern- an array with all the possible classes, i.e. the prediction set. In the implementation of the prediction function we can chose (by commenting the corresponding part of the code) to use majority voting to compute the probabilities of each class or to average the probability sets of all the trees.

For assessing both the accuracy and the effectiveness of the tree bagging implementation and the conformal prediction, several datasets were used. Here is a breakdown of the tested datasets[2]:

1. `Dataset_Consumption_Prediction_24hours.mat`: Matrix whose 24 first columns correspond to daily, non-normalized, consumption vectors and the last column corresponds to the consumption class next day.

2. `Dataset_Consumption_Prediction_48hours.mat`: Matrix whose 48 first columns correspond to daily, non-normalized, consumption vectors for the two previous days we want to predict, and the consumption class next day.

3. `Dataset_Consumption_Prediction_48hours_Week&Day_Before.mat`: Matrix whose 48 first columns correspond to daily, non-normalized, consumption vectors for the previous day and the same day of the week we want to predict but the week before. The last column is the consumption class next day. We included this dataset because in cases such as household consumption it might be more meaningful to understand if it is a normal weekday (from Monday to Thursday), a Friday, or the weekend.

4. `Dataset_Consumption_Prediction_48hours_Week&Day_Before_classEncoded .mat`: Same as the previous matrix but we added the class type of the consumption vectors (the previous day and the same of the previous week) encoded. So, as features it has the 48 columns as consumptions, and $14 = 7 + 7$ because of the encoding of the class types of both days, plus the column of the class of the next day (what we want to predict). We included this dataset because we considered that knowing the consumption types would also be an important addition.

5. `Dataset_Consumption_Prediction_24hours_Normalized.mat`: Matrix whose 24 first columns correspond to daily, normalized, consumption vectors and the last column corresponds to the consumption class next day (the obtained centroids after the normalization).

6. `Dataset_Consumption_Prediction_48hours_Normalized.mat`: Matrix whose 48 first columns correspond to daily, normalized, consumption vectors for the two previous days we want to predict, and the consumption class next day.

7. `Dataset_Consumption_Prediction_48hours_Week&Day_Before_Normalized.mat`: Matrix whose 48 first columns correspond to daily, normalized, consumption vectors for the previous day and the same day of the week we want to predict but the week before. The last column is the consumption class next day.

8. `Dataset_Consumption_Prediction_48hours_Week&Day_Before_classEncoded_Normalized .mat` : Same as the previous matrix but we added the class type of the consumption vectors (the previous day and the same of the previous week) encoded. So, as features it has the 48 columns as consumptions, and $14 = 7 + 7$ because of the encoding of the class types of both days, plus the column of the class of the next day (what we want to predict). We included this dataset because we considered that knowing the consumption types would also be an important addition.

[2]One important remark is that, for the creation of each of these datasets, in case there was some data missing, we decided to discard all the pattern since the amount of other patterns was large enough.

It is important to note that higher accuracy values are expected for the non-normalized datasets, since the corresponding clusters are more differentiated between them (see 2a) according to the amount of power consumed (high consumption, medium consumption, etc.). Because of the shapes of the normalized centroids being more complicated we expect the accuracy to decrease for these cases, because we are also removing the amount of power consumed as something to take into account and focusing more in shape of the consumption itself.

As for the distribution of the data in each configuration, 60% was used for training, 20% for calibration for the conformal prediction, and 20% for testing. Note that the more samples are used for calibrations, the more exact idea of certainty we have about our predictions. If we used n samples for calibration, then for a test prediction we have

$$1 - \alpha \leq \mathbb{P}(Y_{\text{test}} \in \mathcal{C}(\mathbf{X}_{\text{test}})) \leq 1 - \alpha + \frac{1}{n+1}. \tag{2}$$

Furthermore, in order to have a wider set of possible configurations, two settings of feature selection are available on the tree bagging implementation: selecting all the features when providing patterns to train each tree or select a set of random features from the training set of patters to train the tree[3]. For the random feature selection, according to [6], the square root of the number of the features are randomly selected to train each tree.

For the conformal prediction, two settings are also possible when computing the probabilities of each class for a given pattern: on the one hand, the probability of a class is the proportion of trees that output it or, on the other hand, it is the average probability that each tree assigns to it.

5 Results

Applying different configurations of the algorithms to our datasets, the obtained results are the following:

[3]The fact of applying bootstrapping and this random feature selection is what is normally considered as the technique *random forest*.

Table 1: Test results on different datasets and configurations

Dataset	Normal Prediction			Conformal Prediction			
	Num. Trees	**Feature Selection**	**Acc.**	**Probability Method**	α	**Avg. Set Size**	**Acc.**
1	100	Random	0.673	Average	0.1	1.917	0.894
1	100	All features	0.670	Average	0.1	1.974	0.899
1	100	Random	0.673	Majority Voting	0.1	1.889	0.890
1	100	All features	0.670	Majority Voting	0.1	1.953	0.895
2	100	Random	0.679	Average	0.1	1.878	0.891
2	100	All features	0.674	Average	0.1	1.899	0.893
2	100	Random	0.679	Majority Voting	0.1	1.849	0.886
2	100	All features	0.674	Majority Voting	0.1	1.894	0.892
3	100	Random	0.685	Average	0.1	1,892	0,905
3	100	All features	0.683	Average	0.1	1,892	0,904
3	100	Random	0.685	Majority Voting	0.1	1,844	0,896
3	100	All features	0.683	Majority Voting	0.1	1.896	0,903
4	100	Random	0.694	Average	0.1	1.845	0.903
4	100	All features	0.686	Average	0.1	1.906	0.905
4	100	Random	0.694	Majority Voting	0.1	1.812	0.897
4	100	All features	0.686	Majority Voting	0.1	1.91	0.904
5	100	Random	0.447	Average	0.1	3.559	0.896
5	100	All features	0.444	Average	0.1	3.585	0.898
5	100	Random	0.447	Majority Voting	0.1	3.560	0.893
5	100	All features	0.444	Majority Voting	0.1	3.541	0.893
6	100	Random	0.466	Average	0.1	3.375	0.904
6	100	All features	0.465	Average	0.1	3.387	0.900
6	100	Random	0.466	Majority Voting	0.1	3.402	0.905
6	100	All features	0.465	Majority Voting	0.1	3.375	0.896
7	100	Random	0.463	Average	0.1	3.502	0.904
7	100	All features	0.463	Average	0.1	3.511	0.906
7	100	Random	0.463	Majority Voting	0.1	3.451	0.897
7	100	All features	0.463	Majority Voting	0.1	3.418	0.896
8	100	Random	0.461	Average	0.1	3.459	0.904
8	100	All features	0.461	Average	0.1	3.535	0.906
8	100	Random	0.461	Majority Voting	0.1	3.446	0.901
8	100	All features	0.461	Majority Voting	0.1	3.600	0.909

First of all, some remarks about the table. The column 'Avg. Set Size' stands for the averaging the the number of elements that the prediction vector had for the conformal prediction. The second column 'Acc.' that appears in the the Conformal Prediction section stands for the number of cases in which the correct class was found in the prediction dataset. Even though it is not exact, a safety check that the algorithm runs smoothly is that accuracy values close to $1 - \alpha$ are obtained, which is the case.

We can observe that high accuracy values have been obtained for the non-normalized datasets. That is logical taking into account how differentiated the centroids are in 2a, because as mentioned before, these profiles correspond to the gross consumption of the users (high demand, medium demand, etc.) through the day. On the other hand, considerably lower accuracy values were retrieved when using the normalized datasets with the new centroids. This results was also expected since now the profiles have more complex patterns and are not as differentiated as in the previous case, making the classification task more difficult.

Nevertheless, an unexpected result was the decrease in accuracy -despite being close- for dataset 8 with respect to dataset 6 and 7. The highest accuracy for normalized datasets has been obtained when using data from the two previous days. We expected to achieve higher accuracy values providing information of the previous day and the previous week, mainly because of the periodicity in a users consumption through the week (e.g., a user might consume electricity differently on Saturdays than in Thursdays or Fridays and more similarly to the previous Saturday). In addition, the fact of having added the class types for the day before and the previous week also did not improve the results, contrary to what was expected. This result also contrasts with the obtained for the non-normalized datasets, where adding information from the previous week and the encoded classes led to an increase in the accuracy.

As for the conformal prediction concerns, we can observe that set sizes for the non-normalized datasets are much more smaller than for the normalized ones. In the first case, set sizes close to two are obtained, meaning the the model is considerably confident that the correct class will be contained in the set of size 2 with a 90% confidence at least. In the second case, similarly as it happened with the accuracy, the model loses confidence and in order to compensate for this, larger sets have to be provided, averaging from 3 to 4 elements. The smallest average set sizes have been obtained (for the normalized data) when providing information about the consumption during the two previous days, following a similar behaviour as when we analyzed the accuracy.

6 Literature review

Another part of the project has been to analyse the most relevant literature on the applications of load forecasting, with special attention to those novel applications proposed in recent years.

The role of load forecasting in modern power systems extends well beyond conventional scheduling. Recent studies demonstrate that predictions are not only evaluated for accuracy, but are also directly integrated into downstream decision-making procedures. Importantly, these works highlight both regression-based approaches—where continuous demand values are predicted and subsequently fed into optimization routines—and classification-based approaches—where discrete demand states or profiles are forecasted to simplify or enable specific operational strategies.

The work of [14] focuses on regression forecasting and its integration into smart grid operation. In this context, predicted hourly load curves are explicitly embedded into optimization models. For example, demand response programs use these forecasts to determine when and how to schedule flexible loads such as HVAC systems or industrial processes, while tariff design leverages predicted peak hours to generate time-varying electricity prices. Thus, the regression outputs are transformed into actionable control signals for both utilities and end-users, enabling more cost-efficient and adaptive grid management.

In [19], regression methods are applied to the growing challenge of predicting charging demand at electric vehicle (EV) stations. A key novelty lies in the robustness to low-quality data: the model is designed to operate effectively even with missing or noisy observations. The resulting continuous forecasts are then used to allocate charging slots dynamically, anticipate potential transformer overloads, and plan the activation of local battery storage systems. Forecasting, in this case, becomes a central tool for congestion prevention and for the strategic deployment of EV infrastructure in urban networks.

The study by [4] demonstrates the combined use of regression and classification techniques within microgrid energy management systems (EMS). On one hand, regression forecasts provide detailed hourly demand curves that are used to schedule battery charging and discharging, optimize peak shaving, and balance renewable generation with local consumption. On the other hand, classification methods are used to identify operating states—such as high-load vs. low-load days—or to assign daily consumption patterns to pre-defined clusters. These categorical forecasts allow EMS controllers to apply different rule-based strategies depending on the predicted class, reducing the complexity of real-time optimization.

[20] presents an application where classification plays a central role. Instead of predicting exact consumption values, the model outputs discrete demand classes (e.g., high, medium, low). These are fed into a reinforcement learning agent that adjusts electricity prices in real time according to the predicted class. This classification-based approach reduces the dimensionality of the input space, allowing the agent to learn efficient pricing policies that shape user behavior and reduce peak demand. In this setting, the forecast itself is not a numeric curve but a symbolic signal that conditions a learning-based pricing mechanism.

The recent literature in IEEE Transactions on Power Systems and IEEE Transactions on Smart Grid offers concrete pathways by which forecasts are operationalized. For instance, during regime shifts such as the COVID-19 lockdowns, [13] propose adaptive generalized additive models updated through Kalman filtering and lightweight fine-tuning. These regression forecasts are not static outputs: they are continually re-calibrated and injected into national-scale scheduling and reserve sizing routines. Moreover, the authors explore cross-country transfer (e.g., from Italy to France) to anticipate structural breaks in consumption. This illustrates a real-world "forecast $\rightarrow$ dispatch/settlement" loop that remains stable under non-stationarity.

Security and reliability concerns are addressed by [21], who design a Bayesian, attack-resilient load forecasting framework. Here, regression forecasts are produced together with uncertainty intervals inflated in the presence of adversarial perturbations. The EMS does not use the point forecast alone: instead, control actions (such as triggering demand response or updating prices) are gated by the Bayesian credibility of the forecast. When uncertainty is high or suspicious patterns are detected, the EMS can choose to hedge, delay, or reject the action. This creates a security-aware control loop where "forecast + uncertainty" determines whether and how to act.

On the deployment side, [11] introduce an online–offline deep kernel learning pipeline for residential load forecasting. The offline component captures global structure from historical data, while the online stage performs lightweight Bayesian updates to adapt to customer-level drift. These probabilistic regression forecasts are consumed in multiple ways: to settle tariffs, to schedule feeder-level battery systems, and to screen for anomalies at the household level. The architecture is designed for real-time deployment by DSOs or retailers, ensuring production-grade performance even in the presence of volatile behavior and missing data.

Feature engineering at scale is the focus of [8], who propose a multidimensional feature extraction pipeline that combines calendar features, weather variables, temporal locality, and seasonal interactions. Importantly, they include an explicit de-redundancy and selection stage prior to model training. This approach improves interpretability and allows operators to identify which feature families dominate at different horizons or regions. These insights are operationally relevant: they guide sensor placement (e.g., where to install weather stations), inform the design of tariff time windows, and help choose which exogenous variables should feed into EMS pipelines. The regression forecasts are thus not only accurate but also interpretable and actionable.

To ensure adaptivity in streaming operation, [9] propose an online ensemble built on top of batch models using a Passive-Aggressive re-weighting scheme. This layer continuously reallocates weights among pre-trained models as external conditions change (e.g., holidays, events). The forecasts produced are robust for both day-ahead and intraday horizons, without requiring costly full retraining. In practical deployment, this ensemble layer can be embedded immediately before unit commitment or demand response selection, ensuring that forecast accuracy and calibration remain high between major retraining cycles.

Finally, [7] address the spatio-temporal nature of EV charging demand using a graph-convolutional recurrent network. The regression forecasts generated at station level are used to allocate charging slots, prevent transformer overloading, and participate in flexibility markets through bidding strategies. The spatial coupling between stations—based on the road network or geographic proximity—is explicitly modeled, which allows operators to pre-position storage resources or issue price signals in anticipation of local congestion.

Overall, most of these recent works are grounded in regression forecasting. The operational layer is built in different ways: (i) by plugging forecasted curves into optimization problems (generation scheduling, tariff design, battery operation), (ii) by applying policy logic or gating based on uncertainty or feature saliency (security-aware control, sensor and tariff planning), and (iii) by introducing adaptive layers (e.g., online re-weighting) to keep models calibrated under non-stationarity. Classification, when present, is typically used as a derived layer—mapping continuous forecasts into discrete states for use in rule-based or learning-based controllers—rather than as the primary prediction task.

In summary, regression forecasts are primarily used as quantitative inputs to optimization and decision-making frameworks, whereas classification forecasts provide categorical abstractions that simplify controller design and learning. Understanding this distinction clarifies not only the nature of the forecasting task but also the pathway through which forecast outputs influence real-world energy management actions.

Table 2: Summary

Article	**Application Context**	**Forecasting Type**	**How Forecast is Used**
[14]	Smart grid operation and tariff optimization	Regression	Forecasted hourly load curves are input to optimization models for demand response scheduling and to calculate dynamic tariffs reflecting peak hours.
[19]	EV charging stations and local storage	Regression	Predicted charging demand curves guide scheduling of charging slots, transformer capacity planning, and storage usage for peak smoothing.
[4]	Microgrid EMS and peak shaving	Regression + Classification	Regression forecasts control battery scheduling and load balancing; classification of daily load profiles (clustered states) supports rule-based EMS strategies.

Article	**Application Context**	**Forecasting Type**	**How Forecast is Used**
[20]	Smart grid demand response with dynamic pricing	Classification	Demand classes (e.g., high/medium/low) are predicted and fed into a reinforcement learning agent that adjusts electricity prices to influence consumption.
[13]	National STLF under regime shift	Regression	Adaptive GAM + Kalman updates produce corrected curves used directly for generation scheduling, reserve sizing, and settlement during COVID-19; transfer from Italy anticipates breaks.
[21]	Security/ robustness of EMS	Regression (Bayesian)	Attack-resilient forecasts with calibrated uncertainty; EMS gates DR/price updates on posterior credibility and flags adversarial inputs (security-aware control).
[11]	Residential/ retail deployment	Probabilistic regression	Online–offline DKL pipeline; probabilistic outputs drive tariff settlement, feeder-level battery scheduling, and household anomaly screening with low-latency updates.
[8]	Utility-scale STLF	Regression	Multidimensional feature extraction informs which exogenous feeds matter; forecasts feed peak-aware scheduling and guide sensor placement and tariff window selection.
[9]	Streaming operations	Regression ensemble	Online Passive-Aggressive re-weighting over batch models; drop-in layer before unit commitment/DR selection to stay calibrated between retraining cycles.
[7]	EV charging networks	Regression (spatio-temporal)	Station-level demand curves for slot allocation, transformer overload prevention, and flexibility market bids; spatial coupling enables pre-positioning of storage/pricing.
[15]	Residential demand response	Probabilistic regression	Prediction intervals are used to (i) quantify the uncertainty DR can cover, (ii) compute number of consumers needed for target coverage, and (iii) define hourly DR strategies minimizing deviation from purchased energy ($\geq$ 33% reduction in error).

7 Conclusions

The internship at the Basque Center for Applied Mathematics (BCAM) has provided a valuable opportunity to explore both the theoretical and practical aspects of load forecasting -specifically as a classification problem- an area of increasing importance in the context of smart grids and sustainable energy management.

The main achievement of this work was the implementation[4] of a manual implementation of the bagged tree algorithm to predict residential consumption profiles in MATLAB®, plus the option of adjusting the feature selection, which enables the possibility of the algorithm being a random forest too. Furthermore, experimentation with different datasets of consumption data were used, obtaining interesting results as to which kind of features are more important for properly predicting the consumption pattern that there will be in the near feature.

A significant contribution of the project was the incorporation of conformal prediction methods, which enabled not only accurate but also reliable and interpretable forecasts. By providing statistically valid confidence sets, the models went beyond point predictions, offering uncertainty quantification that is essential for real-world decision-making in the energy sector.

In parallel, the literature review shed light on the wide range of applications of load forecasting, from regression-based methods used in tariff design and demand response programs to classification-based approaches applied in consumer segmentation and anomaly detection. This broader perspective helped situate the developed work within the current research landscape, highlighting both the methodological advances and the open challenges in the field.

Overall, this internship has been a highly enriching experience. It strengthened my knowledge in machine learning applied to energy data, particularly in ensemble learning and uncertainty quantification, while also deepening my understanding of the role of forecasting in energy systems. Working within the collaborative and research-driven environment of the BCAM and under the direction of Dr. Santiago Mazuelas has further motivated me to continue pursuing research in this field.

[4] All the code for the project can be found at `https://github.com/LeonelFNR/Electricity_Consumption_Pattern_Prediction.git`.

References

[1] Anastasios N. Angelopoulos and Stephen Bates. *A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification*. 2022. arXiv: 2107.07511 [cs.LG]. URL: https://arxiv.org/abs/2107.07511.

[2] Sean Barker et al. *UMass Smart* Apartment Dataset*. Electrical and weather data for 114 single-family apartments (2014-2016). University of Massachusetts Amherst. Accessed August 2025. 2017. URL: %5Curl%7Bhttps://traces.cs.umass.edu/docs/traces/smartstar/#apartment-dataset%7D.

[3] Cruz E. Borges, Yoseba K. Penya, and Iván Fernández. "Evaluating Combined Load Forecasting in Large Power Systems and Smart Grids". In: *IEEE Transactions on Industrial Informatics* 9.3 (2013), pp. 1570–1577. DOI: 10.1109/TII.2012.2219063.

[4] Mahmoud Fathi, Emiliano Aliprantis, et al. "State-of-the-art review on energy and load forecasting in microgrids using artificial neural networks, machine learning, and deep learning techniques". In: *Energies* 16.12 (2023), p. 4691.

[5] Zhifeng Guo, Jesse R. O'Hanley, and Stuart Gibson. "Predicting residential electricity consumption patterns based on smart meter and household data: A case study from the Republic of Ireland". In: *Utilities Policy* 79 (2022), p. 101446. ISSN: 0957-1787. DOI: https://doi.org/10.1016/j.jup.2022.101446. URL: https://www.sciencedirect.com/science/article/pii/S0957178722001102.

[6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer Series in Statistics. New York, NY: Springer, 2009. ISBN: 978-0-387-84857-0. URL: https://doi.org/10.1007/978-0-387-84858-7.

[7] H. J. Kim and M. K. Kim. "Spatial-Temporal Graph Convolutional-Based Recurrent Network for Electric Vehicle Charging Stations Demand Forecasting in Energy Markets". In: *IEEE Transactions on Smart Grid* (2024). DOI: 10.1109/TSG.2024.0000000.

[8] Nakyoung Kim et al. "Short-Term Electrical Load Forecasting With Multidimensional Feature Extraction". In: *IEEE Transactions on Smart Grid* 13.4 (2022), pp. 2999–3013. DOI: 10.1109/TSG.2022.3158387.

[9] Leandro von Krannichfeldt, Yi Wang, and Gabriela Hug. "Online Ensemble Learning for Load Forecasting". In: *IEEE Transactions on Power Systems* 36.1 (2020), pp. 545–548. DOI: 10.1109/TPWRS.2020.3030130.

[10] Paolo Lazzeroni, Gianmarco Lorenti, and Maurizio Repetto. "A Data-Driven Approach to Predict Hourly Load Profiles From Time-of-Use Electricity Bills". In: *IEEE Access* 11 (2023), pp. 60501–60515. DOI: 10.1109/ACCESS.2023.3286020.

[11] Yuanzheng Li et al. "Residential Load Forecasting: An Online-Offline Deep Kernel Learning Method". In: *IEEE Transactions on Power Systems* (2023). DOI: 10.1109/TPWRS.2023.3299999.

[12] Fintan McLoughlin, Aidan Duffy, and Michael Conlon. "A clustering approach to domestic electricity load profile characterisation using smart metering data". In: *Applied Energy* 141 (2015), pp. 190–199. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2014.12.039. URL: https://www.sciencedirect.com/science/article/pii/S0306261914012963.

[13] David Obst, Joseph de Vilmarest, and Yannig Goude. "Adaptive Methods for Short-Term Electricity Load Forecasting During COVID-19 Lockdown in France". In: *IEEE Transactions on Power Systems* 36.5 (2021), pp. 4754–4763. DOI: 10.1109/TPWRS.2021.3086105.

[14] Mohsin Raza, Mithulan Nadarajah, and Chinthaka Ekanayake. "Review on smart grid load forecasting for smart energy management using machine learning and deep learning techniques". In: *IEEE Transactions on Smart Grid* 12.4 (2021), pp. 3161–3179.

[15] María Carmen Ruiz-Abellón et al. "Applications of Probabilistic Forecasting in Demand Response". In: *Applied Sciences* 14.21 (2024), p. 9716. DOI: 10.3390/app14219716.

[16] Aven Satre-Meloy, Marina Diakonova, and Philipp Grünewald. "Cluster analysis and prediction of residential peak demand profiles using occupant activity data". In: *Applied Energy* 260 (2020), p. 114246. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.114246. URL: https://www.sciencedirect.com/science/article/pii/S0306261919319336.

[17] Dauwe Vercamer et al. "Predicting consumer load profiles using commercial and open data". In: *IEEE Transactions on Power Systems* 31.5 (2015), pp. 3693–3701.

[18] Yi Wang et al. "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges". In: *IEEE Transactions on Smart Grid* 10.3 (2019), pp. 3125–3148. DOI: 10.1109/TSG.2018.2818167.

[19] Yuxuan Wang, Lei Zhang, Yuxuan Zhou, et al. "Short-term electric vehicles charging load forecasting based on deep learning in low-quality data environments". In: *Sustainable Cities and Society* 91 (2024), p. 104011.

[20] Xiaodong Zhou, Hao Wang, and Jian Li. "A dynamic pricing demand response algorithm for smart grid: Reinforcement learning approach". In: *Applied Energy* 306 (2022), p. 117994.

[21] Yihong Zhou et al. "Robust Load Forecasting Towards Adversarial Attacks via Bayesian Learning". In: *IEEE Transactions on Power Systems* 38.2 (2023), pp. 1445–1459. DOI: 10.1109/TPWRS.2022.3212345.