DEGREE PROJECT IN TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2023

Load profiling and customer segmentation for demand-side management

Anne Baril

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Author

Anne Baril, abaril@kth.se
Electrical engineering and computer science
KTH Royal Institute of Technology

Host company

Vattenfall
Supervisor: Vincent Gliniewicz, vincent.gliniewicz@vattenfall.com

Examiner

Carlo Fischione, KTH Royal Institute of Technology

Supervisor

Qianwen Xu, KTH Royal Institute of Technology

Abstract

The energy transition is accompanied by massive electrification of uses and sectors such as transport. As a result, the pressure on the electricity grid is increasing, and the time to connect to the power system is lengthening. Deploying new infrastructure is a laborious and expensive process but there are alternatives to exploit the flexibility of the power grid. The deployment of smart meters opens the door to many applications related to flexibility on the consumer side, to reduce peak loads that threaten grid capacity. Targeting the right consumers for Demand-Side Management (DSM) is a prerequisite to maximizing the chances of success of such programs.

This degree project replicates and adapts the method developed in [14] to segment residential customers. It consists of encoding Daily Load Curves (DLC) using a dictionary of Typical Load Profiles (TLP) and grouping consumers according to the distribution of their TLP. A temporal analysis of the main TLP reveals different consumption behaviors. Customers are segmented into groups that reflect the degree of volatility of their consumption. This enables a classification based on the potential for Energy Efficiency (EE) or Demand Response (D/R) programs. We address the issue of attribute detection using the distribution of TLP of customers. In particular, several classification algorithms are compared to detect TLP characteristic of Electric Vehicle (EV). The obtained load shapes show consumption peaks at night, which may correspond to the charging time of EV.

The method is discussed, especially the choice of the number of load profiles to be included in the dictionary of TLP. It proves to be useful to group consumers with similar consumption profiles and opens the door to applications such as individual household consumption forecasting.

Keywords

Smart meter data, TLP, customer segmentation, DSM, EE, D/R, clustering

Abstract

Energiomställningen kräver en massiv elektrifiering av användningsområden och sektorer som t.ex. transportsektorn. Detta leder till att trycket på elnätet ökar och att tiden för att ansluta sig till elnätet blir allt längre. Att bygga ut ny infrastruktur är en mödosam och dyr process, men det finns alternativ för att utnyttja elnätets flexibilitet. Utplaceringen av smarta mätare öppnar dörren för många tillämpningar som rör flexibilitet på konsumentsidan, för att minska toppbelastningar som hotar nätkapaciteten. Att rikta in sig på rätt konsumenter för DSM är en förutsättning för att maximera chanserna att lyckas med sådana program.

I detta examensarbete replikeras och anpassas den metod som utvecklats i [14] för att segmentera hushållskunder. Den består av att koda DLC med hjälp av ett lexikon av TLP och gruppera konsumenter enligt fördelningen av deras TLP. En tidsmässig analys av de viktigaste TLP avslöjar olika konsumtionsbeteenden. Kunderna delas in i grupper som återspeglar graden av volatilitet i deras konsumtion. Detta möjliggör en klassificering baserad på potentialen för EE eller D/R-program. Vi tar upp frågan om attributdetektering med hjälp av fördelningen av TLP hos kunderna. I synnerhet jämförs flera klassificeringsalgoritmer för att upptäcka TLP som är karakteristiska för EV. De erhållna belastningsformerna visar konsumtionstoppar på natten, vilket kan motsvara laddningstiden för EV.

Metoden diskuteras, särskilt valet av antalet belastningsprofiler som ska ingå i ordlistan för TLP. Metoden visar sig vara användbar för att gruppera konsumenter med liknande förbrukningsprofiler och öppnar dörren för tillämpningar som prognostisering av enskilda hushålls förbrukning.

Nyckelord

Data från smarta mätare, TLP, kundsegmentering, DSM, EE, D/R, klustring

Acknowledgements

I would like to thank my supervisor at Vattenfall, Vincent Gliniewicz, for his support during this degree project. I appreciated your general guidance and advice, as well as your confidence in what I did and your willingness to make this project relevant to Vattenfall. Your feedback helped me see my thesis from the company's perspective.

I would also like to thank my supervisor at KTH, Qianwen Xu. Thank you for your confidence in the master thesis and all the literature you shared, as well as for your very quick response to my inquiries, which helped me move forward in my research. I also appreciated the help of my examiner, Carlo Fischione, who put me in touch with Qianwen Xu.

I would like to express my gratitude to all my colleagues at Vattenfall R&D for their openness, and thanks to whom I came to the office even in the snowy days of the Swedish winter.

Finally, I would like to thank my friends in Sweden, who made this master's degree an unforgettable experience. In particular, I wish Pauline the best of luck in getting the train on time, and I look forward to more adventures with Lucas.

Acronyms

DSO Distribution System Operators

TSO Transmission System Operators

GDPR General Data Protection Regulation

TLP Typical Load Profiles

D/R Demand Response

DSM Demand-Side Management

EE Energy Efficiency

ToU Time-of-use

EV Electric Vehicle

V2G Vehicle-to-grid

DLR Dynamic Line Rating

NTO Network Topology Optimization

Ei Energimarknadsinspektionen - National Regulatory Authority for Energy

NILM Non-Intrusive Load Monitoring

LSTM Long Short-Term Memory

GRU Gated Recurrent Unit

STLF Short-term Load Forecasting

SOM Self-Organizing Map

CFSFDP Fast Search and Find of Density Peaks

RSD Residual Standard Deviation

CDF Cumulative Distribution Function

HAC Hierarchical Agglomerative Clustering

RF Random Forest

BRF Balanced Random Forest

WRF Weighted Random Forest

DLC Daily Load Curves

Contents

Chapter 1

Introduction

The European climate law sets the target for Europe's economy and society to become climate-neutral by 2050. The production and use of energy account for more than 75% of the EU's greenhouse gas emissions. Decarbonizing the EU's energy system while ensuring a secure and affordable energy supply to all is a priority of the European Green Deal. To shift away from fossil fuels and develop renewable energies, several sectors such as transport, construction and industry are increasingly powered by electricity. As a result, providing grid access for new users is no longer straightforward yet even more critical. Transmission and distribution lines cannot be extended quickly enough to meet new peaks in demand and, in any case, it makes little economic sense to design infrastructure specifically for peaks if consumption can be flattened.

At the same time, the widespread deployment of smart meters provides a significant amount of data, not yet fully utilized, to understand consumer behavior and enhance the efficiency of the power system. Fine-grained measurements of consumption provide extensive information on how and when households, businesses, or industries use electricity, which is particularly valuable for Distribution System Operators (DSO) and aggregators when planning grid infrastructures and forecasting consumption. Tailored D/R and EE programs can be designed for specific groups of consumers to leverage user flexibility or willingness to adapt consumption in response to fluctuation in the price of electricity or to incentives.

1.1 Demand-side management and flexibility

In the future, consumers are expected to play a more active role in the power market, investing in distributed energy resources and participating in D/R programs to consume in a sober and economical way. D/R refers to changes in the consumption of a customer in response to the evolution of the price of electricity or incentives to reduce electricity use in times of high demand. The purpose is to adapt the user demand to the grid capacity by moving or changing the energy consumption. EE means using less energy to perform a task [14]. Consumers are also expected to partially participate in wholesale markets through aggregators whose function is to pool electricity supply and sell capacity in the electricity markets.

In general, the consumption behavior of a user is defined with criteria such as the correlation between their consumption and the aggregated load over all users. Customers with load peaks happening at the same time as global peaks are good candidates for incentives to shift consumption from peak hours to off-peak hours. The problem with such a method is that an individual's consumption is highly variable and in practice, it is difficult to reliably estimate such indicators. To analyze the consumption of a residential customer, establish D/R programs and evaluate the potential for flexibility, reduction techniques such as clustering of load curves are necessary.

The potential contribution of the residential sector for load flexibility is the subject of several recent studies. Smart appliances and EV can be piloted to reduce peak loads and relieve pressure on the system. In [19], a method to estimate the load profiles of a group of residential buildings in Italy is developed. A consumption profile is created for each of the most common building typologies. Load time shifting is modelled, taking into account power demand and electricity price and assuming that customers with hourly contracts are sensitive to price. The study shows that the global load profile for all the buildings can be flattened, especially during the cold season. One takeaway of the paper is that the potential for flexibility depends closely on the intrinsic characteristics of the power system, the building types, and the consumers in the region of interest. However, this method would benefit from a more detailed knowledge of the type of appliances used in the buildings and insight into the actual responsiveness of customers to price changes.

Typical Load Profiles (TLP)

A TLP is a representation of the energy consumption patterns of a group of loads, either representing user or utility behaviors over a given time period. It summarizes the temporal variations in energy consumption into a single curve or set of curves, providing a simplified and standardized view of the energy consumption.

1.2 Load clustering and residential customer segmentation

Smart meters record electrical consumption at different time intervals, with frequencies varying from every minute to several hours. This huge amount of data offers new opportunities to study consumption behaviors. Load profiling refers to the classification of load curves or consumers according to their electricity consumption features. Existing studies on load profiling mainly focus on industries and businesses with relatively regular consumption patterns. Residential customers bring new challenges, as load shapes and amplitude may vary greatly between two households and evolve in time. One typical profile is not enough to describe the consumption of a household. Compared with patterns at aggregated levels, smart meter data show high volatility: the orange curve of figure 1.2.1 represents the consumption of a single household, whereas the blue curve is the aggregated behavior at postal area level. Figure 1.2.2 illustrates the variability of the consumption of a randomly chosen residential customer, on the time scale of a week.

Data mining techniques are used to deal with the dimensionality of temporal data such as electrical load curves, which are made of a large number of data points. Methods to explore and reduce the dimension of time series without overwriting their temporal aspect have emerged. Research has focused on discovering patterns in time series, and algorithms for clustering dynamic data have gained momentum. Clustering is an unsupervised technique for classifying data without a priori knowledge, where similar data points are placed into homogeneous groups [3].

Clustering has been used to reduce the dimension of electrical load curves and group customers with similar consumption behaviors. Different tasks have been

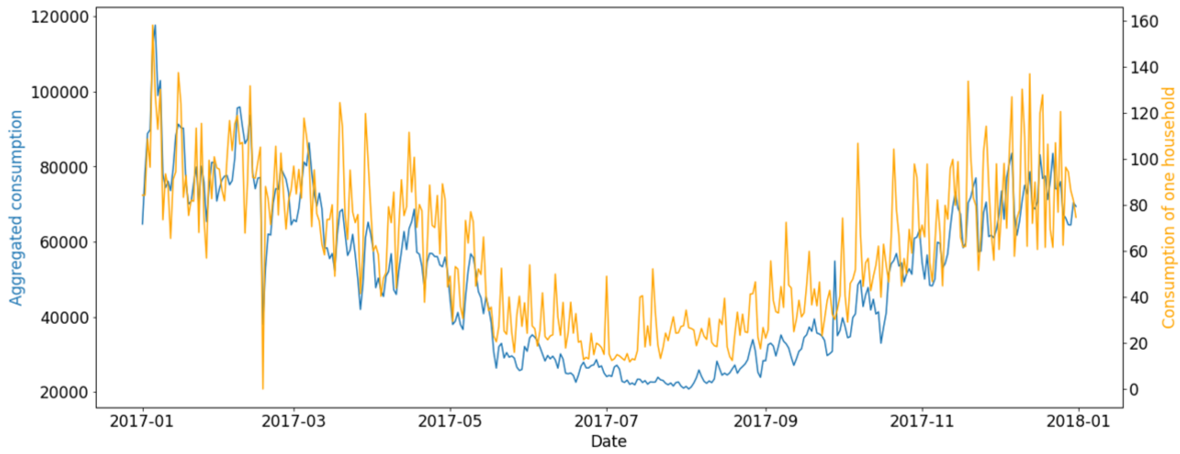


Figure 1.2.1: Electricity consumption (kWh) of a randomly selected household (in yellow) and aggregated consumption at postal level (in blue)

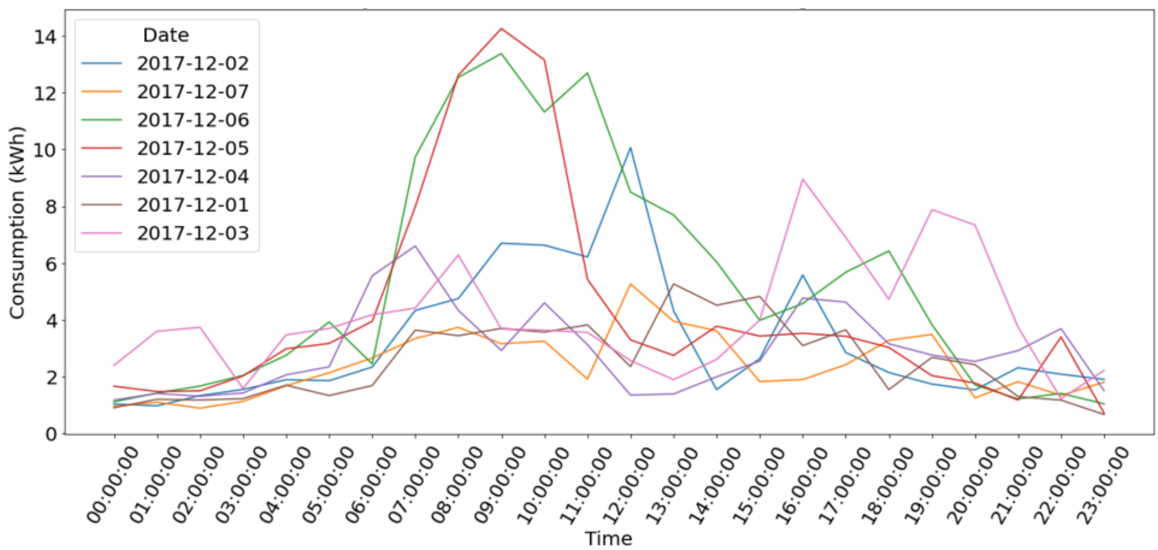


Figure 1.2.2: Electricity consumption patterns (kWh) of a randomly chosen household during one week of December 2017

addressed, among them the creation of TLP, or the extraction of customer features. Historically, customers have mainly been segmented based on socioeconomic factors known through in-home surveys for instance.

1.3 Aim and purpose

The goal of the thesis is to reproduce and adapt a data-driven methodology to segment customers based on their electricity consumption and evaluate its benefits. The method relies on the creation of a large set of TLP and it must be scalable to large data sets.

TLP are one of the most common approaches to study electricity consumption data, as they provide a simplified representation of energy consumption patterns. The applications are numerous, from the anonymization of sensitive data to research institutes, compliant with the requirements of the GDPR [20], to trend analysis and modeling.

Creating groups of consumers with similar consumption lifestyle is a prerequisite to the development of programs that enhance the willingness of targeted customers to change their consumption patterns in response to price-based and incentive-based D/R schemes.

1.4 Research questions

This thesis attempts to answer the following questions:

- How can customers in a large dataset be segmented into groups with similar energy behaviors, using smart meter data?
- Is it possible to create a set of TLP that accurately describe most consumption profiles in a large dataset?

Both magnitude and timings of consumption are relevant to group customers. Thus, multidimensional segmentation is probably a good approach to the problem of guided clustering of consumers. TLP provide a condensed representation of energy consumption data, which allows for spatial and temporal comparisons of common patterns, helps highlight trends in energy consumption and could be used as a preprocessing step for consumption forecasting and load disaggregation models.

By successfully answering these questions, this thesis contributes to the research on load profiling for the evaluation of residential consumers potential in DSM programs and has the potential to help improve consumption models.

1.5 Delimitations

The thesis is written in collaboration with Vattenfall. In the first instance, this research seeks to replicate and adapt the method developed in [14] to create a dictionary of TLP and encode consumers' behavior. Then, the benefits of this method for DSM are discussed.

It should be noted that the data is limited to the Stockholm and Uppsala areas and cover the years 2016 to 2021, an effect of the COVID-19 pandemic on consumption levels can thus not be excluded. Only a subset of all the data available is used, to reduce the running time of the algorithms and to test the method on a spare set. The clustering methodology to create TLP is only tested on residential data and more specifically one- or two-dwelling houses, but should be adaptable for industries and services. The main focus of the work is on the adaptation of a method that is scalable to large datasets, not on the results obtained, even though the clusters will be analyzed and put into perspective in the context of the Swedish power system.

1.6 Outline

The structure of this thesis is as follows. Chapter 2 discusses in further detail the context of the thesis and provides a background on theoretical aspects used in the report. Chapter 3 addresses the data collection, preprocessing and analysis steps. It presents a detailed description of the method used in the thesis and how it can be evaluated. Results are discussed in chapter 4. The conclusions of this work are drawn in Chapter 5, and avenues for further reflection are also presented.

Chapter 2

Background

In this chapter, the background of the degree project is presented together with related work.

2.1 Flexibility in the Swedish grid

The electrical grid in Sweden is divided into three categories: the national grid (transmission grid), regional grids, and local grids (distribution grids). The national grid is managed by the government agency Svenska kraftnät while most of the regional grids are owned by the grid companies E.ON Elnät Sverige, Vattenfall Eldistribution, and Ellevio.

As mentioned earlier, the growing integration of intermittent renewable energies (c.f. Fig. 2.1.1) and the electrification of uses in society with more devices and equipment running on electricity are increasing the pressure on the power system. To ensure stability and delay the addition of new facilities, leveraging system flexibility may become necessary. Flexibility is defined as the possibility of adapting generation or consumption patterns as a consequence of a price signal or incentives. It concerns different actors of the power grid (c.f. Fig. 2.1.2) and can be of different types:

- Technical flexibility, i.e. optimization of the existing infrastructure.
- Market-based flexibility with initiatives such as coordiNet [18], whose aim is to reduce the grid congestion between Transmission System Operators (TSO) and DSO while facilitating the participation of customers in electricity markets.

- Demand-side flexibility, which refers to the portion of demand that can be decreased, increased or shifted over a given period in response to price signals or incentives.

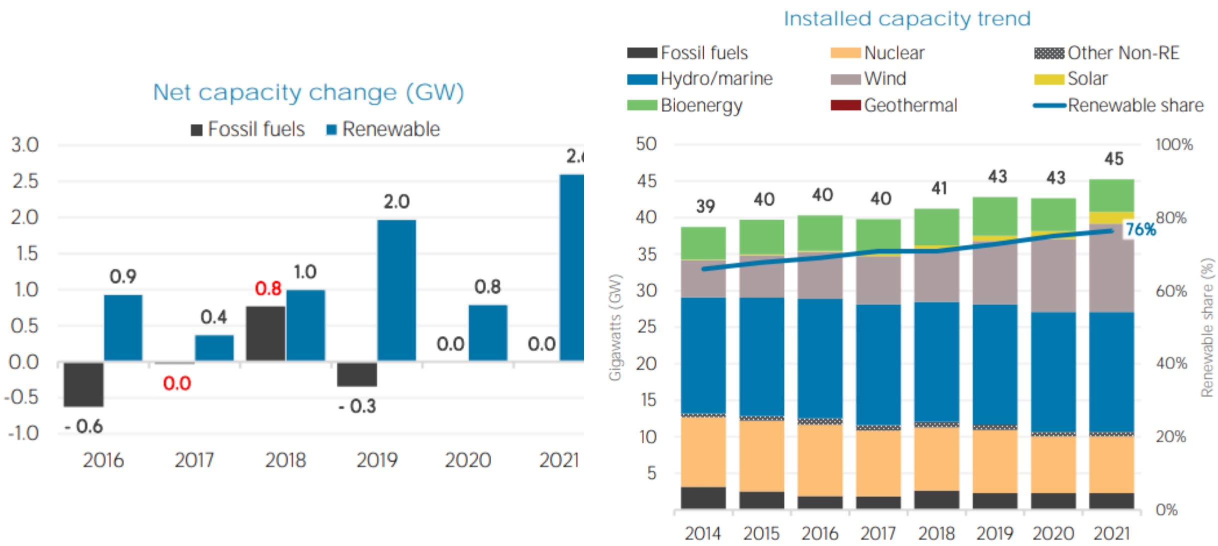


Figure 2.1.1: Evolution of the the share and capacity of renewable energy sources in Sweden [12]

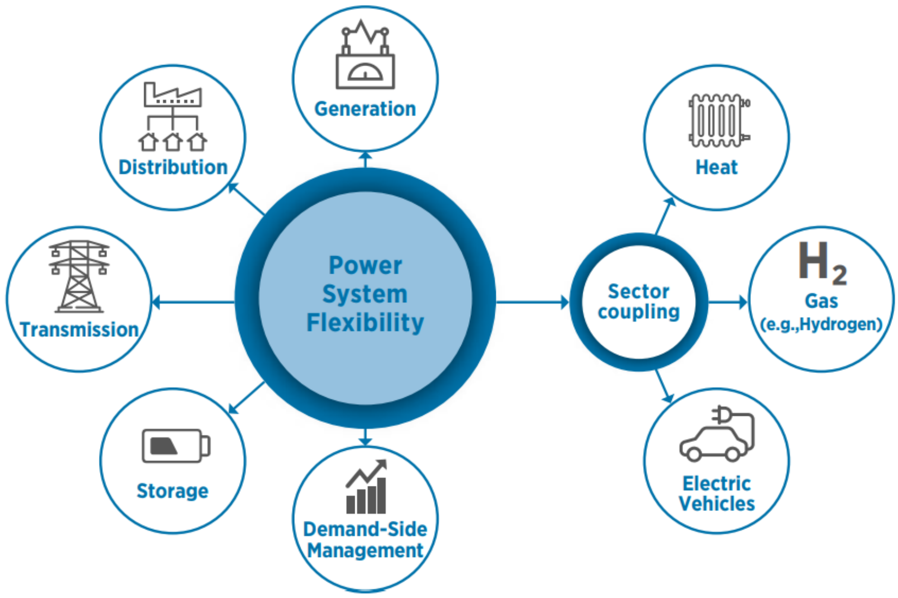


Figure 2.1.2: Power system flexibility enablers [2]

Dynamic Line Rating (DLR) and Network Topology Optimization (NTO) are two examples of technical flexibility, whose aim is to maximise the use of transmission assets at each moment. DLR can help the integration of a higher share of less predictable renewable generation by increasing the grid capacity. Historically, TSO and DSO have used seasonal static thermal ratings for line conductors to calculate their theoretical rather than real ampacity [1]. DLR is the ability to vary the thermal capacity of an overhead power line in real time, depending on varying conditions such as ambient temperature or solar radiation [1]. In practice, dynamic systems use sensors to monitor real-time environmental conditions and maximize power flows. Another way of tackling congestion is through network topology optimization, where the topology is changed by switching on or off transmission lines for instance. The authors of [17] combine network optimization with a weather-based pricing mechanism to evaluate the potential for load curtailment under natural hazards such as windstorms.

In DSM and D/R practices, consumer loads are controlled either technically or through incentives to respond to imbalances in supply and demand. D/R can be defined as "the incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized" [17]. DSM designates a group of technologies such as energy storage or curtailment of loads used to change consumption patterns. DSM drives cost reduction of energy consumption by shifting loads to off-peak periods, thus reducing peak loads and maintaining the balance between electricity demand and supply.

Ref. [13] identifies 6 elements in an energy management system, among which:

- Smart meters which transmit information from the customer to the service provider.
- Renewable energy sources such as solar that can be directly harnessed by so-called prosumers to diminish their electricity bill.
- EV or storage. Plug-in electric vehicles have the potential to behave like distributed energy storage and to discharge power back to the grid at a fast rate. Vehicle-to-grid (V2G) mechanisms may thus be used as spinning reserves [2].
- Load management. Proper scheduling of the use of load is central to optimizing consumer comfort and energy efficiency. Loads can be classified as storable, shiftable, or curtailable, as base load or resulting from self-generation. This will be further developed in the next section.

DSM and D/R are subject to specific constraints to ensure good integration of flexible

[1]Maximum current rating of a power conductor.

[2]The spinning reserve is the amount of capacity which can make up for power shortages or frequency drops within a given period of time

products in the electricity market. Rules control the timing, reliability, and magnitude of customer response. Since trades in electricity markets often involve high volumes, capacity from different consumers may need to be pooled through aggregators. Time-of-use (ToU) tariffs, curtailed load or critical peak pricing are just a few examples of contract types used to monitor consumer behavior.

Residential customers get involved in demand response programs through specific contracts. Ref. [10] distinguish five main types of contracts offered in the electricity market: time of use or dynamic pricing, fixed load or dynamic capping, and direct load control. Constraints can either be imposed by the price of electricity or be defined by a cap on the authorized volume of consumption. In price-based contracts, the tariff for electricity is used to trigger a change in consumption while in incentive-based programs, participants are rewarded depending on load reduction during peak demand. Customers can also choose between static or dynamic contracts. ToU pricing is a static price-based contract whereas dynamic load capping is a dynamic volume-based contract (c.f. Fig. 2.1.3). Eventually, a customer may cede control over specific appliances in a control-based contract.

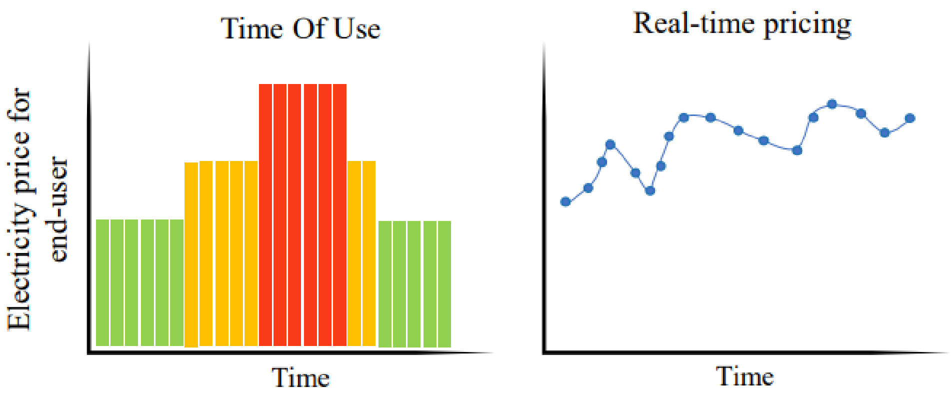


Figure 2.1.3: Example of dynamic contracts: ToU and real-time-based pricing [9]

From the contract provider's perspective, the design of optimal tariffs is a difficult problem. Regulations prohibit from offering specific tariffs to each household. Customers have to be able to choose between keeping their current pricing or shifting to a new tariff. Still, an optimal pricing model should take into account the expected response of households to contract options. Quantifying the number of hours at a time that electricity consumption can be reduced or the frequency to which consumers are willing to shift their habits is key. Each contract comes indeed with its own set of benefits and risks for consumers. Risks can be financial, especially when pricing is dynamic, and non-financial with a loss of autonomy and privacy for control-based contracts. A customer with a price-based contract may end up paying more than if they had a fixed contract while a volume-based contract may curtail consumption at given times and limit the use of some appliances.

Overall, with technical improvement, the Swedish grid is becoming more like a smart grid that supports bidirectional flows of information and electricity. EV encompass the challenges and potential that are the future of the power grid with on the one hand increasing electricity load, but on the other, new solutions such as storage through plug-in mechanisms that may help smooth load profiles [23].

2.2 Smart meter data, time series and Typical Load Profiles

One of the building blocks of load management systems is data feedback from smart meters. Sweden was one of the first countries in Europe to spread out smart meters. As smart meters are owned by DSO in Sweden, the Energimarknadsinspektionen - National Regulatory Authority for Energy (Ei) has developed regulations on minimum requirements for smart meters to ensure equal treatment of consumers. Those requirements aim at providing more information to both consumers and DSO to respectively increase their awareness and help them be more efficient. By 2025, all smart meters should register active energy every hour or fifteen minutes in a way that protects consumer privacy and data security [11].

A set of observations such as electricity measurements taken at specific times is called a time serie. Electricity consumption is a discrete-time serie measured at fixed-time intervals. Figure 2.2.3 shows the electricity consumption of a household over several years. It is easy to detect some outliers where consumption abruptly drops to 0. It also highlights the annual periodicity of the signal, varying with the seasons. Zooming on monthly consumption would also exhibit a weekly pattern. Hourly measurements give a total of 24 data points per consumer per day. Each consumer can be characterized by their load mix, that is to say, the collection of all their daily load curves.

The classification of consumers according to their loads is the topic of many studies trying to evaluate customers' potential to participate in demand response. It may be possible to reduce power demand through load control or increase it to receive

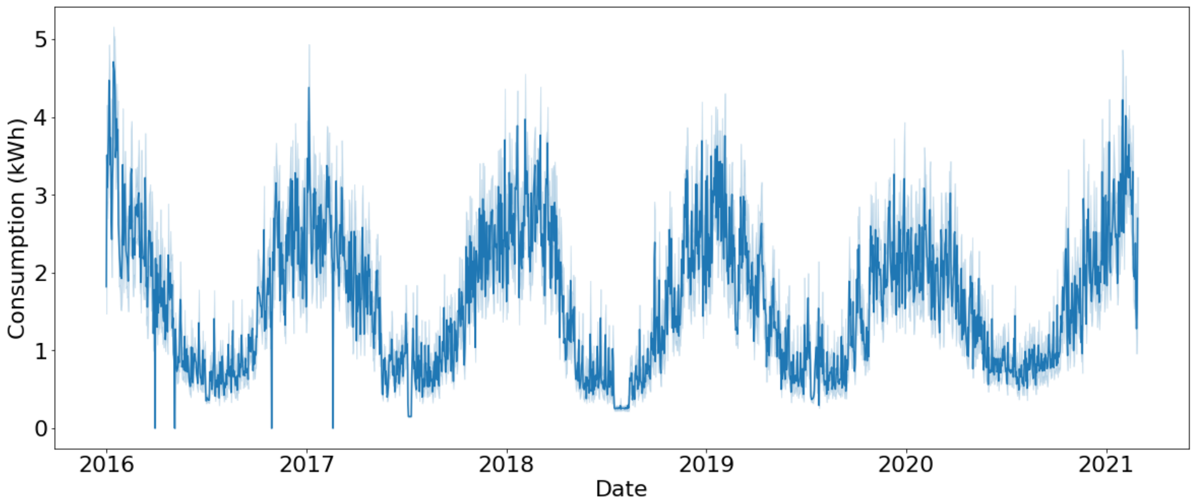


Figure 2.2.1: Load curve of a randomly chosen household between 2016 and 2021

the surplus from variable electricity production [16]. However, not all consumers show the same potential for demand-side flexibility. Load profiling is defined as the classification of load curves or consumers according to electricity consumption behaviors [24]. The goal is to segment consumers based on the flexibility of their load to evaluate their responsiveness to different signals. Data about which and when appliances are being used is usually not available. Load profiling studies are manifold and have various research objectives: from the grouping of similar consumption profiles, TLP can be created that describe a specific behavior or user group. Load profiling can also help identify information about consumers, such as socioeconomic indicators or insight into the type of appliances that they own. With the massive roll-out of smart meters, household clustering could be shifted from an attribute-based approach to a shape-based one [22].

The deployment of D/R programs requires a better knowledge of customers. Several studies have attempted to predict load curves, others try to identify specific appliances from load shapes. The authors of [26] present an algorithm to recognize classical appliances (refrigerators, air conditioners, washing machines, water dispensers, etc.). In [10], loads are classified into different categories, depending on their degree of flexibility (c.f. Fig. 2.2.2):

- Non-storable vs storable loads
- Non-shiftable vs shiftable loads (laundry, dishwasher, vacuum cleaner)
- Base load (instant power is needed and cannot be interrupted such as fire alarms

or freezer) vs curtailable loads (the service can be interrupted instantly such as TV or computer).

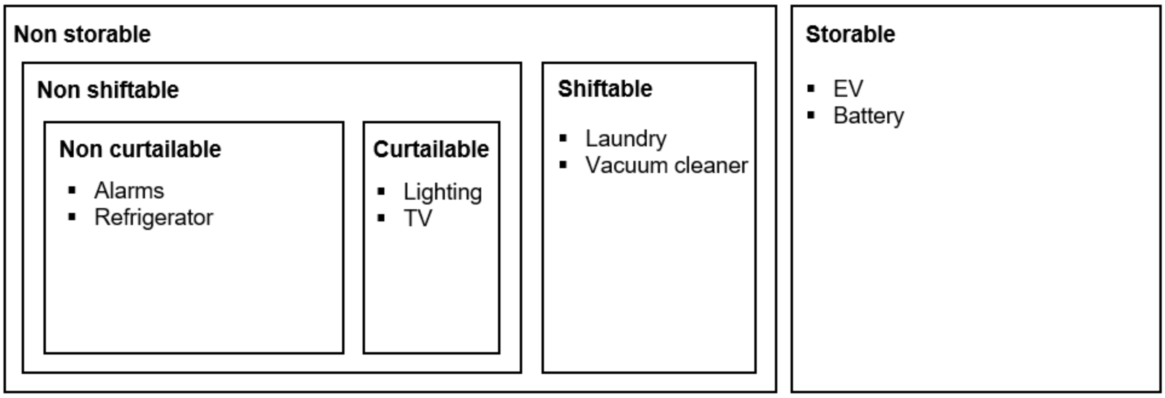


Figure 2.2.2: Classification of loads

Once typical load profiles have been identified, several load management techniques can be applied [13] including:

- Peak shaving to flatten load at the time of peak consumption.
- Valley filling to increase load during off-peak hours to improve load factor[3].
- Load shifting.

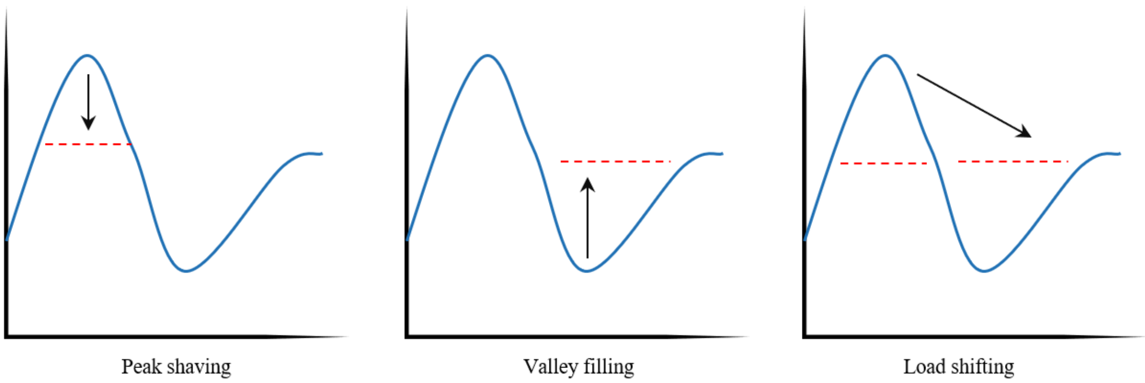


Figure 2.2.3: Load management techniques: peak shaving, valley filling and load shifting

In [16], several indicators for measuring the flexibility gain of load shifting are introduced. Among them, the maximum shiftable rate of an appliance activity encompasses both the shiftable potential and penetration rate[4] of the appliance.

[3]The load factor is the average load divided by the peak load in a specified time period. It is an indicator of the efficiency of energy usage. A high load factor means that the electric system is used more efficiently. $f_{load} = \frac{\text{average load}}{maximumloadinagiventimeperiod}$)

[4]Percentage of targeted customers who own the appliance.

Theoretically, the charging load of an EV can be fully shifted. However, the current infrastructure does not support a significant use of smart control meaning that, in practice, few charging loads can be shifted.

2.3 Machine learning techniques for consumption characterization

With access to smart meter data, machine learning algorithms are increasingly being used for flexibility applications. Among others, Hidden Markov Models have been used for Non-Intrusive Load Monitoring (NILM) also known as energy consumption data disaggregation, to evaluate the contribution of each appliance to the total electricity demand. More recently, deep learning models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) neural networks have been tested for flexibility prediction based on indoor and outdoor temperatures and heat pump consumption [6].

During the last decade, many authors have worked on inferring the relationship between customer characteristics and their consumption, either estimating load profiles according to consumer information or deriving information about consumers from smart meter data. The authors of [25] identify four challenges related to the depiction of consumer behavior: pattern recognition, personal price design, socio-demographic information identification, and household behavior coding. The main applications are for individual and aggregated forecasting of loads. Ref. [25] gives an overview of how smart meter data analytics is used to model electricity consumer behavior. As mentioned, one area of research is the relationship between the socioeconomic status of individuals and their consumption. The authors propose an automatic feature extraction method based on deep learning to derive socio-demographic characteristics from consumers such as sex, age, employment, or social class. One of the main challenges of load clustering is to detect the high similarity of loads slightly shifted in time. Convolutional neural networks is a good model for load profiling as the features computed in the convolutional layer are invariant to small shifts, allowing stable features to be obtained from varying load profiles. Besides, many factors such as weather conditions or the day of the week can affect load profiles. The correlations between electricity consumption and these factors are nonlinear and can be modeled by neural networks with multiple layers.

2.4 Clustering

Clustering consists in finding similarities between data according to characteristics of the data and grouping similar objects into clusters so that data within the same group are related and data in different groups are unrelated. Good clusters have high intra-cluster similarity and low inter-cluster similarity. Clustering can be used as a stand-alone tool to get further insights about the data or as a preprocessing step for other algorithms. Load profiling is divided into direct-clustering-based and indirect-clustering-based approaches. The most popular clustering techniques are k-means, hierarchical clustering, and Self-Organizing Map (SOM), which can be directly used to group load curves. Indirect clustering uses features extracted from input data before clustering [24].

2.4.1 Clustering techniques

Review of the main approaches

The main clustering approaches are partitioning, hierarchical clustering, density-based and grid-based approaches. In this report, we are interested in the first two families.

Partitioning consists in building several partitions of the data and then evaluating them against a criterion such as inertia (within-cluster sum-of-squares). k-means is one of the most well-known partitioning algorithms that requires predefining the number of clusters and runs in linear time. Data points are clustered trying to create groups of equal variance, minimizing the inertia:

$$\sum_{i,k \text{ s.t. } s_i \in Cl_k} \|s_i - C_k\|^2$$

Each cluster is described by the mean (centroid C_k) of the data points $\{s_i\}$ in the cluster Cl_k. Centroids are not necessarily points in the dataset, contrary to medoids[5]. The points in a cluster are those that are closest to its centroid than any other centroid.

[5]Medoid: point in the cluster closest to the centroid.

Algorithm 1 `k-means`

Input Data points $\{s_i\}$, number of clusters K
Output Results of the clustering: clusters $\{Cl_k\}_{k\in\{1,\ldots,K\}}$ and centroids $\{C_k\}_{k\in\{1,\ldots,K\}}$

```
Random initialization of centroids {C_k}_{k∈{1,...,K}}
while any of the {s_i} changes cluster do
    Attribute each point s_i to the closest centroid C_k
    Recompute all cluster centers {C_k}_{k∈{1,...,K}}
end while
return Clustering results
```

The k-means algorithm always converges but not always to a global optimum. The performance of the algorithm depends tightly on the initialization of the cluster centers. The k-means++ scheme initializes centroids distant from each other, generally producing better results than random initialization [4]. The k-means algorithm scales well to large numbers of samples but it is sensitive to noise and outliers.

Hierarchical clustering is a family of clustering algorithms generally divided into two types:

- Bottom-up approach or agglomerative: the algorithm starts with each data point in its own cluster. Neighboring clusters are then iteratively joined.
- Top-down approach or divisive: in the beginning, all data are in same cluster which is then split recursively.

One of the advantages of hierarchical clustering is that the number of clusters does not need to be defined beforehand. Besides, it is possible to represent results with a dendrogram, which makes it interpretable. However, Hierarchical Agglomerative Clustering (HAC) has a time complexity of $\mathcal{O}(n^3)$ and is unstable.

To decide which clusters to combine first, a measure of similarity between clusters is needed. The linkage criterion determines the distance between clusters as a function of the pairwise distances between data points. The lower level metric (distance) determines which data points are closest neighbors, whereas the linkage criterion acts on the shape of clusters. Four linkage criteria are implemented in the Python library "scikit-learn":

- Ward is similar to the k-means objective function: it minimizes the sum of squared differences within all clusters.
- Complete linkage minimizes the maximum distance between data points

belonging to different clusters.

- Average linkage minimizes the average of the distances between data points belonging to different clusters.
- Single linkage minimizes the distance between the closest data points of pairs of different clusters.

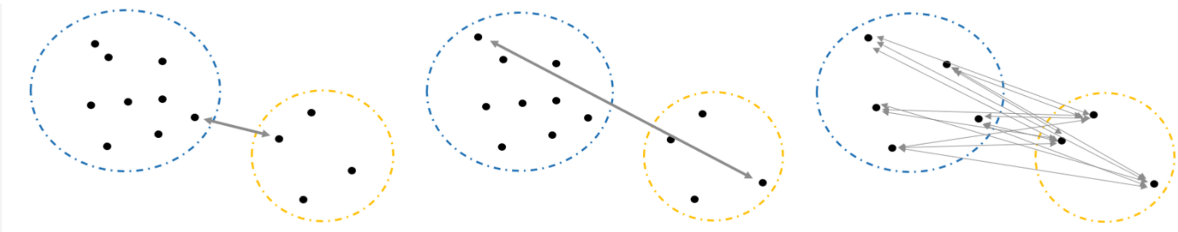


Figure 2.4.1: Single, complete and average linkages

Single, average and complete linkage can be used with any affinity matrix or distances such as $l1$, $l2$ or cosine distance. The metric should maximize the distance between points in different clusters and minimize the distance within each cluster.

Evaluation of clusters

The quality of the clustering can be evaluated with the similarity measure used for clustering and the ability to discover hidden patterns. It can be based on the shape of clusters, their stability (sensitivity to noise), or domain knowledge. Direct evaluation in the relevant application is the best way to measure the effectiveness of clustering for a given task, but it can be expensive to perform.

The accuracy of the clustering can be evaluated according to two types of measures:

- An external index, such as cluster purity, in case a ground truth such as a class label is given.
- An internal criterion, which measures the quality of the clusters without using external information. Objective functions in clustering generally seek to maximize intra-cluster similarity (points within a cluster are similar) while minimizing low inter-cluster similarity (instances of different clusters are dissimilar). The Silhouette coefficient S measures how close a point is to its

neighbors within the same cluster compared to other clusters:

$$S = \max_i \frac{b(i) - a(i)}{max(a(i), b(i))}$$

where $a(i)$ is the mean distance between data point i and its cluster neighbors and $b(i)$ is the minimum of the mean distance between data point i and the data points in another cluster.

2.4.2 Clustering of residential loads and customers

Existing studies on load profiling mainly focus on industries and businesses with relatively regular consumption patterns. Residential customers bring new challenges, as their load patterns may vary greatly between two customers but also evolve over time. The daily consumption of a customer can therefore not be described with only one typical profile. Clustering is an effective way to address the high dimensionality of smart meter data.

In [22], a method to analyze DLC of residential customers and improve D/R targeting is developed. A reduction technique is applied that identifies the main time periods of activity during the day. Instead of using highly variable load curves directly, their reduced representations are grouped into clusters. The results are then used to help in the tailoring of D/R programs for households.

In [25], a clustering method based on transitions between consumption behaviors is developed. The dynamics of electricity consumption are used as a factor for clustering and are represented with a time-dependent Markov model. The underlying assumption is that future consumption behaviors are related to current states. The method tackles the challenge of the high dimensionality of fine granularity datasets with a distributed clustering algorithm based on a divide-and-conquer approach, where adaptive k-means is applied at local sites and a modified Fast Search and Find of Density Peaks (CFSFDP) method is performed at global sites.

Load forecasting is a challenging task, especially at the household level. Forecasting is conducted at different time horizons, ranging from day-ahead with Short-term Load Forecasting (STLF) to a couple of years ahead with LSTM. [21] shows the value of load profiling for short-term forecasting of the consumption of residential customers. One way of forecasting system load is first to use smart meter data for individual household forecasting and then to aggregate the predicted loads to build a high-level forecasting model. In this paper, low-level (i.e. smart meter) data are first clustered by intraday consumption behavior with an emphasis on coincident demand, highlighting customers who contribute most to the total consumption at specific times. The system load is built using averaging of clusters on each segment of the day.

2.5 Classification and feature importance

In this degree project, the focus is on non-parametric supervised learning methods for classification. The aim is not so much to perform classification tasks as to extract the characteristic features of a given group. For this purpose, we use algorithms for which it is possible to estimate the relative importance of the features with respect to the prediction of the target variable. In order to improve the quality of the predictors, two assembly methods have been tried.

2.5.1 Ensemble methods

The goal of ensemble methods is to combine the predictions of several base estimators to improve robustness. Two families of ensemble methods exist:

- Averaging methods output the averaged prediction of a set of classifiers independently built. Bagging methods and random forests are two examples of algorithms that help reduce the variance of predictions.
- Boosting methods like AdaBoost combine weak models built sequentially to gradually reduce the bias of the combined estimator.

Base classifiers of ensemble methods are different due to different samplings of training data or parameter values.

A decision tree predicts the value of a target variable by learning simple decision rules from the data. Algorithms to construct trees choose a variable at each node that best splits the sample according to a metric which measures the homogeneity of the target variable within the created subsets. For instance, information gain can be used to decide which feature to split on at each node. It measures the reduction in entropy[6] before and after the split, i.e. the entropy reduction due to the addition of

[6] Given a discrete random variable $\mathcal{X}$ with values in χ and a probability distribution $p : \chi \to [0, 1]$, the

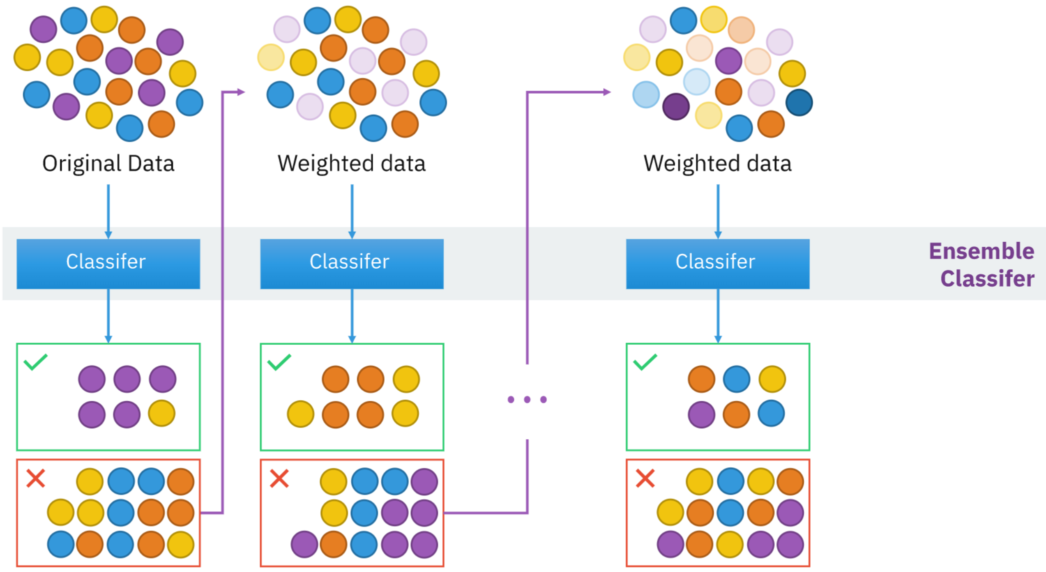


Figure 2.5.1: Illustration of a boosting method with parallel learners and weighted datasets [5]

a new attribute. As the tree grows, the decision rules become more complex. In a classification tree, leaves represent class labels and branches represent a combination of characteristics that lead to the label. A small decision tree is often a weak classifier, with predictions slightly better than random guesses. Random forest is a perturb-and-combine averaging algorithm based on decision trees.

The boosting algorithm AdaBoost trains individual classifiers sequentially, iteratively re-weighting the training examples so that each classifier is trained based on knowledge of the performance of previously trained classifiers. The training data is re-weighted to emphasize the hard cases, i.e. instances that were misclassified in the previous step. The final classifier is a weighted sum of the component classifiers where each classifier is weighted by the quality of their individual predictions.

2.5.2 Classification evaluation

There are several ways of evaluating the performance of a classification model. A confusion matrix summarises the results of the classification:

Accuracy is the proportion of correctly classified instances to the total number of instances:

$$acc = \frac{TP + TN}{TP + FP + FN + TN}$$

entropy $\mathcal{H}$ of $\mathcal{X}$ is defined as $\mathcal{H}(X) := -\sum_{x \in \mathcal{X}} p(x) \log p(x)$.

		True class 1	True class 0
Prediction	1	True Positive (TP)	False Positive (FP)
	0	False Negative (FN)	True Negative (TN)

Table 2.5.1: Confusion matrix

It is an intuitive metric, but it can be misleading if classes are imbalanced. More generally, the choice of the metric is above all a matter of the problem to be addressed. If a poisonous mushroom recognition program is developed, it is imperative that a mushroom that is not edible is not classified as such, i.e. the priority is to reduce the number of false negatives. It is less serious to wrongly classify a mushroom poisonous even if it is not (false positive). Precision (true positive instances over the total number of positive instances) measures the ability to avoid false positives while sensitivity (true positive rate, i.e. true positive instances over the total number of positive instances) is a measure of the estimator's ability to find all the positive instances. The $F1$ score combines precision and sensitivity.

2.5.3 Feature importance

There are several common methods for evaluating feature importance in machine learning models. The tree-based models implemented in the scikit-learn library have a built-in feature importance attribute. The features that are located at the top of the trees are those that impact the largest number of inputs. The fraction of the sample that a feature influences can be used to measure the relative importance of that feature. In scikit-learn, the importance of the feature is calculated using the fraction of inputs and the decrease in impurity[7] due to splitting the data at the node of the feature. Then the feature importance is averaged over the different trees. This method suffers from two main shortcomings: on the one hand, it evaluates the importance of the features based on statistics performed on the training set. This means that this technique gives an evaluation of the importance of the feature to classify points in the training set, however it is unclear how this generalizes to other datasets. On the other hand, it gives more importance to features with high cardinalities.

The permutation method does not have these shortcomings. The permutation importance consists in measuring the impact on predictions of randomly shuffling the

[7]Impurity is a measure of homogeneity of the labels at a node.

values of a feature. If a feature is important, randomly shuffling its value should have a negative impact on the model results. Recursive feature removal is a similar approach in which the least important features are gradually removed. The model is re-trained on a reduced number of features and its performance with and without a feature is compared.

2.5.4 Imbalanced datasets

By using different subsets of data or features, ensemble classifiers improve performance over basic classifiers. However, class imbalance affects predictions and classifiers tend to favor majority classes. Techniques to deal with this problem include oversampling and undersampling. In the first case, new samples are generated in the underrepresented class using data preprocessing such as sampling with replacement. Algorithms such as ADASYN, which focuses on difficult examples, and SMOTE use interpolation to create new samples. In the latter case, samples are generated from the original set instead of being selected. Clustering is used to reduce the number of samples in over-represented classes: instead of training on the original data, the centroids of the clusters are used for the classification task. Subsampling generally gives better results than oversampling but can lead to a loss of information. In [7], two methods are proposed to obtain a balanced random forest: weighted and balanced random forests.

- The weighted random forest has a greater penalty for misclassification of minority class instances.
- The balanced random forest combines ensemble learning with a reduced sampling of the majority class.

Chapter 3

Methodology

This chapter presents the data used in the thesis and gives a theoretical description of methodologies and methods applied in the degree project.

The aim of the work is to adapt and assess the DSM-related benefits of the method from [14] to characterize the energy lifestyle of residential consumers from their electricity consumption records. The steps of the method are presented in figure 3.0.1. Smart meter data are decomposed into daily load shapes (normalized load curves) and daily total usage. Load shapes are associated to TLP. The first step of the method, taking inspiration from [14], is thus to create a set of typical load shapes that cover the variety of uses of residential consumers. Then, DLC are encoded with the help of the dictionary of TLP. Features, such as vectors of TLP frequency, are extracted from this representation. The method is tested on several tasks, including the creation of groups of users with similar consumption behavior and the identification of information about customers. In broad terms, the following steps are taken:

- Smart meter data are preprocessed to obtain DLC. Households are divided into different samples.
- DLC are decomposed into a load shape (normalized consumption) and total daily consumption. A dictionary of TLP is built based on a sample of the data. Modifications are proposed to speed up the running time of the algorithms used in [14].
- Distribution vectors are created to describe the frequency of TLP in the consumption records of residential customers and group users with similar

consumption habits.

- Detection of attributes and characterization of customers relevant to the development of DSM are tested.

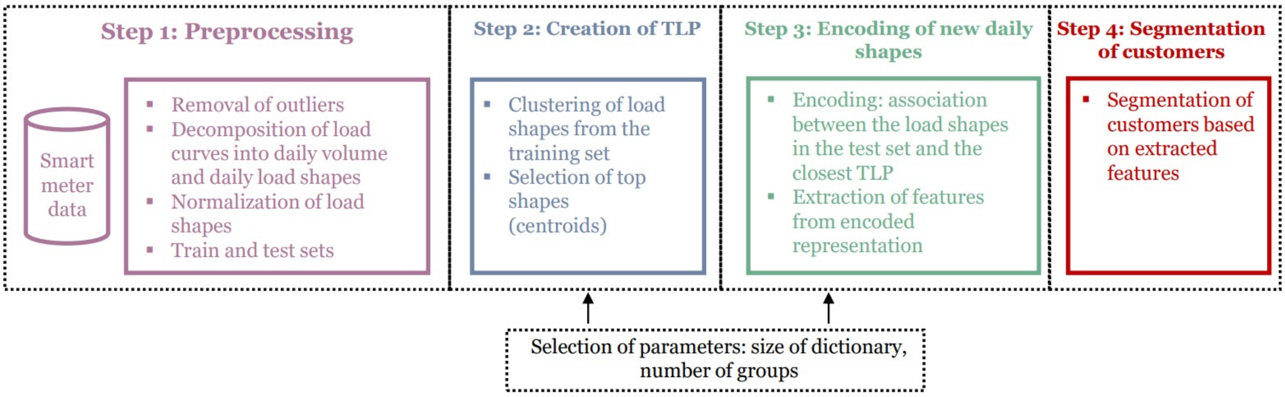


Figure 3.0.1: Steps of the segmentation task [22]

3.1 Data presentation and preprocessing

3.1.1 Presentation of smart meter data

The dataset is made of the consumption records of one- or two-dwelling houses. The measurements correspond to the period between 2016-01-01 and 2021-02-28, or a maximum of 1,460 DLC per smart meter. Two sets are created, one to create the dictionary of TLP, the other to evaluate the methodology. Each is made of the consumption records of 1,000 smart meters, without overlap between the two sets. In the beginning, only the data until 2020 were kept to avoid the potential confounding effect of COVID-19 on consumption. However, for the purposes of the study concerning the profiles of the users of EV, most of whom were purchased in 2020 or later, the data set was finally extended to 2021. The customers are distributed in two electricity network regions, Uppsala (UPP) and Huvudsta (HUV) (cf. Fig. 3.1.1).

3.1.2 Removal of missing data and outliers

The first step in preprocessing is to deal with missing values and outliers. A missing value means a null consumption. The clustering methodology used in the project takes as input the daily consumption curves of the households, i.e. a list of 24 values per day and per customer. A simple way to solve the problem of missing data is with

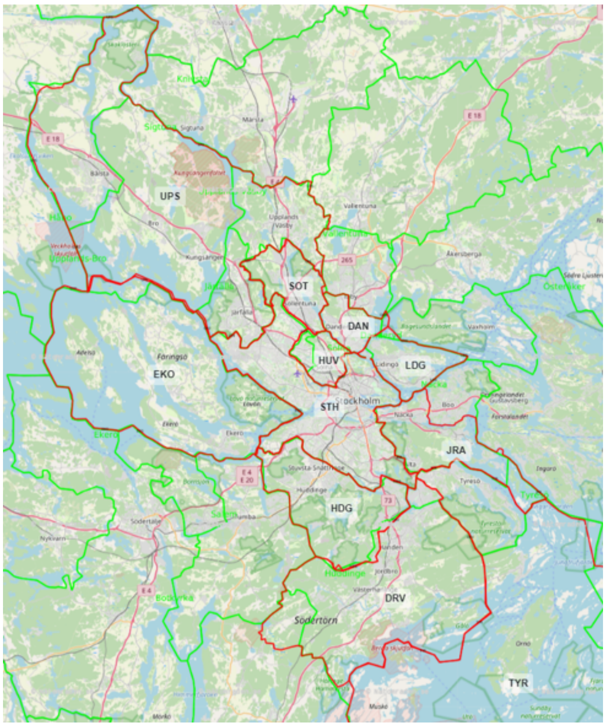


Figure 3.1.1: Local grid areas in the Stockholm region. Credits: `Nätområden.se`

interpolation, which consists in taking the values on both sides of the missing point and averaging them. However, this technique is problematic when a smart meter stops recording or transmitting data for several hours. As the data set available is very large, the final decision is to delete the missing values, i.e. all the daily consumption lists where at least one hour of consumption has not been recorded. There are few isolated missing points, but rather day or week-long periods when consumption is not recorded. Thus, there is little loss of data due to the use of a list of 24 values rather than isolated measurements.

The second problem is to identify outliers and implement an appropriate strategy to remove or replace them. The method adopted consists in filtering the extreme consumption values in relation to seasonal averages. The 5% most extreme values are removed. The reason behind this is that marginal behaviours are not of great interest for consumer segmentation. Thus, one can filter out these extreme consumptions. Moreover, the data set is very large, the results are therefore not impacted by the withdrawal of a few DLC.

3.1.3 Analysis of consumption data

To deal with the huge size of the data set, made of a billion DLC, two samples of 1,000 smart meters are randomly created. The training data set is used to build a set of TLP. The other is needed to test the methodology and evaluate the quality of the dictionary. Figure 3.1.2 shows the box plot of daily household consumption for each month of the year (January is 1 and December is 12) in the training data set. The months are divided into two seasons: summer goes from April to September and winter from October to March. The graph is created after removal of outliers because unexpectedly high levels of consumption distort the estimates. The creation of typical load patterns should not be negatively impacted by the removal of extreme values.

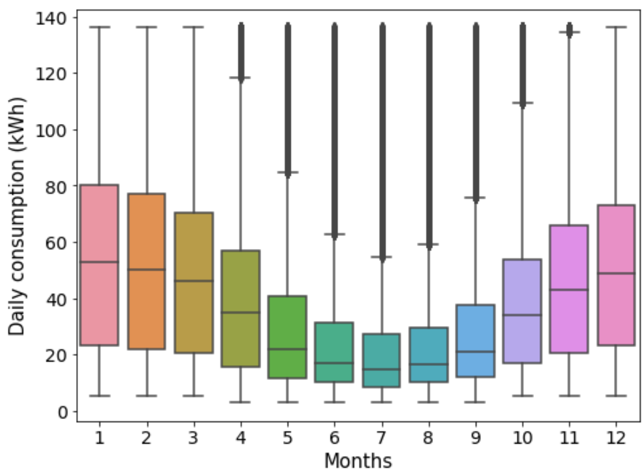


Figure 3.1.2: Distribution of daily consumption (kWh) over the months of the year

Figure 3.1.3 shows the distribution of the daily consumption of the DLC of the training set. It exhibits a long tail in the direction of high consumption, while most daily consumption is spread around 15 kWh.

The statistics in 3.1.1 and 3.1.2 describe household daily consumption (in kWh) during winter and summer. Consumption ranges from a median value of 20 kWh and an average of 32 kWh in the summer compared to 49 kWh, respectively 65 kWh, in the winter. Consumption is significantly higher in the winter months than in the summer. The average and median consumption values are more than twice as high in winter as in summer. The main focus of energy consumption is heating, with significantly higher consumption during the Swedish winter, mainly because of colder temperatures.

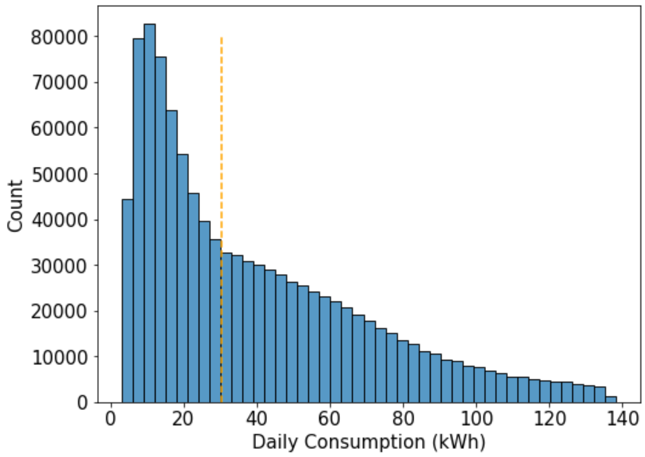


Figure 3.1.3: Distribution of daily consumption and median consumption (dotted line)

Mean	Standard Deviation	25% quantile	Median	75% quantile
65.13	69.79	21.42	49.00	81.43

Table 3.1.1: Household daily consumption (kWh) for the winter season

Mean	Standard Deviation	25% quantile	Median	75% quantile
32.48	38.87	10.69	19.89	40.23

Table 3.1.2: Household daily consumption (kWh) for the summer season

3.2 Encoding of Daily Load Curves

In this part, we explain how Kwac et al. [14] transform the consumption data into a standardized reduced representation, which allows comparison of consumption patterns between groups of customers. Our contribution consists in a simplification of the overall methodology and the implementation of approximation-algorithms to save time. Our suggestions come from tests carried out following the general framework developed in [14] but relaxing the cluster coherence constraints in the clustering algorithms.

The main steps of the encoding, developed by Kwac et al. in [14] and summarized in figure 3.2.1, consist in:

1. Decomposition of load curves into daily load shapes and daily total usage (volume of consumption)
2. Creation of a set of TLP
3. Matching DLC to their closest TLP

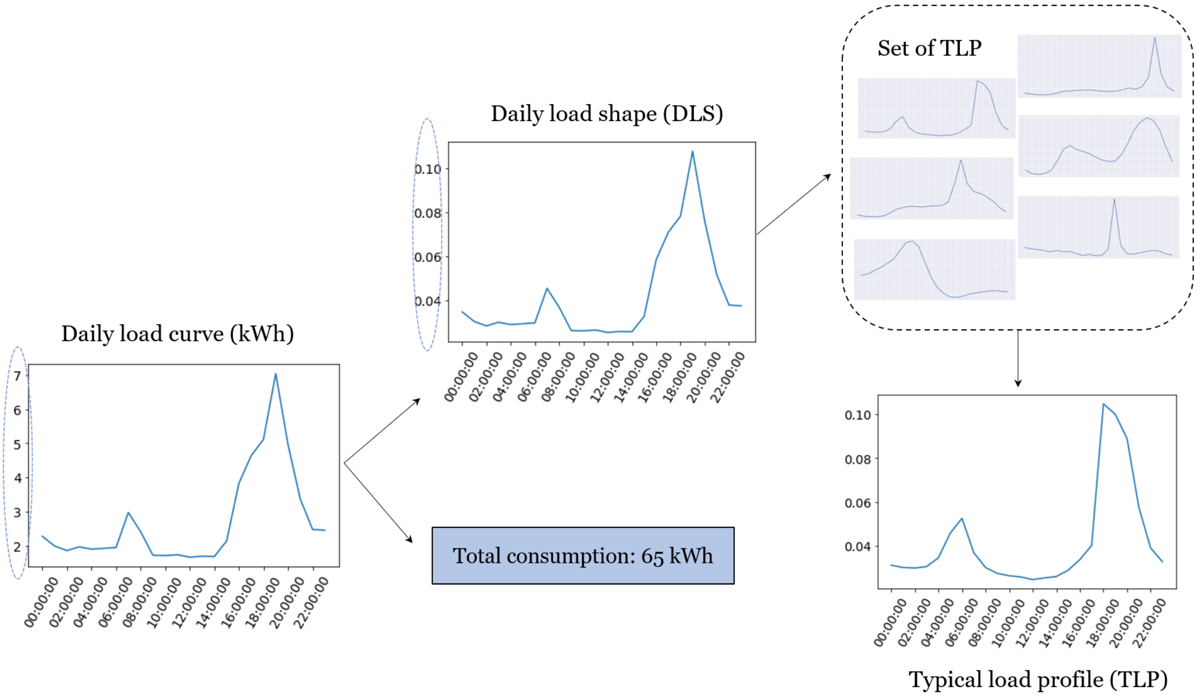


Figure 3.2.1: Encoding of DLC

3.2.1 Method of Kwac et al. [14] to create the set of Typical Load Profiles

Kwac et al. [14] create the set of TLP in two steps. First, adaptive k-means is used to create consistent clusters of loads. Then, correlated clusters are merged with HAC. The general framework of the method to build a dictionary of relevant TLP is studied in this section. A simplification of the intermediate algorithms, which reduces the computation time, is proposed in the next section.

Let us note $l_{i,j}$ the load curve of customer i on day j and $s_{i,j}$ their load shape.

$$l_{i,j} = \{m_{ij,0}, ..., m_{ij,23}\} \text{ and } s_{i,j} = \{nm_{ij,0}, ..., nm_{ij,23}\}$$

where m_{ijt} is the hourly consumption of customer i on day j during hour t and nm_{ijt} is the normalized hourly consumption of customer i on day j during hour t such that $nm_{ijt} = \frac{m_{ijt}}{\sum_{t'} m_{ijt'}}$.

The dictionary of TLP is created to encode loads. The encoding consists in the association of load shapes to TLP of the dictionary. The set of TLP is built on a sample of the data, both to speed up calculation and to ensure that the resulting dictionary is able to describe the consumption profiles of unseen households. It must satisfy two main criteria:

- It must have good coverage. This means that the majority of load shapes in the dataset are sufficiently close to at least one typical shape (and preferably only one). There may be noisy behaviors, which are not expected to be closed to a TLP. The notion of coverage is tightly connected to the coherence of patterns within a cluster, i.e. to the intra-cluster similarity and also the generalisability of the model: how well does it perform when we move from classifying loads in the building dataset to loads in an evaluation dataset?

- The dictionary must also be consistent. The TLP obtained by creating several dictionaries from different samples of households should be close. Consistency is linked to the notion of robustness of the model, which depends on the size of the sample selected to build the TLP. In our case, the training sample had to be small enough to keep the calculation time low, but big enough to represent diverse behaviors.

TLP are created with the help of two clustering algorithms. Load shapes are divided into groups, whose centers, which are an average of the shapes in the cluster, constitute the TLP. The main difficulty of this method is to determine the optimal number T of clusters according to the two main criteria. To have good coverage, a dictionary must contain enough TLP to cover the variety of consumption profiles. If the number of clusters is too small, data that represent very different consumption behaviors will end up in the same group. On the other hand, if the number of clusters is too large, there is a risk of overfitting to perfectly match the shapes of the sample used to create the dictionary. The dictionary will thus be poorly suited to describe other samples.

Kwac et al.'s method to produce TLP is based on a two-step procedure (c.f. "Method" of Fig. 3.2.2). Daily load shapes of the training samples are divided into clusters using adaptive k-means. A first set of TLP is made with the cluster centers. The size of the set is reduced by successively merging the clusters of the closest TLP until the desired size of the dictionary is reached. The goal of the first step is to find a number K of clusters that guarantees good coverage, i.e. that the curves within a cluster are close to each other. In the second step, a hierarchical clustering algorithm is used to group together clusters that are highly correlated.

The two clustering algorithms used to create TLP in [14], adaptive k-means and hierarchical clustering, are reviewed. The focus of the method is on defining the right

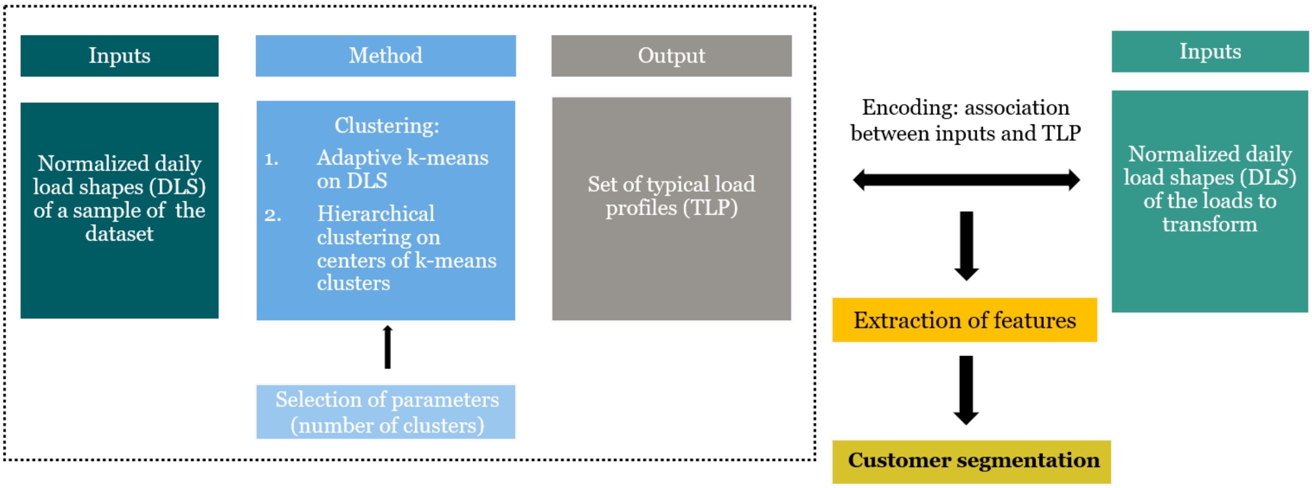


Figure 3.2.2: Encoding of load curves for customer segmentation

number of TLP.

1. Adaptive k-means

The modified version of k-means clustering proposed in [14] does not require specifying a priori the final number of clusters K but rather a threshold θ which controls the distance between the elements of a cluster k and the cluster center $C_k = \{\overline{m}_{k,0}, ..., \overline{m}_{k,23}\}$, where $\forall i \in \{0, ..., 23\}$, $\overline{m}_{k,i}$ is the mean of all the i_{th} coordinates of the load shapes in k.

For each cluster Cl_k resulting from the initialization, if there exists a load shape s in Cl_k such that the normalized distance between s and the cluster center C_k is higher than the threshold θ, then the cluster Cl_k is divided using a 2-means clustering. The number of clusters K is adjusted accordingly and k-means is run once again, using the current centers updated with the result of 2-means for the initialization. This procedure is repeated until the condition 3.1 is satisfied for all the load shapes or the number of clusters exceed an upper limit.

Let us note $\|.\|_2$ the Euclidean norm such that $\forall x \in \mathbb{R}^n$, $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$. Given that a TLP is not null, the threshold condition for i, j is:

$$\min_k \frac{\|s_{ij} - C_k\|_2^2}{\|C_k\|_2^2} \leq \theta \tag{3.1}$$

The squared distance between the load shape and the centroid is divided by the norm of the centroid to ensure that the threshold condition 3.1 guarantees the same

Algorithm 2 `Adaptive k-means`

Input Daily load shapes $\{s_{ij}\}$, minimum and maximum numbers of clusters min_k and max_k, threshold θ
Output Results of the clustering: final number of clusters K, clusters $\{Cl_k\}_{k\in\{1,\dots,K\}}$ and centroids $\{C_k\}_{k\in\{1,\dots,K\}}$

$K \leftarrow min_k$ ▷ Number of clusters
$N_v \leftarrow max_k$ ▷ Number of clusters violating 3.1
Random initialization of centroids ▷ scikit-learn 'k-means++' initialization
while $N_v \neq 0$ **do**
  if $K > max_k$ **then**
    return Failure to converge
  else
    Run K-means algorithm with $\{C_k\}_{k\in\{1,\dots,K\}}$ as initial centroids to set/update $\{Cl_k\}_{k\in\{1,\dots,K\}}$ and $\{C_k\}_{k\in\{1,\dots,K\}}$
    $N_v \leftarrow 0$
    for k = 1 to K **do**
      if $\exists s_{ij} \in Cl_k$ s.t. 3.1 is violated **then**
        $N_v \leftarrow N_v + 1$
        Run 2-means algorithm to divide Cl_k into Cl_{k_1} and Cl_{k_2}
      end if
    end for
    $K \leftarrow K + N_v$
    Update $\{C_k\}_{k\in\{1,\dots,K\}}$ with the new cluster centers
  end if
end while
return Clustering results

intra-cluster purity over all groups. For non-trivial solutions, θ should be chosen in $[0, 2]$.

2. Hierarchical clustering

As mentioned before, the first algorithm guarantees low cluster inertia. However, there is a risk of creating many small and highly correlated clusters whose resulting centers exhibit the same behavior, thus inducing redundancy in the typical load profiles. The second algorithm, therefore, aims at assembling the closest centroids to reduce the size of the final dictionary. The hierarchical clustering algorithm is applied to the centroids of the k-means clusters and not to the load curves directly: it is therefore not costly to calculate the pair-wise distances between all the cluster centers, whereas it would have been impossible to apply such a method on the initial dataset. The algorithm takes as input the final size T of the dictionary. The choice of T is made by taking the minimal number of clusters that allows a coverage of 95% of the load shapes, respecting the θ condition.

Algorithm 3 `Hierarchical clustering`

Input Size of the dictionary T, number of clusters K, clusters $\{Cl_k\}_{k\in\{1,\ldots,K\}}$ and centroids $\{C_k\}_{k\in\{1,\ldots,K\}}$ returned by algorithm 2
Output Typical load profiles $\{C_k\}_{k\in\{1,\ldots,T\}}$

while $K > T$ **do**
  $K \leftarrow K - 1$
  Find the two closest centers C_i and C_j
  $Cl_i \leftarrow Cl_i \cup Cl_j$ ▷ Merge corresponding clusters Cl_i and Cl_j of sizes n_i and n_j
  $C_i \leftarrow \frac{n_i C_i + n_j C_j}{n_i + n_j}$ ▷ Compute the barycentre of the original centroids weighted by their sizes
  Update C_i and delete C_j.
end while
return Typical load profiles $\{C_k\}_{k\in\{1,\ldots,T\}}$

The distance matrix between centroids is initiated at the beginning of the algorithm and then updated at each step with the new distances between the centroids from the merged clusters and the rest of the centers.

Discussion about the method of Kwac et al. [14]

The method developed in [14] is a two-step process: the adaptive k-means clustering insures that loads inside a cluster are not too far from the center. This is done by imposing a threshold θ on the distance between load shapes inside a cluster and the cluster center. Correlated clusters are then merged by hierarchical clustering on the centroids. The advantage of this method over classical clustering approaches is that a reasonable size for the dictionary is determined at the same time as it is constructed. Indeed, in k-means clustering, the number of clusters k has to be pre-assigned, which is impractical with time-series because data sets are very large and checks for determining the number of clusters are complicated.

The first step allows reducing the dimension of the original dataset with k-means which offers the possibility to control the quality of the resulting clusters thanks to the θ parameter. Then, hierarchical clustering can be used on the centroids returned in the first step. This algorithm has an expensive time complexity in $\mathcal{O}(K^3)$ which makes it impractical for direct classification of load shapes. It has the disadvantage of producing unbalanced cluster sizes[1], but it is particularly useful for applications requiring a comparison of the similarity of different pairs of instances, as in the case of nearest neighbor search. It is therefore an ideal candidate to correct the results of the k-means algorithm. By combining two clustering algorithms, the method developed in [14] aims at building a stable and coherent dictionary.

One of the shortcomings of the adapted k-means is that the algorithm only stops under two conditions:

- Either the number of clusters K exceeds k_{max},
- Either the number of violated clusters N_v is equal to 0.

However, when N_v is close to 0, the procedure can become very slow. Let's imagine the case when there is an outlier in the data and $N_v = 1$. The algorithm continues to run until this outlier and all other curves finally meet the threshold condition. To do this, at each new iteration, N_v additional clusters are created and the k-means algorithm runs to reassign each curve to a cluster. The number of clusters evolves very slowly because N_v is small. However, each iteration is time-consuming. Moreover, in the example, the quality of the clustering does not improve as new clusters are created. Therefore, the algorithm could be improved by adding a convergence criterion to stop it. Two options could be explored:

- One possible solution is to use a tolerance value to terminate the algorithm if the

[1]"Rich get richer" behavior c.f. 2.3. Clustering — scikit-learn 1.2.0 documentation

difference between the centroids of two consecutive iterations is smaller than a threshold value.

- Alternatively, the algorithm is terminated when the inertia between two successive iterations is less than a given value.

In place of the adaptive k-means, we use an approximation algorithm and stop it when N_v reaches a lower bound.

3.2.2 Adaptation of the method of Kwac et al. [14]

Our approach is to simplify the overall methodology to create the set of TLP. The encoding method inspired by [14] is reduced to the following steps:

- Generation of a set of load curves large enough to cover the variety of consumptions in the initial dataset. The goal is to quickly create a set of varied load profiles. Two approaches are developed: a simple k-means algorithm and a clustering algorithm inspired by the θ-adaptive k-means (see algorithm 4).
- Clustering of the most similar loads to further reduce the size of the dictionary and make it more stable. As in [14], hierarchical clustering is used to group centroids. Algorithm 5 is proposed as an alternative to 3 in case the number of centroids produced during step 1 is too high.

We take example from the θ-adaptive k-means to propose a clustering approximation, sub-optimal compared to the k-means algorithm but faster to obtain a set of diversified TLP. Clusters are initialized using k-means with a small k to create a rough division of loads and then to successively redivide the clusters until the desired number of clusters is obtained or until the iterative solutions converge (see algorithm 4).

When K is small (of the order of 100), it is possible to use the algorithm 3 to further reduce the size of the dictionary and cluster the centroids obtained in the first step. On the other hand, when K is large (of the order of 10,000), the procedure is costly because it is necessary to initially calculate all the pair-wise distances between centroids. Then, at each iteration, the distances between the new center resulting from the merging of the nearest neighbors and the remaining centroids need to be updated. Contrary to the algorithm 3, the algorithm 5 does not take into account the weights of each centroid to adjust the pair-wise distance matrix after merging the two nearest neighbors. Here, the classical hierarchical clustering algorithm is applied and weights are only used a

Algorithm 4 `Approximation of adaptive k-means`

Input Daily load shapes $\{s_{ij}\}$, initial and final numbers of clusters K_0 and K_f, threshold θ, tolerance ϵ, tolerance for the number of violated clusters ϵ_{N_v}
Output Results of the clustering: clusters $\{Cl_k\}_{k\in\{1,\dots,K\}}$ and centroids $\{C_k\}_{k\in\{1,\dots,K\}}$

$K \leftarrow K_0$
Run K-means algorithm to set $\{Cl_k\}_{k\in\{1,\dots,K\}}$ and $\{C_k\}_{k\in\{1,\dots,K\}}$
$N_v \leftarrow K_f$
while $N_v > \epsilon_{N_v}$ and $K < K_f$ **do**
    $N_v \leftarrow 0$
    for k = 1 to K **do**
        if $\exists s_{ij} \in Cl_k$ s.t. 3.1 is violated **then**
            $N_v \leftarrow N_v + 1$
            Run 2-means algorithm to divide Cl_k into Cl_{k_1} and Cl_{k_2} ; compute C_{k_1} and C_{k_2}
        end if
    end for
    if $\sum\limits_{k,i\in\{1,2\}} d(C_k, C_{k_i}) < \epsilon$ **then** $\triangleright$ d: Distance between centers of two consecutive iterations
        return $\{Cl_k\}_{k\in\{1,\dots,K\}}$ and $\{C_k\}_{k\in\{1,\dots,K\}}$
    end if
    $K \leftarrow K + N_v$
    Update clusters and centroids with results from 2-means clustering
end while
return Clustering results

posteriori to adjust the final calculation of the typical load profiles.

Algorithm 5 `Approximation of hierarchical clustering`

Input Size of the dictionary T, clusters $\{Cl_k\}_{k\in\{1,\dots,K\}}$, sizes of clusters $\{n_k\}_{k\in\{1,\dots,K\}}$ and centroids $\{C_k\}_{k\in\{1,\dots,K\}}$

Output Typical load profiles $\{C_k\}_{k\in\{1,\dots,T\}}$

1: Run hierarchical clustering to create T groups of centroids

2: **return** The barycentres of each group weighted by the sizes $\{n_k\}_{k\in\{1,\dots,K\}}$

3.3 Features extraction and customer segmentation

Once the dictionary is built, it is possible to encode new load shapes by assigning them to the closest TLP. One can also keep track of those that do not satisfy the condition 3.1, to see if some patterns are not correctly captured by the TLP.

Load shapes of customers are classified according to their associated TLP. For each household i, a distribution vector of size T is built, which gives the frequency of each TLP. The variability of a customer's consumption can be expressed by their entropy E, defined by the following formula:

$$E := -\sum_{k\in\{Cl_i\}} p(C_k)\log p(C_k)$$

where $\{Cl_i\}$ is the set of the clusters where the load shapes of i have been distributed, and $p(C)$ is the frequency of the load shape C among all the load shapes of i. The entropy is maximal when a consumer has an equal probability of having any of the TLP, that is to say, if their consumption patterns are very diverse and unpredictable. On the contrary, it is minimal if only one TLP is enough to describe the consumption of the user.

From the dictionary, two types of analysis are conducted:

- On an aggregated level, shape analysis of the prevalent TLP provide information on the general behavior of users such as their main timings of consumption, and highlight potential spatial and seasonal patterns.

- On the household level, each consumer is characterized by the sequence of TLP resulting from the encoding of their DLC and by the sequence of total daily volume of consumption. Features can be extracted from those two representations to segment customers sharing the same behavior.

The analysis of TLP provide criteria for grouping customers. As mentioned, shape analysis of TLP gives an insight on the timings of consumption, whether it be in the morning, during daytime, at night, in the evening, or a combination of those. Users are usually characterized by their peak consumption: the load factors and the degree of utilization measure the efficiency of the electrical usage of a household. Therefore, load shapes are grouped according to the timing of their peaks. To do so, the function *find_peaks* of the library "SciPy" looks for local maxima through comparison of neighboring values.

Kwac et al. [14] cluster households based on the frequency distribution of their TLP. Consumer are represented by a vector of dimension T equal to the size of the dictionary, such that each coordinate gives the frequency of the load profile in the daily load shapes of the customer. Distribution vectors are clustered with the k-means algorithm. If T is too large, it may be necessary to reduce the size of vectors before clustering. A simple way to proceed is to keep the top N load shapes ordered by cluster size. Centroids of low-dimensional clusters are then filtered out.

3.4 Attributes detection: focus on Electric Vehicles

The electricity consumption of the residential sector depends on many factors: type of heating system, use of certain appliances, etc. One of the research subject associated to the deployment of smart meter data is load disaggregation. Detecting the signature of a device is extremely difficult, even with high frequency smart meter data. Hourly measurements are not enough for NILM. In this part, we try to detect the TLP associated to EV using the distribution vectors presented earlier.

3.4.1 Presentation of the data

The method studied in this degree project is used to identify EV owners. Data about households with an EV in the Uppsala region is provided by the Swedish Transport Agency (STA). The test set is made of 1075 non-EV and 176 EV owners (c.f. Tab 3.4.1).

	Non EV (0)	EV (1)	Total
Sample	1075 (86%)	176 (14%)	1251
Train set	798 (85%)	140 (15%)	938
Validation set	277 (88%)	36 (12%)	313

Table 3.4.1: Distribution of the two classes in the sets

The consumption data is filtered to keep only the data following the date of purchase of the vehicle. However, there is not any available information about the discontinuation of the use of an EV. Therefore, in case several EV have been purchased, only the data after the date of purchase of the last vehicle are kept.

The evolution of consumption over the year is similar for EV and non-EV owners (see Fig. 3.4.1). The difference in consumption between the two groups is relatively bigger during the summer though (c.f. Tab. 3.4.2 and 3.4.3). The figures should be taken with caution as the data for households without an EV are more evenly distributed between 2016 and 2021 than the data for those with an EV. In fact, a number of confounding factors can explain the differences in consumption between the classes (EV vs non-EV). It may be that EV owners are more likely to have solar panels for instance. As consumption is lower in the summer, notably due to the reduction in heating, EV consumption represents a larger share of the total need of household, even though seasonal factors can also affect the consumption of EV.

Mean	Standard Deviation	25% quantile	Median	75% quantile
66.39	63.54	24.81	51.06	84.74

Table 3.4.2: EV owners' daily consumption (kWh) for the winter season

Mean	Standard Deviation	25% quantile	Median	75% quantile
44.79	46.04	15.91	31.34	56.08

Table 3.4.3: EV owners' daily consumption (kWh) for the summer season

Figure 3.4.2 shows the distribution of daily consumption of EV owners across all

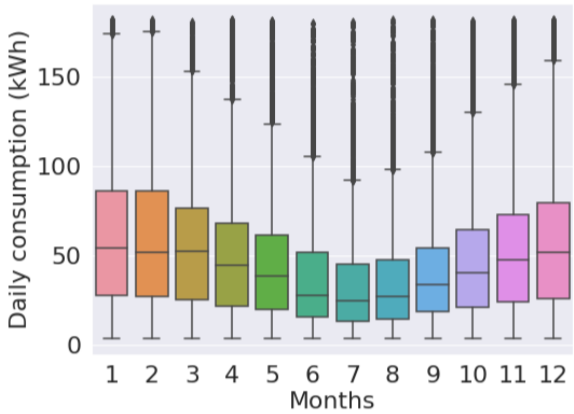


Figure 3.4.1: Distribution of daily consumption of EV owners over the months of the year

seasons. The main differences are the smoother distribution and the presence of some households with very high consumption among EV.

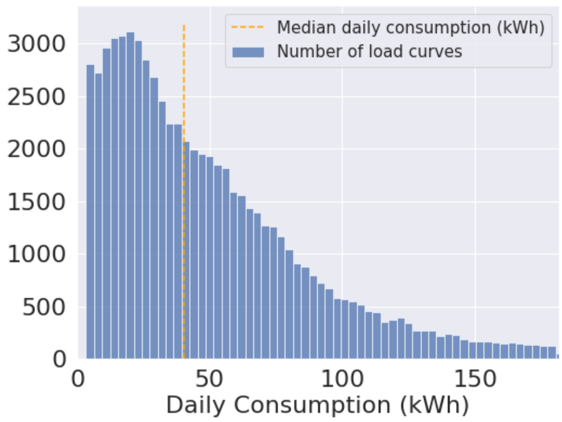


Figure 3.4.2: Distribution of daily consumption of EV owners

3.4.2 Identification of load shapes of Electric Vehicle owners

Applications of the DLC-reduction method include customer segmentation and attribute detection. In this section, we propose a method to identify the most helpful TLP to distinguish classes of customers such as EV owners versus non-EV owners. Four classification models implemented with the Python library "scikit-learn" are compared: Random Forest (RF), Weighted Random Forest (WRF), Balanced Random Forest (BRF) and AdaBoost. The classification models take probability distribution vectors as input and return a label: 0 for non-EV and 1 for EV.

Feature importance is used to select the most important TLP for the detection of EV owners. Impurity-based and permutation-based feature importances are compared.

Permutation importance is evaluated on the validation dataset by randomly shuffling the values of each feature, one after the other. The aim, beyond testing the relevance of distribution vectors for the recognition of certain groups, is to detect the TLP associated with a charging vehicle. The frequency of occurrence of the selected TLP are compared for households with and without an EV. The most characteristic patterns of EV owners are derived.

A frequent problem in the detection of specific attributes is the imbalance between the different classes. The data set of EV contains only about 200 users whose relevant consumption is recorded over a few years. With the Python library "scikit-learn", it is possible to re-balance the classes (see section 2.5). WRF uses the same model as RF with *class_weight = balanced_subsample*.

To evaluate the different models, several scores are compared:

- Out of bag evaluation is a score computed on the training set, using an out-of-bag estimate.
- The validation score is the mean accuracy on the validation set.
- Precision measures the ability of not labeling as positive a sample that is not.
- Recall is the ability to find all the positive instances.
- The F1-score is the harmonic mean of precision and recall.

3.5 Evaluation of the methods

3.5.1 Evaluation of the Typical Load Profiles

The quality of the dictionary is evaluated based on:

1. An intrinseque evaluation, looking at the correlation between pairs of TLP.
2. An evaluation of the encoding of data in the test set, made of the consumption records of 1,000 smart meters.

The Pearson correlation coefficient between the TLP x and y measures their linear correlation and is given by:

$$r_{x,y} = \frac{\sum_{t\in\{0,\dots,23\}}(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t\in\{0,\dots,23\}}(x_t - \bar{x})^2}\sqrt{\sum_{t\in\{0,\dots,23\}}(y_t - \bar{y})^2}}$$

where $\bar{x}$ (resp. $\bar{y}$) is the average value over the day of TLP x (resp. y).

For each load shape $s = [s_0, ..., s_{23}]$ and their associated TLP $C = [C_0, ..., C_{23}]$, the estimated threshold $\hat{\theta}$ and Residual Standard Deviation (RSD) $\sigma(s - C)$ are defined as:

$$\hat{\theta} = \frac{\|s - C\|_2^2}{\|C\|_2^2}$$

$$\sigma(s - C) = \sqrt{\frac{1}{24}\sum_{i\in\{1,\dots,24\}}(s_i - C_i)^2}$$

These two indicators allow a quick comparison with the results of article [14]. The estimated threshold is useful in case algorithm 2 is used. The RSD is a measure of the error between DLC and TLP.

3.5.2 Comparison of behaviors between two groups

The dictionary is used to conduct a temporal analysis of consumption behaviors. In chapter 4, consumption patterns during weekdays and weekends are compared. The significance of the difference in frequency of a TLP between weekdays and weekends is evaluated with a two proportion z-test. It is used to test a difference between the frequencies in two samples of data. The null hypothesis H_0 is that the frequency of the TLP is equal during weekdays and weekends. The alternative hypotheses are:

- H_1: the frequencies are not equal.
- H_2: the frequency of the TLP is lower during weekdays than weekends.
- H_3: the frequency of the TLP is lower during weekends than weekdays.

The statistic z is given by the following formula:

$$z = \frac{p_1 - p_2}{\sqrt{p(1-p)(1/n_1 + 1/n_2)}}$$

where p_1 and p_2 are the frequencies during weekdays and weekends, n_1 and n_2 are the sample sizes, and p is the total frequency:

$$p = \frac{p_1 n_1 + p_2 n_2}{n_1 + n_2}$$

If the p-value is less than the significance level α, then the null hypothesis is rejected (i.e. the frequency of the TLP is not the same during weekdays and weekends). Common choices for α are 0.10, 0.05, and 0.01.

Chapter 4

Results

Chapter 4 presents and discuss the results of the thesis. First, the quality of the dictionary of TLP is evaluated in terms of coherence and coverage. Then, the TLP are analyzed to characterize the general consumption patterns of residential households of the dataset. We look at the results of load shapes encoding at household level: a visual analysis of the clustering results is performed. After that, several features are proposed to group customers with similar energy behavior. Among others, the method is tested on the task of classifying EV/non-EV for the identification of relevant TLP. Eventually, the interest of the method for DSM applications is discussed.

4.1 Analysis of the dictionary of Typical Load Profiles

The set of TLP is built on a sample of 1,000 smart meters or 1 million DLC. In the first place, we tried to use the algorithms proposed in [14], but running them several times to find the right parameters was too time-consuming. Therefore, we explored two approaches using the approximation algorithms introduced in the previous chapter.

4.1.1 Parameters selection and comparison of methods

Two methods were tested and compared to build the dictionary of TLP:

- Use of algorithm 2 which provides a set of 10,000 load shapes then reduction of the size of the dictionary using algorithm 5, an adaptation of hierarchical clustering. An optional reduction step can be added by selecting top shapes.

- Use of the k-means algorithm in the first step with $k = 500$, applied to daily load shapes. The size of the dictionary obtained is reduced using the hierarchical clustering algorithm with $T = 200$.

The choice of the parameters is justified afterwards.

First method: adaptive k-means and hierarchical clustering, top shapes

To determine the value of the parameter θ, we look at the evolution of the number of clusters according to the threshold θ. Depending on apriori knowledge of the dataset, it is possible to impose a given threshold but the goal here is to find a balance between the precision of the clustering and the number of clusters.

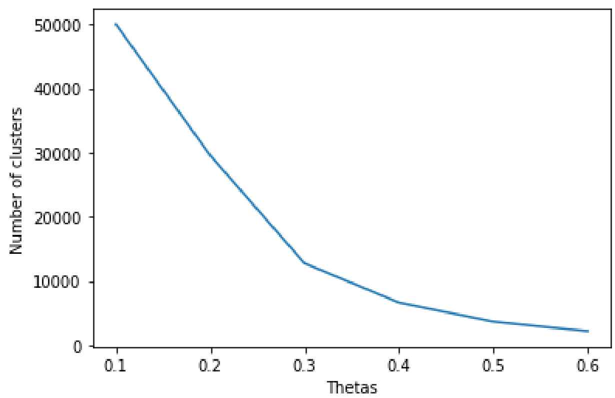


Figure 4.1.1: Number of clusters given θ

The smaller the θ, the closer the load shapes in a cluster are to the center, but the greater the number of clusters. We want to reduce the dimension of the dataset to efficiently characterize customers' consumption. The consumption profiles are variable and, while the aim is to capture the diversity of profiles, the TLP should still generalize well to other households. Therefore, the inflection point at 0.3 is chosen for θ. The execution time of this step is 1,25 hours and 11,949 clusters are obtained. When the algorithm is interrupted, 10 clusters still do not meet the condition 3.1. The reason is that convergence can be unnecessarily slow when N_v is close to 0, so we prematurely stop the algorithm to prevent useless oscillations.

Figure 4.1.2 shows how the number of clusters that do not respect the condition evolves during the successive iterations: the first iterations of the algorithm lead to an exponential growth in the number of clusters, because they all contain at least one load violating 3.1. This highlights the sensitivity of the method to initialization and the presence of outliers. An outlier may force a cluster to divide without improving the clustering results. The initialization of the approximation algorithm is essential because k-means is not executed on the whole data set when K changes. Clusters are divided to improve intra-clustering similarity but the method disregards inter-cluster dissimilarity. With more computing resources, it would be interesting to compare the evolution of the number of violated clusters with different initializations. Indeed, the faster the convergence and the better the clustering, since the relative importance of the approximation part of dividing clusters is reduced compared to the initialization. The k-means algorithm is not deterministic either, but several initializations are compared and the best is selected, looking at the inertia of the initial clustering. In our case, out of 400 clusters, only 4 respect condition 4.1.2. The number of violated cluster reaches a peak at 2,500 when $K = 5,500$ and then starts decreasing until almost all clusters respect the constraint.

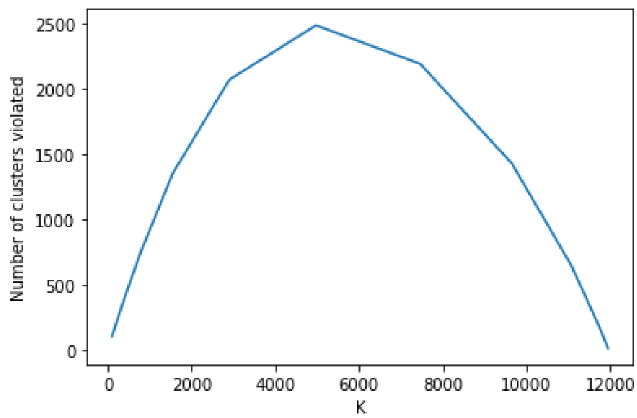
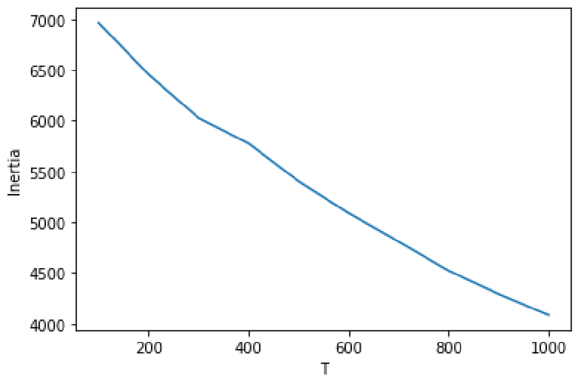


Figure 4.1.2: Number of violated clusters with respect to K

On top of figure 4.1.2, the evolution of the ratio of the number of violated clusters over the total number of clusters would be a good indicator of the speed of the algorithm. Certainly, this method ensures a certain proximity of the loads within a cluster but it does not guarantee that the loads are associated with the closest cluster. Nevertheless, given the large amount of data and the size of the resulting set of TLP, one can hope that the impact of this mismatch is not too great. The task is to create a first set of diverse TLP and not to directly cluster loads. In practice, this step is mainly used to generate varied load shapes, which can then be gathered into more meaningful representations with the second algorithm. Thus, it may be sufficient to use an approximation algorithm for this part.

The complexity due to the computation of the pair-wise distances does not allow us to use the hierarchical clustering algorithm on the 11,949 centroids. Instead, we use the approximation provided by the algorithm 5 to reduce the size of the dictionary. To choose the final size T, we plot the evolution of the inertia as a function of the number of clusters (c.f. Fig 4.1.3). The linear relationship between the inertia and the number of clusters does not help decide on T. Instead, we look at the percentage of loads that violate the condition of proximity to the centre. From figure 4.1.3, $T = 600$ appears to be a good candidate.



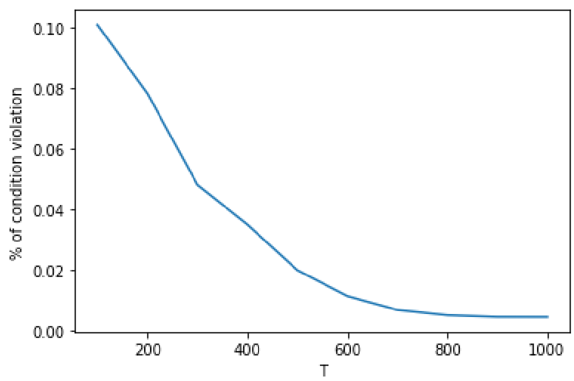


Figure 4.1.3: Evolution of the total inertia of clusters of load shapes of the test set using the successive centroids as TLP and the number of loads not respecting the θ-condition with T

However, the number of load profiles is still very large. The quality of the dictionary is evaluated against another sample of data, also composed of the load shapes of 1000 households over the period 2016-2021. Each load is associated with the closest TLP. The Cumulative Distribution Function (CDF) in figure 4.1.4 shows that 130 shapes cover more than 80% of the data. Only the top 130 shapes are kept.

Second method: k-means and hierarchical clustering

The previous method indicates that a set of about a hundred load shapes is enough to characterize a majority of the consumption profiles of the dataset. If load shapes are reduced to the timing of peak consumption and the focus is on profiles with one or two peaks, which constitute a large part of the observed profiles, then a dictionary of $24 \times 24 = 576$ load shapes would be needed to encapsulate all the potential combinations. Of course, the peaks are not necessarily concentrated in one hour but

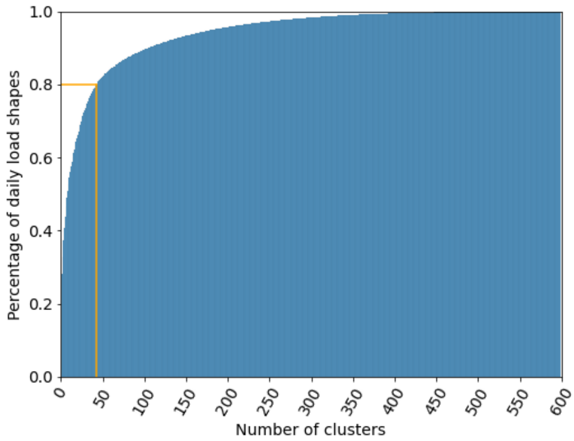


Figure 4.1.4: Coverage of the data

may be spread over a longer period, and some combinations are unlikely. Nevertheless, this gives the intuition that with 500 TLP, it is possible to cover a large diversity of consumption behaviors.

Moreover, hierarchical clustering can only be performed on a relatively small set of centroids. Thus, a maximum of 500 clusters at the end of the first step is set. Two approaches are compared: the traditional k-means algorithm with $k = 500$ and the adapted algorithm where the parameter θ is set to 0.3 but where the algorithm is interrupted when the number of clusters exceeds 500. The top shapes and the inertia obtained in both cases are close but the adaptive method is two times faster.

4.1.2 Analysis of the dictionary

The TLP were evaluated against several criteria, including:

- Clustering metrics such as inertia.
- The quality of the encoding of test data.

The results of the different methods are quite similar. This is not surprising since the general framework of the methodology is the same: first a reduction with a form of k-means algorithm that produces a large set of TLP and then a clustering of the most similar shapes. Thus, the interest of the method presented in [14] lies in the conjunction of the two reduction steps to obtain a dictionary of reasonable size covering a varied panel of consumption without creating too specific shapes. About a hundred TLP may seem excessive. For comparison, we have also built a dictionary of 50 typical shapes. This dictionary is large enough to create groups of consumers with similar behaviors. However, it performs poorly for attribute detection, let alone load disaggregation, the goal of which is to identify the specific signature of a device. Creating a sufficiently large dictionary allows for the detection of specific usages. A final dictionary with between 100 and 200 load shapes is a good compromise for the different use cases. The results of the second method are presented, as it does not need to filter out marginal shapes unlike method 1 where only top shapes are retained.

Figure 4.1.5 shows the distributions of estimated threshold $\hat{\theta}$ and RSD $\sigma(s - C)$. A clear majority of the load shapes respect the threshold condition ($\theta = 0.3$). The RSD measures the dispersion of the differences between the 24 daily measurements of the the load shapes and the TLP. Most values are below 0.03.

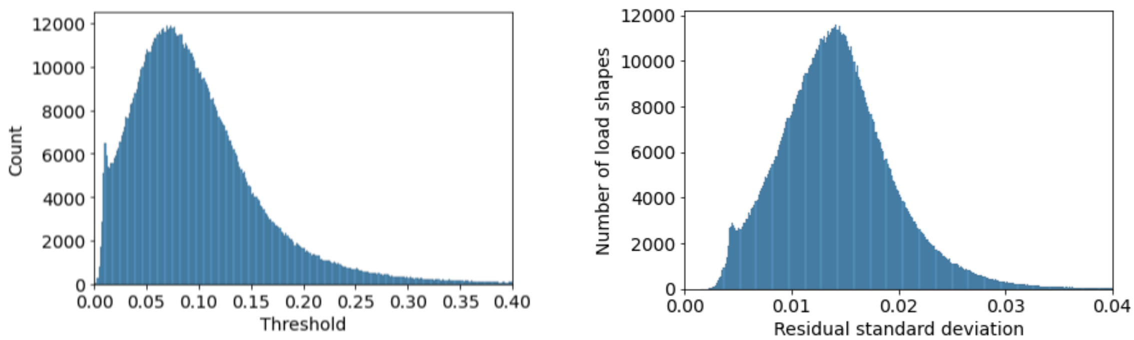


Figure 4.1.5: Distributions of estimated threshold $\hat{\theta}$ and residual standard deviation

Figure 4.1.6 shows the distribution of the number of TLP per household. This indicator should be taken with caution because it does not reflect the frequency with which a household's consumption follows a given pattern. In parallel, we give the number of households that a cluster represents, and the same remark applies. Some consumption patterns are common to all households, while others are more specific. Similarly, some households have very consistent profiles with a few TLP, while others have very anarchic behaviors. Households with few load shapes may be more predictable and easier to target for specific DSM programs, while those that exhibit more fluctuations in their behavior may be more flexible.

Only 20 clusters have more than 10,000 load shapes (0.97% of the data). The largest cluster has 28.3% of the data (291,814 load shapes). The number of clusters necessary to cover a given percentage of the data is shown in figure 4.1.7: 43 clusters are enough to cover 80% of the load shapes.

Figure 4.1.8 shows the distribution of the Pearson correlation coefficients between the

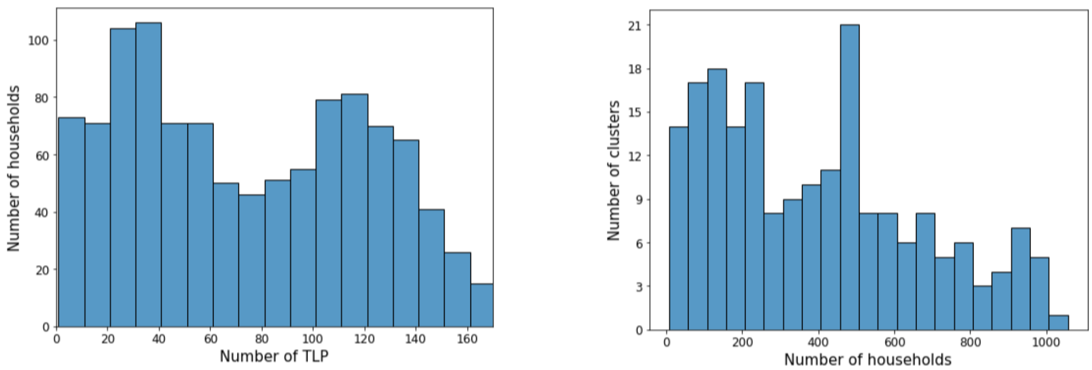


Figure 4.1.6: Distributions of the number of TLP per household and the number of households per cluster

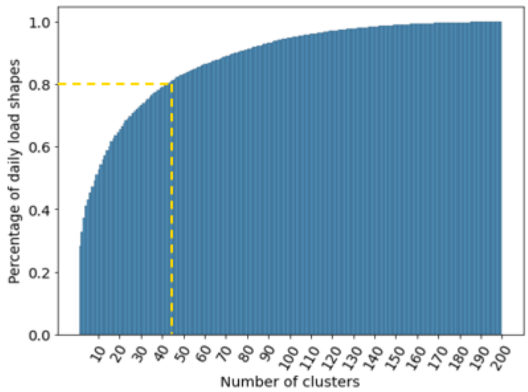


Figure 4.1.7: Coverage of the data (CDF)

pairs of TLP. 13% of the pairs of load shapes have a correlation higher than 0.50. One way to further reduce the size of the dictionary would be to filter on highly correlated pairs to keep only one of the load shapes. However, the analysis of correlated shapes show that even though peaks occur at the same time, overall densities may be different (c.f. Fig. 4.1.9), as the peak intensity is different. Therefore, correlation is not used to filter load shapes, especially since the agglomerative clustering step is already based on a similar approach. Still, it gives insight about simultaneity of peaks and explains the bias towards positive coefficients, as consumption profiles are not symmetrical in the morning and in the evening.

Conversely, negatively correlated load shapes correspond to loads with opposite peak timings, as can be seen in figures 4.1.10 and 4.1.11. Shape 0 has a peak consumption around 14:00, whereas the peak for shape 1 goes from 02:00 to 04:00. Shape 2 corresponds to daytime consumption, while shape 3 is a combination of evening and night consumption.

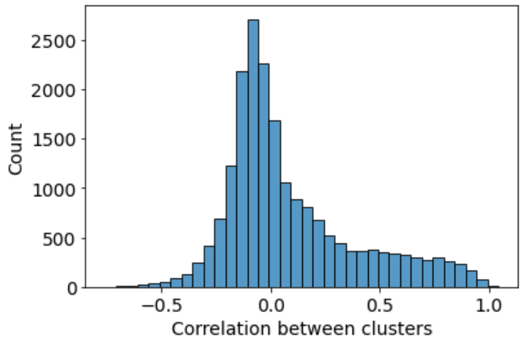


Figure 4.1.8: Distributions of correlation coefficients between TLP

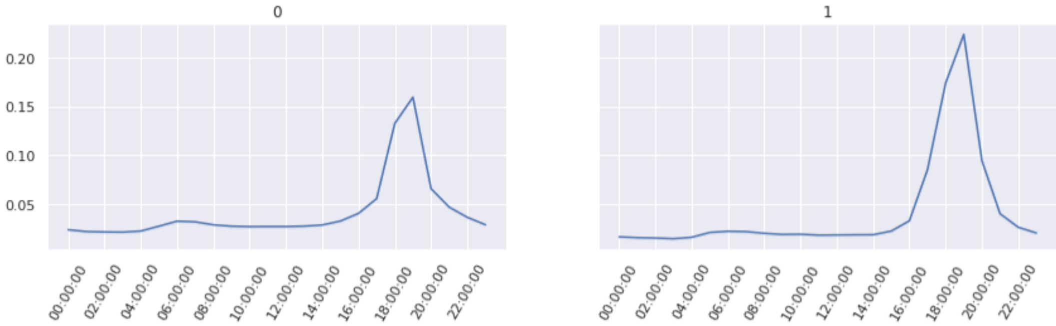


Figure 4.1.9: Example of two highly correlated (> 0.99) load shapes

Figure 4.1.10: Example 1: negatively correlated (< -0.75) load shapes

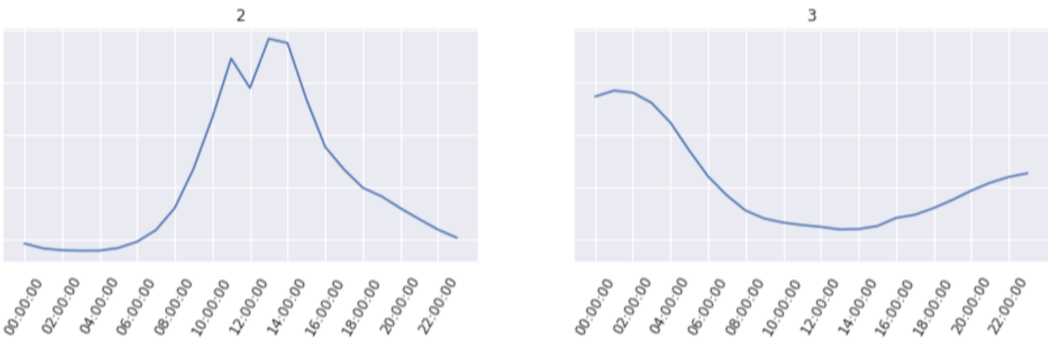


Figure 4.1.11: Example 2: negatively correlated (< -0.75) load shapes

Once the TLP are evaluated, the value of data encoding can be illustrated through different applications:

- Consumption analysis at aggregated level:
- Creation of groups of customers with similar consumption profiles.
- Prediction of EV ownership.

4.2 Shape analysis: study of the Typical Load Profiles

Label	7	1	33	9	55	36	95	20
Size	291814	46621	46104	37885	22601	21308	20947	19404
Frequency	0.283	0.045	0.045	0.037	0.022	0.021	0.021	0.019

Label	31	166	101	17	15	68	14	45	34
Size	18134	17734	17235	15554	15390	15104	14611	13930	11547
Frequency	0.018	0.017	0.017	0.015	0.015	0.015	0.014	0.014	0.011

Table 4.2.1: Size and frequency of the clusters of top TLP
Label 7 corresponds to shape 1, label 1 to shape 2, ... and label 45 to shape 16.

Figure 4.2.1 shows the 16 main load shapes. A load shape can be seen as a probability density. The most frequent shape (shape 0) allows to easily verify that the integral is equal to 1 (the daily average is about 0.04, that is to say over 24 hours a total of 0.96). The shape corresponding to a double peak is found several times in the morning and late afternoon, with different times for the main peak (shapes 7, 8, 11, 12). Shape 0 represents 28% of the loads (see Tab. 4.2.1). It can correspond to a vacant dwelling, but not only since the median consumption associated to this shape is approximately 50 kWh (see Fig. 4.2.2). Other top shapes account for 1-5% of the loads of the test set. The second most important shape represents a late afternoon consumption and the third a morning consumption. Both TLP 5 and 13 show a morning peak, slightly delayed in time. The degree of utilization (ratio of maximum to minimum consumption) is better for shape 5 than for shape 13, because the peak is less marked and consumption is more spread out. Shapes 7 and 8 are highly correlated (with a correlation coefficient of 0.66) and are very similar. However, the peak of shape 7 is at 12:00 and the peak of shape 8 is at 13:00. The average consumption associated with shape 8 is lower than that of shape 7.

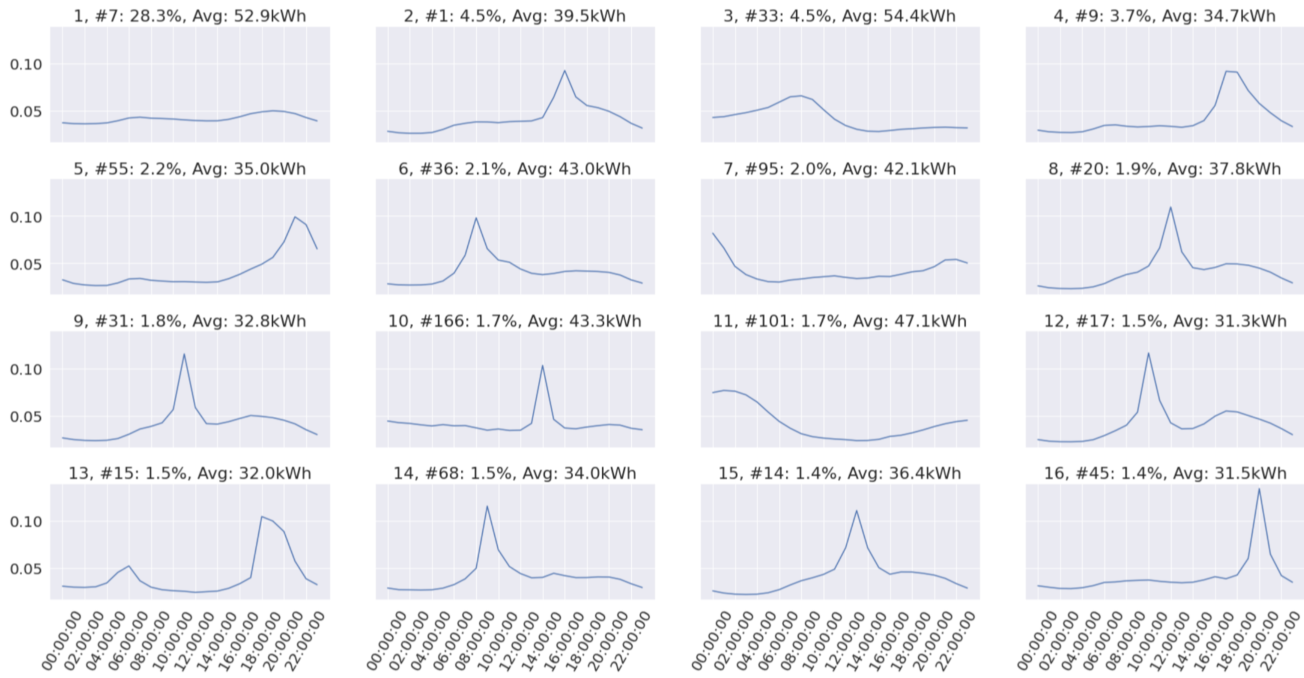


Figure 4.2.1: Top 16 load shapes

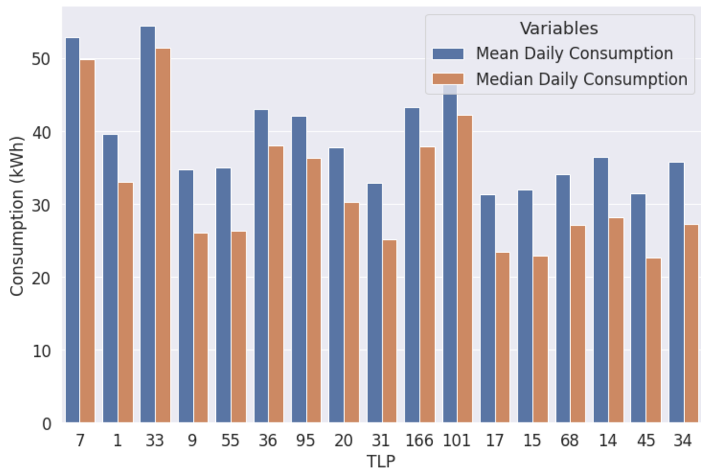


Figure 4.2.2: Volume of daily consumption (kWh) associated to the main load shapes

Figure 4.2.3 shows the peaks found on the top 8 shapes, with a prominence of 0.06 times the amplitude of the signal. Finding peaks on a large data set is not an easy task, but a systematic check could be performed on the final dictionary. Specifying an absolute minimum height can be useful to avoid detecting small fluctuations (c.f. label 0). On the contrary, if the goal is to know when people consume, rather than to identify appropriate peaks, then comparison with neighboring values is the right approach, but the time window needs to be defined.

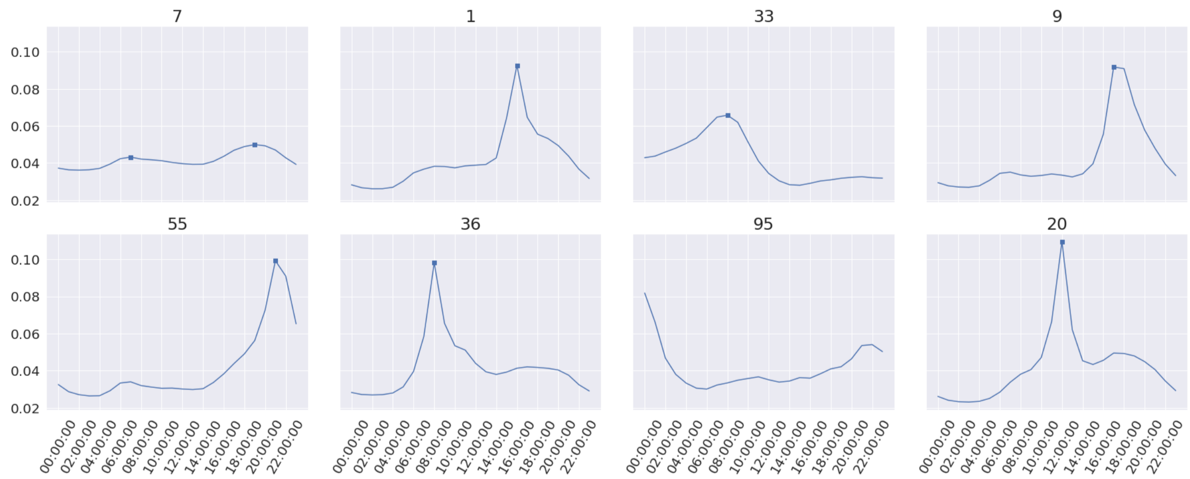


Figure 4.2.3: Identified peaks of the top 8 shapes

With the dictionary, it is possible to perform temporal and spatial analyses and compare the main shapes in different areas or across seasons. If the consumption of a substation can be tracked, it becomes possible to see which patterns are dominant and when they occur. Using weather data, one could see whether decreasing or increasing temperature has an impact on load shape. These are just a few examples of the type of analyses that can be conducted with the set of TLP. All the data in our data set are collected in a small area, so the focus is on temporal analysis.

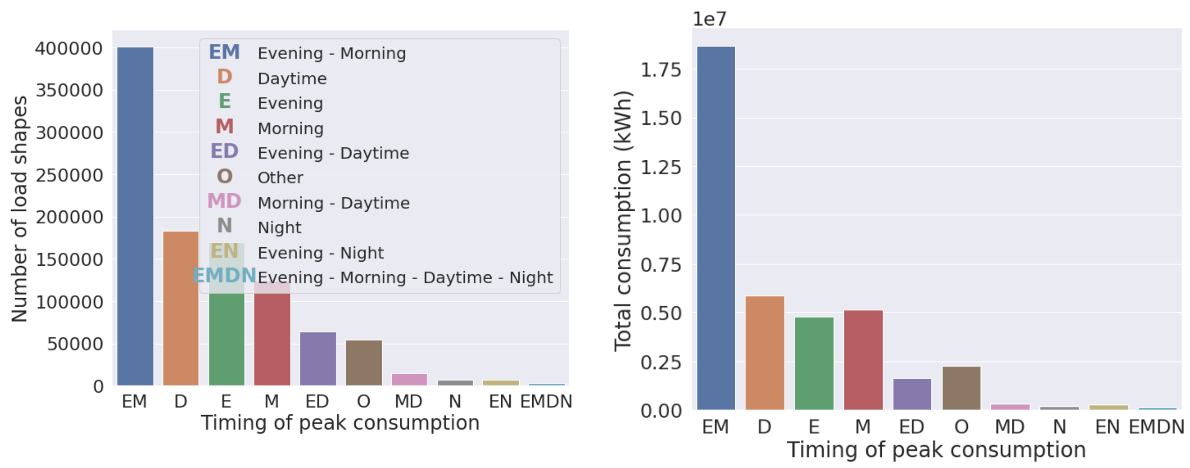


Figure 4.2.4: Timings of peaks (number of shapes and volume of consumption in kWh)

Figures 4.2.4 shows the results of the shape analysis conducted on the whole test set. Load shapes were divided into different categories based on the timing of consumption: morning (06:00 - 11:00), daytime (11:00 - 17:00), evening (17:00 - 22:00), night (00:00 - 06:00 and 22:00 - 00:00) and a combination of those. The most classic shape, which also accounts for the highest total volume of consumption, is by far a combination of evening and morning consumption. This is an expected result, as households are usually not at home during the day on weekdays, that is, 5 days a week.

4.2.1 Temporal analysis: comparison of weekdays and weekends

In this section, the result of comparing weekday and weekend consumption is presented. To continue the above analysis, figure 4.2.5 shows that a larger proportion of loads are classified as daytime consumption during the weekends than during weekdays. The same is true for the combination of daytime and evening consumption.

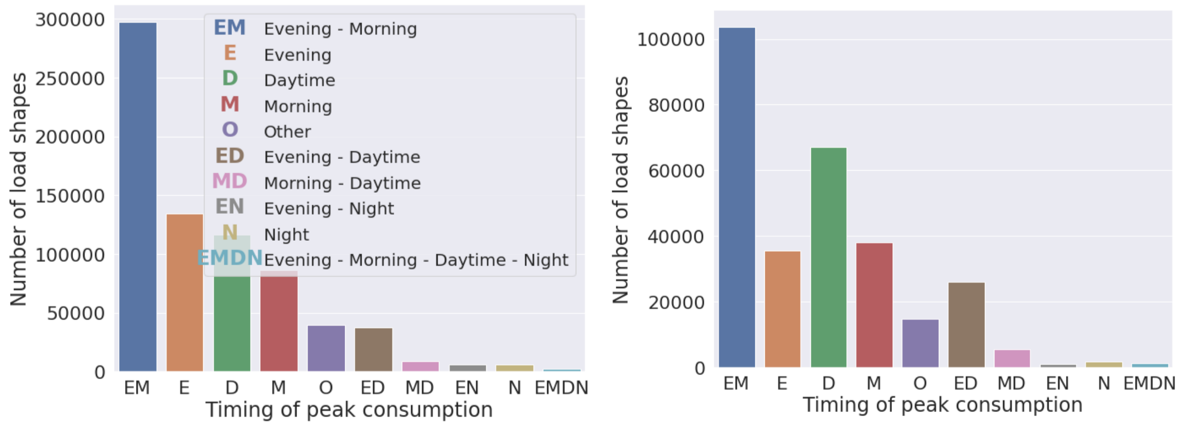


Figure 4.2.5: Timings of peaks during weekdays (left) and weekends (right)

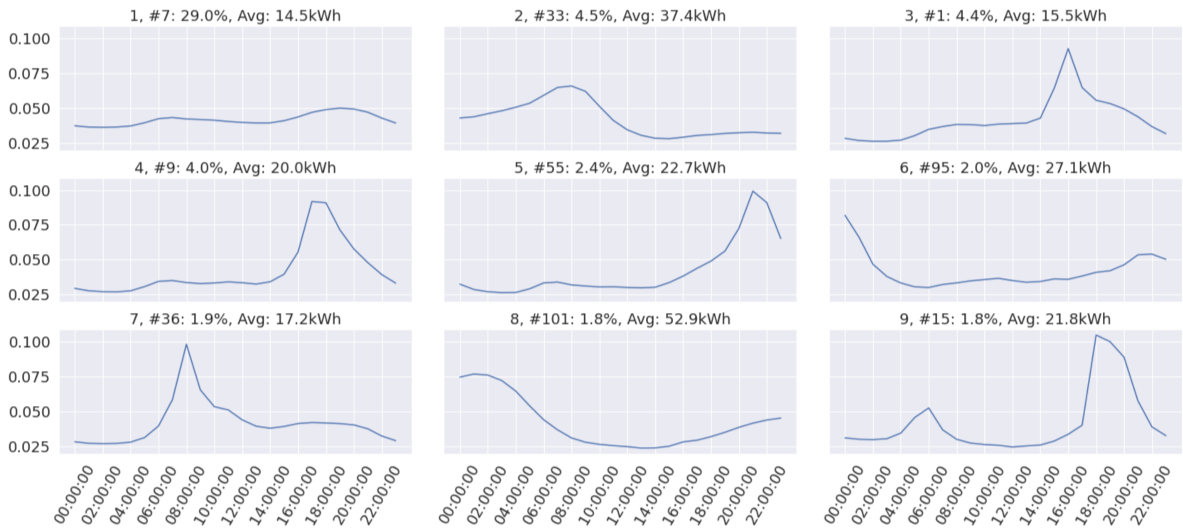


Figure 4.2.6: Top 9 TLP during weekdays

Most of the main shapes are common to both weekdays and weekends (see Fig. 4.2.6 and 4.2.7), but their frequency may be different (Fig. 4.2.10). The vacant housing shape

Figure 4.2.7: Top 9 TLP during weekends

is the dominant shape on both weekdays and weekends. In contrast to weekdays, the main shapes on weekends do not show a drop in consumption like shapes 5 and 7 of the weekdays. Load shape 3 in the weekend with a peak in the middle of the day is not among the top shapes during weekdays and conversely, load shape 4 in the weekday, with a flat consumption during the day and a peak in the evening is really typical of working days.

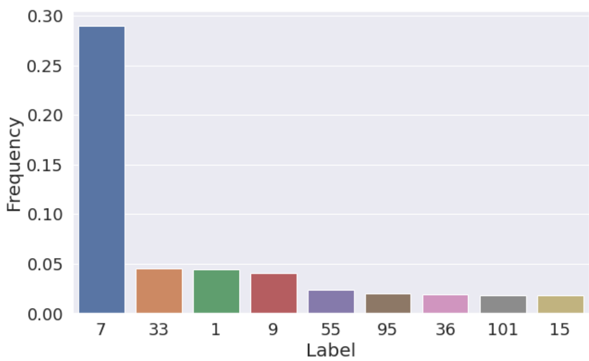


Figure 4.2.8: Weekdays

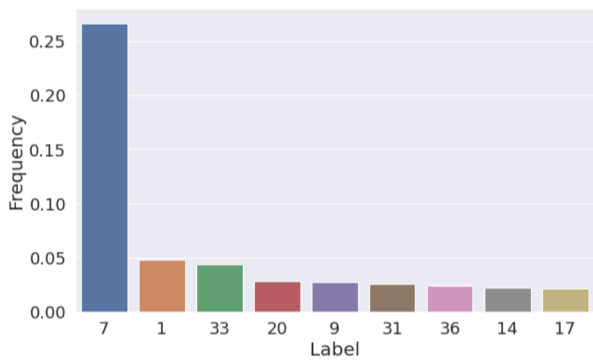


Figure 4.2.9: Weekends

Figure 4.2.10: Frequency of the top 9 TLP

For a significance level $\alpha = 0.05$, only 13.5% of the TLP have similar frequencies during weekdays and weekends (Tab. 4.2.2). 45.5% of TLP are more frequent on weekends than on weekdays, showing that consumption patterns during the weekends are diverse. Of the top 20 weekday shapes and the top 20 weekend shapes, only one has a similar proportion in both groups.

	Number of TLP
$P(C_i \mid weekdays) = P(C_i \mid weekends)$	27 (13.5 %)
$P(C_i \mid weekdays) < P(C_i \mid weekends)$	82 (41 %)
$P(C_i \mid weekdays) > P(C_i \mid weekends)$	91 (45.5 %)

Table 4.2.2: Distribution of the results of the z-test with $\alpha = 0.05$

4.3 Customer segmentation

Frequency analyses of the clusters associated with each TLP provide information about the temporal and spatial characteristics of residential consumption in the Stockholm area. In the previous section, the focus was on the main TLP. One of the goals of creating the TLP is to characterize household consumption and better target customers for D/R and EE programs. Therefore, this section examines the distribution of TLP in daily household consumption. It is then possible to deduce the main patterns of customers. A temporal analysis can also be conducted at household level to see how a specific user consumes during vacations, weekends or at different times of the day or year.

4.3.1 Encoding of the energy consumption of customers

The encoding of the consumption data, associated with the TLP, allows to significantly reduce the size of the data set. Figure 4.3.1 shows the top 16 load shapes of a randomly selected customer. The first group looks noisy but it corresponds to the top profile over all households, which is flat (c.f. Fig. 4.3.2). For comparative purposes, the TLP associated with the 4 main shapes are shown in figure 4.3.2. The TLP labeled according to the results of the previous shape analysis provide some insight on the type of consumption profile of the customer: the first load shapes show a more important consumption in the evening. By describing each TLP precisely (timing of peak consumption, intensity of peaks, frequency of occurrence for all users, etc.), customer behaviors could be segmented on different criteria.

Figure 4.3.1: Top 16 load shapes of a randomly chosen household

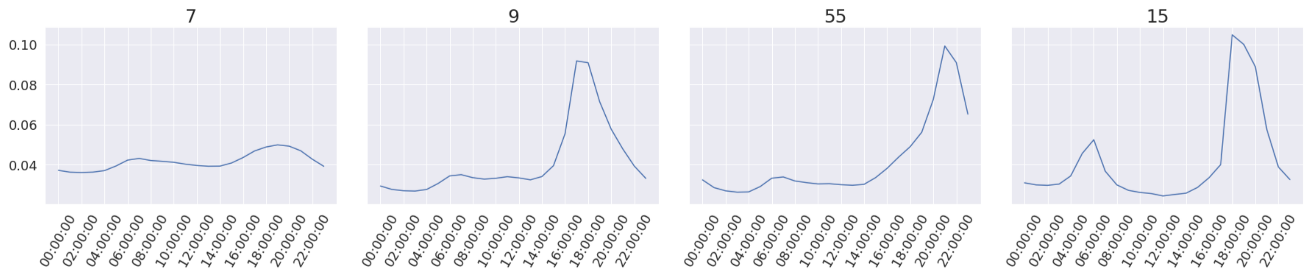


Figure 4.3.2: Typical load profiles associated to the first row of Fig. 4.3.1

4.3.2 Segmentation based on entropy and volume of consumption

This section focuses on a method to automatically group consumers according to the distribution of their TLP. Figure 4.3.3 shows the distribution of the entropy of all the households in the test set. At aggregated level, the daily distribution does not differ significantly from the overall distribution (see Fig. 4.3.4).

The purpose of the distribution vectors is to group customers with similar consumption patterns. The k-means algorithm is used for clustering with $k = 4$ chosen thanks to the built-in score (c.f. Fig. 4.3.5).

Figure 4.3.6 shows two scatter plots corresponding to summer and winter of household normalized entropy on the x-axis and normalized consumption volume on the y-axis. The households are colored by group (see Fig. 4.3.6). This confirms that customers

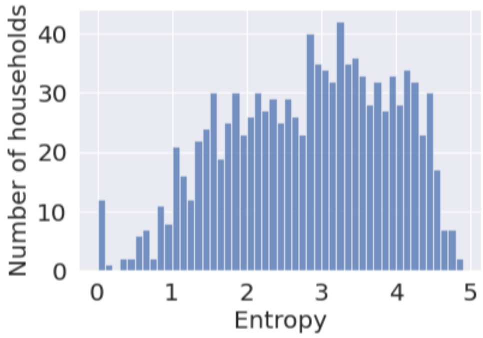


Figure 4.3.3: Distribution of entropy

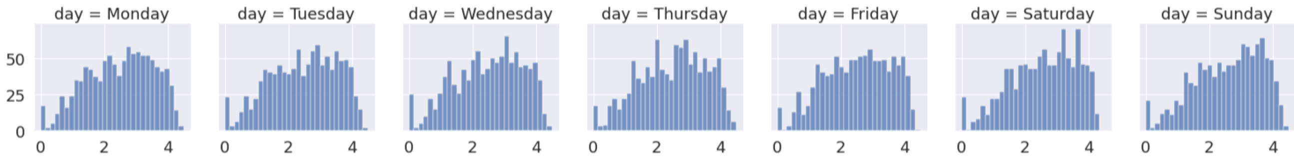


Figure 4.3.4: Distribution of entropy for each day

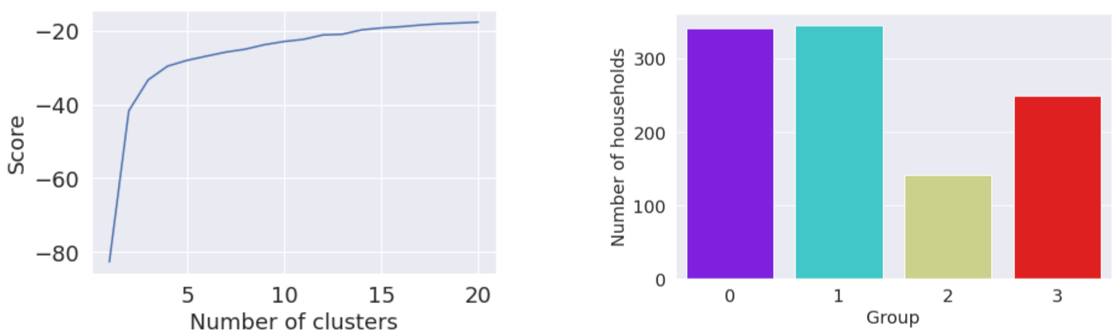


Figure 4.3.5: k-means score and sizes of the groups

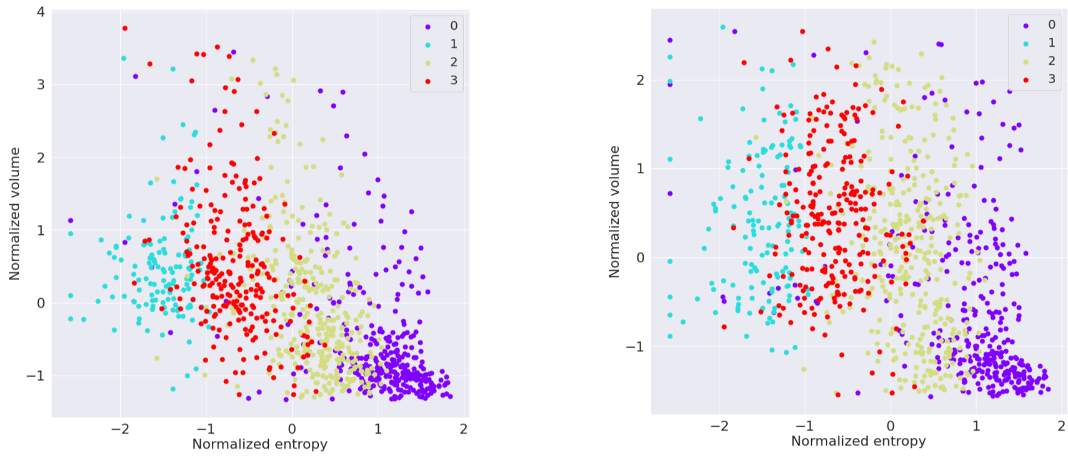


Figure 4.3.6: Scatter plot of consumers' normalized entropy vs normalized volume of consumption in the summer (left) and the winter (right)

are clustered based on their entropy only. However, the graph also shows that during the summer, the households with the most volatile consumption are also those with the lowest total consumption. One hypothesis is that holidays disrupt the usual consumption patterns of households. Distribution vector-based clustering segment households on the variability of their usage, expressed as the frequency of their daily patterns.

4.4 Detection of Electric Vehicles

Table 4.4.1 and figure 4.4.1 show a report of the classification models' performances and their respective confusion matrices. From the comparison of out of bag evaluation and validation score, it is clear that AdaBoost (*estimator* = *DecisionTreeClassifier*, *n_estimators* = 100) overfits during training. In the results, EV is considered as the positive class.

	RF	WRF	BRF	AdaBoost
Out of bag evaluation	0.91	0.90	0.89	0.99
Validation score	0.93	0.93	0.88	0.91
Precision	0.82	1.00	0.49	0.62
Recall	0.50	0.39	0.69	0.58
F1-score	0.62	0.56	0.57	0.60

Table 4.4.1: Evaluation of the different classifiers

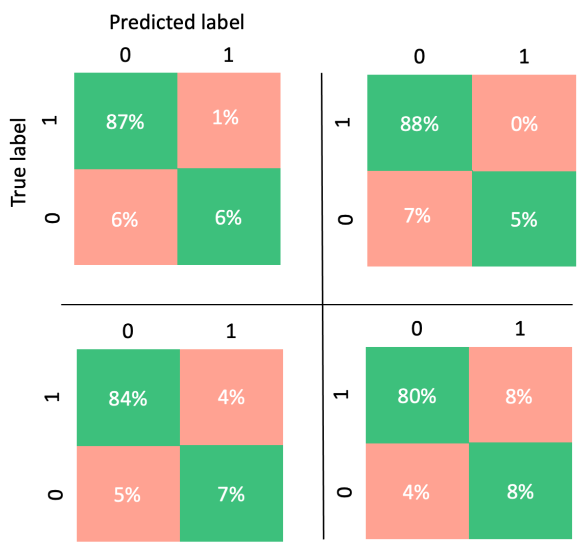


Figure 4.4.1: Confusion matrices (from left to right, top to bottom : RF, WRF, BRF and AdaBoost)

Comparing F1-scores, RF is the best model to identify the features specific to EV. We present the results obtained with this model. Obviously, the classification scores obtained here are very low. This will be discussed in the next section. For the moment, the focus is not so much on the classification task as in the task of recognising the TLP associated with EV owners. Of course, the more reliable the classification, the more relevant the features that are important for the classification.

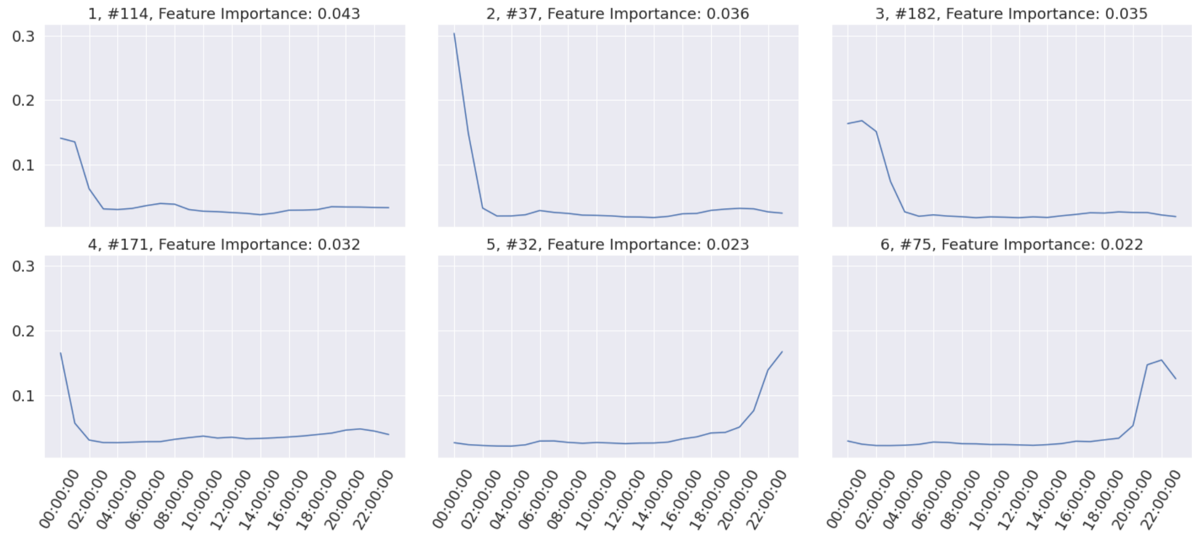


Figure 4.4.2: Top 5% load shapes selected on impurity-based importance to recognize EV with RF

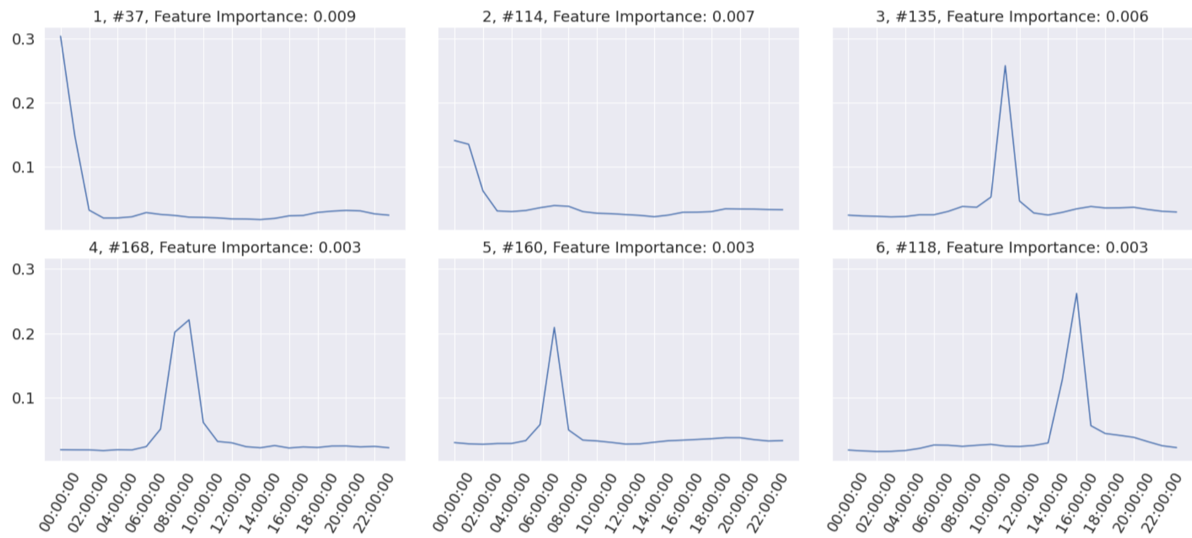


Figure 4.4.3: Top 6 load shapes based on permutation importance

Two ways of evaluating feature importance are tested: impurity-based feature importance and permutation importance. The features obtained are not the same (see Fig. 4.4.2 and 4.4.3), but all of them correspond to a single peak and a flat consumption during the rest of the day. Figure 4.4.2 shows the top 5% features (i.e. load shapes) ordered by impurity-based feature importance, filtered to keep only those which are more frequent among EV owners than among non-EV owners. As expected, most shapes correspond to night peaks, probably when vehicles are charging. Figure 4.4.3 shows the top 6 load shapes (for comparison with the previous method) based on permutation importance, which are more frequent among EV owners. Only single peaks are found. This is also expected as vehicles are not charged several times during the day.

It is not clear which method is more effective in selecting model features, but a combination of the two approaches provides a more comprehensive view of the patterns of interest. More generally, this section presents a method to detect features but does not pretend to do user classification. Again, the only features given to the models are the distributions of TLP and not the TLP themselves, which may allow for refinement of the classification models and better results. Taking into account the volume of consumption would help greatly as well. Besides, the validation set includes few EV owners, which may be detrimental to the quality of the results.

4.5 Discussion of the results

The results include an analysis of the set of TLP created by load clustering, the study of the top consumption patterns in the test set, a comparison of consumption patterns during weekdays and weekends and the detection of shapes that may correspond to the charging of EV. Looking at the results rather than the methodology, it is particularly interesting to review the last application.

As mentioned in the previous section, the classification scores obtained in the EV recognition task are not high. Once again, only the distribution of TLP was taken into account. Nevertheless, the methodology allows us to distinguish load profiles that are those we would expect for EV. Moreover, if the goal is to improve the classification, taking into account the volume of consumption of the users should certainly allow to obtain a higher F1-score. It should also be noted that very little information was known on the use of EV by consumers.

Chapter 5

Discussion

In this chapter, the methods and results are discussed. The method presented in the degree project consists in creating a dictionary of TLP and then encoding the consumption data of the households to obtain a simplified representation of their smart meter data. The segmentation of customers is based on the similarity between the distributions of their TLP. Distribution vectors are also used to detect load shapes characteristic of EV. The discussion covers the preprocessing of the data, the creation of the dictionary of TLP, the segmentation method, the analysis of the TLP and the interpretation of the results obtained.

5.1 Methodology

The data pre-processing step removes values that are extreme in comparison to the dataset. It is also possible to detect outliers from a smart meter by comparing each value with the mean and standard deviation of all consumption records over a moving time window whose size is predefined. However, comparing neighboring values is not effective when the values are missing over a long period of time, which is usually the case. This is the reason why we prefer simply removing global outliers rather than outliers for a given user.

The encoding method is based on the creation of the dictionary of TLP. The approach of reducing the size of the data with a k-means algorithm and then clustering the highly correlated shapes provides good results. The simple k-means used in the project allows to significantly reduce the computing-time compared to the adaptive k-means method used in [14].

The main difficulty in creating the set of TLP is to estimate the number of load profiles needed. The size of the dictionary should be closely related to the final application of the data encoding. If TLP are to be used for load forecasting, a high level of accuracy is important, which justifies a high number of TLP. The method presented in [14] proposes to use adaptive k-means so as not to choose a priori the number of clusters, but the problem is deferred to the choice of the parameter θ whose interpretation is not obvious. This is another reason why we choose to use a simple k-means algorithm.

The TLP are used to encode DLC. Distribution vectors reflecting the importance of each TLP in the consumption habits of a user are created for each consumer, and are used for the segmentation of customers. This approach is interested as it highlights the variability of consumption but it could be improved to reflect the actual distance between load profiles, by taking as input the actual TLP (and not only the distribution vectors) or the distances between them. Assume that load profiles 1 and 2 represent the same behavior slightly offset in time. Household i has consumption similar to load profile 1, and household j has similar consumption although closer to load profile 2. The two consumers have similar energy behaviors but are assigned to different TLP and their distribution vectors do not highlight this similarity. To improve this naive approach, the distance between load profiles should be used in the clustering process, instead of simply taking into account the frequency of each pattern. Kwac and al. use the Earth mover distance, which corresponds to the minimal cost to transform one shape into another [15]. The cost is the normalized usage needed to transform the shape multiplied by the number of hours of the shift. This metric could be used for the construction of the dictionary as well, as a distance for hierarchical clustering when grouping similar shapes together. Then, the similarity measure between shapes needs to be integrated in the method for customer segmentation. Again, the Earth mover distance could be used to express the cost to transform a distribution vector from one household to another, taking into account the distance between the shapes, and their frequency.

5.2 Applications

The approach to customer segmentation should depend on the objective of the program and the type of customer targeted. In this project, variability is used as a criterion for classifying customers. Indeed, we assume that stable consumers have a predictable consumption pattern, which facilitates the identification of their potential for flexibility, whereas more volatile consumers are more likely to change their behavior and are therefore good targets for EE programs. However, other criteria can be used to segment consumers.

The analysis of the top TLP provides information on preferences in consumption at different moments. The method used for peak timing analysis, where peaks are detected automatically, requires the specification of many parameters such as height, width and spacing between peaks. In practice, the shapes of consumption peaks are very different, making it difficult to define appropriate parameters. It is not possible to accurately detect peaks on every DLC, whereas it is possible to manually classify the TLP according to the time when peaks occur. Therefore, profiling improves the reliability of the study of timing of consumption.

The profiling method developed in the project, while offering accurate insight into the behavior of domestic consumers, also provides an efficient way to anonymize data. Smart meter data is indeed classified as personal data and, as such, is protected by the General Data Protection Regulation (GDPR). TLP do not show specific details about individual energy usage, thus protecting the privacy of consumers.

5.3 Results

The results reflect the behavior of one or two residential houses in the Stockholm area over 5 years. Theoretically, the method of grouping and comparing consumers does not require additional information on households. However, this relatively homogeneous group of consumers may cover different realities. In the case of a two-dwelling house, the level of consumption and the variety of profiles is expected to be greater than for a smart meter recording data from only one household. To be more precise, customers could be divided according to their consumption level first and then TLP created for each group. This would improve the accuracy of the TLP and the encoding scheme.

Chapter 6

Conclusion

In this chapter, the conclusions of the degree project and the potential development of the work are presented.

6.1 Conclusive summary

This work focuses on the technical aspects of a method for segmenting electricity consumers according to their consumption habits, in order to facilitate the implementation of D/R programs. In practice, this method relies on the construction of a dictionary of TLP that allows to reduce the size of the consumption data while keeping a sufficient degree of accuracy to obtain a characterization of the variability of the habits of a household, rather than averaging the DLC, and risking to integrate marginal behaviors.

This project takes up and adapts the methodology presented by Kwac et al. in [14] by following the general framework of the method but proposing simplifications to save time on the different steps. The results are consistent with those obtained in the paper on another residential consumption dataset. A shape analysis of the TLP in the dictionary is performed, to illustrate the direct use of profiling, then a segmentation based on the variability of users' consumption is presented, which relies, as proposed in [14], on the encoding of household consumption data, using the dictionary of TLP to associate a DLC with a TLP, and then on the clustering of TLP distributions in the encoded data. This results in four groups of consumers that can be distinguished by the degree of variability in their consumption profiles. The interest of the segmentation from the perspective of D/R programs is discussed. Finally, the method is evaluated with respect to its potential for attribute detection thanks to classification algorithms used to identify features relevant to recognize EV owners.

The conclusion of this work is that the 2-step clustering method allows to create a set of TLP of good quality, in terms of coverage and consistency, whose size has to be adjusted (e.g. selecting the main shapes) according to the final application. The same level of accuracy is not expected for customer segmentation as for individual consumption prediction. The method of size reduction of consumption data allows to adjust the desired degree of accuracy while filtering out marginal profiles, which are not interesting from the perspective of D/R. The encoding of smart meter data allows to generate a detailed report of the energy behavior of a customer, highlighting the top shapes that are characteristic of their consumption and the temporality of these shapes. Moreover, this simplified representation of energy usage patterns makes it easier to compare energy usage across different users. It opens the doors to new applications such as attribute detection and individual forecasting, thanks to the high level of accuracy of the set of TLP. Finally, the method presented in the project offers insights into the sharing of consumption data, classified as sensitive, for research purposes.

6.2 Future Work

The methodology presented in this degree is designed to deal with the variety of residential customer profiles. The next step is to see if it can be applied to other consumer groups (services and industry) and if it is relevant. Small businesses may be good candidates. The interest would be less in modelling the variation over time of the consumption of a particular shop than in modelling the diversity of consumption of different shops.

The methodology opens the door to the modeling of individual consumption, which is increasingly relevant for smart grid planning. The performances of models taking raw data or TLP as input should be compared to evaluate the benefits of profiling. Ref. [21] uses a methodology for intraday load forecasting consisting in clustering of customers based on their energy behavior, forecast of the load of each group and then aggregation of the results of the group forecasts. It shows that error can be reduced, compared to models taking raw data as input. As mentioned in [14], Markov chains could be used for load shape prediction, where the states of the model would be TLP and the observations could include weather or electricity prices.

Tariff optimization is needed to incentivize customers to reduce peak loads that threaten the stability of the grid. To optimize models, the response of households to price fluctuation must be modelled. Thus, a future step could be to classify TLP according to whether the peak consumption occurs during peak or off-peak hours. From this, the preference of users to consume during peak hours can be expressed, as well as the potential for peak shaving, if willingness to change behavior can't be measured. The objective of [8] is to propose an optimal ToU tariff, using principal-agent theory. It is assumed that households seek to maximize their utility, which is not just a function of the price of electricity but also of user flexibility, as nobody wants to have to cook in the middle of the night. The pricing methodology models consumer preferences. In [8], three parameters are used to characterize consumer flexibility: α measures the preference to consume during peak hours, β the willingness or ability to shift consumption to off-peak hours, and λ, the importance placed on electricity relative to other goods of consumption. Better knowledge of the consumers can allow for better estimation of the α and β parameters.

NILM must identify the most discriminative features from the smart meter data. Deep learning techniques are generally capable of automatically extracting important features. Unsupervised solutions minimize the need for data transformation. Yet, feature quality impacts model performance and choosing the right data is critical. Feature selection reduces the complexity of the model, saving computation time and improving generalization ability. Even though most recent models can handle data with huge dimensionality, it can be useful to transform input data, using load profiling for example. As with load forecasting, it would be interesting to study the improvement that load profiling can bring to disaggregation models. However, this application remains the most uncertain, and should be tested on easy-to-identify devices, especially since the current literature on NILM focuses on active power data from smart meters.

6.3 Final Words

This data profiling method offers an interesting technical approach to the problem of the dimensionality of smart meter data, which record data at ever closer time intervals. It is of great interest for the segmentation of consumers according to their consumption habits. Nevertheless, the main way of leveraging user flexibility for DSM purposes is still to offer tariffs aiming at reducing peaks. The optimization of these models is a difficult problem, with an asymmetry of information between the supplier and the households, whose behavior is uncertain. Load profiling allows to quickly capture the type of consumption of the customers and thus to improve the modeling of their behavior. In the context of micro and smart grids, the modeling of individual consumption may become a key issue.

Bibliography

[1] Agency, International Renewable Energy. “Innovation landscape brief: Dynamic line rating”. In: *Abu Dhabi* (2020). DOI: ISBN978-92-9260-182-9.

[2] Agency, International Renewable Energy. “Power System Flexibility for the Energy Transition, Part 1: Overview for policy makers”. In: (2018). DOI: ISBN978-92-9260-089-1.

[3] Aghabozorgi, Saeed, Seyed Shirkhorshidi, Ali, and Ying Wah, Teh. “Time-series clustering – A decade review”. In: *Information Systems* 53 (2015), pp. 16–38. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2015.04.007. URL: https://www.sciencedirect.com/science/article/pii/S0306437915000733.

[4] Arthur, David and Vassilvitskii, Sergei. *k-means++: The Advantages of Careful Seeding*. Technical Report 2006-13. Stanford InfoLab, June 2006. URL: http://ilpubs.stanford.edu:8090/778/.

[5] *Boosting (machine learning)*. https://www.irena.org/Data/Energy-Profiles.

[6] Brusokas, Jonas, Pedersen, Torben, Siksnys, Laurynas, Zhang, Dalin, and Chen, Kaixuan. “HeatFlex: Machine learning based data-driven flexibility prediction for individual heat pumps”. In: June 2021, pp. 160–170. DOI: 10.1145/3447555.3464866.

[7] Chen, Chao and Breiman, Leo. “Using Random Forest to Learn Imbalanced Data”. In: *University of California, Berkeley* (Jan. 2004).

[8] Chen, Sijie, Alan Love, H., and Liu, Chen-Ching. “Optimal Opt-In Residential Time-of-Use
Contract Based on Principal-Agent Theory”. In: *IEEE Transactions on Power Systems* 31.6 (2016), pp. 4415–4426. DOI: 10.1109/TPWRS.2016.2518020.

[9] Deng, Ruilong, Yang, Zaiyue, Chow, Mo-Yuen, and Chen, Jiming. "A Survey on Demand Response in Smart Grids: Mathematical Models and Approaches". In: *IEEE Transactions on Industrial Informatics* 11.3 (2015), pp. 570–582. DOI: 10.1109/TII.2015.2414719.

[10] He, Xian, Keyaerts, Nico, Azevedo, Isabel, Meeus, Leonardo, Hancher, Leigh, and Glachant, Jean-Michel. "How to engage consumers in demand response: A contract perspective". In: *Utilities Policy* 27 (Dec. 2013), pp. 108–122. DOI: 10.1016/j.jup.2013.10.001.

[11] Huang, Yalin, Grahn, Elin, Wallnerström, Carl Johan, Jaakonantti, Lena, and Johansson, Tommy. "Smart meters in Sweden – Lesson learned and new regulations". In: Dec. 2018.

[12] *IRENA Energy Profiles*. https://en.wikipedia.org/wiki/Boosting_%28machine_learning%29. Last updated on: 24th August, 2022.

[13] Kumar Beniwal, Rajender. "Chapter 2 - An introduction to the smart grid-II". In: *Advances in Smart Grid Power System*. Ed. by Anuradha Tomar and Ritu Kandari. Academic Press, 2021, pp. 33–45. ISBN: 978-0-12-824337-4. DOI: https://doi.org/10.1016/B978-0-12-824337-4.00002-3. URL: https://www.sciencedirect.com/science/article/pii/B9780128243374000023.

[14] Kwac, Jungsuk, Flora, June, and Rajagopal, Ram. "Household Energy Consumption Segmentation Using Hourly Data". In: *IEEE Transactions on Smart Grid* 5.1 (2014), pp. 420–430. DOI: 10.1109/TSG.2013.2278477.

[15] Kwac, Jungsuk, Flora, June, and Rajagopal, Ram. "Lifestyle Segmentation Based on Energy Consumption Data". In: *IEEE Transactions on Smart Grid* 9.4 (2018), pp. 2409–2418. DOI: 10.1109/TSG.2016.2611600.

[16] Li, Pei-Hao and Pye, Steve. "Assessing the benefits of demand-side flexibility in residential and transport sectors from an integrated energy systems perspective". In: *Applied Energy* 228 (2018), pp. 965–979. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.06.153. URL: https://www.sciencedirect.com/science/article/pii/S0306261918310237.

[17] Li, Yanlin, Xie, Kaigui, Wang, Lingfeng, and Xiang, Yingmeng. "Exploiting network topology optimization and demand side management to improve bulk power system resilience under windstorms". In: *Electric Power Systems Research* 171 (2019), pp. 127–140. ISSN: 0378-7796. DOI: https://doi.org/

10.1016/j.epsr.2019.02.014. URL: https://www.sciencedirect.com/science/article/pii/S0378779619300707.

[18] Madina, Carlos, Gomez-Arriola, Ines, Santos-Mugica, Maider, Jimeno, Joseba, Kessels, Kris, Trakas, Dimitris N., Chaves, José Pablo, and Ruwaida, Yvonne. "Flexibility markets to procure system services. CoordiNet project". In: *2020 17th International Conference on the European Energy Market (EEM)* (2020), pp. 1–6.

[19] Mancini, Francesco, Romano, Sabrina, Lo Basso, Gianluigi, Cimaglia, Jacopo, and Santoli, Livio de. "How the Italian Residential Sector Could Contribute to Load Flexibility in Demand Response Activities: A Methodology for Residential Clustering and Developing a Flexibility Strategy". In: *Energies* 13.13 (2020). ISSN: 1996-1073. DOI: 10.3390/en13133359. URL: https://www.mdpi.com/1996-1073/13/13/3359.

[20] McKenna, Eoghan, Richardson, Ian, and Thomson, Murray. "Smart meter data: Balancing consumer privacy concerns with legitimate applications". In: *Energy Policy* 41 (2012). Modeling Transport (Energy) Demand and Policies, pp. 807–814. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2011.11.049. URL: https://www.sciencedirect.com/science/article/pii/S0301421511009438.

[21] Quilumba, Franklin L., Lee, Wei-Jen, Huang, Heng, Wang, David Y., and Szabados, Robert L. "Using Smart Meter Data to Improve the Accuracy of Intraday Load Forecasting Considering Customer Behavior Similarities". In: *IEEE Transactions on Smart Grid* 6.2 (2015), pp. 911–918. DOI: 10.1109/TSG.2014.2364233.

[22] Rajabi, Amin, Eskandari, Mohsen, Jabbari Ghadi, Mojtaba, Ghavidel, Sahand, Li,
Li, Zhang, Jiangfeng, and Siano, Pierluigi. "A pattern recognition methodology for analyzing residential customers load data and targeting demand response applications". In: *Energy and Buildings* 203 (2019), p. 109455. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2019.109455. URL: https://www.sciencedirect.com/science/article/pii/S037877881931103X.

[23] Villar, José, Bessa, Ricardo, and Matos, Manuel. "Flexibility products and markets: Literature review". In: *Electric Power Systems Research* 154 (2018),

pp. 329–340. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2017.09.005. URL: https://www.sciencedirect.com/science/article/pii/S0378779617303723.

[24] Wang, Yi, Chen, Qixin, Hong, Tao, and Kang, Chongqing. "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges". In: *CoRR* abs/1802.04117 (2018). arXiv: 1802.04117. URL: http://arxiv.org/abs/1802.04117.

[25] Wang, Yi, Chen, Qixin, and Kang, Chongqing. "Clustering of Consumption Behavior Dynamics". In: *Smart Meter Data Analytics: Electricity Consumer Behavior Modeling, Aggregation, and Forecasting*. Singapore: Springer Singapore, 2020, pp. 225–246. ISBN: 978-981-15-2624-4. DOI: 10.1007/978-981-15-2624-4_10. URL: https://doi.org/10.1007/978-981-15-2624-4_10.

[26] Yan, Da, Jin, Yuan, Sun, Hongsan, Dong, Bing, Ye, Zi, Li, Zhaoxuan, and Yuan, Yanping. "Household appliance recognition through a Bayes classification model". In: *Sustainable Cities and Society* 46 (2019), p. 101393. ISSN: 2210-6707. DOI: https://doi.org/10.1016/j.scs.2018.12.021. URL: https://www.sciencedirect.com/science/article/pii/S221067071830489X.

TRITA-EECS-EX-2023:105