Jul, 2025

GRINS DISCUSSION PAPER SERIES DP N° 30/2025 ISSN 3035-5576

Identifying Electricity Consumption Patterns by Using High-Frequency Data: Volatility and Self-Organizing Maps

DP N° 30/2025

Authors:
F. Atzori, L. Corazzini, M. Guerzoni, M. Mantovani

Jul, 2025

GRINS DISCUSSION PAPER SERIES DP N° 30/2025 ISSN 3035-5576

Identifying Electricity Consumption Patterns by Using High-Frequency Data: Volatility and Self-Organizing Maps

F. Atzori, L. Corazzini, M. Guerzoni, M. Mantovani

ACKNOWLEDGEMENTS

This study was funded by the European Union - NextGenerationEU, in the framework of the GRINS - Growing Resilient, INclusive and Sustainable project (GRINS PE00000018). The views and opinions expressed are solely those of the authors and do not necessarily reflect those of the European Union, nor can the European Union be held responsible for them.

CITE THIS WORK

Title: Identifying Electricity Consumption Patterns by Using High-Frequency Data: Volatility and Self-Organizing Maps. Publication Date: 2025.

In this report, we present the results of a two-level clustering approach that combines self-organizing maps and K-means to identify distinct household electricity consumption patterns. To address the high dimensionality of the data and better capture both short- and long-term volatility in energy usage, we apply a discrete wavelet transformation to the data expressed in standardized first differences. The resulting transformed dataset serves as input for the clustering methodology. Despite the initial homogeneity of the households in the dataset, four distinct clusters emerge, exhibiting marked differences in energy consumption and intra-day volatility. Moreover, our analysis uncovers a strong positive association between volatility and energy usage: households in clusters characterized by greater intra-day variability consistently consume more electricity. The available socio-economic data further enable us to profile households in the most volatile clusters: they are typically residents of detached houses, rely heavily on electricity for water heating, and are subscribed to energy plans offering reduced tariffs during nighttime hours and weekends. These findings underscore the importance of targeting consumption volatility in demand-side management strategies. Addressing short-term fluctuations in electricity usage may be a key lever for improving efficiency and reducing overall demand.

Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

GRINS FOUNDATION

GRINS FOUNDATION

Discussion Paper Series

Identifying Electricity Consumption Patterns by Using High-Frequency Data: Volatility and Self-Organizing Maps

Discussion paper n. 30/2025

F. Atzori, L. Corazzini, M. Guerzoni, M. Mantovani

ISSN 3035-5567

Identifying Electricity Consumption Patterns by Using High-Frequency Data: Volatility and Self-Organizing Maps

F. Atzori, L. Corazzini, M. Guerzoni, M. Mantovani

Discussion Paper DP n° 30/2025

In this report, we present the results of a two-level clustering approach that combines self-organizing maps and K-means to identify distinct household electricity consumption patterns. To address the high dimensionality of the data and better capture both short- and long-term volatility in energy usage, we apply a discrete wavelet transformation to the data expressed in standardized first differences. The resulting transformed dataset serves as input for the clustering methodology. Despite the initial homogeneity of the households in the dataset, four distinct clusters emerge, exhibiting marked differences in energy consumption and intra-day volatility. Moreover, our analysis uncovers a strong positive association between volatility and energy usage: households in clusters characterized by greater intra-day variability consistently consume more electricity. The available socio-economic data further enable us to profile households in the most volatile clusters: they are typically residents of detached houses, rely heavily on electricity for water heating, and are subscribed to energy plans offering reduced tariffs during nighttime hours and weekends. These findings underscore the importance of targeting consumption volatility in demand-side management strategies. Addressing short-term fluctuations in electricity usage may be a key lever for improving efficiency and reducing overall demand.

We thank Affective (Torben Emmerling, Laura Fontanesi, and Daniel Seyffardt) for supporting us during the development of the study. We thank the Swiss electricity provider for sharing ideas and giving us access to the energy consumption dataset. We are in debt with the energy sector research for extracting and preparing the dataset used in the present study. This study was funded by the European Union - NextGenerationEU, in the framework of the GRINS -Growing Resilient, INclusive and Sustainable project (GRINS PE00000018 – CUP C93C22005270001). The views and opinions expressed are solely those of the authors and do not necessarily reflect those of the European Union. The European Union can be held responsible for them.

Identifying Electricity Consumption Patterns by Using High-Frequency Data: Volatility and Self-Organizing Maps[*]

F. Atzori, L. Corazzini, M. Guerzoni, M. Mantovani

Policy Summary

In this report, we present the results of a two-level clustering approach that combines self-organizing maps and K-means to identify distinct household electricity consumption patterns. To address the high dimensionality of the data and better capture both short- and long-term volatility in energy usage, we apply a discrete wavelet transformation to the data expressed in standardized first differences. The resulting transformed dataset serves as input for the clustering methodology.

Despite the initial homogeneity of the households in the dataset, four distinct clusters emerge, exhibiting marked differences in energy consumption and intra-day volatility. Moreover, our analysis uncovers a strong positive association between volatility and energy usage: households in clusters characterized by greater intra-day variability consistently consume more electricity. The available socio-economic data further enable us to profile households in the most volatile clusters: they are typically residents of detached houses, rely heavily on electricity for water heating, and are subscribed to energy plans offering reduced tariffs during nighttime hours and weekends.

These findings underscore the importance of targeting consumption volatility in demand-side management strategies. Addressing short-term fluctuations in electricity usage may be a key lever for improving efficiency and reducing overall demand.

1 Introduction

This report documents our analysis of electricity consumption patterns in a large dataset of households in Switzerland. The objective is to identify meaningful groups of consumers based solely on their electricity consumption behavior, using high-frequency smart meter data. This represents the first step in a broader project aimed at improving energy efficiency among final consumers. The insights from the segmentation exercise are intended as inputs for randomized controlled trials (RCTs) designed to test personalized interventions to reduce energy demand.

Electricity consumption is known to exhibit both regularities and substantial heterogeneity across households (e.g. Jones et al., 2015). Certain patterns tend to recur within specific groups of consumers; for instance, families with children often show pronounced evening peaks, while single-person households or remote workers may display flatter or midday-shifted usage profiles. Other group-level differences arise from structural factors such as dwelling size, heating systems, and appliance ownership, which influence baseline consumption levels and usage timing. At the same time, broader differences—such as in responsiveness to weather, seasonality, or price changes—highlight the importance of behavioral variation. Capturing this complexity is essential for understanding energy demand and designing targeted interventions that account for the diverse ways in which households consume electricity.

A wide range of interventions has been tested to reduce residential electricity demand, from economic incentives such as time-of-use pricing and rebates, to behavioral strategies like feedback, social comparisons, and personalized messaging. One of the most well-known examples is the Opower program in the United States, which delivered home energy reports comparing household usage to that of similar neighbors, leading to persistent reductions in electricity consumption of around 1.5–3% on average Allcott (2011). In Europe, similar approaches have gained traction.

[*]We thank Affective (Torben Emmerling, Laura Fontanesi, and Daniel Seyffardt) for supporting us during the development of the study. We thank the Swiss electricity provider for sharing ideas and giving us access to the energy consumption dataset. We are in debt with the energy sector research for extracting and preparing the dataset used in the present study. This study was funded by the European Union - NextGenerationEU, in the framework of the GRINS -Growing Resilient, INclusive and Sustainable project (GRINS PE00000018 – CUP C93C22005270001). The views and opinions expressed are solely those of the authors and do not necessarily reflect those of the European Union. The European Union can be held responsible for them.

In the United Kingdom, the nationwide smart meter rollout has enabled more granular feedback and time-of-use tariffs. Trials combining smart meters with real-time feedback apps (such as the Low Carbon London project) showed that consumers could reduce peak demand by 10–15% under time-varying pricing, especially when supported by clear information and reminders (Schofield et al., 2014). In Italy, where time-of-use pricing has been in place for most households since 2010, studies show modest reductions in consumption during peak hours, but emphasize the importance of awareness and understanding: many households were initially unaware of the tariff structure, reducing its effectiveness. Behavioral nudges and clearer communication have since been tested to enhance responsiveness (Torriti, 2012). In Switzerland, real-time feedback on appliance-specific consumption, combined with environmental framing, induced significant reductions in energy use (Tiefenbeck et al., 2018). In the Netherlands, a reputation-based mechanism was found to be effective, while simple private communication was not (Delmas and Lessem, 2014).

These findings align with the growing consensus that one-size-fits-all approaches are suboptimal, and that personalized, data-informed strategies can improve both engagement and energy-saving outcomes. They also reinforce the importance of segmenting consumers to account for their diverse behaviors and responsiveness to price and non-price signals. This report outlines the methodology behind our novel clustering procedure, which identifies groups based solely on consumption data, and presents the characteristics of the resulting consumer segments.

We adopt a two-level clustering approach that combines the Self-Organizing Map (SOM) methodology with the K-means algorithm to identify energy consumption profiles and investigate their determinants. The SOM methodology uncovers patterns in high-frequency consumption data, while the K-means algorithm organizes consumers into distinct groups with similar usage behavior. We apply this methodology to hourly electricity consumption data obtained from smart meters installed in 6,254 Swiss households located in the canton of Zürich, covering the period from September 1, 2023, to August 30, 2024. After identifying temporally homogeneous and recurrent consumption clusters, we investigate their determinants, characteristics, and correlates.

We proceed in four steps. First, we transform and standardize the data. This allows us to focus on changes in consumption over time rather than on raw levels, and to compare behavior across households regardless of their baseline energy usage. This step is necessary because electricity consumption levels show high variance across households and tend to follow a scale-invariant distribution. In particular, we take the first difference of the data and apply a z-score transformation to normalize consumption across households. Our analysis is then conducted on the intra-day, normalized volatility of electricity consumption.

Second, we further simplify the data and reduce its dimensionality in a way that captures both short-term and long-term changes in electricity use. For this, we apply a wavelet transformation. This technique breaks down each household's consumption pattern into different time scales, enabling us to observe both rapid fluctuations and slower trends over the entire study period. A wavelet is like a small, wiggly wave used to decompose the time series of electricity usage into components representing different time scales—for example, daily cycles versus long-term trends. One can think of it as zooming in and out on a map: zoomed in, you see detailed local roads (short-term patterns); zoomed out, you see highways and city layouts (long-term trends). Wavelets decompose the data to reveal when in time changes occur, and how significant they are—something traditional methods like Fourier transforms handle less effectively for non-repeating or irregular signals (Percival and Walden, 2000). In our project, we use Haar wavelets, which act like a sliding window comparing values in chunks (e.g., electricity use in the first half of an hour versus the second half) to detect jumps or shifts. By stacking these comparisons over time, we can isolate short blips in usage (like a kettle boiling) from broader trends (like weekday vs. weekend habits).

Third, we use each household's wavelet-based energy pattern as input to group similar consumers. We begin with the SOM methodology (Kohonen, 1990), which helps uncover natural groupings in the data without requiring prior assumptions. SOM projects our high-dimensional data onto a 2D grid, where each point represents a typical usage pattern, and similar households are placed close together. The grid adapts over time to best represent the data structure. The SOM method is data-driven and unsupervised: it identifies structure on its own and requires only the size of the grid, not the number of desired groups. In our context, SOM maps the landscape of energy behaviors before we define the group boundaries. Next, we apply a K-means algorithm to draw clear boundaries and form actual clusters. The algorithm starts with a prespecified number of clusters, assigns each data point (i.e., household) to the nearest center, and then iteratively updates the centers until the groupings stabilize. The K-means algorithm is fast and intuitive, but can be sensitive to initial conditions and may struggle with high-dimensional or noisy data. That's why combining SOM and K-means is effective: SOM provides a robust and informative initialization for the clustering procedure. Together, they allow us to identify meaningful and reliable clusters of energy users.

Finally, the fourth step of our analysis examines how the clusters identified in the previous steps differ in terms of (i) time dynamics, levels, and volatility of electricity consumption patterns, and (ii) socio-economic characteristics of the corresponding households.

Our analysis yields three main findings. First, after addressing the issue of scale invariance, our clustering procedure successfully identifies four distinct clusters that differ significantly in their temporal energy consumption profiles. This is a noteworthy result, considering that our methodology is entirely agnostic and that our dataset is relatively homogeneous due to the selection criteria used during data collection. Second, the primary factor contributing to the formation of well-defined clusters is the intra-day volatility in energy consumption. Third, we document a strong relationship between volatility and overall consumption levels: households in clusters characterized by higher volatility also tend to use more energy. In this regard, the available socio-economic information sheds light on the main characteristics of households in highly volatile clusters: they typically live in detached houses, make extensive use of electricity for water heating, and are subscribed to energy contracts with reduced tariffs during nighttime hours and weekends.

2 Background

Mitigating global climate change is one of the most pressing challenges of the twenty-first century. To keep global warming below 1.5°C, extraordinary reforms are required from governments, industry, and communities (The Core Writing Team, 2023; Shukla et al., 2022). In addition to regulations and incentive-based policy interventions, understanding how to stimulate environmentally friendly behavior among citizens is essential to reducing CO_2 emissions. People can contribute to climate change mitigation by adjusting their lifestyles—such as traveling more sustainably, consuming less, adopting a plant-based diet (Wynes and Nicholas, 2017), and, most relevant to this report, changing their energy consumption habits.

Academic interest in behavioral approaches has grown rapidly over the last two decades. Today, the literature evaluating behavioral interventions in energy use is vast. Classical studies (e.g. Allcott, 2011; Allcott and Rogers, 2014) assess the impact of Home Energy Reports (HERs), which provide households with feedback comparing their energy use to that of neighbors, alongside conservation tips. These randomized field experiments typically demonstrate reductions in energy consumption. While initial reports induce immediate reductions, effects tend to diminish between report deliveries, and sustained delivery is necessary to support habit formation and persistent behavioral change. While social nudges may, in some cases, lead to crowding-out effects, HERs and peak-event reminders appear to have additive effects when combined, with little evidence of negative interactions (Brandon et al., 2019). Reviews and meta-analyses broadly confirm the positive average effects of such interventions—typically yielding 2%–4% reductions in electricity use—yet also highlight significant heterogeneity across studies (Andor and Fels, 2018; Nisa et al., 2019; Mertens et al., 2022; Karlin et al., 2015).

A growing body of research suggests that effective interventions must be targeted. Allcott and Kessler (2019) consider the welfare costs borne by nudge recipients and show that the welfare gains of HERs could double if the reports were directed at the most responsive households instead of sent universally. Costa and Kahn (2013) demonstrate how ideological differences lead to heterogeneous responses to energy reports. Feedback should not only be personalized but also delivered in real-time, when energy-intensive decisions are being made (Tiefenbeck et al., 2018). As argued by Van Valkengoed et al. (2022), understanding the determinants of behavior is fundamental, and interventions should be designed to address specific behavioral drivers. In this respect, tailoring behavioral interventions to the characteristics of households and their recurring consumption patterns is one of the most promising and fascinating areas for future research (Karaliopoulos et al., 2022).

To design such interventions, one first needs to identify relatively homogeneous groups of households. Several studies have performed segmentation exercises based on electricity consumption patterns (see the recent review in Michalakopoulos et al., 2024). Self-Organizing Maps (SOM) have been successfully applied across a range of fields, including energy systems and time-series analysis. In the context of electricity usage, SOM and K-means clustering have been used to group consumers with similar behavioral patterns, identify inefficiencies in buildings, and support energy management and policy design.

Räsänen and Ruuskanen (2008) applied SOM and K-means clustering to segment approximately 8,000 Finnish electricity customers by annual consumption patterns and building characteristics, suggesting how segmentation can support personalized feedback. Liu et al. (2012) adopted a similar method and identified four distinct consumer profiles differing in average daily load, peak usage, and seasonal patterns—offering valuable insights for tariff design and demand-side management. Talei et al. (2023) used SOM to analyze operational data from a highly efficient office building in Houston, USA, identifying inefficiencies with an estimated 4.6% energy-saving potential. Majidi and Smith (2023) proposed a hybrid SOM–K-means approach to analyze smart meter data from London households, evaluating clustering quality using silhouette scores. Abdelaziz et al. (2024) developed a hybrid framework combining SOM with deep learning models optimized via a genetic algorithm. Applied to public building energy data, the model improved both classification and prediction of usage patterns, highlighting the potential of integrating SOM with modern AI tools for smart energy management. McLoughlin et al. (2015) compared different unsupervised clustering algorithms to group households into usage profiles and linked those to household characteristics, showing that electricity usage can be explained by individual-level attributes. Al Khafaf et al. (2020) proposed an entropy-based clustering index to determine the optimal number of electricity user clusters, applying it to datasets from Australia and Ireland. Several studies compare the performance of different clustering techniques (e.g. Chicco et al., 2006; Yilmaz et al., 2019), and some use both smart meter and survey data for clustering (e.g. Gouveia and Seixas, 2016).

Our methodology combines three instruments: wavelet transformation, SOM, and the K-means algorithm. While some of the studies mentioned above used subsets of these tools, none have integrated all three. More importantly, these previous approaches typically overlook the scale-free nature of electricity consumption distributions, clustering users based on *levels* of usage. In contrast, we analyze *changes* in electricity consumption over time. We do so by differencing the data and applying z-score normalization, which allows us to focus on the *volatility* of consumption rather than its raw level. This approach enables the identification of behaviorally meaningful clusters that reflect dynamic energy use patterns rather than static consumption quantities.

3 Dataset Characteristics and Pre-processing

3.1 Overview and Selection Criteria

We partnered with a Swiss electricity provider[1]. The dataset used in this analysis was extracted and transferred by an energy sector research company in March 2025.

To ensure meaningful clustering, we deliberately adopted a conservative approach by selecting a dataset of households that is, ex ante, relatively homogeneous. Specifically, we asked the Swiss electricity provider and energy sector research company to provide a representative sample of households drawn randomly (with stratification based on house characteristics, number of occupants, and heating type). This sample adheres to the following selection criteria:

- Demo users and users with business tariffs are excluded;
- Households must have been clients since at least September 1, 2023;
- Electricity consumption must have been continuously recorded via a smart main meter (with an asset type code starting with 1, indicating a residential household meter) since September 1, 2023;
- Users in the bottom decile of energy consumption are excluded.

Only households with a complete, uncorrupted time series of electricity consumption data and socio-demographics data are included in the analysis. The resulting dataset consists of 225,460,898 electricity consumption records (in kWh) measured at 15-minute intervals from the main household meter for 6,254 households located in the canton of Zürich, covering the period from September 1, 2023, to August 30, 2024. The mean consumption per 15-minute interval is 0.15 kWh (standard deviation: 0.341).

Unit	Time Unit	N. Obs	Households	Mean	Median	Std. dev.	Skewness
kWh	15-min.	225.460.898	6,254	0.154	0.05	0.341	7.68

Table 1: Descriptive statistics of the electricity consumption dataset. Statistics refers to average energy consumption per quarter hour.

[1] In line with the NDA agreements made with the company, all references in this report to the electricity supply company (hereafter “Swiss electricity provider”), his energy product (renamed as "Energy Product") and the data management company (hereafter “energy sector research company”) have been anonymized.

To reduce dataset size and focus on temporal dynamics, we aggregate 15-minute readings into hourly sum:[2]

$$kWh_{i,h} = \sum_{j=1}^{4} kWh_{i,h,j}, \tag{1}$$

where i indexes households, h denotes the hour, and j refers quarter-hour observations within that hour.

Unit	Time Unit	N. Obs	Households	Mean	Median	Std. dev.	Skewness
kWh	Hour	225.460.898	6,254	0.617	0.226	1.26	8.21

Table 2: Descriptive statistics of the electricity consumption dataset. Statistics refers to average energy consumption per hour.

As discussed in the introduction, we first show that scale invariance is a relevant feature of the dataset and then transform the data accordingly. The analysis is carried out on intra-day, normalized volatility of electricity consumption, obtained via a z-score transformation of the first differences. Next, we apply Haar wavelet transformation at the household level to reduce dimensionality and isolate both short- and long-term volatility patterns. Then, we cluster households using SOM, followed by K-means refinement. Finally, we explore the determinants of the resulting clusters using available household-level attributes.

3.2 Data Preprocessing

Scale Invariance. Scale invariance is a common property in distributions generated by human behavior, characterized by heavy tails. In this context, clustering based solely on absolute levels may obscure important temporal features such as volatility and time-of-day patterns in energy use.

A probability distribution $f(x)$ is said to be *scale-invariant* if, for any positive constant $c > 0$, the distribution of the transformed variable $Y = cX$ has the same functional form:

$$f_Y(y) = \frac{1}{c} f\left(\frac{y}{c}\right). \tag{3.1}$$

More specifically, a power-law distribution that satisfies the following homothetic relation is scale-invariant:

$$f(x) \propto x^{-\alpha}, \quad \text{for some } \alpha > 0. \tag{3.2}$$

Figure 1 visualizes the scale-invariance property in our dataset through a histogram of electricity consumption values (Figure 1a) and a log-log plot (Figure 1b). The heavy-tailed nature of the distribution and the near-linear shape of the log-log plot provide *prima facie* evidence of scale invariance.

First-difference transformation. In order to mitigate the dominance of absolute levels due to the scale-invariance of the original data, we compute the first differences between consecutive hourly data points:

$$dkWh_{i,h} = kWh_{i,h} - kWh_{i,h-1}. \tag{2}$$

This transformation also allows us to directly measure intra-day volatility and identify behavioral patterns such as consumption spikes and dips.

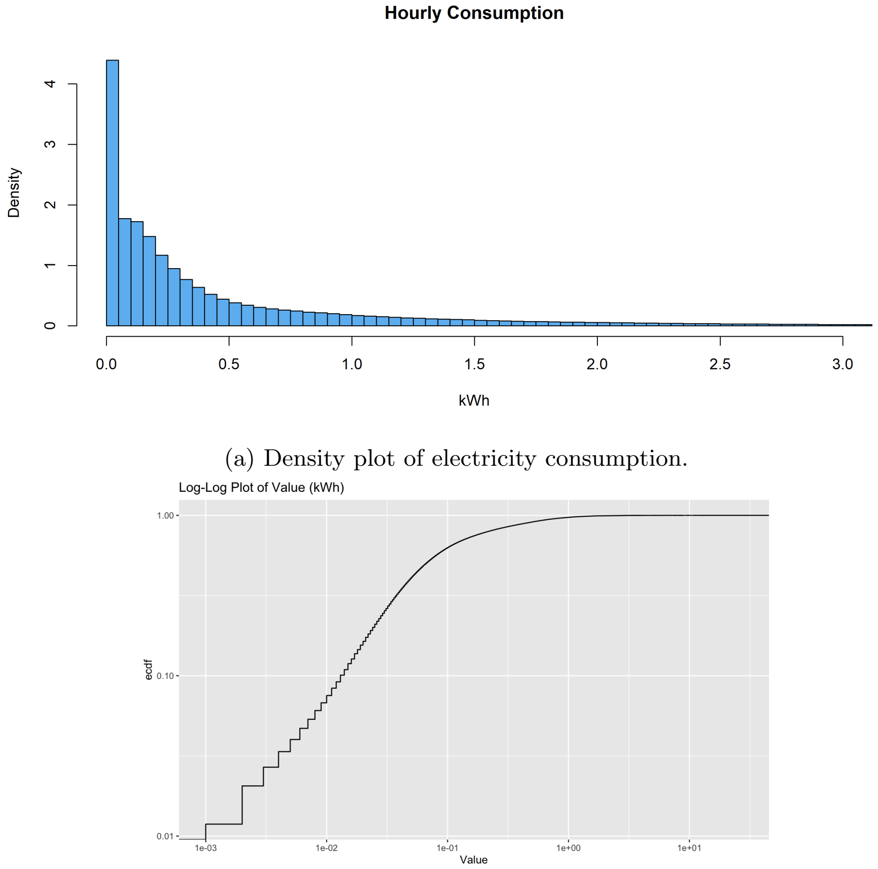
Standardization To enhance comparability across households and account for outliers, we normalize $dkWh_{i,h}$ using a z-score transformation:

$$Z_{i,h} = \frac{dkWh_{i,h} - \mu_i}{\sigma_i}, \tag{3}$$

where μ_i and σ_i are the mean and standard deviation of $dkWh_{i,h}$ for each household i. Figures 2a and 2b illustrate average hourly consumption and normalized z-scores across all households and days.

The hourly consumption shows a typical daily pattern: lower usage in the mid-day hours and higher consumption in the evening and early morning. The z-scores highlight periods of high volatility, providing additional insights into consumption dynamics across time.

[2] The R ((Posit team, 2025)) and STATA ((StataCorp, 2025)) codes used in the following analysis are reported in the appendix.



(a) Density plot of electricity consumption.

(b) Log-log rank-size plot of electricity consumption.

Figure 1: Visualization of scale-invariance of electricity consumption.

4 Clustering Procedure

4.1 Discrete Wavelet Transform (DWT)

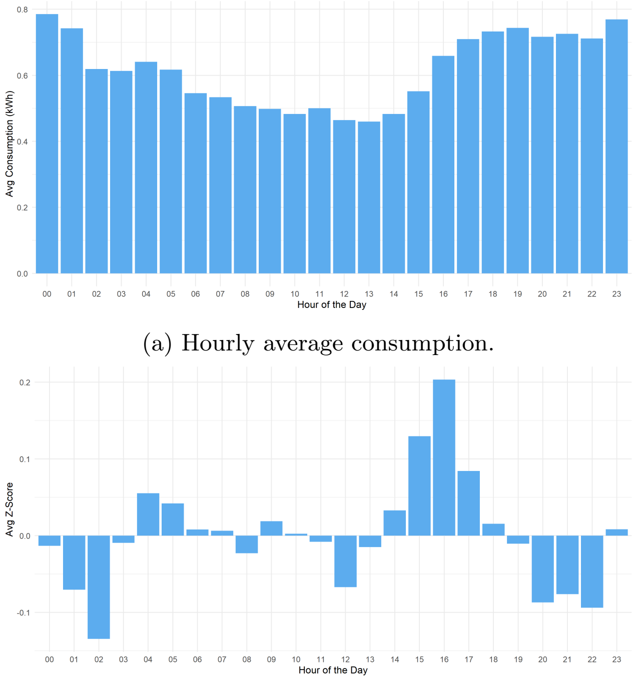
The wavelet transformation employs a flexible windowing approach that adapts over time, enabling it to resolve low-frequency information using a window function whose radius increases with time (and decreases with frequency). As a result, it provides fine time resolution for short-duration, high-frequency components, and fine frequency resolution for long-duration, low-frequency components. Through multi-resolution decomposition, data, functions, or operators can be separated into components of different scales.

Various wavelet types have been proposed for different analytical goals. For instance, complex wavelets such as the Morlet wavelet are well-suited for analyzing periodic or scale-specific time series, while real-valued wavelets like the Mexican hat and Haar wavelets are more appropriate for detecting discontinuities, singularities, or abrupt signal changes.

We employ the Haar DWT for dimensionality reduction, enabling efficient representation of signal data in a compact form that supports subsequent analysis. DWT is particularly effective in capturing essential signal features across multiple frequency scales, simplifying the original data while preserving its key characteristics.

The core concept behind wavelet analysis is to express a signal as a linear combination of functions derived from a single "mother wavelet," alongside scaling functions. This framework allows wavelets to simultaneously capture frequency and temporal information, making them especially suitable for analyzing non-stationary signals or those exhibiting abrupt shifts. Unlike the Fourier transform, which uses infinitely extended sine and cosine waves, wavelets have finite support, enabling localized and accurate identification of temporal changes in signals.

In our analysis, we apply DWT to a sequence of standardized hourly measurements, $Z_{i,h}$, at the household level, decomposing the signal into approximation coefficients $C_i(k;j_0)$ and detail coefficients $\Gamma_i(k;j_0)$. The approximation coefficients capture low-frequency components and overall signal trends, whereas the detail coefficients represent high-frequency components and localized



(a) Hourly average consumption.

(b) Z-scores of hourly consumption differences.

Figure 2: Average electricity consumption by hour of day.

variations. Mathematically, this decomposition is described as:

$$C_i(k; j_0) = \sum_{h=1}^{N} Z_{i,h}\, \phi_h(k; j_0), \quad k = 1, \ldots, K, \tag{4}$$

$$\Gamma_i(k; j_0) = \sum_{h=1}^{N} Z_{i,h}\, \psi_h(k; j_0), \quad k = 1, \ldots, K, \tag{5}$$

where:

- h indexes the hourly measurements, ranging from 1 (corresponding to the first hour, 00:00–01:00 on September 1, 2023) to N (the final hour, 23:00–00:00 on August 31, 2024);
- $C_i(k; j_0)$ and $\Gamma_i(k; j_0)$ denote the approximation and detail coefficients, respectively, for household i;
- $\phi_h(k; j_0)$ and $\psi_h(k; j_0)$ represent the scaling and wavelet functions associated with decomposition level j_0 and coefficient index k;
- the decomposition level j_0 is chosen as the maximum admissible level that can be applied consistently across all households in the dataset.

The combined use of standardized signals ($Z_{i,h}$), rather than raw energy consumption compressed into wavelet coefficients, offers several advantages. First, it reduces the dominance of absolute magnitudes, making frequency analysis more robust. Second, it enhances the detection of abrupt changes, as standardized signals clearly highlight sudden variations. Finally, it enables the capture of patterns at various temporal resolutions, thereby providing deeper insights into signal behavior.

Through recursive decomposition, the DWT efficiently produces a compact set of coefficients suitable for further analysis. Specifically, our analysis yields a total of 8,745 detail coefficients, which are subsequently used as input data in the next section for cluster analysis.

Table 3 represents the decomposition of a one-year time series into different time-scale components using wavelet analysis. Each row in the table corresponds to a different level j of the decomposition, where the number of coefficients is calculated as:

$$\Gamma_i(k; j_0) \approx \frac{8745}{2^j}.$$

This division by two is motivated by the fact that the wavelet transformation, applied recursively, starts with the original hourly data (8,745 hours in a year) and applies a pair of filters—a low-pass filter for the approximation (the "slow" or low-frequency part) and a high-pass filter for the details (the "fast" or high-frequency part). After filtering, the algorithm downsamples the results by retaining every other sample (one in, one out), as adjacent samples are considered redundant. This downsampling halves the number of observations at each level. We can therefore interpret the coefficients as follows:

- Level 01 ($j = 1$: With $\frac{8745}{2^1} \approx 4372$ coefficients, this level captures the finest scale of the data, reflecting hourly variations.
- Level 02 ($j = 2$: With $\frac{8745}{2^2} \approx 2186$ coefficients, it represents variations occurring approximately every 4 hours (sub-daily changes).
- Level 03 ($j = 3$): With $\frac{8745}{2^3} \approx 1093$ coefficients, this level approximates the day/night transition.
- Level 04 ($j = 4$): With $\frac{8745}{2^4} \approx 547$ coefficients, the decomposition captures daily variations in the data.
- Levels 05–10 ($j = 5, \ldots, 10$): These levels capture progressively coarser scales of the data, corresponding to multi-day trends, weekly patterns, bi-weekly trends, monthly fluctuations, and seasonal or quarterly variations.

Level j	$\Gamma_i(k; j_0)$	$C_i(k; j_0)$	**Explanation**
01	$8745/2^1 \approx 4372$	-	Hourly variation
02	$8745/2^2 \approx 2186$	-	4-hour change (sub-daily)
03	$8745/2^3 \approx 1093$	-	Approximate day/night transition
04	$8745/2^4 \approx 547$	-	Daily variation
05	$8745/2^5 \approx 273$	-	Multi-day trends
06	$8745/2^6 \approx 137$	-	Short-term trend
07	$8745/2^7 \approx 68$	-	Weekly patterns
08	$8745/2^8 \approx 34$	-	Bi-weekly trends
09	$8745/2^9 \approx 17$	-	Monthly fluctuations
10	$8745/2^{10} \approx 9$	$A_{10} \approx 9$	Seasonal/quarterly trends

Table 3: Temporal granularity of the wavelet coefficients, ranging from the highest frequency (two-hourly) to the lowest frequency.

In summary, the table not only displays the number of coefficients at each decomposition level but also associates these levels with the specific time scales they represent. This wavelet decomposition enables analysis of the data at various resolutions, ranging from high-frequency hourly variations to lower-frequency seasonal trends.

4.2 Self-Organizing Maps (SOM)

SOM is a type of neural network particularly well-suited for clustering and visualizing high-dimensional data. The SOM algorithm maps input data onto a two-dimensional grid of neurons, producing a structured and intuitive representation. A typical SOM consists of an input layer and a Kohonen layer: each unit in the Kohonen layer (usually arranged in a 2D grid) represents a weight vector that competes during the training process to best match the input data. The SOM learns through competitive learning, where the best-matching unit (BMU) is updated along with its neighborhood, preserving the topological relationships of the input space on the output map. This feature makes SOM especially effective for visualizing complex data structures and for identifying clusters or patterns that may not be apparent in the raw data.

Given the set of wavelet detail coefficients $\Gamma_i(k; j_0)$, a grid of neurons is selected (a 20 × 20 grid, considering the dataset size), where each neuron represents a point in the data space[3]. Each neuron has a fixed position on the SOM grid and an associated weight vector $\mathbf{w}$, which has the

[3] The choice of grid size depends on the number of observations and the desired output resolution. A $20 \times 20 = 400$ grid is appropriate given the size of the dataset, and 400 points provide a suitable basis for the subsequent K-means analysis.

same dimension as the input data (i.e., the number of wavelet coefficients). In this specific case, given the length K of the coefficients, each neuron n, with $n = 1, \dots, N$, will have a weight vector:

$$w_n = (w_1, w_2, \dots, w_K) .$$

The SOM proceeds iteratively by presenting each observation from the dataset to the model one at a time. Each time a new data point is introduced to the SOM, every neuron computes the distance between its weight vector and the input vector. The neuron with the closest weight vector is identified as the Best Matching Unit (BMU); subsequently, the weights of the BMU, as well as those of its neighboring neurons on the grid, are updated according to the following equation:

$$w_n(t+1) = w_n(t) + \eta(t) \cdot h_{\tilde{w},n}(t) \cdot \Big(x - w_n(t)\Big) \tag{6}$$

where:

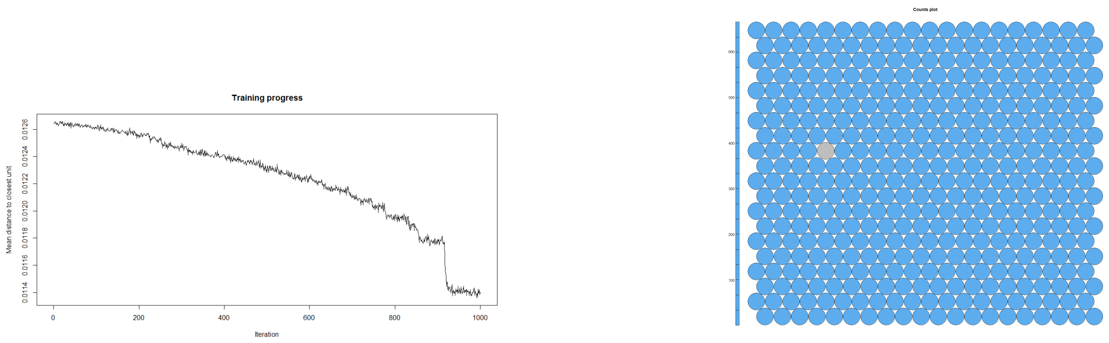
- $w_n(t)$ is the weight vector of neuron n at time t;
- $\eta(t)$ is the learning rate, which is decreasing over time.
- $h_{c,n}(t)$ is the neighborhood function centered on the winning neuron $\tilde{w}$;
- x is the input vector (wavelet coefficients).

After a predetermined number of iterations (1000 in this case), the algorithm terminates. The output of SOM includes, among others, two essential visualizations:

- Training Progress: Figure 3a shows how the difference between neuron weights and the corresponding input samples decreases over the course of the iterations, eventually converging toward a minimum value. This behavior indicates effective SOM training.

- Neuron Counts: Figure 3b illustrates the distribution of samples across neurons. An ideally uniform distribution suggests that the grid size is appropriate. Only one neuron remains unassigned, further supporting the adequacy of the chosen map size.

The two-dimensional topology of SOM is significant as it visually preserves the proximity relationships among data points, effectively representing similar consumption patterns near each other on the grid. Subsequently, rather than using raw data, we input the representative vectors of each of the 400 SOM clusters as initial conditions to the K-means algorithm. This approach solves the problem of sensitivity to initial conditions. Additionally, leveraging the SOM topology, the resulting K-means clusters can be intuitively visualized within the structured two-dimensional space generated by SOM.



(a) Training Progress

(b) Neurons Counts

Figure 3: SOM Visualisation

4.3 K-Means Analysis

K-means is a widely used unsupervised clustering algorithm that partitions a dataset into k clusters by minimizing the within-cluster variance. The algorithm iteratively assigns each data point to the nearest cluster centroid and then updates the centroids based on the current assignments. Despite its simplicity and efficiency, K-means suffers from two notable limitations. First, it is sensitive to the initial placement of centroids, which can lead to convergence to local minima rather than the global optimum. This implies that different runs of the algorithm may yield different results depending on the initialization. Second, K-means operates in the original input space, which can hinder result interpretation, especially in the presence of high-dimensional or noisy data. In particular, the algorithm offers limited support for visualizing the clustering structure in a meaningful way.

After training the SOM, we apply the K-means clustering algorithm to classify the identified energy consumption patterns. Specifically, the representative vectors (or codebook vectors) derived from the SOM are used as inputs for K-means clustering. Each of these vectors corresponds to the weight vector of a neuron and represents a distinct pattern captured by the SOM. Combining K-means with SOM provides a robust solution to the aforementioned issues. By first projecting the high-dimensional input data onto a two-dimensional topological map via the SOM, the intrinsic structure of the data is preserved and visualized more effectively. The SOM organizes similar data points into neighboring regions of the grid, reducing dimensionality and noise. When K-means is subsequently applied to the SOM output — typically the weight vectors or activation patterns — the initialization problem is mitigated due to the pre-structured nature of the input, and the resulting clusters become easier to interpret both numerically and graphically. This hybrid approach thus leverages the strengths of both methods: the topology-preserving mapping of SOM and the compact cluster formation of K-means.

The primary objective of the K-means algorithm in this context is to partition the SOM prototype vectors into groups by minimizing the variance within each cluster. Initially, each vector is assigned to the nearest cluster centroid based on Euclidean distance. The algorithm then iteratively recalculates the centroids as the average of all vectors assigned to each cluster. This process continues until the cluster assignments converge, indicating that the clusters have stabilized.

4.4 Determining the Optimal Number of Clusters

An essential step in clustering analysis is determining the appropriate number of clusters. To this end, we employ two complementary methods: the elbow technique and silhouette analysis.

The *elbow technique* visually assesses the optimal number of clusters by plotting the explained variance as a function of the number of clusters. The optimal number corresponds to the point where the rate of increase in explained variance begins to level off significantly. As shown in Figure 4, we identify four clusters as the optimal choice, since adding more clusters beyond this point yields only marginal improvements.

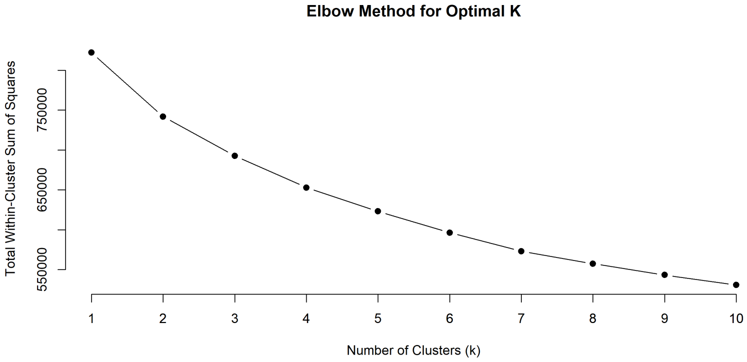


Figure 4: Elbow Method Result

To further validate this result, we perform a *silhouette analysis*. Specifically, the silhouette score quantifies how well each point is assigned to its cluster by comparing its cohesion (similarity with elements of the same cluster) to its separation (dissimilarity with elements of the nearest neighboring cluster). For each point p, the silhouette score $s(p)$ is defined as:

$$s(p) = \frac{b(p) - a(p)}{\max\{a(p), b(p)\}}$$

where $a(p)$ is the average distance between point p and all other points in the same cluster (intra-cluster distance), and $b(p)$ is the minimum average distance between point p and the points in any other cluster (nearest-cluster distance). The resulting score ranges from -1 to 1, where a value close to 1 indicates that the point is well clustered, a value near 0 suggests it lies between two clusters, and negative values imply misclassification. As illustrated in Figure 5, the highest silhouette score is achieved with four clusters, confirming the result suggested by the elbow technique.

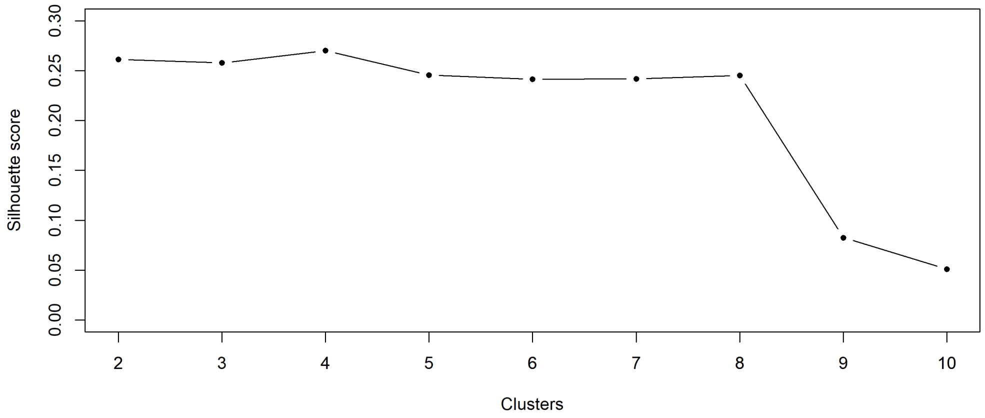


Figure 5: Silhouette Score Results

5 Analysis of the Clusters

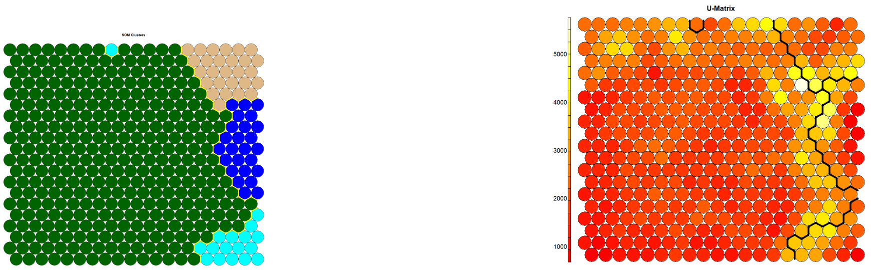
5.1 Identified Clusters and SOM Visualization

Cluster	Mean Hourly	SD Hourly	Unique ID Count
1	0.842	1.25	336
2	0.600	1.27	5291
3	0.818	1.35	353
4	0.812	1.20	274

Table 4: Statistics by cluster

The final clustering results are visualized directly on the trained SOM. Figure 6a clearly illustrates the spatial distribution of the four clusters identified by K-means on the SOM grid. One large cluster (cluster 2, colored in green, comprising 5,291 households) covers a significant portion of the map, while three smaller clusters — cluster 1 (blue, 336 households), cluster 3 (sand, 353 households) and cluster 4 (cyan, 274 households) — occupy distinct and separate regions. Notably, the clusters display clear spatial coherence, although one neuron from the cyan cluster appears slightly displaced near the blue cluster, further confirming the robustness of the clustering approach.

Additionally, Figure 6b presents the U-Matrix, which visually represents the distances between neurons on the map. Warmer colors indicate shorter distances (i.e., greater similarity between neurons), while cooler colors represent larger distances (i.e., lower similarity). This visualization aids in interpreting cluster boundaries and assessing their compactness and separation.



(a) SOM Mapping with 4 Clusters

(b) UMatrix: Distance between neurons

Figure 6: SOM results and visualisation

5.2 Differences in Energy Consumption Patterns Across Clusters

We now examine differences in energy consumption time patterns across the four clusters identified through the two-step procedure described in the previous subsections.

Figure 7 shows, for each cluster, the evolution of mean of $Z_{i,h}$ over time.

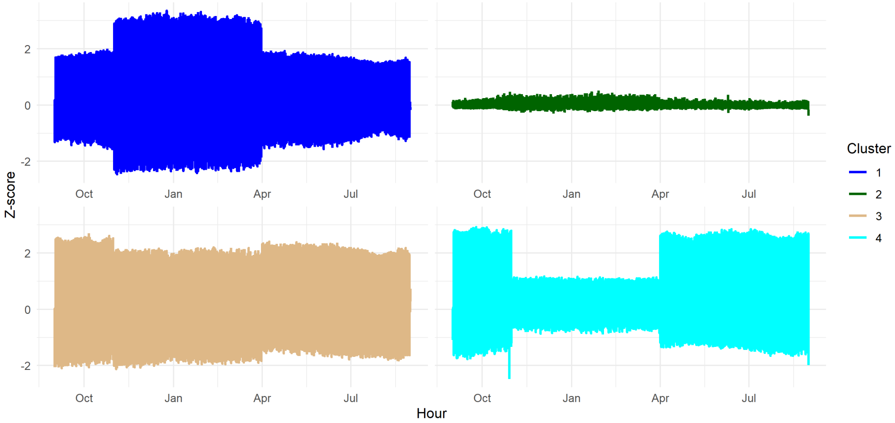


Figure 7: Mean of $Z_{i,h}$ in the four clusters over time.

Two main observations emerge from the analysis. First, Z-scores are more volatile in clusters 1, 3, and 4 than in cluster 2. Second, we observe marked seasonal differences in volatility across these clusters. In particular, the most notable contrast occurs between cluster 1 and cluster 4: while cluster 1 exhibits high volatility from late autumn to early spring and more stable patterns during the rest of the year, cluster 4 shows the opposite — greater volatility from late spring to early autumn.

These differences are further illustrated in Figure 8, which shows the average energy consumption levels for each cluster during four representative weeks — one for each season (January 14–21; April 7–14; June 30 – July 6; September 30 – October 6). The conclusions remain robust when alternative weeks are selected.

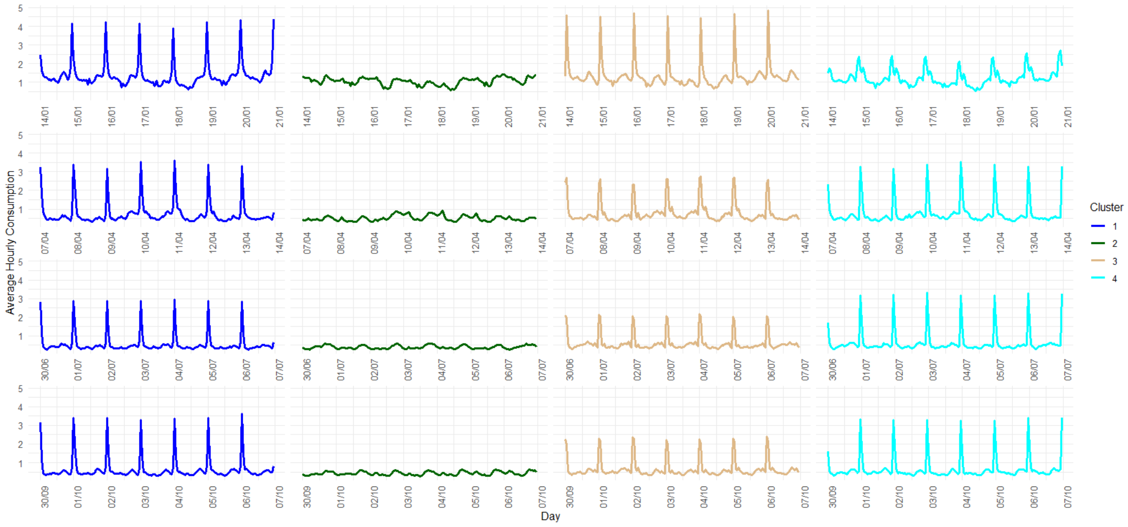


Figure 8: Average energy consumption levels of the four clusters in the four representative weeks. The week begins on Monday and ends on Sunday. In the graph, there is an extra day on the x-axis to represent midnight between the two days.

The graphical analysis of hourly consumption variations across the clusters allows us to draw several key conclusions. First, it confirms the presence of season-specific volatility patterns, particularly for clusters 1 and 4. Specifically, cluster 1 displays greater consumption variability during the January week compared to July, while cluster 4 shows the reverse trend — greater stability in January and higher volatility during the summer week. Second, the analysis highlights recurring daily peaks in energy consumption, especially in clusters 1, 3, and 4, which are consistently observed across seasons.

A more detailed intraday analysis, presented in Figure 9 and based on the Fridays of the representative weeks as reference days, reveals that these peaks tend to occur during nighttime hours, particularly for clusters 1 and 4. During daytime and typical working hours, the differences between clusters become less pronounced.

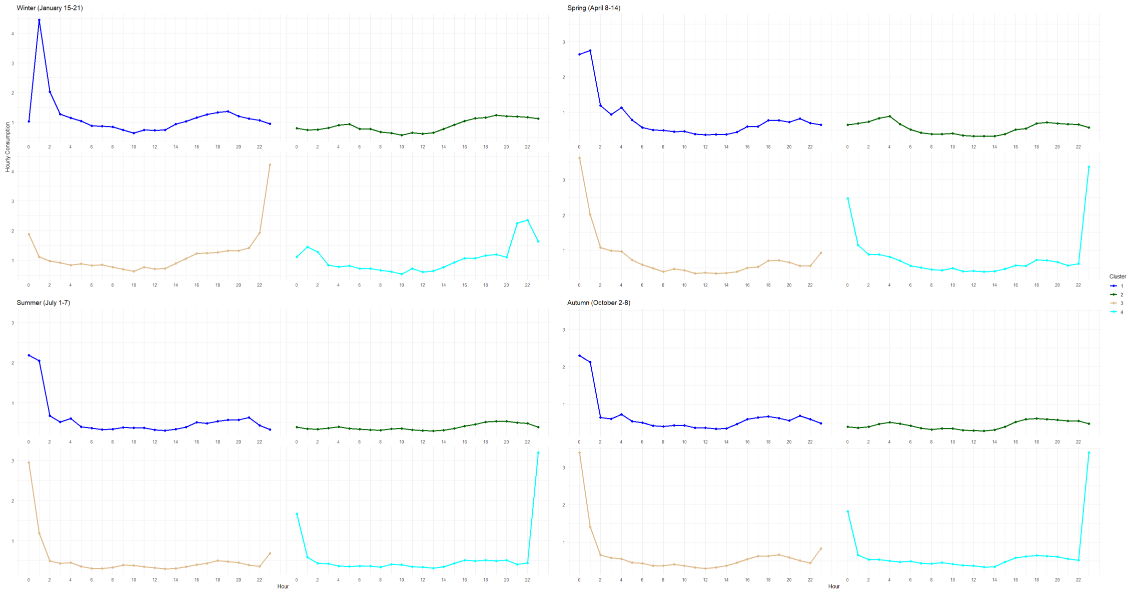


Figure 9: Average intraday energy consumption profiles for the four clusters, based on Fridays from the four representative weeks

To further validate these insights, we analyze the wavelet detail coefficients that most effectively distinguish between clusters in the SOM output. To rigorously identify the coefficients driving inter-cluster differences, we compute the centroid vectors — defined as the average weight vectors representing the characteristic patterns of each cluster. We then calculate the squared differences between the centroid vectors across clusters, focusing on comparisons of clusters 1, 3, and 4 with respect to cluster 2. This allows us to quantify the divergence for each detail coefficient and determine at which temporal scales the differences are most pronounced.

Figure 10 illustrates, across decomposition levels $j = 1, \ldots, 10$, the distribution of squared differences in wavelet detail coefficients between cluster 2 (the reference cluster) and each of the remaining clusters: cluster 1, cluster 3, and cluster 4. For simplicity, each graph is labeled with the number of the cluster being compared to the reference group.

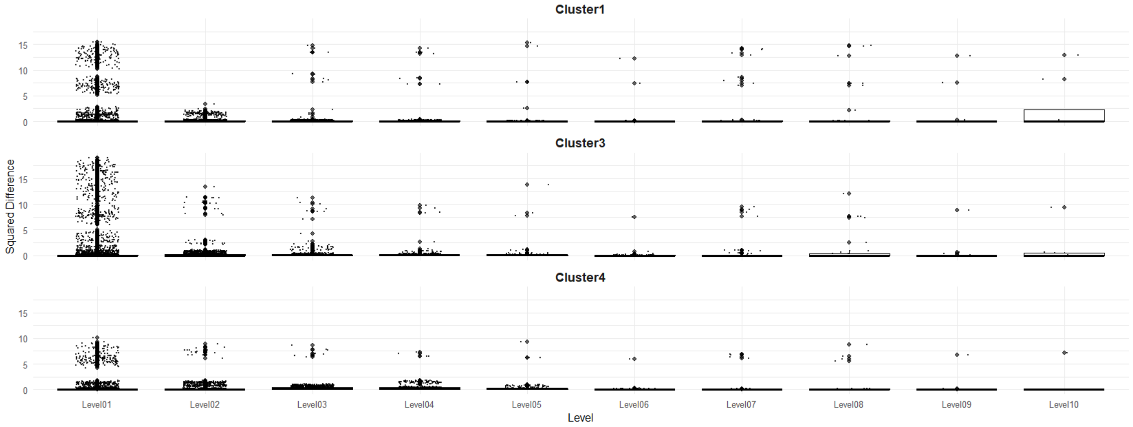


Figure 10: Distribution of squared differences in wavelet detail coefficients between clusters 1, 3, and 4 and the reference cluster 2, across decomposition levels $j = 1, \ldots, 10$.

As illustrated by Table 3, at lower levels (e.g., $j = 1$ or $j = 2$), the detail coefficients capture high-frequency components, such as hourly or sub-daily variations. As the level increases, the wavelet decomposition isolates lower-frequency trends, including weekly, bi-weekly, and seasonal patterns. Therefore, larger squared differences at a given level suggest greater dissimilarity between the two clusters at that specific time scale.

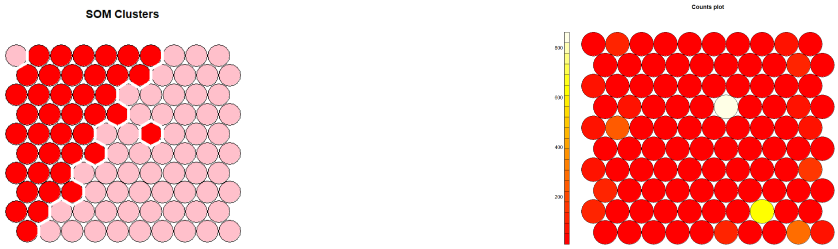
The main result shown by Figure 10 is that, for all three pairwise comparisons, the distribution of squared differences in wavelet detail coefficients at level 1 exhibits the highest number of points with extremely large values. This suggests that the differences between clusters are primarily driven by high-frequency components.

5.3 Robustness Analysis: Evaluating the Effect of Re-Clustering

Despite the ex-ante homogeneity of the households due to the selection criteria, the clustering analysis successfully identified four distinct groups exhibiting substantially different energy consumption patterns over time. Among these, cluster 2 is the largest, encompassing approximately 85% of the households.

As a robustness check to validate our clustering approach, we repeat the analysis using the SOM (with a training length of 500 iterations) and K-means (with 100 initializations), starting with a re-clustering of the households originally assigned to cluster 2. We then iteratively apply the same procedure to the largest resulting cluster from each step, continuing until further partitioning fails to reveal meaningful or interpretable subgroups.

After two iterations, the process converges to a final scenario (illustrated in Figure 11) consisting of only two clusters. The first cluster comprises 1,476 households and is characterized by an average hourly energy consumption of 0.212 kWh (standard deviation: 0.355), while the second cluster includes 3,346 users with an average hourly energy consumption of 0.105 kWh (standard deviation: 0.218).



(a) SOM Mapping

(b) SOM Counts

Figure 11: Results of the re-clustering robustness check.

Relative to the initial results, the final implementation of the analysis exhibits limited explanatory power and a substantial deterioration in the quality of household segmentation.

First, the silhouette score significantly declines, reaching a value of only 0.0562 in the last iteration, suggesting that the clustering becomes less meaningful. As a benchmark, consider that the average silhouette score across cluster numbers from two to ten in the initial clustering is 0.21, which drops to just 0.03 in the final repetition.

Second, the sum of squared distances also drops sharply — from 532,200.5 in the first iteration to 13,401.49 in the second (and final) one — indicating a much smaller contribution of the additional clustering step to the overall clustering quality.

6 Clustered Energy Use and Socio-Economic Drivers

As previously mentioned, our clustering approach is entirely data-driven and model-agnostic. Therefore, to better interpret the resulting clusters, it is essential to examine the specific characteristics of the corresponding households and assess whether the identified groups differ along relevant socio-economic dimensions.

To this end, for those households that provided explicit informed consent for research purposes, the Swiss electricity provider supplied the following socio-economic information: (i) type of housing associated with the energy contract, (ii) property ownership status, (iii) number of occupants, (iv) type of space heating, (v) type of water heating, (vi) household postal code, and (vii) electricity product specified in the household's contract.

Building on this information, the following analysis proceeds in two steps. First, we descriptively examine the socio-economic profiles of the four clusters and assess, through parametric methods, to what extent these variables explain the segmentation produced by the clustering. Second, controlling for socio-economic characteristics, we investigate whether there are significant differences in electricity consumption levels, $kWh_{i,h}$, across the clusters.

6.1 A Socio-Economic Characterization of the Identified Clusters: Descriptive and Parametric Evidence

Table 5 reports the socio-economic characteristics of the full sample, as well as for each cluster individually.[4]

	Overall	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
Composition and Energy Consumption					
N. Households	6254	336	5,291	353	274
$KWh_{i,h}$	0.768	0.842	0.600	0.818	0.812
	(1.26)	(1.25)	(1.27)	(1.35)	(1.20)
Demographics					
Occupants	2.55	2.57	2.54	2.64	2.62
	(1.07)	(1.04)	(1.07)	(1.08)	(1.07)
Building age	45.23	54.43	44.51	43.25	49.94
	(43.76)	(57.10)	(42.35)	(47.51)	(44.89)
Housing Characteristics					
Property (Owner)	77.14%	84.64%	75.48%	89.33%	84.47%
Property (House)	62.42%	78.26%	59.37%	80.79%	78.41%
Energy Product					
Energy Product 1	79.37%	89.86%	77.56%	84.45%	94.70%
Energy Product 2	15.35%	6.67%	16.89%	10.06%	3.03%
Heating					
Heat Pump	48.94%	51.75%	47.73%	62.81%	52.29%
Fossil Fuels	35.37%	37.72%	35.44%	30.31%	37.02%
Electric	2.79%	2.92%	2.94%	1.56%	1.15%
Other	12.90%	7.90%	13.89%	5.31%	9.54%
Water Heating					
Heat Pump	34.89%	12.51%	38.34%	19.06%	15.27%
Fossil Fuels	19.96%	2.92%	23.06%	1.25	3.82%
Electric	27.54%	78.36%	18.73%	75.31	76.72%
Other	17.61%	6.14%	19.87%	4.38%	4.20%

Table 5: Socio-economic characteristics (with standard deviation in parentheses when applicable) of the households in the identified clusters.

Cluster 2, the largest in size, is associated with the lowest average hourly electricity consumption (0.600 kWh), while cluster 3 exhibits the highest average consumption (0.818 kWh) as well as the largest standard deviation (1.35), indicating greater variability within the group.

The average number of household occupants remains relatively stable across clusters, with cluster 3 showing a slightly higher value. Cluster 2 stands out for having the lowest percentage of homeowners (75.48%) and the lowest share of houses (59.37%) relative to apartments. In contrast, cluster 3 has the highest proportion of both homeowners and houses. No particularly notable differences emerge between clusters 1 and 4 in this regard.

In terms of energy product choices and related behaviors, cluster 2 shows the lowest proportion of households subscribed to the Energy Product 1 contract, whereas cluster 4 exhibits the highest share of subscribers to this energy product. Moreover, cluster 4 also records the highest percentage of households subscribed to the Energy Product 2 contract.[5]

[4] The number of households, as well as the mean and standard deviation of $KWh_{i,h}$ reported in each column, refer to the complete dataset (6,254 households) used to construct the four clusters. The remaining descriptive statistics on socio-economic dimensions, along with the parametric analysis presented below, refer to the 6,093 households (342 in cluster 1, 5,169 in cluster 2, 320 in cluster 3, and 262 in cluster 4) for which the selected complete socio-economic information is available.

[5] Energy Product 1 is a specific energy contract that applies a standard tariff rate (HT) from 7:00 am to 8:00 pm on weekdays (Monday to Friday), and a reduced tariff rate during nighttime hours and weekends. The *Energy Product 2* contract applies a tariff scheme paid by grid operators to generators for electricity fed into the public

Regarding house heating, heat pumps represent the most common technology in the dataset. In this dimension, cluster 2 again stands out for having the lowest proportion of households equipped with a heat pump.

Finally, interesting differences emerge with respect to water heating. Cluster 2 records the highest percentage of households using a heat pump and the lowest using electricity for water heating. The opposite holds for clusters 1, 3, and 4: they rely more heavily on electricity for water heating and show a lower percentage of households with heat pumps compared to cluster 2.

We now proceed to a parametric assessment of these observed differences using a multinomial logit model. The model estimates how socio-economic characteristics influence the likelihood of a household belonging to clusters 1, 3, or 4, relative to cluster 2 — which serves as the reference category due to its large size and representativeness. To enhance the interpretability of the results, we adopt a parsimonious specification of the model by selecting a subset of relevant independent variables. The estimation results are presented in Table 6.

	Cluster 1	Cluster 3	Cluster 4
Energy Product			
Energy Product 1	1.421***	1.041***	2.149***
	(0.202)	(0.172)	(0.284)
Heating			
Heat pump	0.070	0.157	-0.083
	(0.148)	(0.154)	(0.160)
Electric	-1.057***	-1.807***	-2.876***
	(0.355)	(0.475)	(0.606)
Water Heating			
Heat pump	0.219	1.185***	0.703**
	(0.280)	(0.154)	(0.160)
Electric	2.814***	3.447***	3.160***
	(0.227)	(0.282)	(0.261)
Household			
Number of occupants	-0.071	-0.062	-0.037
	(0.061)	(0.062)	(0.068)
Renter (vs. Owner)	0.041	-0.152	0.055
	(0.831)	(0.474)	(0.788)
House (vs. Apartment)	0.694***	0.667***	0.753***
	(0.184)	(0.183)	(0.198)
Building $>$ 1989	0.166	0.558***	0.214
	(0.137)	(0.140)	(0.145)
Constant	-5.665***	-6.283***	-6.992***
	(0.336)	(0.368)	(0.408)
Observations		6,093	
Pseudo R-squared		0.2080	
$LR - \chi^2$		1163.67	
$Prob > \chi^2$		0.0000	

Notes. This table reports estimates (robust standard errors in parentheses) of a multinomial regression. The dependent variable is the cluster identifier of the household (cluster 2 is the reference category). Apart from the number of occupants, all of the remaining independent variables are dummies built by using the available socio-economic information at the household level. Significance levels are denoted as follows: ***p<0.01, **p<0.05, *p<0.1.

Table 6: Multinomial Logit Regression.

Subscribing to the Energy Product 1 contract increases the probability of transitioning from the reference category (cluster 2) to any of the other three clusters.

With regard to house heating, using electricity acts as an attractor for the reference category, as it increases the probability of moving from clusters 1, 3, or 4 to cluster 2. However, the effect of using electricity changes when it is employed for water heating. In this case, it increases the likelihood of transitioning from cluster 2 to any of clusters 1, 3, and 4. Conversely, the use of a heat pump for water heating significantly increases the probability that a household belongs to clusters 3 or 4, relative to cluster 2.

Turning to housing characteristics, residing in a house (as opposed to an apartment) raises the likelihood of being in any of the other three clusters compared to the reference category.

The effect of building age is more nuanced. Living in a newer building significantly increases the probability of belonging to cluster 3 rather than cluster 1, as indicated by the positive and statistically significant coefficient for cluster 3. However, the impact on cluster 2 is less clear-cut.

grid from private systems, such as photovoltaic installations. For systems with a capacity of less than 100 kW, the tariff is generally market-based and may vary depending on the operator and prevailing market conditions.

Although the coefficient for cluster 2 is negative—suggesting that newer buildings may reduce the likelihood of being in cluster 2 compared to cluster 1—the effect is only marginally significant. This indicates that the result should be interpreted with caution. Therefore, while building age has a strong and reliable influence on the likelihood of belonging to cluster 3, its effect on cluster 2 is weaker and less robust.

6.2 Cluster-Based Energy Use Differences

The final part of the analysis explores differences in energy consumption levels across clusters over time through a parametric approach. To manage the high dimensionality of the dataset, regressions are estimated using data from the four representative weeks only. Table 7 reports the results from a set of random effects panel regressions, where the dependent variable is the energy consumption level, $kWh_{i,h}$.

From the first five columns, we find that, relative to the reference category (Cluster 2), all other clusters are associated with higher levels of energy consumption. This effect proves to be robust across the representative weeks, with the exception of the winter period, during which the coefficients remain positive but are not statistically significant. Interestingly, column (5) indicates that the winter week is also associated with the highest overall level of energy consumption throughout the year.

As extensively discussed in the clustering analysis, the clusters differ significantly in terms of the volatility of energy consumption over time. Moreover, we have shown that intra-day volatility represents the main dimension exploited by the SOM procedure to group households. Based on these findings, it is reasonable to expect a positive relationship between energy consumption and its volatility.

To test this hypothesis, we introduce a measure of intra-day volatility in energy consumption, denoted as $Z_{i,h}^{\text{abs}}$, which is defined analogously to $Z_{i,h}$ but takes the absolute value in the numerator (Figure 12):

$$Z_{i,h}^{\text{abs}} = \frac{|dkWh_{i,h} - \mu_i|}{\sigma_i} \tag{7}$$

By ensuring that the distance from the mean is always positive, $Z_{i,h}^{\text{abs}}$ captures how much an observation deviates from the individual mean in terms of standard deviations, regardless of the direction (positive or negative). The following figure replicates Figure 12, this time using $Z_{i,h}^{\text{abs}}$ to emphasize intra-day volatility.

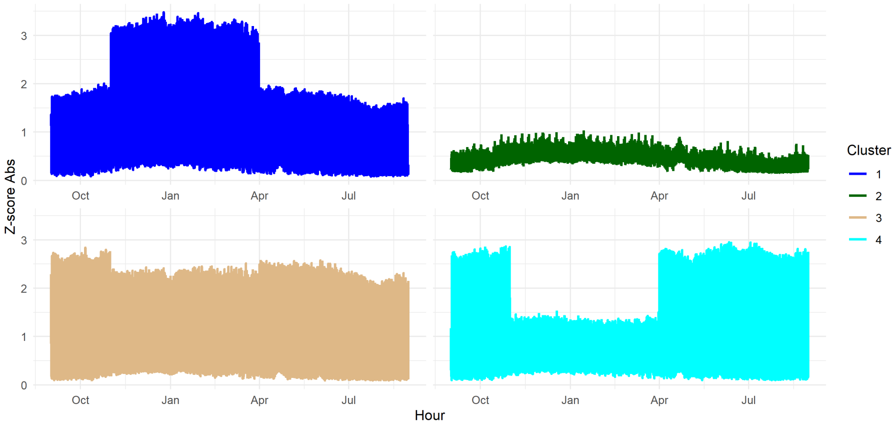


Figure 12: Mean of $Z_{i,h}^{\text{abs}}$ in the four clusters over time.

In line with the observations made for Figure 7, clusters 1, 3, and 4 exhibit higher levels of volatility compared to cluster 2. Moreover, the volatility of energy consumption in clusters 1 and 4 displays a pronounced seasonal pattern: in cluster 1, volatility is elevated between November 2023 and April 2024, whereas in cluster 4, the most volatile periods are September–November 2023 and April–August 2024.

Table 7: Energy consumption levels across clusters and volatility: parametric results

	Dependent variable: hourly									
	Winter	Spring	Summer	Autumn	Gen1	Winter	Spring	Summer	Autum	Gen2
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Cluster 1	0.300***	0.265***	0.200***	0.219***	0.246***	0.061	0.105***	0.078***	0.083***	0.082***
	(0.075)	(0.033)	(0.028)	(0.028)	(0.037)	(0.059)	(0.026)	(0.021)	(0.023)	(0.027)
Cluster 3	0.307***	0.227***	0.147***	0.203***	0.221***	0.118**	0.078***	0.037*	0.080***	0.078***
	(0.074)	(0.032)	(0.028)	(0.028)	(0.036)	(0.058)	(0.025)	(0.021)	(0.022)	(0.027)
Cluster 4	0.166**	0.244***	0.230***	0.225***	0.216***	0.065	0.098***	0.105***	0.089***	0.089***
	(0.083)	(0.036)	(0.031)	(0.031)	(0.040)	(0.065)	(0.029)	(0.024)	(0.025)	(0.030)
Week 15					$-$0.540***					$-$0.523***
					(0.001)					(0.001)
Week 27					$-$0.666***					$-$0.649***
					(0.001)					(0.001)
Week 40					$-$0.633***					$-$0.614***
					(0.001)					(0.001)
Constant	1.050***	0.513***	0.396***	0.425***	1.056***	2.684***	0.596***	$-$0.073**	0.087**	1.270***
	(0.018)	(0.008)	(0.007)	(0.007)	(0.009)	(0.091)	(0.040)	(0.033)	(0.035)	(0.042)
Controls	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes
Observations	1,050,672	1,050,672	1,050,672	1,050,672	4,202,688	1,023,624	1,023,624	1,023,624	1,023,624	4,094,496
F Statistic	33.843***	143.698***	118.948***	148.704***	263,720.500***	3,810.566***	2,858.640***	1,888.264***	1,906.556***	257,611.300***

Notes. This table reports estimates (robust standard errors in parentheses) of random effects panel models performed in the four representative weeks. The dependent variable is $kWh_{i,h}$. Cluster 1, 3, and 4 are dummies that assume a value of 1 in the corresponding cluster and 0 o/w (cluster 2 is the reference category). Spring, Summer, and Autumn are dummies that assume a value of 1 in the corresponding representative week and 0 o/w. Significance levels are denoted as follows: ***p<0.01, **p<0.05, *p<0.1.

7 Conclusion

We present the results of a two-level clustering approach that combines Self-Organizing Maps (SOM) with the K-means algorithm to identify energy consumption profiles and analyze their underlying determinants. Despite the strong ex-ante homogeneity of the households in our dataset — primarily due to the restrictive and conservative selection criteria adopted during data extraction — our analysis successfully identifies four distinct household clusters.

We further characterize these clusters by examining differences in both energy consumption levels and intra-day volatility. In this regard, our findings indicate that, relative to the larger and more homogeneous reference category (cluster 2), households in the remaining three clusters exhibit significantly higher energy consumption levels and greater intra-day consumption volatility. This evidence points to a positive association between volatility and energy use — a relationship that is formally confirmed through parametric analysis controlling for relevant household-level socio-economic factors.

In essence, our data-driven and agnostic clustering approach uncovers a novel behavioral insight that warrants further investigation: volatility in electricity use appears to be a key driver of elevated energy consumption, not merely a correlated outcome. One plausible explanation lies in appliance-induced variability: households exhibiting greater consumption fluctuations are likely operating a wider array of high-power appliances—such as electric ovens, heating systems, or washing machines—whose intermittent usage contributes to both short-term spikes and persistently higher energy demand. Similarly, behavioral complexity in high-consumption households suggests that more intensive electricity use corresponds to a broader range of individual routines and preferences, which generate frequent quarter-hourly variations that accumulate over time.

Moreover, electrified households tend to experience sharper peaks and troughs in demand, particularly when electricity is used for heating, cooking, or vehicle charging—activities that inherently introduce cyclical consumption patterns and reduce efficiency in demand responsiveness. Additionally, automated systems (e.g., smart thermostats or scheduled appliances) may further amplify these short-run variations, reinforcing the structural link between volatility and elevated energy use.

Taken together, these explanations suggest that volatility is not simply a byproduct of consumption behavior, but rather a key determinant of it. As such, future research should explore how demand-side management strategies might address consumption volatility as a pathway to improving energy efficiency and reducing household electricity demand.

References

Abdelaziz, A., Elsayed, N., and Fathy, M. (2024). A hybrid model of self-organizing map and deep learning with genetic algorithm for managing energy consumption in public buildings. *Journal of Cleaner Production*, 434:140040.

Al Khafaf, N., Jalili, M., and Sokolowski, P. (2020). A novel clustering index to find optimal clusters size with application to segmentation of energy consumers. *IEEE transactions on industrial informatics*, 17(1):346–355.

Allcott, H. (2011). Social norms and energy conservation. *Journal of public Economics*, 95(9-10):1082–1095.

Allcott, H. and Kessler, J. B. (2019). The welfare effects of nudges: A case study of energy use social comparisons. *American Economic Journal: Applied Economics*, 11(1):236–276.

Allcott, H. and Rogers, T. (2014). The short-run and long-run effects of behavioral interventions: Experimental evidence from energy conservation. *American Economic Review*, 104(10):3003–3037.

Andor, M. A. and Fels, K. M. (2018). Behavioral economics and energy conservation–a systematic review of non-price interventions and their causal effects. *Ecological economics*, 148:178–210.

Brandon, A., List, J. A., Metcalfe, R. D., Price, M. K., and Rundhammer, F. (2019). Testing for crowd out in social nudges: Evidence from a natural field experiment in the market for electricity. *Proceedings of the National Academy of Sciences*, 116(12):5293–5298.

Chicco, G., Napoli, R., and Piglione, F. (2006). Comparisons among clustering techniques for electricity customer classification. *IEEE Transactions on power systems*, 21(2):933–940.

Costa, D. L. and Kahn, M. E. (2013). Energy conservation "nudges" and environmentalist ideology: Evidence from a randomized residential electricity field experiment. *Journal of the European Economic Association*, 11(3):680–702.

Delmas, M. A. and Lessem, N. (2014). Saving power to conserve your reputation? the effectiveness of private versus public information. *Journal of Environmental Economics and Management*, 67(3):353–370.

Gouveia, J. P. and Seixas, J. (2016). Unraveling electricity consumption profiles in households through clusters: Combining smart meters and door-to-door surveys. *Energy and Buildings*, 116:666–676.

Jones, R. V., Fuertes, A., and Lomas, K. J. (2015). The socio-economic, dwelling and appliance related factors affecting electricity consumption in domestic buildings. *Renewable and Sustainable Energy Reviews*, 43:901–917.

Karaliopoulos, M., Tsolas, L., Koutsopoulos, I., Halkidi, M., Van Hove, S., and Conradie, P. (2022). Beyond clustering: Rethinking the segmentation of energy consumers when nudging them towards energy-saving behavior. *ACM SIGEnergy Energy Informatics Review*, 2(4):28–43.

Karlin, B., Zinger, J. F., and Ford, R. (2015). The effects of feedback on energy conservation: A meta-analysis. *Psychological bulletin*, 141(6):1205.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

Liu, H., Mäkinen, E., and Räsänen, T. (2012). Electricity consumption time series profiling: a data mining application in energy industry. In *Advances in Data Mining. Applications and Theoretical Aspects (ICDM 2012)*, volume 7377 of *Lecture Notes in Computer Science*, pages 214–227. Springer.

Majidi, F. and Smith, J. (2023). A hybrid som and k-means model for time series energy consumption clustering. In *Proc. Nat. Conf. on Data Mining in Engineering and Life Sciences*, pages 57–64.

McLoughlin, F., Duffy, A., and Conlon, M. (2015). A clustering approach to domestic electricity load profile characterisation using smart metering data. *Applied Energy*, 141:190–199.

Mertens, S., Herberz, M., Hahnel, U. J., and Brosch, T. (2022). The effectiveness of nudging: A meta-analysis of choice architecture interventions across behavioral domains. *Proceedings of the National Academy of Sciences*, 119(1):e2107346118.

Michalakopoulos, V., Sarmas, E., Papias, I., Skaloumpakas, P., Marinakis, V., and Doukas, H. (2024). A machine learning-based framework for clustering residential electricity load profiles to enhance demand response programs. *Applied Energy*, 361:122943.

Nisa, C. F., Bélanger, J. J., Schumpe, B. M., and Faller, D. G. (2019). Meta-analysis of randomised controlled trials testing behavioural interventions to promote household action on climate change. *Nature communications*, 10(1):4545.

Percival, D. B. and Walden, A. T. (2000). *Wavelet methods for time series analysis*, volume 4. Cambridge university press.

Posit team (2025). *RStudio: Integrated Development Environment for R.* Posit Software, PBC, Boston, MA.

Räsänen, T. and Ruuskanen, M. (2008). Reducing energy consumption by using self-organizing maps to create more personalized electricity use information. *Applied Energy*, 85(9):830–840.

Schofield, J., Carmichael, R., Tindemans, S., Woolf, M., Bilton, M., and Strbac, G. (2014). Residential consumer responsiveness to time-varying pricing: Report a3 for the "low carbon london" lcnf project.

Shukla, P. R., Skea, J., Slade, R., Al Khourdajie, A., van Diemen, R., McCollum, D., Pathak, M., Some, S., Vyas, P., Fradera, R., et al. (2022). Climate change 2022: Mitigation of climate change. *Contribution of working group III to the sixth assessment report of the Intergovernmental Panel on Climate Change*, 10:9781009157926.

StataCorp (2025). *Stata 18 Base Reference Manual.* College Station, TX. Version 18.5.

Talei, H., Hussein, M. I., and Elkamel, A. (2023). Identifying energy inefficiencies using self-organizing maps: Case of a highly efficient certified office building. *Applied Sciences*, 13(3):1666.

The Core Writing Team (2023). *Climate Change 2023: Synthesis Report.* Intergovernmental Panel on Climate Change (IPCC), first edition.

Tiefenbeck, V., Goette, L., Degen, K., Tasic, V., Fleisch, E., Lalive, R., and Staake, T. (2018). Overcoming salience bias: How real-time feedback fosters resource conservation. *Management science*, 64(3):1458–1476.

Torriti, J. (2012). Price-based demand side management: Assessing the impacts of time-of-use tariffs on residential electricity demand and peak shifting in northern italy. *Energy*, 44(1):576–583.

Van Valkengoed, A. M., Abrahamse, W., and Steg, L. (2022). To select effective interventions for pro-environmental behaviour change, we need to consider determinants of behaviour. *Nature Human Behaviour*, 6(11):1482–1492.

Wynes, S. and Nicholas, K. (2017). The climate mitigation gap: Education and government recommendations miss the most effective individual actions. *Environmental Research Letters*, 12.

Yilmaz, S., Chambers, J., and Patel, M. K. (2019). Comparison of clustering approaches for domestic electricity load profile characterisation-implications for demand side management. *Energy*, 180:665–677.

Appendix

Code

Where not otherwise indicated, the software used for data analysis is R(Posit team, 2025) otherwise is Stata®(StataCorp, 2025).

```
#R Studio
R version 4.4.2 (2024-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 10 x64 (build 19044)

#STATA
StataNow/MP 18.5 for Mac (Apple Silicon)
Revision 26 Feb 2025
Total physical memory: 18.00 GB
version 18.5
. display "`c(os)'"
MacOSX
. display "`c(osdtl)'"
15.3.2
. display "`c(stata_version)'"
18.5
```

Packages

```
# Packages
library(tidyverse); library(dplyr); library(data.table); library(lubridate)
library(sf); library(stringi); library(ggplot2); library(viridis)
library(RColorBrewer); library(scales); library(wavelets)
library(NbClust); library(cluster); library(factoextra); library(kdensity)
library(sm); library(patchwork); library(mice); library(gridExtra); library(moments)
library(plm); library(stargazer); library(kohonen)
```

Dataset Management

```
# ----------------------------------------
# Dataset Management
# ----------------------------------------

# Read CSV file
energy_data <- read_csv("C:/Documents/merged_data2.csv",
                        col_types = cols(timestamp = col_character()))

# Convert timestamp column to POSIXct format for time-based operations
energy_data$timestamp_parsed <- ymd_hms(energy_data$timestamp, tz = "UTC")

# Get the minimum and maximum timestamps in the dataset
min_date <- min(energy_data$timestamp_parsed, na.rm = TRUE)
max_date <- max(energy_data$timestamp_parsed, na.rm = TRUE)

# Print the min and max timestamps
cat("Min date:", format(min_date, "%Y-%m-%d %H:%M:%S"),
    "\nMax date:", format(max_date, "%Y-%m-%d %H:%M:%S"), "\n")

# Check data completeness per ID (each ID should cover the full date range)
complete_series <- energy_data %>%
  group_by(ID) %>%
  summarise(Start = min(timestamp_parsed),
            End = max(timestamp_parsed)) %>%
  filter(Start == min_date & End == max_date)

# Count the number of IDs with complete data
num_complete_series <- nrow(complete_series)
cat("Number of IDs with a complete series:", num_complete_series, "\n")

# Filter dataset to keep only complete series
energy_data <- energy_data %>%
  filter(ID %in% complete_series$ID)
```

```r
# ----------------------------------------
# Variable Creation
# ----------------------------------------

# Aggregate hourly energy consumption
hourly_data <- energy_data %>%
  mutate(hour = floor_date(timestamp_parsed, "hour")) %>%
  group_by(ID, hour) %>%
  summarize(hourly = sum(kwh, na.rm = TRUE), .groups = "drop")

# Compute hourly differences for each ID
data_diff <- hourly_data %>%
  arrange(ID, hour) %>%
  group_by(ID) %>%
  mutate(diff = hourly - lag(hourly)) %>%
  filter(!is.na(diff)) %>%
  ungroup()

# Compute Z-score normalization for differences
data_z_norm <- data_diff %>%
  group_by(ID) %>%
  mutate(z_diff = (diff - mean(diff, na.rm = TRUE)) / sd(diff, na.rm = TRUE)) %>%
  ungroup()

# Compute absolute Z-score
data_z_norm_abs <- data_diff %>%
  group_by(ID) %>%
  mutate(z_diff = abs(diff - mean(diff, na.rm = TRUE)) / sd(diff, na.rm = TRUE)) %>%
  ungroup()

# ----------------------------------------
# Summary Statistics
# ----------------------------------------

#Reference in the Report: Table 1
summary_table <- energy_data %>%
  summarize(
    mean_kwh = mean(kwh, na.rm = TRUE),
    sd_kwh = sd(kwh, na.rm = TRUE),
    skewness_kwh = skewness(kwh, na.rm = TRUE),
    median_kwh = quantile(kwh, 0.50, na.rm = TRUE),
  )

# ----------------------------------------
# Scale Invariance
# ----------------------------------------

#Reference in the Report: Figure 1
hist(hourly_data$hourly,
     main = "Hourly Consumption",
     xlab = "kWh",
     col = "lightblue",
     breaks = 100,
     xlim = c(0, 3),
     freq = FALSE)

plot(log(hourly_data$hourly + abs(min(hourly_data$hourly)) + 1),
     log(seq_along(hourly_data$hourly)),
     main = "Log-Log Plot", xlab = "log(Consumption)", ylab = "log(Rank)",
     col = "blue", pch = 16)

# ----------------------------------------
# Visualization
# ----------------------------------------

#Reference in the Report: Figure 2
ggplot(hourly_avg, aes(x = hour_only, y = mean_hourly)) +
  geom_col(fill = "steelblue2") +
  labs(x = "Hour of the Day", y = "Avg Consumption (kWh)") +
  theme_minimal()

hourly_avg_z <- data_z_norm %>%
  mutate(hour_only = format(hour, "%H")) %>%
  group_by(hour_only) %>%
  summarize(mean_hourly_zdiff = mean(z_diff, na.rm = TRUE))
```

```r
ggplot(hourly_avg_z, aes(x = hour_only, y = mean_hourly_zdiff)) +
  geom_col(fill = "steelblue2") +
  labs(x = "Hour of the Day", y = "Avg Z-Score") +
  theme_minimal() +
  theme_minimal()
```

Wavelet Analysis

```r
# ---- APPLY DISCRETE WAVELET TRANSFORM (DWT) ----
# Compute wavelet coefficients for each individual
wavelet_list <- data_z_norm %>%
  group_by(ID) %>%
  group_split() %>%
  lapply(function(df) {
    # Determine the maximum allowed levels for DWT based on data length
    max_levels <- min(10, floor(log2(nrow(df))))

    # Apply Discrete Wavelet Transform (DWT) using Haar filter
    dwt_res <- dwt(df$z_diff, filter = "haar", n.levels = max_levels)

    # Store wavelet coefficients in a dataframe
    data.frame(ID = df$ID[1], coeffs = unlist(dwt_res@W))
  })

# Combine wavelet coefficient results into a dataframe
wavelet_df <- do.call(rbind, wavelet_list)
length(unique(wavelet_df$ID))

# ---- CONVERT WAVELET COEFFICIENTS INTO MATRIX ----
# Reshape wavelet coefficients for Self-Organizing Map (SOM) analysis
wavelet_df <- wavelet_df %>%
  group_by(ID) %>%
  mutate(variable = paste0("W", row_number())) %>%
  ungroup()

# Create a matrix where rows represent IDs and columns represent wavelet coefficients
coeffs_matrix <- reshape2::acast(wavelet_df, ID ~ variable,
                                 value.var = "coeffs")

rows_with_na <- which(apply(coeffs_matrix, 1, function(x) any(is.na(x))))
print(rows_with_na)

# Store original IDs
coeffs_matrix_clean <- coeffs_matrix[complete.cases(coeffs_matrix), ]
original_IDs <- rownames(coeffs_matrix_clean)
```

SOM

```r
# ---- TRAIN SELF-ORGANIZING MAP (SOM) ----
# Create the SOM Grid
set.seed(123)
som_grid <- somgrid(xdim = 20, ydim=20, topo="hexagonal")
# Train the SOM
som_model <- som(coeffs_matrix_clean,
                 grid=som_grid,
                 rlen=500,
                 alpha=c(0.05,0.01),
                 keep.data = TRUE)

length(som_model$unit.classif)

# Extract SOM reference vectors (codebook)
som_codebook <- som_model$codes
som_codebook <- do.call(rbind, som_model$codes)

# Plot the SOM before clustering
#Reference in the Report: Figure 3
plot(som_model, type = "counts")
plot(som_model, type="changes")
```

Silhouette Analysis

```r
# ---- DETERMINE OPTIMAL NUMBER OF CLUSTERS (ELBOW METHOD) ----

# Compute within-cluster sum of squares (WSS) for different k values (1 to 10)
wss_values <- sapply(1:10, function(k) {
  kmeans(som_codebook, centers = k, nstart = 50)$tot.withinss
})

# Plot the elbow curve to visualize the optimal number of clusters (k)
#Reference in the Report: Figure 4
wss <- plot(1:10, wss_values, type = "b", pch = 19, frame = FALSE,
            xlab = "Number of Clusters (k)",
            ylab = "Total Within-Cluster Sum of Squares",
            main = "Elbow Method for Optimal K")
# Add x-axis labels for clarity
axis(1, at = 1:10, labels = 1:10)

# ---- SILHOUETTE METHOD FOR OPTIMAL K ----

# Compute silhouette scores for k values from 2 to 10
silhouette_scores <- sapply(2:10, function(k) {
  km <- kmeans(som_codebook, centers = k, nstart = 50)  # Perform k-means clustering
  sil <- silhouette(km$cluster, dist(som_codebook))  # Compute silhouette scores

  if (is.matrix(sil)) {
    return(mean(sil[, 3]))  # Return the mean silhouette score if valid
  } else {
    return(NA)  # Return NA if silhouette score computation fails
  }
})

# Plot silhouette scores for different k values
#Reference in the Report: Figure 5
scores <- plot(2:10, silhouette_scores, type = "b", pch = 20,
    xlim = c(2, 10), ylim = c(0, 0.3), xaxt = "n")
# Add x-axis labels for clarity
axis(1, at = 1:10, labels = 1:10)
```

k-means Clustering

```r
# ---- APPLY K-MEANS CLUSTERING TO SOM CODEBOOK ----
# Define the number of clusters
k_optimal <- 4

# Perform k-means clustering on the SOM codebook
kmeans_result <- kmeans(som_codebook, centers = k_optimal, nstart = 50)

# Get the Best Matching Unit (BMU) index for each data point
bmu_index <- som_model$unit.classif

# Create a mapping between ID and BMU
id_som_mapping <- data.frame(ID = original_IDs, BMU = bmu_index)

# Create a dataframe for SOM neuron clusters
neuron_clusters <- data.frame(BMU = 1:nrow(som_codebook),
                              cluster = kmeans_result$cluster)

# Merge ID-BMU mapping with cluster assignments
id_cluster_mapping <- merge(id_som_mapping, neuron_clusters, by = "BMU")

# Merge cluster assignments with the original data
data_with_clusters <- merge(hourly_data_clean, id_cluster_mapping, by = "ID")
datadiff_with_clusters <- merge(data_z_norm, id_cluster_mapping, by = "ID")

# Plot SOM with cluster assignments
#Reference in the Report: Figure 5
plot(som_model, type = "mapping", main = "SOM Clusters",
     bgcol = cluster_colors[kmeans_result$cluster], pch = NA)
add.cluster.boundaries(som_model, kmeans_result$cluster, col = "yellow")

plot(som_model, type = "dist.neighbours", main = "U-Matrix")
```

```
add.cluster.boundaries(som_model, kmeans_result$cluster)
```

Re-clustering

```
# ----------------------------------------
# First Iteration
# ----------------------------------------

cluster_2_ids <- id_cluster_mapping$ID[id_cluster_mapping$cluster == 2]
coeffs_matrix_cluster2 <- coeffs_matrix_clean[rownames(coeffs_matrix_clean)
                                              %in% cluster_2_ids, ]

set.seed(124)
som_grid_cluster2 <- somgrid(xdim = 10, ydim = 10, topo = "hexagonal")

som_model_cluster2 <- som(coeffs_matrix_cluster2,
                          grid = som_grid_cluster2,
                          rlen = 500,
                          alpha = c(0.05, 0.01),
                          keep.data = TRUE)

som_codebook_cluster2 <- do.call(rbind, som_model_cluster2$codes)

# Determines the optimal number of clusters
silhouette_scores2 <- sapply(1:10, function(k) {
  km <- kmeans(som_codebook_cluster2, centers = k, nstart = 50)
  sil <- silhouette(km$cluster, dist(som_codebook_cluster2))

  if (is.matrix(sil)) {
    return(mean(sil[, 3]))
  } else {
    return(NA)
  }
})
scores2 <- plot(1:10, silhouette_scores2,
type = "b", pch = 20, xlim = c(1, 10),
ylim = c(0, 0.7), xaxt = "n")
axis(1, at = 1:10, labels = 1:10)
mean(silhouette_scores2, na.rm = TRUE)

# K-means
k_optimal_cluster2 <- 3
kmeans_result_cluster2 <- kmeans(som_codebook_cluster2,
centers = k_optimal_cluster2, nstart = 100)

cluster_color2 <- c("red", "pink", "yellow")

# New clusters to IDs
bmu_index_cluster2 <- som_model_cluster2$unit.classif

id_som_mapping_cluster2 <- data.frame(ID = rownames
(coeffs_matrix_cluster2), BMU = bmu_index_cluster2)

neuron_clusters_cluster2 <- data.frame(BMU = 1:nrow
(som_codebook_cluster2), cluster = kmeans_result_cluster2$cluster)

id_cluster_mapping_cluster2 <- merge
(id_som_mapping_cluster2, neuron_clusters_cluster2, by = "BMU")

# Merge with dataset
datadiff_with_clusters_cluster2 <- merge
(hourly_data_clean, id_cluster_mapping_cluster2, by = "ID")

# ----------------------------------------
# Second Iteration
# ----------------------------------------

cluster_2_ids_new <- id_cluster_mapping_cluster2$ID
[id_cluster_mapping_cluster2$cluster == 1]

coeffs_matrix_cluster2_new <- coeffs_matrix_cluster2
[rownames(coeffs_matrix_cluster2) %in% cluster_2_ids_new, ]
```

```
set.seed(1245)
som_grid_cluster2_new <- somgrid(xdim = 10, ydim = 10, topo = "hexagonal")

som_model_cluster2_new <- som(coeffs_matrix_cluster2_new,
                              grid = som_grid_cluster2_new,
                              rlen = 500,
                              alpha = c(0.05, 0.01),
                              keep.data = TRUE)

som_codebook_cluster2_new <- do.call(rbind, som_model_cluster2_new$codes)

# Determines the optimal number of clusters
silhouette_scores_new <- sapply(1:10, function(k) {
  km <- kmeans(som_codebook_cluster2_new, centers = k, nstart = 50)
  sil <- silhouette(km$cluster, dist(som_codebook_cluster2_new))

  if (is.matrix(sil)) {
    return(mean(sil[, 3]))
  } else {
    return(NA)
  }
})

# k-means
k_optimal_cluster2_new <- 2
kmeans_result_cluster2_new <- kmeans(som_codebook_cluster2_new,
centers = k_optimal_cluster2_new, nstart = 100)

cluster_color2_new <- c("red", "pink")

# Plot SOM
#Reference in the Report: Figure 11
plot(som_model_cluster2_new, type = "mapping", main = "SOM Clusters",
     bgcol = cluster_color2_new
     [kmeans_result_cluster2_new$cluster], pch = NA)
add.cluster.boundaries(som_model_cluster2_new,
kmeans_result_cluster2_new$cluster, col = "white")

# New clusters to IDs
bmu_index_cluster2_new <- som_model_cluster2_new$unit.classif

id_som_mapping_cluster2_new <- data.frame(ID = rownames
(coeffs_matrix_cluster2_new), BMU = bmu_index_cluster2_new)

neuron_clusters_cluster2_new <- data.frame(BMU = 1:nrow
(som_codebook_cluster2_new), cluster = kmeans_result_cluster2_new$cluster)

id_cluster_mapping_cluster2_new <- merge(id_som_mapping_cluster2_new,
neuron_clusters_cluster2_new, by = "BMU")

## Adjusted Rand Index and INERTIA ##
diff_sse <- abs(kmeans_result$tot.withinss - kmeans_result_cluster2$tot.withinss)
print(diff_sse)

diff_sse2 <- abs(kmeans_result_cluster2$tot.withinss - kmeans_result_cluster2_new$tot.withinss)
print(diff_sse2)
```

Z-score Analysis

```
# ----------------------------------------
# Analysis on Z-Score
# ----------------------------------------

# Compute the average Z-score per cluster and hour
cluster_hourly_mean <- datadiff_with_clusters %>%
  group_by(cluster, hour) %>%
  summarise(mean_diff = mean(z_diff, na.rm = TRUE), .groups = "drop")

# Save a plot of average hourly z-score by cluster
#Reference in the Report: Figure 7
ggplot(cluster_hourly_zmean, aes(x = hour, y = mean_diff, color = factor(cluster))) +
  geom_line(size = 1) +
  labs(title = "",
       x = "Hour",
       y = "Z-score",
       color = "Cluster") +
  theme_minimal() +
  scale_color_manual(values = cluster_colors) +
  scale_x_datetime(date_labels = "%b")+
  facet_wrap(~ cluster, scales = "free_x")+
  theme(strip.text = element_blank())
dev.off()

# Save a plot of average hourly z-score abs by cluster
#Reference in the Report: Figure 12
png("Average_ZABS_by_Cluster.png", width = 3000, height = 1500, res = 300)
ggplot(cluster_hourly_zmean, aes(x = hour, y = mean_diff_abs, color = factor(cluster))) +
  geom_line(size = 1) +
  labs(title = "",
       x = "Hour",
       y = "Z-score Abs",
       color = "Cluster") +
  theme_minimal() +
  scale_color_manual(values = cluster_colors) +
  scale_x_datetime(date_labels = "%b")+
  facet_wrap(~ cluster, scales = "free_x")+
  theme(strip.text = element_blank())
dev.off()
```

Level Analysis

```
cluster_hourly_mean <- data_with_clusters %>%
  group_by(cluster, hour) %>%
  summarise(mean_avg = mean(hourly, na.rm = TRUE), .groups = "drop")

# Add time components for seasonal analysis
cluster_hourly_mean <- cluster_hourly_mean %>%
  mutate(hour = hour(hour),
         week = week(hour),
         year = year(hour),
         month = month(hour),
         weekday = wday(hour),
         weekday_name = weekdays(hour))

# Select specific weeks to represent different seasons
filter_data_by_week <- cluster_hourly_mean %>%
  filter(((week == 3) |  # Winter
          (week == 15) | # Spring
          (week == 27) | # Summer
          (week == 40))) # Autumn

# Save a seasonal volatility analysis plot
#Reference in the Report: Figure 8
ggplot(filtered_data, aes(x = hour, y = mean_avg, color = factor(cluster))) +
  geom_line(size = 1) +
  facet_wrap(~ week + cluster, scales = "free_x") +
  scale_color_manual(values = cluster_colors) +
  scale_x_datetime(
    name = "Day",
    date_breaks = "1 day",
    date_labels = "%d/%m"
  ) +
  labs(
    y = "Average Hourly Consumption",
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(strip.text.x = element_blank(),
        axis.text.x = element_text(angle = 90, hjust = 1))

# Extract data for each season on a specific weekday (Thursday)
winter_data <- filter(cluster_hourly_mean, week == 3 & weekday == 5)
spring_data <- filter(cluster_hourly_mean, week == 15 & weekday == 5)
summer_data <- filter(cluster_hourly_mean, week == 27 & weekday == 5)
autumn_data <- filter(cluster_hourly_mean, week == 40 & weekday == 5)
```

```r
# Function to create seasonal plots
#Reference in the Report: Figure 9
create_season_plot <- function(data, season) {
  ggplot(data, aes(x = hour, y = mean_diff, color = factor(cluster), group = cluster)) +
    geom_line(size = 1) +
    geom_point(size = 2) +
    scale_x_continuous(breaks = seq(0, 23, by = 1)) +
    scale_color_manual(values = cluster_colors) +
    labs(title = paste(season, "(Week", unique(data$week), ", Thursday)"),
         x = "Hour",
         y = "Z-score Average",
         color = "Cluster") +
    theme_minimal() +
    facet_wrap(~ cluster, scales = "free_x")
}

# Create and save seasonal plots
winter_plot <- create_season_plot(winter_data, "Winter")
spring_plot <- create_season_plot(spring_data, "Spring")
summer_plot <- create_season_plot(summer_data, "Summer")
autumn_plot <- create_season_plot(autumn_data, "Autumn")
```

Centroid Analysis

```r
# Total number of variables
total_vars <- 8760

# Levels and corresponding group sizes
divisors <- 2^(1:10)
group_sizes <- floor(total_vars / divisors)

# Initialize the category vector as "Others"
var_category <- rep("Others", total_vars)

# Assign categories using numerical limits as names
start_idx <- 1
for (j in seq_along(group_sizes)) {
  end_idx <- min(start_idx + group_sizes[j] - 1, total_vars)

  # Add leading zero for levels 1-9
  level_label <- sprintf("Level%02d", j)  # Creates "Level01", "Level02", ..., "Level10"

  var_category[start_idx:end_idx] <- level_label
  start_idx <- end_idx + 1
}

# Assign variable names
names(var_category) <- colnames(coeffs_matrix_clean)

# Create a dataframe to store category counts
category_counts <- data.frame()

# Set cluster IDs to use
cluster_names <- c(1, 3, 4)

for (i in seq_along(cluster_names)) {
  cluster_id <- cluster_names[i]  # Use correct cluster ID

  # Get top N variable names for the current cluster
  top_vars_i <- colnames(diff_squared)[order(diff_squared[i, ], decreasing = TRUE)[1:n_top]]

  # Remove NAs
  top_vars_i <- top_vars_i[!is.na(top_vars_i)]

  # Count how many variables of each category are in the top N
  category_dist <- table(var_category[top_vars_i])

  # Add counts to the dataframe
  for (cat in names(category_dist)) {
    category_counts <- rbind(category_counts, data.frame(
      Cluster = paste0("Cluster", cluster_id),
      Category = cat,
      Count = as.numeric(category_dist[cat])
    ))
  }
}

# Ensure all categories are present
all_cats <- c(paste0("Level", sprintf("%02d", 1:10)), "Others")

# Debug print
print(all_cats)  # Check that "Level01", ..., "Level10" are present
print(unique(category_counts$Category))  # Check present categories

# Fill in missing category-cluster combinations with zero counts
for (cat in all_cats) {
  if (!(cat %in% category_counts$Category[category_counts$Cluster == paste0("Cluster", cluster_id)])) {
    category_counts <- rbind(category_counts, data.frame(
      Cluster = paste0("Cluster", cluster_id),
      Category = cat,
      Count = 0
    ))
  }
}

# View first few rows of final result
print(head(category_counts))

# Prepare a long-format dataframe containing squared differences for all clusters
long_diff_df <- data.frame()

cluster_ids <- c(1, 3, 4)
cluster_labels <- paste0("Cluster", cluster_ids)

for (i in seq_along(cluster_ids)) {
  cluster_index <- i  # index in diff_squared matrix
  cluster_name <- cluster_labels[i]

  temp_df <- data.frame(
    Variable = colnames(diff_squared),
    DiffValue = diff_squared[cluster_index, ],
    Level = var_category[colnames(diff_squared)],
    Cluster = cluster_name
  )

  long_diff_df <- rbind(long_diff_df, temp_df)
}

# Ensure the Level column is treated as an ordered factor
long_diff_df$Level <- factor(long_diff_df$Level, levels = paste0("Level", sprintf("%02d", 1:10)))

# Optionally remove "Others" and NA levels
long_diff_df <- long_diff_df[!is.na(long_diff_df$Level) & long_diff_df$Level != "Others", ]

#Reference in the Report: Figure 10
ggplot(long_diff_df, aes(x = Level, y = DiffValue, fill = Level)) +
  geom_boxplot(alpha = 0.6) +
  geom_jitter(color = "black", size = 0.4, alpha = 0.7, width = 0.2) +
  scale_fill_viridis_d() +
  facet_wrap(~ Cluster, ncol = 1) +
  theme_minimal() +
  labs(
    title = "",
    x = "Level",
    y = "Squared Difference"
  ) +
  theme(
    legend.position = "none",
    strip.text = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 14, face = "bold")
  )
```

Multinomial (STATA)

```stata
****************************************************
* Summary Statistics
```

```
**************************************************
summarize customerid
summarize typeofhousehold
summarize numberofoccupants
describe numberofoccupants typeofhousehold
summarize numberofoccupants

**************************************************
* Tabulations
**************************************************
tabulate typeofhousehold
tabulate typeofhousehold numberofoccupants
tabstat numberofoccupants, by(typeofhousehold) stat(sum)
tabstat numberofoccupants, by(typeofhousehold) stat(mean)

**************************************************
* Create Occupants Dummy Variable
**************************************************
gen occupants_dummy = .
replace occupants_dummy = 1 if numberofoccupants == 1
replace occupants_dummy = 2 if numberofoccupants == 2
replace occupants_dummy = 3 if numberofoccupants == 3
replace occupants_dummy = 4 if numberofoccupants == 4
replace occupants_dummy = 5 if numberofoccupants > 5

**************************************************
* Heating Type Analysis
**************************************************
tabulate typeofhousehold typeofheating
contract typeofhousehold typeofheating
tabulate typeofheating
tabulate typeofheating, sort

**************************************************
* Encode Categorical Variables
**************************************************
encode typeofheating, generate(heating)
encode typeofhousehold, generate(household)
encode waterheatingmethod, generate(waterheating)
encode propertyownershipstatus, generate(status)

**************************************************
* Label Categorical Variables
**************************************************
label list
label values heating
tabulate household heating
label values waterheating
tabulate household waterheating
label define status_label 1 "Owner" 2 "Renter"
label values status status_label

tabulate propertyownershipstatus_id typeofhousehold, column
tabulate heating_id household_id, row
tabulate heating_id household_id, column
tabulate waterheating_id household_id, column

**************************************************
* Year of Construction and Building Age
**************************************************
tabulate yearofconstructionofthebuilding, sort

* Clean and convert year
list year if real(year) == .
replace year = "" if year == "NULL"
list year if real(year) == .
destring year, replace

* Generate building age
gen building_age = 2025 - year

* Categorize building age
gen age_dummy = .
replace age_dummy = 1 if building_age < 30
replace age_dummy = 2 if building_age >= 30 & building_age < 40
replace age_dummy = 3 if building_age >= 40 & building_age < 50
replace age_dummy = 4 if building_age >= 50 & building_age < 60
replace age_dummy = 5 if building_age >= 60

label define age_dummy_label ///
    1 "Less than 30" ///
    2 "30-39" ///
    3 "40-49" ///
    4 "50-59" ///
    5 "60+"
label values age_dummy age_dummy_label

****************************************************
* Electricity Product
****************************************************
encode selectedelectricityproduct, gen(product)
list selectedelectricityproduct product
tabulate product_id typeofhousehold, column

****************************************************
* Product Categories (Drop low obs)
****************************************************
bysort product: gen obs_count = _N
drop if obs_count < 100
drop obs_count

gen prod_1 = (str_product == "5")
label variable prod_1 "Energy Product 1"

gen prod_2 = (str_product == "9")
label variable prod_2 "Energy Product 2"

****************************************************
* Heating Grouping
****************************************************
gen heat_group = .
replace heat_group = 1 if inlist
(typeofheating, "heat pump", "heatpumpground",
"heatpumpgroundseparatemeter", "heatpumpseparatemeter")
replace heat_group = 2 if inlist
(typeofheating, "electric", "electricstorage")
replace heat_group = 3 if inlist
(typeofheating, "other", "central heating", "district heating",
"pellets", "woodblocks", "oil", "gas")

label define heat_lbl 1 "Heat pump" 2 "Electric" 3 "Other"
label values heat_group heat_lbl

****************************************************
* Water Heating Grouping
****************************************************
gen waterheat_group = .
replace waterheat_group = 1 if inlist(waterheatingmethod, "heat
pump", "heatpumpground", "heatpumpgroundseparatemeter",
"heatpumpseparatemeter")
replace waterheat_group = 2 if inlist(waterheatingmethod,
"electric")
replace waterheat_group = 3 if inlist(waterheatingmethod, "other",
"central heating", "district heating", "pellets", "woodblocks",
"solar", "solarvacuumtubecollector", "oil", "gas")

label define waterheat_lbl 1 "Heat Pump" 2 "Electric" 3 "Other"
label values waterheat_group waterheat_lbl

****************************************************
* Multinomial Logit Model
****************************************************
#Reference in the Report: Table 4

fvset base 3 heat_group
fvset base 3 waterheat_group

mlogit cluster prod_1 i.heat_group i.waterheat_group
numberofoccupants i.status i.household young, robust
```

Regressions

```
# Note
# merged_data combines the clustering variables from R with the
# metadata (socio-demographic variables) analyzed on STATA.

#Reference in the Report: Table 7
==============================
# Hourly dependent variable
==============================
merged_data <- data_with_clusters %>%
  left_join(metadata, by = c("ID" = "id")) %>%
  filter(week %in% c(3, 15, 27, 40))

week_labels <- as.character(unique(merged_data$week))
column_labels <- c("General 1", "General 2",
                   paste0("Simple ", week_labels),
                   paste0("Complex ", week_labels))

merged_data$week <- as.factor(merged_data$week)
merged_data$cluster <- as.factor(merged_data$cluster)
merged_data$status <- as.factor(merged_data$status)
merged_data$household <- as.factor(merged_data$household)
merged_data$heat_group <- as.factor(merged_data$heat_group)
merged_data$waterheat_group <- as.factor(merged_data$waterheat_group)

merged_data$cluster <- relevel(as.factor(merged_data$cluster), ref = "2")

#With control variables
mod_general1 <- plm(
  hourly ~ cluster + week + numberofoccupants +
    Netz400F + heat_group + waterheat_group +
    status + household + young,
  data = merged_data,
  index = c("ID", "hour"),
  model = "random",
  random.method = "walhus"
)

# Fit weekly models (same as general1 but estimated separately by week)
mods1 <- list()
column_labels <- c("General")

mods1[[1]] <- mod_general1

weeks_to_include <- c(3, 15, 27, 40)
i <- 2

for (w in weeks_to_include) {
  data_week <- subset(merged_data, week == w)

  mod_week <- plm(
    hourly ~ cluster + numberofoccupants +
      Netz400F + heat_group + waterheat_group +
      status + household + young,
    data = data_week,
    index = c("ID", "hour"),
    model = "random",
    random.method = "walhus"
  )

  mods1[[i]] <- mod_week
  column_labels[i] <- paste0("Week ", w)
  i <- i + 1
}

#Without control variables
# --- GENERAL MODEL 1: Hourly (without z_diff_abs) ---
mod_generalNO <- plm(
  hourly ~ cluster + week,
  data = merged_data,
  index = c("ID", "hour"),
  model = "random",
  random.method = "walhus"
)
```

```r
# Fit weekly models (same as general1 but estimated separately by week)
modsNO <- list()
column_labels <- c("General")

# Aggiungi modello generale
modsNO[[1]] <- mod_generalNO

# Aggiungi modelli settimanali
i <- 2
for (w in c(3, 15, 27, 40)) {
  data_week <- subset(merged_data, week == w)

  mod_week <- plm(
    hourly ~ cluster,
    data = data_week,
    index = c("ID", "hour"),
    model = "random",
    random.method = "walhus"
  )

  modsNO[[i]] <- mod_week
  column_labels[i] <- paste0("Week ", w)
  i <- i + 1
}
```