Explainable Clustering of Building-Energy Time Series: From Traditional Methods to Deep Representation Learning

Sarra Kallel

A Thesis

in

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

December 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Sarra Kallel**

Entitled: **Explainable Clustering of Building-Energy Time Series: From Traditional Methods to Deep Representation Learning**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair and Examiner
Dr. Abdessamad Ben Hamza

_____ Examiner
Dr. Honghao Fu

_____ Supervisor
Dr. Nizar Bouguila

_____ Supervisor
Dr. Manar Amayri

Approved by _____
Chun Wang, Chair
Department of Concordia Institute for Information Systems Engineering

_____ 2025

_____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Explainable Clustering of Building-Energy Time Series: From Traditional Methods to Deep Representation Learning

Sarra Kallel

The increasing deployment of smart meters and IoT sensing infrastructures has produced large volumes of high-resolution building-energy data, offering new opportunities for understanding consumption behavior, improving operational efficiency, and supporting energy planning. Extracting meaningful structure from this data remains challenging due to its high dimensionality, nonlinear temporal patterns, and the absence of labels. Existing clustering approaches often struggle with instability, sensitivity to preprocessing choices and distance metrics, and inconsistencies across internal validation indices, limiting their ability to reliably characterize underlying consumption behavior patterns. Moreover, most clustering studies treat the process as a black box, providing little insight into why specific profiles are grouped together, which restricts their usefulness for practitioners and decision-makers.

To address these limitations, this research develops a comprehensive analytical framework that combines traditional clustering methods, deep time-series representation learning, and explainable artificial intelligence to analyze building-energy load profiles. Cluster quality across all analyses is assessed using five internal validation indices. First, we evaluate K-Means, K-Medoids, Fuzzy C-Means, and Gaussian Mixture Models, both with and without dimensionality reduction, examining their robustness under variations in intra- and inter-cluster characteristics, including outliers, overlapping profiles, density shifts, skewness, kurtosis, and sub-clustering. Multiple techniques are used to estimate the optimal number of clusters for each method. Decision-tree–based explanation models, specifically axis-aligned and sparse oblique, are applied to produce human-explainable rules linking profile features to cluster assignments. Second, we develop a deep time-series clustering

pipeline across seven encoder architectures, five representation-learning losses, and seven clustering losses on both univariate and multivariate building-energy datasets, tested with two different cluster configurations. To determine whether the additional complexity of deep clustering is justified, we compare its performance against four traditional clustering algorithms. To overcome the sensitivity of deep clustering to hyperparameters, we integrate Population-Based Training as an evolutionary optimization strategy. Explainability is incorporated through prototype–criticism analysis, providing representative and atypical profiles that summarize each cluster's internal structure.

The results show that dimensionality reduction has minimal impact on overall clustering quality but can enhance separability in overlapping or variable-density settings. The explainability analysis further revealed a trade-off between completeness and simplicity: axis-aligned trees achieved full cluster coverage at the cost of greater rule complexity, whereas sparse oblique trees produced simpler rules but occasionally failed to cover specific clusters. Overall, deep clustering methods consistently capture more coherent and better-separated patterns than traditional algorithms, while the proposed interpretability modules offer clear and actionable explanations of consumption behavior. Together, these contributions provide a scalable, unsupervised, and transparent approach for transforming raw building-energy time series into meaningful behavioral archetypes, enabling improved energy management, personalized feedback, and data-driven planning.

Acknowledgments

I would like to sincerely thank my supervisors, Dr. Manar Ayamri, and Dr. Nizar Bouguila, for their guidance, support, and invaluable insights throughout this work.

My deepest gratitude goes to my parents for their unwavering love and encouragement.

Finally, I thank my husband for his constant support, patience, and belief in me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The way humanity consumes energy has become one of the most defining patterns of the modern era. As cities expand, industries grow, and lifestyles become increasingly dependent on technology, global electricity demand continues to surge at an unprecedented rate [1]. What was once a predictable and centralized energy system has evolved into a dynamic, data-rich network of consumers, buildings, and distributed resources [2]. Within this transformation, the buildings sector has emerged as one of the most energy-intensive domains responsible for roughly 34 % of total global energy demand in 2022 and nearly 37 % of energy-related CO_2 emissions [3]. Recent data show that in 2024, electricity use in buildings expanded nearly four times faster than the previous year, reflecting the combined impact of rising cooling needs, digitalization, and the electrification of heating systems [1]. Because buildings directly influence both operational energy use and indirect emissions, improving their efficiency has become one of the most effective strategies for reducing greenhouse gases and mitigating climate change.

The way we consume energy carries profound environmental consequences. Electricity generation remains dominated by fossil fuels which still supply around 60 % of global electricity [1]. Their combustion releases large quantities of carbon dioxide and other greenhouse gases, intensifying global warming and extreme weather events. Beyond climate impact, the energy system contributes to resource depletion through extraction [4], air pollution from particulate and nitrogen-oxide emissions [5], and significant freshwater use for cooling in thermal and nuclear power plants [6]. These interconnected pressures underscore that the energy challenge is not only technical but

ecological, linking data analytics and system efficiency directly to planetary health.

Amid these challenges, the rise of digital technologies has created new opportunities for insight. Smart meters, embedded sensors, and IoT-based monitoring systems now record electricity, heating, and cooling data at fine temporal resolutions, transforming the built environment into a continuously measured ecosystem. These high-resolution data streams capture the hidden dynamics of daily consumption like the morning peaks, the nighttime drops, and the irregular fluctuations shaped by behavior, climate, and building design [7, 8]. When analyzed effectively, they reveal patterns that can guide personalized efficiency programs, adaptive control systems, and large-scale policy interventions [9–11]. In this context, data has become not merely a record of past usage but a strategic resource for shaping future sustainability.

Clustering has emerged as one of the most powerful analytical tools for uncovering these hidden structures. By grouping similar load profiles, it transforms vast, unlabeled datasets into organized representations of consumption behavior. Clusters can reveal distinct user archetypes, operational regimes, or temporal patterns, insights that are indispensable for planning, optimization, and policy design [12–14]. The benefits of such analysis ripple across all layers of the energy ecosystem. At the individual level, clustering enables households to understand their own consumption behavior relative to peers, fostering awareness and empowering users to adopt targeted energy-saving measures [15, 16]. For facility and building managers, it highlights inefficiencies, detects anomalies, and identifies characteristic operational modes that inform predictive maintenance and adaptive control strategies [17]. For energy suppliers and grid operators, clustering supports accurate load forecasting, dynamic pricing, and demand-response coordination, ensuring system stability and efficient resource allocation [18]. And for governments and regulators, aggregated cluster insights form the basis of data-driven policies, fair tariff structures, and evidence-based sustainability planning. Clustering bridges profile-level and system-wide views , turning raw data into actionable knowledge that benefits every stakeholder in the energy chain.

The rapid growth of data analytics and artificial intelligence now makes it possible to move beyond descriptive grouping toward adaptive and explainable understanding. Advances in machine learning, deep learning, and explainable AI (XAI) have enabled the transformation of complex energy data into structured and transparent knowledge. The convergence of these fields signals a

new paradigm, one in which data-driven models can uncover latent consumption regimes, support predictive decision-making, and remain explainable to human experts. In an age where algorithms increasingly shape the operation of cities and buildings, making sense of the invisible rhythms of energy is not only a technical challenge but a moral imperative. Understanding how and why energy is consumed is the foundation upon which sustainable, intelligent, and equitable energy systems will be built.

1.1 Problem Statement

Despite the unprecedented availability of high-resolution energy data and the growing urgency to understand the complex dynamics of building consumption, current analytical practices remain insufficient for the scale and sophistication of modern energy systems. Daily load profiles are inherently high-dimensional, noisy, behavior-driven, and strongly modulated by climate, equipment, and operational routines. Extracting meaningful structure from such data is extremely challenging. Traditional analytical methods cannot capture the temporal structures, nonlinear relationships, and overlapping behavioral regimes embedded in these complex load profiles. Even when clustering is applied, existing approaches often produce unstable results that vary with scaling choices, initialization, hyperparameters, or the number of clusters. Internal validation indices frequently disagree or behave inconsistently, offering little guidance on model selection. Moreover, because most methods operate as black boxes, stakeholders are left without clear explanations of why buildings fall into one group rather than another, limiting trust, explainability, and practical adoption.

At the same time, the rise of deep learning and representation learning has opened new possibilities for modeling temporal patterns directly from raw data. However, deep time-series clustering remains an emerging field with open challenges. Current techniques suffer from sensitivity to architecture choices, difficulty tuning complex losses, lack of stability across datasets, and limited understanding of how different pretext tasks or encoder types affect clustering quality. Most critically, the field lacks a unified, systematic, and transparent framework that can compare traditional clustering with deep clustering, evaluate performance across multiple internal indices, and explain the discovered behavioral archetypes in a way that is meaningful to practitioners, energy planners,

and policymakers.

Therefore, the central problem is that although high-resolution building-energy datasets contain rich information about consumption behavior, there is still no robust, scalable, and explainable analytical framework capable of reliably extracting, validating, and explaining the latent consumption regimes they contain. This methodological gap prevents the energy community from turning abundant data into actionable insights

1.2 Contributions

The main contributions of this research are summarized as follows:

- We provide a comprehensive and explainable framework for clustering residential electricity demand profiles by systematically evaluating a wide set of clustering algorithms and integrating post-clustering explainability. We compare hard clustering methods (K-Means, K-Medoids) to soft clustering techniques (Fuzzy C-Means and Gaussian Mixture Models), and we assess their performance using five Cluster Validity Indices under varying data characteristics, including outliers, overlapping profiles, differential density, skewness, kurtosis, and sub-clustering. Unlike previous studies that treat clustering as a black box, we incorporate explainability through axis-aligned and sparse oblique decision trees to generate transparent, human-readable rules that explain why each household is assigned to a specific cluster. We additionally investigate the effect of dimensionality reduction on clustering performance using PCA and provide a feature-mapping strategy to link reduced dimensions back to original time intervals for interpretability. Overall, this research offers an end-to-end methodology that unifies clustering quality assessment with explainability, enabling clearer understanding of electricity consumption behaviors in smart-meter data and supporting more informed decision-making in energy planning and management.

This research was published in *Sensors* as part of the Special Issue *Intelligent Sensors and Artificial Intelligence in Building* under the title "Clustering and Interpretability of Residential Electricity Demand Profiles" [19].

- We investigate deep time-series clustering for building-energy by developing an end-to-end unsupervised framework that jointly learns latent features and cluster assignments from raw energy data. We evaluate seven deep architectures, five representation-learning losses, and seven clustering losses on two real-world datasets, univariate and multivariate. We benchmark these deep pipelines against traditional clustering baselines and assess model quality using five internal evaluation indices. To address the high sensitivity of deep clustering to hyperparameters, we apply Population-Based Training (PBT) [20] as an evolutionary optimization strategy to tune the top-performing models for each dataset. Finally, to overcome the black-box nature of deep learning, we integrate explainable AI through prototype–criticism analysis, highlighting representative and atypical consumption profiles for each cluster. Overall, this research demonstrates that deep clustering pipelines outperform traditional methods and provide explainable, operationally meaningful insights into building-energy consumption patterns.

This research was submitted to *Sustainable Cities and Society: Advances* entitled "Explainable Deep Representation Learning for Clustering Building-Energy Time Series" [21].

1.3 Thesis Overview

The rest of the thesis is structured as follows:

- **Chapter 2** presents the literature review supporting this work. It covers time-series clustering taxonomies and core algorithmic families, prior research on clustering energy data, key concepts in explainable artificial intelligence, and the internal metrics used to evaluate clustering quality.

- **Chapter 3** introduces the first empirical contribution, focusing on the clustering and explainability of residential electricity demand profiles. The chapter details the methodology, including preprocessing, dimensionality reduction, clustering algorithms, and explainability analysis, followed by a comprehensive discussion of results.

- **Chapter 4** presents the second contribution, developing a deep time-series clustering framework that integrates representation learning, hyperparameter optimization, and explainable AI. It outlines the datasets, pipeline design, baseline methods, evaluation criteria, Population-Based Training configuration, and final results.

- **Chapter 5** concludes the thesis by summarizing the main findings and contributions.

Chapter 2

Litterature Review

To situate our work within existing research, we begin by reviewing the main families of time-series clustering methods and how different approaches represent temporal data. We then examine how clustering has been applied in the energy domain, highlighting insights specific to building load analysis. Because understanding cluster assignments is as important as discovering them, we also introduce key concepts from Explainable AI that support interpretability in this context. Finally, we summarize the internal validation metrics most commonly used to assess clustering quality in unlabeled settings.

2.1 Time-Series Clustering: Taxonomies and Core Families

Different taxonomies have been proposed in the literature for classifying time-series clustering methods. For instance, [22] distinguishes between shape-based, feature-based, and model-based approaches. Similarly, another framework [23] organizes methods into three broad categories depending on whether they operate directly on raw data, indirectly on features extracted from the raw data, or indirectly on models learned from the raw data. A simpler classification is provided in [24], which reduces these to just two categories: raw data–based methods and feature-based methods.

More recently, a more general taxonomy has been proposed in [25], which groups time-series clustering methods into four major categories: distance-based, distribution-based, subsequence-based, and representation learning–based methods. Each of these categories can be further divided

into subgroups depending on the specific techniques employed. In this review, we consider two of these categories distance-based, and distribution-based.

Distance-based methods

Distance-based methods cluster time series by defining a dissimilarity measure between two sequences. Similarity reflects how close two series are to each other according to a chosen metric, which returns a distance value for each pair of objects [26]. The quality of clustering results is therefore highly dependent on the choice of dissimilarity measure, making this a critical design decision. Among the most widely used measures are Euclidean distance and Dynamic Time Warping (DTW), both of which remain the standard baselines in time-series clustering. Although numerous alternative similarity measures have been proposed, many studies emphasize that clustering research often contributes more through the introduction of new distance functions than through fundamentally new clustering algorithms [27]. However, a key limitation is that many of these proposed measures are not systematically benchmarked against strong baselines. For example, in a comparative study of eleven similarity measures against Euclidean distance and DTW using two datasets, results confirmed that none of the alternatives consistently outperformed the baselines, and several performed substantially worse [27]. Even so, Euclidean distance, despite performing relatively well overall, is highly sensitive to scaling, shifting, and misalignments, meaning that a simple stretch or temporal shift can produce a disproportionately large distance in Euclidean space [28]. To mitigate temporal misalignment, DTW is commonly adopted. However, directly integrating DTW within K-means clustering can be problematic. In this context, [29] demonstrated that applying DTW within K-means can lead to failures due to issues with DTW-based averaging, while combining K-medoids with DTW remains more stable and produces meaningful clustering of multimedia and energy-related time series.

Once pairwise distances are computed, clustering can be performed using either partitional approaches such as K-means, K-medoids, K-Shape or Fuzzy C-Means or hierarchical approaches, which may be agglomerative or divisive depending on whether clusters are merged or split. However, hierarchical clustering typically suffers from quadratic computational complexity, making it less suitable for large-scale time-series datasets [22]. These distance-based algorithms form the

foundation for both traditional and modern clustering paradigms, as they define how similarity between time-series profiles is quantified.

K-means is one of the most popular and widely adopted clustering algorithms. It is known as a hard clustering algorithm [30], meaning it assigns each data point exclusively to one cluster without any shared membership across clusters. The algorithm partitions data into distinct groups by minimizing the sum of squared distances between data points and their respective cluster centroids. However, K-means is sensitive to outliers because extreme values heavily influence the mean, leading to suboptimal clustering results [31]. To overcome this limitation, the K-medoids algorithm was introduced as another hard clustering approach. Unlike K-means, it defines cluster centers (medoids) as actual data points instead of the mean, which makes it less sensitive to extreme values. Each data point is assigned to the nearest medoid based on a specified distance metric [32], and using medoids instead of means substantially reduces the impact of outliers on the clustering structure [33].

While hard clustering methods are computationally efficient, they may fail to capture overlapping or ambiguous cluster boundaries, especially in datasets with gradual transitions or noisy behavior. To address this issue, soft clustering [30] techniques have been introduced, allowing each data point to belong to multiple clusters with varying probabilities. The Fuzzy C-Means (FCM) algorithm [34] is one of the most widely used soft clustering approaches. It assigns each data point a degree of membership to different clusters based on the distance between the point and the cluster centers. Points closer to a centroid receive higher membership values, whereas those farther away are assigned lower ones. This probabilistic framework allows FCM to better handle overlapping or transitional patterns common in energy time-series data.

Distribution-based methods

Unlike distance-based approaches, distribution-based methods group time series by modeling their underlying statistical properties rather than relying on explicit dissimilarity measures. One approach is to fit probabilistic models and then cluster series based on their parameter estimates. The Gaussian Mixture Model (GMM) [35] is a prominent example of this family. GMM assumes that data are generated from a mixture of Gaussian distributions and estimates the probability that

each point belongs to each cluster component, thus providing a probabilistic interpretation of cluster membership that captures uncertainty and continuous transitions more effectively than hard partitioning. As a result, GMM is considered a soft clustering algorithm, since each data point can belong to multiple clusters with different probabilities rather than being assigned to a single group.This model-based formulation allows GMM to represent clusters of different shapes and densities, governed by the estimated mean and covariance of each Gaussian component. Another approach is to detect dense regions directly in the data space. Among these, DBSCAN [36] is widely used because it does not require the number of clusters to be specified in advance, can identify clusters of arbitrary shape, and remains robust in the presence of noise or outliers. OPTICS [37] extends DBSCAN by recovering clusters across multiple density levels, making results less dependent on parameter tuning.

Overall, distribution-based methods are effective for capturing complex or non-linear cluster structures, particularly when data are noisy or large-scale. Their performance, however, depends on how well the chosen model or representation reflects the true properties of the series.

2.2 Clustering Energy Data

Clustering has been deployed across energy to benchmark buildings, discover usage patterns, and derive archetypes that can inform decision-making . Tien et al. [38] provide a comprehensive review of machine learning and deep learning applications in the built environment, focusing on building energy efficiency and indoor environmental quality. The paper surveys methods for energy forecasting and management, HVAC optimization, occupancy detection, fault detection and diagnosis, and comfort and air quality prediction. It highlights how AI techniques are increasingly employed to improve energy performance without compromising occupant comfort, but also notes that most studies remain at the experimental stage with limited deployment in real buildings. Importantly, the review situates clustering and other unsupervised methods as key tools for load profiling, anomaly detection, and occupancy inference, linking them directly to energy efficiency strategies.

For building benchmarking, Gao and Malkawi [39] propose an “intelligent clustering” workflow that groups buildings by multi-dimensional attributes (area, schedules, glazing, climate, etc.) and

then benchmarks members against the cluster centroid which they called ”pole”, explicitly arguing that proximity in this feature space yields fairer peer comparisons than single-feature typologies. They motivate clustering as the core step before benchmarking and implement the pole idea operationally for Energy Use Intensity comparisons. In a more recent benchmarking system, BEEM (Singapore) [40], clustering is used at two stages in the pipeline: first to form peer groups, and then to map continuous performance predictions to human-readable letter grades via univariate clustering. The system couples CatBoost for predictive accuracy with LIME for explainability, and reports significant error reductions over linear baselines.

Beyond benchmarking, clustering is a key method for finding daily and seasonal load shapes. Iglesias and Kastner [41] conduct a study of similarity measures for time-series clustering of building electricity use, comparing Euclidean, Mahalanobis, Pearson-correlation, and DTW within fuzzy c-means, and propose “clustered-vector balance” as a validation tool to judge representativeness of discovered patterns highlighting how the distance choice changes the clustering results. At the distribution level, Damayanti et al. [42] cluster feeder-level daily load profiles with k-means, fuzzy c-means, and k-harmonic means, select the number of clusters with the Davies–Bouldin Index, and derived load and loss factors useful for grid planning, with k-harmonic means yielding the most reliable clusters.

At the residential segment, Toussaint and Moodley [43] argue that internal indices alone are insufficient for constructing useful customer archetypes. They extend validation by embedding expert-driven criteria, ensuring the resulting archetypes are both representative and operationally meaningful. The authors develop a library of representative daily load profiles that form the basis for South African household archetypes. Finally, extending beyond single-carrier electricity contexts, Guo et al. [44] present a three-stage adaptive pattern-mining framework for multi-energy loads. Their approach first segments load sequences using an Autoplait-inspired method, then applies clustering and motif selection to extract recurring consumption patterns across multiple carriers. Results show that this adaptive pipeline captures the dynamics of multi-energy demand more effectively than clustering raw time series.

2.3 Explainable Artificial Intelligence

Early AI systems were relatively transparent and easy to interpret, whereas recent years have seen the emergence of opaque decision-making systems, notably deep neural networks [45]. As these systems increasingly make decisions that were previously entrusted to humans, they must be able to explain themselves [46]. Practitioners remain hesitant to deploy models that lack interpretability, operational tractability, and reliability. In energy and building analytics, stakeholders want to know why a building, a time period, or a customer was assigned to a given cluster, which features drove that assignment, and how a different operating profile would change the outcome. Understanding how these models make decisions is therefore crucial.

The literature distinguishes explainability from interpretability. We adopt the following working definition of explainability: "Given an audience, an explainable AI system produces details or reasons that make its functioning clear or easy to understand" [45]. Interpretability refers to transparency that is intrinsic to the model class, where a human can follow the computation without auxiliary tools, as in sparse linear models, small decision trees, or prototype- and rule-based models [45, 47]. In this review we use explainability as the broader umbrella while acknowledging that interpretability by design is desirable when it does not compromise performance.

Studies such as [42] and [48] applied clustering to energy consumption data without incorporating post-clustering interpretability techniques, leaving energy analysts with limited insight into why clusters were formed. A recent survey on interpretable clustering [49] categorizes explainable clustering approaches into pre-clustering, in-clustering, and post-clustering techniques, reviewing how existing methods incorporate interpretability at different stages. However, this survey remains theoretical and does not provide an empirical comparison of clustering algorithms, limiting its ability to assess the practical effectiveness of explainability methods. While some studies attempted to integrate explainability using decision trees, such as [50], these efforts were often restricted to a single decision tree model, without evaluating how different tree structures impact interpretability. [51] introduced an optimization-driven tree-based clustering model, but it did not compare Axis-Aligned vs. Sparse Oblique Decision Trees, leaving a gap in understanding the trade-offs between

full cluster coverage and rule simplicity. While some prior studies have investigated explainability in clustering [49], [50], [51], none have systematically compared multiple clustering methods on electricity consumption data while simultaneously evaluating different decision tree models for post-clustering interpretability.

XAI techniques can be applied at different stages, to different model classes, and to explain different targets. In this literature review, we organize the space along three axes: model scope, timing, and explanation approach. Under model scope, methods are either model specific or model agnostic. Model-specific techniques leverage details of a particular architecture or algorithm, while model-agnostic techniques treat the predictor as a black box and can be applied to any machine learning model. Timing distinguishes post hoc from built-in explanations: post hoc methods are applied after training and do not alter the learned model, while built-in approaches arise from design choices that make the model transparent by default. The explanation approach describes the form of the explanation. Local explanations account for a single prediction or a single cluster assignment, while global explanations summarize overall model behavior or cluster semantics. Other widely used approaches include contrastive explanations that focus on why one outcome rather than another, counterfactuals that ask what changes would move a point to a different cluster, rule-based summaries, saliency-based attributions, and prototype-based explanations that reason with representative examples. We adopt this taxonomy in what follows. Table 2.1 compiles the most frequently cited methods under each category with representative references.

Table 2.1: Comparison of Explainable AI methods.

Method	Type	Explanation	Question it answers	Research Works
Saliency map	Post hoc, local, model-agnostic	A saliency map is a visualization that tells you which parts of the input most strongly affect the model's output for a specific prediction.	Which input features most influence this model output, and how sensitive is the output to changes in each feature?	[52] [53]
Class Activation Map (CAM)	Post hoc, local, model-specific (CNN with Global Average Pooling)	Highlights the regions of the input that contributed the most to a model's prediction for a specific class.	Where in the input did the network focus to make this particular class prediction?	[54]
Gradient-weighted Class Activation Mapping (Grad-CAM)	Post hoc, local, model-specific (any conv-layer model)	Uses the gradients of the output w.r.t. feature maps to produce a heatmap showing which regions most strongly support the chosen class prediction.	Which regions of the input had the strongest positive influence on the model's prediction for this specific class?	[55]
Local Interpretable Model-agnostic Explanation (LIME)	Post hoc, local, model agnostic	Explains why a complex model made one prediction by building a simple surrogate model around that example.	If I slightly remove or change each feature, how does the model's prediction locally change?	[56]
Shapley Additive Explanations (SHAP)	Post hoc, local or global, model-agnostic	Uses Shapley values to fairly distribute a prediction across features, showing each feature's contribution from a baseline to the actual result.	How much did each input feature contribute positively or negatively to this specific model prediction compared to a baseline output?	[57]
Anchor explanations	Post hoc, local, model-agnostic, rule-based	Finds if–then rules ("anchors") that guarantee the model keeps making the same prediction under small input changes.	What minimal set of conditions guarantees (with high probability) that the model would still make this same prediction?	[58]
Local Rule-based Explanations (LORE)	Post hoc, local, model-agnostic, rule- and counterfactual-based	Learns an interpretable rule around the instance and provides counterfactual examples to flip the prediction.	Which rule-like conditions made the model choose this prediction, and what minimal changes would switch the prediction to another class?	[59]
Counterfactuals	Post hoc, local, model-agnostic	Finds a minimal, realistic change to the input that yields a different output.	What would need to be different about this input for it to be assigned to a different outcome?	[60]

Continued on next page

Table 2.1 continued from previous page

Method	Type	Explanation	Question it answers	Research Works
CEM: Contrastive Explanation Method	Post hoc, local, typically model-specific, contrastive	Finds pertinent positives (minimally sufficient features) and pertinent negatives (smallest changes that flip the prediction).	What features were minimally sufficient to justify this prediction, and what small changes would have flipped it?	[61]
CLUE	Post hoc, local, model-specific to encoder–decoder pipelines	Searches for the nearest point in latent space where the model is confident about a different prediction, then decodes it back to input space.	What is the closest, most realistic alternative version of this input that would make the model confidently predict a different outcome?	[62]
Prototypes and criticisms	Post hoc, global, model-agnostic, example-based	Summarizes model behavior by selecting prototypes (representative examples) and criticisms (unusual or poorly represented examples).	Which examples best summarize this class/cluster/model behavior, and which are least typical and deserve extra attention?	[63]

2.4 Internal Metrics

- Silhouette Score (SH): This index evaluates clustering quality by contrasting intra-cluster cohesion and inter-cluster separation. Scores range from -1 to 1, with higher values indicating more well-defined clusters [64]. For each sample i, the silhouette coefficient is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}}, \tag{1}$$

where $a(i)$ is the average distance between i and all other points in its assigned cluster (cohesion), and $b(i)$ is the minimum, over all other clusters, of the average distance between i and the points in that cluster (separation). The overall silhouette score is then:

$$\mathrm{SH} = \frac{1}{n} \sum_{i=1}^{n} s(i). \tag{2}$$

The silhouette value thus compares how close i is to its own cluster against the best alternative cluster.

- Calinski-Harabasz (CH) Index: Also known as the Variance Ratio Criterion, this index evaluates the ratio of between-cluster dispersion to within-cluster dispersion.

Formally, the CH score is computed as follows:

$$\mathrm{CH} = \frac{\sum_{i=1}^{k} n_i \left\| m_i - m \right\|^2}{k-1} \Bigg/ \frac{\sum_{i=1}^{k} \sum_{j=1}^{n_i} \left\| d_{ij} - m_i \right\|^2}{n-k}. \tag{3}$$

In this expression, n_i denotes the number of data points assigned to cluster i, and m_i is the centroid (mean vector) of that cluster. The vector m represents the overall mean of all n observations in the dataset, while k is the total number of clusters. The term d_{ij} refers to the j-th data point in cluster i, and its contribution to the denominator is given by its distance to the corresponding cluster centroid m_i. Each cluster centroid m_i is computed as the arithmetic

mean of the data points belonging to that cluster:

$$m_i = \frac{1}{n_i} \sum_{j=1}^{n_i} p_{ij}, \tag{4}$$

where p_{ij} denotes the j-th observation assigned to cluster i. Higher scores suggest compact, well-separated clusters [65].

- Davies-Bouldin (DB) Index: This index assesses clustering quality based on the average similarity ratio between each cluster and its most similar counterpart. Formally, it is calculated as:

$$\mathrm{DB} = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \frac{D_i + D_j}{\|m_i - m_j\|}, \tag{5}$$

where D_i denotes the diameter (or within-cluster scatter) of cluster i, and m_i is its centroid. The numerator $D_i + D_j$ captures the combined dispersion of clusters i and j, while the denominator $\|m_i - m_j\|$ measures the distance between their centroids, reflecting how well the two clusters are separated. As a result, the Davies–Bouldin index jointly encodes cluster compactness and separation. Lower DB values indicate better-defined clusters with greater separation [66].

- Dunn's (DI) Index: This index identifies compact and well-separated clusters by maximizing the minimum inter-cluster distance and minimizing the maximum intra-cluster distance. The Dunn Index (DI) is defined as:

$$\mathrm{DI} = \frac{\min\limits_{1 \leq i < j \leq k} \delta(m_i, m_j)}{\max\limits_{1 \leq q \leq k} D_q}, \tag{6}$$

where $\delta(\cdot, \cdot)$ denotes the distance measure used to quantify dissimilarity between cluster representatives (typically the centroids m_i and m_j of clusters i and j), and D_q is the diameter of cluster q. The numerator captures the smallest separation between any pair of clusters, while the denominator reflects the largest within-cluster dispersion across all clusters. As a consequence, higher values indicate superior clustering performance [67].

- Xie-Beni (XB) Index: Commonly used in fuzzy clustering, this index compares within-cluster

scatter to between-cluster separation. Lower XB values reflect better clustering quality [68]. Formally, it is computed in the following way:

$$\mathrm{XB} = \frac{\sum_i \sum_{x \in K_i} \delta^2(x, m_i)}{n \, \min_{i,j:\, i \neq j} \delta^2(m_i, m_j)}. \tag{7}$$

Chapter 3

Clustering and Explainability of Residential Electricity Demand Profiles

3.1 Introduction

Smart meters, building automation systems, and energy sensors allow continuous tracking of electricity usage, providing large volumes of data that can be analyzed. The integration of machine learning and clustering techniques with these smart infrastructures enables data-driven optimization of energy consumption, improving energy efficiency and reducing peak demand. However, making sense of such vast, high-dimensional energy datasets remains a challenge, requiring effective data mining techniques to extract valuable insights.

To address these challenges, data mining [69], also known as knowledge discovery from data [70], has emerged as a process for extracting insights from large and complex datasets. It involves the use of data analysis and discovery algorithms to identify specific patterns or models within the data [71]. Among its many techniques, clustering is the process of grouping objects with minimal or no prior knowledge of their relationships within the data. It aims to uncover underlying patterns or classes, grouping similar objects into the same cluster while ensuring they differ from objects in other clusters [28]. These clusters help identify consumption patterns, enabling customized energy-saving strategies, demand response programs, and better infrastructure planning. However, clustering energy data presents challenges such as high dimensionality, outliers, overlapping clusters, and

varying densities, which can affect clustering accuracy and explainability.

Several studies have explored clustering techniques for energy data, but they often have limitations that affect their applicability. Some research focuses on explainable clustering methods but fails to evaluate a broad set of clustering algorithms, lacks explainability mechanisms, or does not explore dimensionality reduction effects on clustering performance. [51] introduced an optimization-driven tree-based clustering model, which directly constructs tree-based clusters. While their method ensures inherent explainability by structuring clusters as leaves in a decision tree, it does not compare traditional clustering methods such as K-Means, K-Medoids, Fuzzy C-Means, or GMM. Additionally, their study does not focus on clustering electricity consumption data, nor does it analyze how different decision tree models affect post-clustering explainability. In contrast, [50] explored clustering explainability but focused primarily on K-Means and K-Median, failing to evaluate soft clustering techniques like GMM or FCM. Other research, such as [72] and [73] applied clustering to electricity consumption data but only tested a limited number of clustering methods, restricting their ability to assess which techniques best segment real-world energy consumption patterns. Even studies that incorporated ensemble clustering techniques, such as [74], relied on a single clustering approach (GMM), without evaluating how different clustering techniques handle variations in electricity demand.

Beyond the choice of clustering algorithms, another major limitation in prior research is the lack of explainability, which results in clustering being treated as a black-box method with no clear justification for why certain households belong to specific clusters. Studies such as [42] and [48] applied clustering to energy consumption data without incorporating post-clustering explainability techniques, leaving energy analysts with limited insight into why clusters were formed. A recent survey on explainable clustering [49] categorizes explainable clustering approaches into pre-clustering, in-clustering, and post-clustering techniques, reviewing how existing methods incorporate explainability at different stages. However, this survey remains theoretical and does not provide an empirical comparison of clustering algorithms, limiting its ability to assess the practical effectiveness of explainability methods. While some studies attempted to integrate explainability using decision trees, such as [50], these efforts were often restricted to a single decision tree model, without evaluating how different tree structures impact explainability. [51] introduced an

optimization-driven tree-based clustering model, but it did not compare Axis-Aligned vs. Sparse Oblique Decision Trees, leaving a gap in understanding the trade-offs between full cluster coverage and rule simplicity. While some prior studies have investigated explainability in clustering [49–51] none have systematically compared multiple clustering methods on electricity consumption data while simultaneously evaluating different decision tree models for post-clustering explainability.

This study fills this gap by providing an empirical comparison of both clustering and explainability techniques, ensuring that results are explainable and applicable to real-world energy planning.

In addition to explainability challenges, previous research often overlooked the impact of dimensionality reduction on clustering performance. Many studies apply dimensionality reduction techniques before clustering without testing how dimensionality reduction affects cluster quality. For example, [73] used PCA but did not test whether clustering performance was improved or degraded by reducing feature dimensions. Similarly, [74] performed clustering on high-dimensional electricity load data without testing whether reducing dimensionality could help improve segmentation accuracy. This gap is significant because dimensionality reduction may either enhance clustering efficiency or lead to information loss, depending on the dataset.

Furthermore, the evaluation of clustering methods in prior studies was often limited to a single or some Cluster Validity Index (CVI), making it difficult to assess the robustness of the clustering results. While [73] provided a comprehensive analysis of multiple CVIs, most previous research relied solely on Silhouette Score, Davies-Bouldin Index (DBI), or Dunn Index (DI), without testing their reliability under different data conditions. Since some CVIs are sensitive to specific data characteristics like density variations, skewness, and overlapping clusters, it is crucial to assess multiple CVIs to ensure that clustering evaluations are robust and not biased by a single metric.

The rest of the chapter is organized into 2 sections. In section 2, we introduce and explain the considered methodology while in section 3, we present and discuss research findings.

3.2 Methodology

To analyze electricity consumption patterns and enhance clustering explainability, this study follows a structured methodology that integrates data pre-processing, clustering, cluster validation,

and explainability techniques. Figure 3.1 provides an overview of the complete workflow, outlining the key steps involved in the analysis.

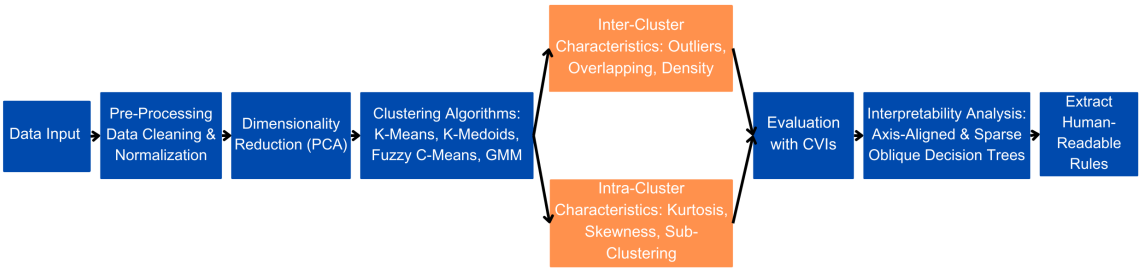


Figure 3.1: Methodological framework outlining data preprocessing, clustering, inter-/intra-cluster evaluation, and explainability analysis.

3.2.1 Data and Pre-processing

This study analyzes electric load demand profiles using a publicly available dataset from the Electricity Load Diagram (EL) Dataset, part of the UCI Machine Learning Repository [75]. The dataset contains electricity consumption data recorded at 15-minute intervals over a four-year period (2011–2014). As smart buildings and connected energy systems become more common, such datasets provide insights into electricity usage, facilitating demand prediction, anomaly detection, and energy efficiency improvements. Initially, data was collected from 370 Portuguese consumers, but due to late participation and missing records, a pre-filtering process was applied. Only households with at least 900 days of complete data were retained, reducing the sample size to 315 households. To ensure data consistency, specific days were excluded from the dataset. The last Sundays of March and October were removed since they coincide with daylight-saving time changes in Portugal and the U.S., potentially introducing inconsistencies in consumption patterns. Since the main objective of this study is to cluster daily load demand profiles rather than the data itself, removing these days was considered appropriate. Rather than using raw daily records, which may contain seasonal fluctuations or temporary anomalies, a representative daily profile was computed for each

household. This was achieved by calculating the median consumption value for each time slot across all available days, ensuring that only valid and complete daily profiles were considered. Using the median instead of the mean prevents outliers (e.g., extreme consumption days) from distorting the profile, making clustering results more stable. To enable meaningful comparisons across households, L2 normalization was applied to the median daily profiles. This transformation scales each household's profile vector to have a unit norm, ensuring that the clustering process emphasizes the shape and distribution of consumption patterns rather than absolute electricity usage levels. The L2 normalization is defined as:

$$x_{\text{norm}} = \frac{x}{\|x\|_2} = \frac{x}{\sqrt{\sum_{t=1}^{T} x_t^2}}, \tag{8}$$

where x represents the original median daily profile vector containing T time slots per day (here $T = 96$). The resulting vector x_{norm} is the normalized profile used for clustering, ensuring that differences in overall consumption magnitude do not dominate the identification of structural patterns in the data.

To visualize the daily consumption patterns across all households, Figure 3.2 presents a comparison of 315 representative daily profiles in the original feature space. Each curve corresponds to a single household, showcasing the variability in energy usage over 24 hours. The overlapping patterns reveal common consumption trends, such as morning and evening peaks, while also highlighting individual differences in household energy behavior. This visualization provides an overview of how normalized load demand profiles differ across consumers, reinforcing the need for clustering to identify distinct consumption patterns.

3.2.2 Dimensionality Reduction

Each consumer's electric load demand profile is structured as a time-series vector with 96 distinct values, where each value represents electricity consumption over a 15-minute interval within a 24-hour period. Since there are four intervals per hour, the total number of dimensions is computed as $24 \times \frac{60}{15} = 96$. This vectorized representation effectively captures each consumer's daily consumption pattern. However, high-dimensional data presents challenges, particularly the curse of dimensionality, which lowers the performance of clustering algorithms by introducing noise and

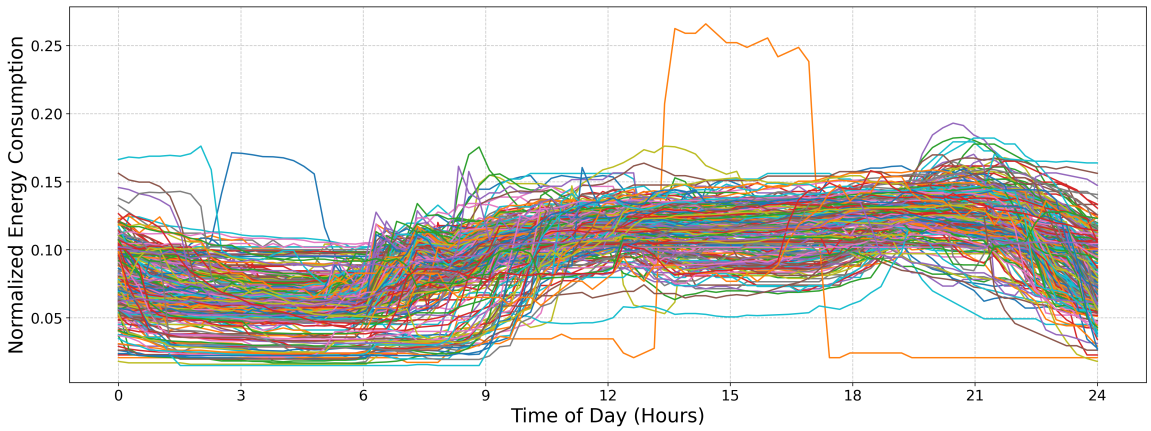
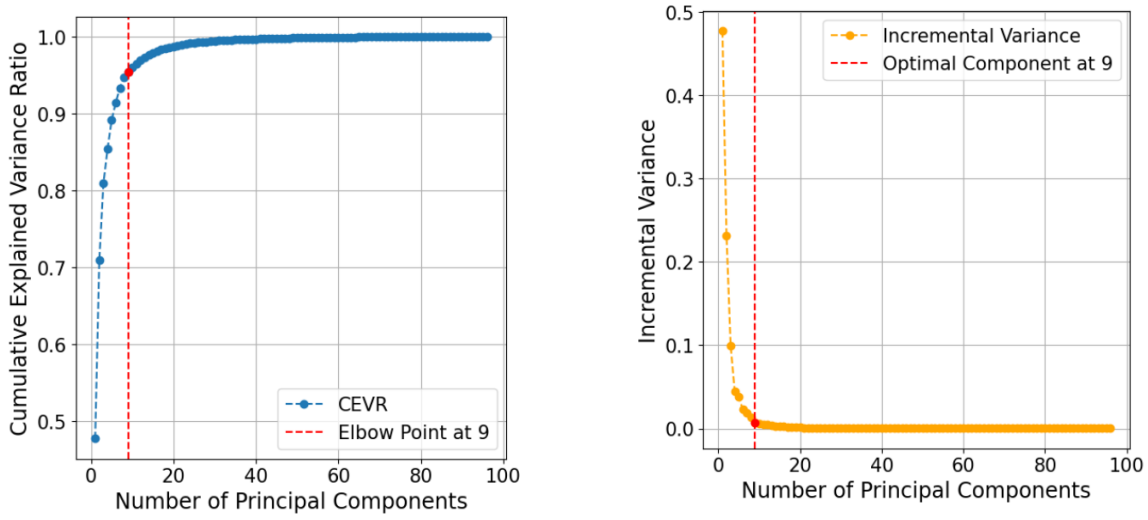


Figure 3.2: Comparison of representative daily profiles across 315 households in the original feature space. Each curve represents a single household's normalized electricity consumption over a 24-hour period.

increasing computational complexity [76]. To address these challenges, Principal Component Analysis (PCA) [77], a widely used dimensionality reduction technique, was applied. To ensure explainability, a feature mapping approach was employed, where the contribution of each original feature to the principal components was analyzed. This was done by examining the PCA loadings, which indicate how strongly each 15-minute interval influences a given principal component. The top contributing features for each principal component were identified and stored in a structured table to establish a meaningful connection between the reduced dimensions and the original electricity consumption patterns. Dimensionality reduction technique was applied to EL dataset after standardizing the data. Standardization is essential to ensure that all features contribute equally to the analysis. The main goal of applying PCA is to reduce the number of dimensions while keeping the most significant variance within the data. For this purpose, the optimal number of dimensions needs to be identified. The optimal number of principal components was determined using the Cumulative Explained Variance Ratio (CEVR) method. It identifies the number of components required to capture a substantial portion of the total variance [78]. To further validate the optimal number selected, incremental variance analysis was conducted. This method works by assessing the additional variance explained by each subsequent component until the incremental gain falls below a predefined threshold. The two combined methods ensure an accurate determination of the number of components to consider and enhance the efficiency of dimensionality reduction process. Applying PCA

could potentially enhance the efficiency and explainability of subsequent clustering analysis.



(a) (b)

Figure 3.3: (**a**). Plot of number of principal components versus CEVR. Elbow point is the optimal number of reduced dimensions while performing the PCA. (**b**). Plot of number of principal components versus Incremental Variance. Optimal component is the optimal number of reduced dimensions while performing PCA.

3.2.3 Clustering Algorithm

With the challenges of high-dimensional data now reduced, the next step is to use clustering algorithms to uncover patterns within customer profiles. Four clustering algorithms were applied to analyze customer consumption patterns: K-means, K-medoids, Fuzzy C-means, and GMM to EL dataset with and without dimensionality reduction. Unlike the previous study, which primarily relied on K-means for the EL dataset with dimensionality reduction and Fuzzy C-means on another dataset with dimensionality reduction only, this study tests a broader range of algorithms.

Each of these algorithms was chosen based on its ability to address specific challenges in clustering electricity load demand profiles: K-means is one of the most popular clustering algorithms. It was selected as a baseline due to its efficiency and widespread use in clustering time-series data [79]. It is known as a hard clustering algorithm [30], meaning it assigns each data point exclusively to one cluster without any shared membership across clusters. This method partitions data into distinct groups by minimizing the sum of squared distances between data points and their respective

cluster centroids. However, K-means is sensitive to outliers because an extreme value heavily influences the mean, leading to suboptimal clustering results [31]. To address this challenge, K-medoids was employed, which is another hard clustering algorithm. However, it considers cluster centers (medoids) instead of the mean, which is not affected by extreme values. It assigns each data point to the nearest medoid based on a specified distance metric [32]; in this research, Euclidean distance was chosen. Using the medoid reduces the impact of outliers on the clustering structure [33].

Knowing that hard clustering methods may not effectively capture clusters with outliers and overlapping characteristics, soft clustering [30] techniques were adopted to improve the clustering analysis. In soft clustering, data points can belong to multiple clusters with varying probabilities. Fuzzy C-means [34] is a soft clustering algorithm that assigns each data point a degree of membership to different clusters. The membership value is calculated based on the distance between the data point and randomly chosen centers. Higher membership is associated with points that are closer to a cluster center, while other distant points receive lower values. This approach helps to identify clusters with overlapping and shared features. To further validate our assumptions about soft clustering, GMM [35] were applied.

To apply the clustering algorithms discussed, the optimal number of clusters needs to be identified as a first step. For k-means, the optimal number of clusters was determined using both the elbow heuristic [80] method and the gap statistic [81] method. The elbow method calculates the sum of squared distances from each data point to its assigned cluster centroid. This metric, known as the within-cluster sum of squares (WSS), measures the compactness of clusters, with lower values indicating tighter, more well-defined groupings. When comparing datasets with different dimensionalities, it is more appropriate to use the average WSS, which normalizes the sum of squared distances by the number of data points. This approach makes the comparison fair by reducing the impact of higher-dimensional spaces, where distances naturally increase. As a result, it provides a more accurate way to assess clustering quality. The optimal number of clusters is the point that balances the sum of squared distances and model complexity. Essentially, it is the point where adding another cluster results in only a marginal improvement in variance reduction. The second method is the gap statistic, which determines the optimal number of clusters by comparing the clustering quality of the actual dataset to that of a randomly distributed reference dataset. It evaluates how much

more compact the clusters in the real data are compared to randomly scattered points, ensuring that the chosen number of clusters reflects meaningful structure rather than random variations. The gap statistic is computed by first running clustering on the real dataset and calculating WSS. A random dataset with the same size and range is then generated, and clustering is applied to obtain its WSS. The gap statistic is defined as:

$$\text{Gap}(k) = E^*[\log(WSS_k)] - \log(WSS_k) \tag{9}$$

where WSS_k is the within-cluster sum of squares for k clusters in the real dataset, and $E^*\left[\log(WSS_k)\right]$ is the expected log WSS for the random dataset. A larger gap indicates that the real data exhibits stronger clustering structure. The ideal number of clusters is where the gap between real and random clustering is largest [81].

For k-medoids, gap statistic was used to determine the optimal number of clusters. For fuzzy c-means, the Dunn index [82] was used to select the optimal number of clusters. The Dunn index calculates the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance. Finally, for GMM, silhouette score [82] was employed, which calculates the average distance between points within the same cluster compared to points in neighboring clusters.

3.2.4 Intra-Cluster and Inter-Cluster Analysis

To analyze clustering performance, the next step is to examine structural properties within and between clusters to identify influencing factors. By examining inter-cluster characteristics which are outlier, overlapping, and density, and intra-cluster characteristics which are skewness, kurtosis, and sub-clustering, adjustment could be done to enhance cluster quality. This study evaluates the impact of data characteristics on clusters using CVIs.

Inter-Cluster Analysis

Inter cluster characteristics define relationships between clusters. In this study, outlier, overlapping and, differential density are analyzed.

- Outliers; are defined as single-point clusters in the clustering results. These points are included in the clustering process but are not expected to significantly influence the CVI scores. To verify this assumption, outliers were identified through manual inspection and subsequently removed. CVI scores were then compared before and after the removal of these single-point clusters to evaluate any potential impact on the clustering validation metrics.

- Overlapping clusters; occur when certain data points share characteristics with more than one cluster, making it difficult to assign them definitively to a single cluster. This typically arises when clusters are not well-separated, with some data points located near the boundaries of two or more clusters. To address this issue, overlapping points were removed from the dataset to evaluate whether reducing overlap could improve cluster quality. For fuzzy c-means without and with DR, GMM without DR, and k-means without DR, overlapping points were identified and removed through manual inspection. For GMM with DR, k-medoids with and without DR, and k-means with DR, a distance-based approach was applied. This method calculates the Euclidean distance between each data point and the cluster centers. Points found to be close to multiple centers within a specified threshold were considered overlapping and removed. The threshold was manually adjusted to balance effectiveness in reducing overlap while maintaining data integrity.

- Differential Density; refers to clusters having varying densities, often reflecting natural consumption patterns. Density is defined as the number of data points per unit volume and was calculated as the ratio of points in a cluster to its diameter. It was hypothesized that increasing the density of clusters would improve CVI scores. To evaluate this, the density of the densest cluster was increased by adding additional points while keeping the cluster diameter constant. The effect of this adjustment on CVI scores was then analyzed to assess the relationship between cluster density and clustering quality.

Intra-Cluster Analysis

Intra-cluster characteristics refer to the internal structural properties of clusters. In this study, these characteristics are assessed through the analysis of kurtosis, skewness, and sub-clustering.

- Central Kurtosis; refers to data points being tightly clustered around the center of a cluster, with fewer points near the edges. Statistically, kurtosis measures the tailedness of a distribution. When more points are concentrated near the center, the distribution becomes leptokurtic, increasing similarity within the cluster. This increased density is expected to enhance CVI scores by making clusters more compact. To evaluate this, $k\%$ of the data points in each cluster were shifted closer to the center, where $k \in \{25, 50, 75, 100\}$. The shifted points were generated on a d-dimensional hypersphere, where d corresponds to the data dimensionality. The radius of the hypersphere was set to the cluster radius divided by m, where m is a discrete value ranging from 5 to 20, depending on the dimensionality of the data. The effect of these adjustments on CVI scores was analyzed to assess the impact of central kurtosis on clustering performance.

- Skewness; occurs when the mean is positioned away from the cluster's center, resulting in an asymmetrical distribution. While kurtosis increases density near the center, skewness shifts the majority of data points toward one side of the cluster, causing the mean to deviate from the geometric center. With the cluster's diameter remaining constant, increasing skewness is expected to either maintain or improve CVI scores. To evaluate this, $k\%$ of the data points in each cluster were rearranged farther from the center and closer to the mean, where $k \in \{25, 50, 75, 100\}$. To preserve the natural structure of each cluster, no new points were added to empty regions. Adjustments were performed around the mean to introduce skewness without creating artificial patterns inconsistent with real-world data distributions. During implementation, the distance between the mean and the center was calculated for all clusters. If this distance exceeded the cluster radius divided by n, skewness adjustments were applied, where n is a discrete value ranging from 2 to 5, depending on the cluster size. The rearranged points were positioned on a d-dimensional hypersphere, where d corresponds to the data dimensionality. The radius of the hypersphere was set to the cluster radius divided by m, where m is a discrete value ranging from 5 to 20, based on the data dimensionality.

- Sub Clustering; occurs when a cluster splits into two or more smaller clusters. This suggests that the cluster structure is not optimal and that dividing it into smaller, distinct clusters may

yield better results. Such a division indicates that the sub-clusters do not share significant characteristics or features, warranting their distinction as separate clusters. The presence of sub-clusters is expected to worsen CVI scores due to reduced cohesion and separation within the clusters. To evaluate the impact of sub-clustering, the mean and center of each cluster were analyzed, as these are considered dense points. The distance between the mean and the center of each cluster was calculated and compared to a threshold defined for each algorithm. If this distance exceeded the threshold, sub-clustering was identified. In such cases, points were rearranged to move closer to the nearest dense point (mean or center). These rearrangements were performed with adjustments affecting $\{25, 50, 75, 100\}$ of the points within the cluster.

3.3 Results and Discussion

3.3.1 Principal Component Analysis

PCA served as an important step in simplifying the dataset while retaining its most significant features. Figure 3.3 illustrates the Cumulative Explained Variance Ratio (CEVR) across the principal components, with an elbow point at the ninth component. This point captures 95% of the total variance. Any additional components beyond the elbow point yield minimal variance, indicating diminishing returns. To further validate the selection of nine components, Figure 3.3 showcases an incremental variance analysis. This analysis confirms that components beyond the ninth contribute less than a practical threshold of 0.01 variance. This finding reinforces that nine components represent the optimal number to retain in this study. Since a dimensionality reduction technique was applied to the EL dataset, a feature mapping was generated during PCA to link each principal component to the original features. This mapping facilitates the interpretation of decision rules in subsection 3.3.4 and is presented in Table 3.1.

Table 3.1: Mapping of Principal Components to top features and time intervals.

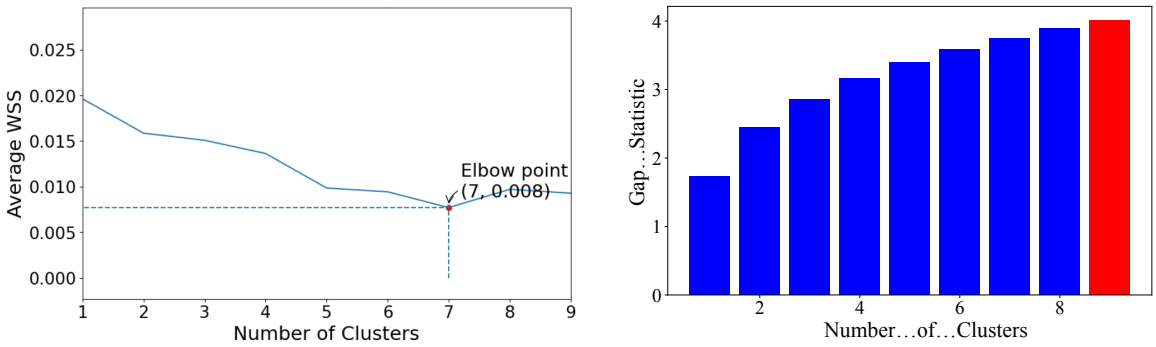
PC (Feature Index)	Top Contributing Features	Time Intervals
PC1	Features {3,4,5,6,7}	00:45, 01:00, 01:15, 01:30, 01:45
PC2	Features {79,80,82,83,84}	19:45, 20:00, 20:30, 20:45, 21:00
PC3	Features {39,40,76,77, 78}	09:45, 10:00, 19:00, 19:15, 19:30
PC4	Features {69,70,71,72, 73}	17:15, 17:30, 17:45, 18:00, 18:15
PC5	Features {27,28,29, 30, 48}	06:45, 07:00, 07:15, 07:30, 12:00
PC6	Features {0,1,69, 70, 72}	00:00, 00:15, 17:15, 17:30, 18:00
PC7	Features {89,90,91, 92,52}	22:15, 22:30, 22:45, 23:00, 13:00
PC8	Features {51,52,50, 49, 88}	12:45, 13:00, 12:30, 12:15, 22:00
PC9	Features {23,24,22, 88, 87}	05:45, 06:00, 05:30, 22:00, 21:45

3.3.2 Clustering Algorithms

- K-Means; For the EL dataset without DR, the optimal number of clusters was determined using two methods. The elbow heuristic suggested 7 clusters while the gap statistic recommended 9 clusters as shown in Figure 3.4. To correct this difference, 8 clusters were selected as the optimal number. Applying k-means with 8 clusters produced the visualization shown in Figure 3.7 , which reveals overlapping clusters and the presence of an outlier. For the EL dataset with DR, the same process was followed. The elbow heuristic again suggested 7 clusters while the gap statistic indicated 9 clusters presented in Figure 3.5 . Based on both methods, 8 clusters were again selected as the optimal number. The clustering results, displayed in Figure 3.8, also show overlapping clusters and the presence of an outlier in cluster 4. K-Means, as a hard clustering algorithm, is known to be sensitive to outliers. This sensitivity is evident in the clustering results presented, where outliers were present alongside overlapping clusters. The reliance on the mean as the cluster center causes outliers to significantly influence the cluster boundaries, resulting in less accurate cluster definitions.

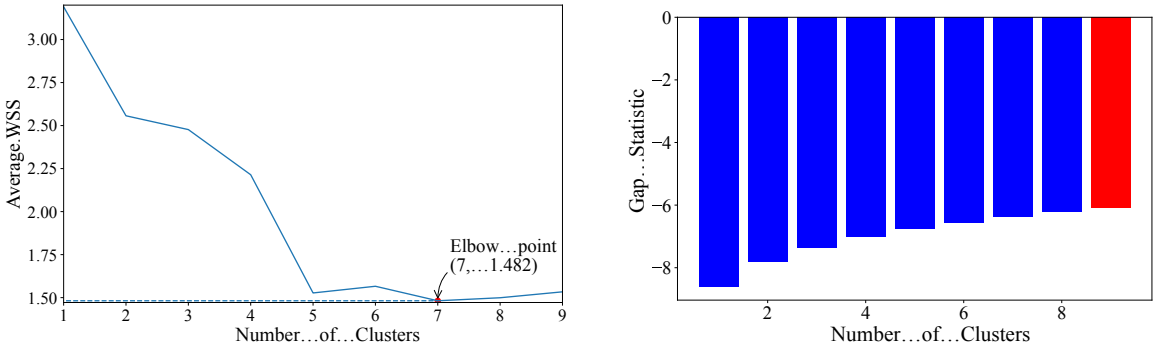
- K-Medoids; To address the identified outlier issue, the k-medoids algorithm was applied. For the EL dataset without DR, the gap statistic suggested an optimal number of clusters equal to 9 while for EL dataset with DR, the gap statistics proposed 8 clusters (see Figure 3.6). After applying k-medoids clustering with 9 clusters and with 8 clusters, the results, shown in Figure 3.7 and Figure 3.8 respectively, demonstrated effective handling of the outlier. However,

overlapping still exists in both cases. K-medoids as a hard clustering algorithm solved the outlier issue. The removal of outliers demonstrates its effectiveness in correcting this limitation of k-means. However, overlapping clusters persisted, indicating that while k-medoids can handle outliers, it does not correctly address the challenge of overlapping. This underscores a limitation of hard clustering methods, as they rely on strict boundaries that cannot account for shared characteristics between clusters.



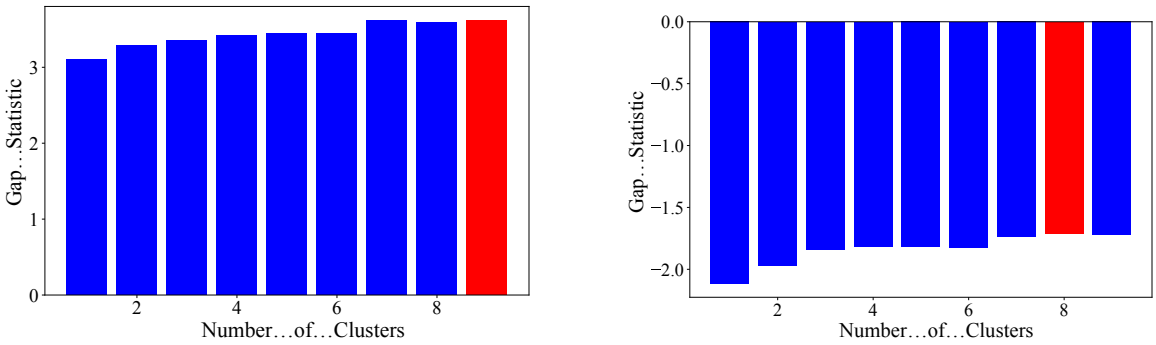
(a) (b)

Figure 3.4: (**a**). Plot of number of clusters versus average within sum of squares. Elbow point is the optimal number of clusters while performing K-Means on EL without dimensionality reduction. (**b**). Value of gap-statistic for different number of clusters to identify the optimal number of clusters for the K-Means algorithm on EL without dimensionality reduction. Optimal value of gap-statistic is highlighted in red.



(a) (b)

Figure 3.5: (**a**). Plot of number of clusters versus CEVR. Elbow point is the optimal number of clusters while performing K-Means on EL with dimensionality reduction. (**b**). Value of gap-statistic for different number of clusters to identify the optimal number of clusters for the K-Means algorithm on EL with dimensionality reduction. Optimal value of gap-statistic is highlighted in red.



(a) (b)

Figure 3.6: Value of gap-statistic for different number of clusters to identify the optimal number of clusters for the K medoids algorithm on EL. Optimal value of gap-statistic is highlighted in red.(**a**). El dataset without dimensionality reduction (**b**). EL dataset with dimensionality reduction.

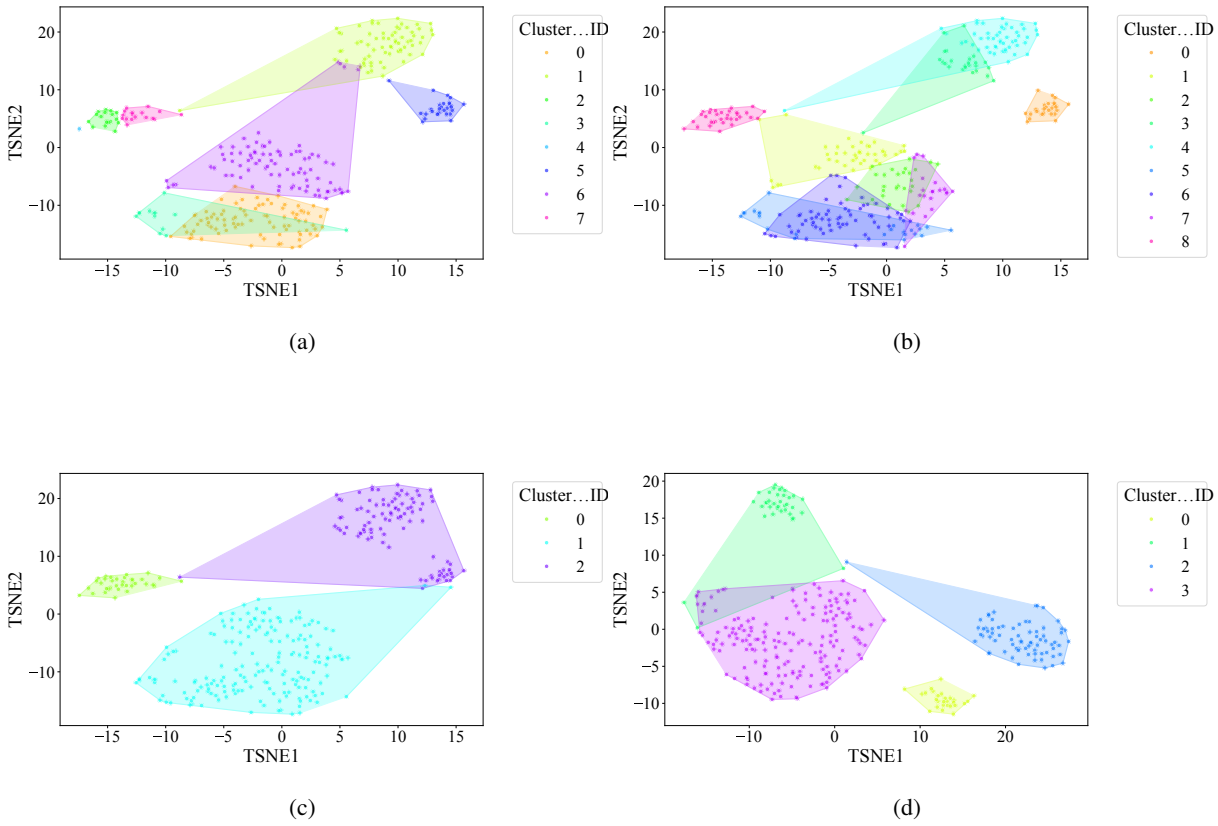


Figure 3.7: The t-SNE plot illustrates baseline clustering results using different algorithms on the EL dataset without dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means, and (**d**) GMM.

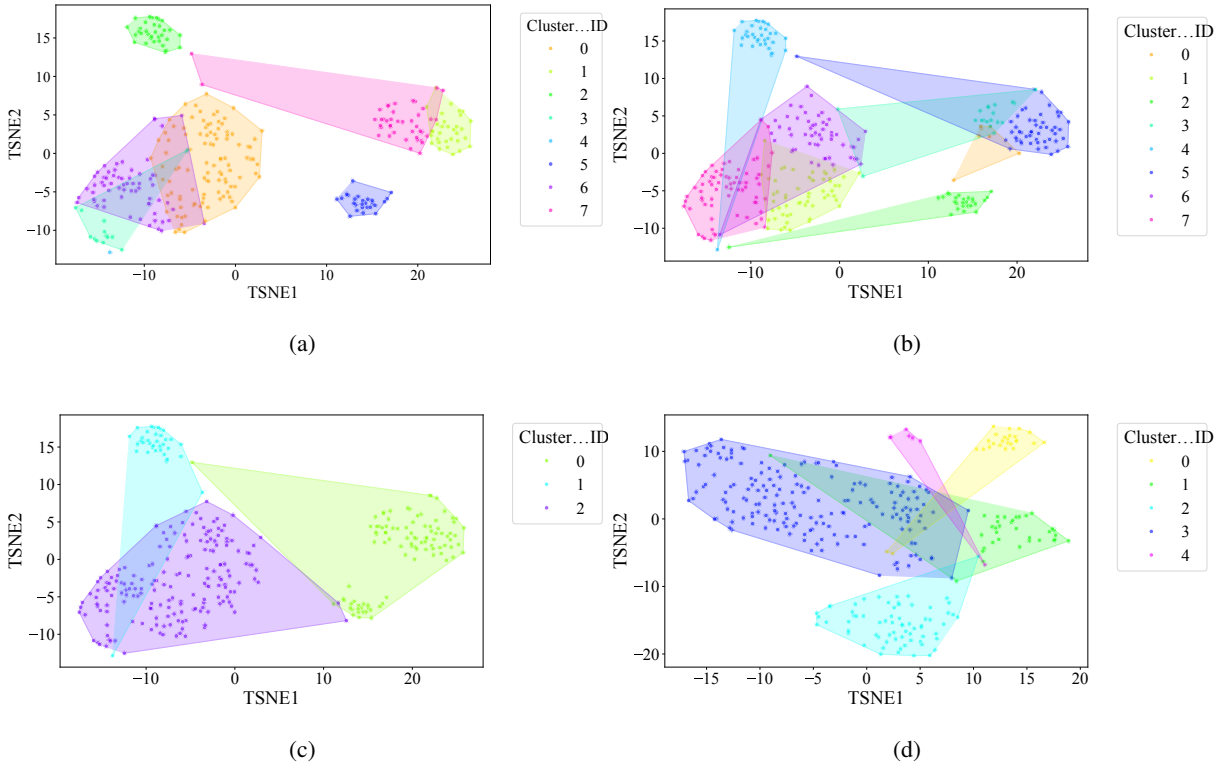


Figure 3.8: The t-SNE plot illustrates baseline clustering results using different algorithms on the EL dataset with dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means, and (**d**) GMM.

- Fuzzy C-Means; To address the issue of overlapping clusters, the soft clustering algorithm fuzzy c-means was applied. For EL dataset without DR, the Dunn Index identified the optimal number of clusters as 3, with a fuzzifier value of 1.2. The clustering results, presented in Figure 3.7, show that fuzzy c-means minimized overlapping compared to hard clustering techniques. For the EL dataset with DR, the Dunn Index similarly indicated an optimal cluster count of 3, with a slightly adjusted fuzzifier value of 1.1. The clustering results, displayed in Figure 3.8, show no outliers and minimal overlapping, although the overlap was slightly greater than in the non-DR case.

  Fuzzy c-means resulted in a marked reduction in overlap. The probabilistic nature allows it to handle data points that fall near cluster boundaries. This results in smoother transitions between clusters and a better representation of complex data structures. Furthermore, no

outliers were observed, which highlight its robustness. Dimensionality reduction had no noticeable effect on these results, as similar improvements were observed for both datasets with and without DR.

- Gaussian Mixture Model; The second soft clustering algorithm applied was the GMM, where the silhouette score was used to determine the optimal number of clusters. For the EL dataset without DR, the SH identified 4 as the optimal number of clusters. The clustering results, presented in Figure 3.7, show well-defined clusters with minimal overlap and no outliers, demonstrating the capability of GMM to handle overlapping clusters effectively. For the EL dataset with DR, SH indicated 5 as the optimal number of clusters, as shown in Figure 3.8. The clustering results reveal no outliers but show a slightly higher degree of overlap compared to the dataset without DR. The presented results further validate the advantages of soft clustering. Like fuzzy c-means, results from GMM demonstrated reduced overlap compared to hard clustering methods and the complete absence of outliers. GMM's ability to model the data probabilistically improves its flexibility in representing real-world data, particularly in high-dimensional spaces. Dimensionality reduction similarly showed no significant impact on the results, as the improvements in clustering quality were consistent across both datasets.

  Findings from both the EL datasets, with and without dimensionality reduction, demonstrated that soft clustering is better than hard clustering in terms of outlier and overlapping. While k-means and k-medoids struggle with overlapping clusters, soft clustering algorithms, fuzzy c-means and GMM, effectively reduce overlap and eliminate outliers. This reinforces the strength of soft clustering approaches in handling complex data distributions, where clusters may share characteristics or where outliers significantly impact results. The absence of a noticeable effect from dimensionality reduction suggests that the algorithms themselves played a more critical role in improving clustering quality than the preprocessing step.

To further explore the characteristics of each cluster, representative daily energy consumption profiles were visualized for each clustering algorithm. Figure 3.9 offers insight into how energy usage varies across different clusters throughout the day, revealing general trends such as morning

and evening peaks or fluctuations in consumption.

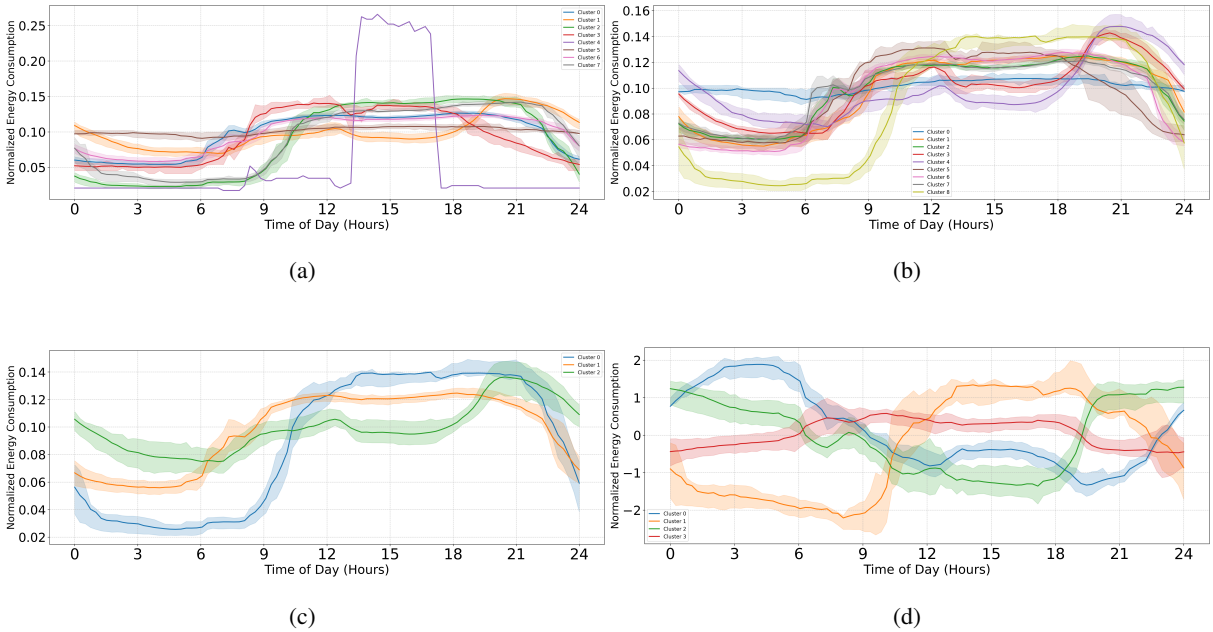


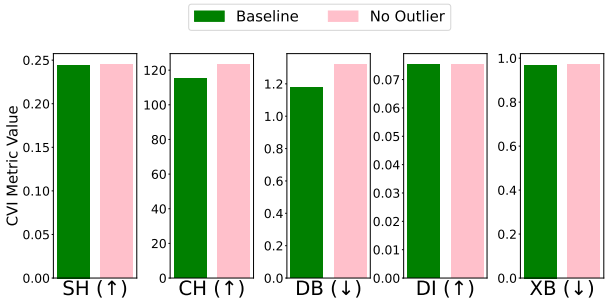
Figure 3.9: Comparison of Representative Daily Profiles Across Clusters using EL dataset without dimensionality reduction.(**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means, and (**d**) GMM.

The figure highlights key differences in clustering behavior across algorithms. In K-Means (See Figure 3.9a), the presence of an outlier is evident, as represented by the purple cluster, which exhibits an unusual spike around midday, a characteristic not observed in the other clustering methods. Additionally, overlapping between clusters is noticeable, particularly in K-Means (See Figure 3.9a) and K-Medoids (See Figure 3.9b), where multiple clusters share similar consumption patterns, leading to ambiguity in boundary definitions. In contrast, Fuzzy C-Means (See Figure 3.9c) and GMM (See Figure 3.9d) exhibit more distinct and structured cluster separations. However, while these visualizations provide a high-level overview of cluster behavior, the underlying rationale behind cluster assignments remains a black box. There is no explicit understanding of the factors driving these groupings or the specific features that most strongly influence cluster formation. To bridge this explainability gap, a combination of Axis-Aligned and Sparse Oblique Decision Trees is employed to extract meaningful classification rules, offering a transparent and structured way to understand and justify the clustering process.

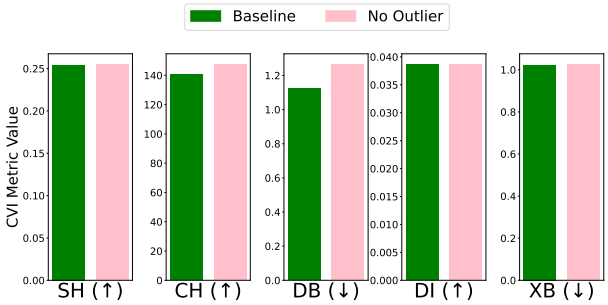
3.3.3 Inter and Intra Cluster characteristics

Inter-Cluster Characteristics

- Outlier; The impact of outlier removal on clustering performance was assessed by comparing CVI metrics for the baseline and outlier-removed cases. Among the algorithms, only k-means identified outliers. In both the EL dataset without and with DR, SH, DI, and XB metrics remained unchanged after outlier removal. However, the CH score improved, while the DB index deteriorated unexpectedly in both cases (see Figure 3.10). Most CVIs remained unaffected, except for CH and DB. CH improved due to enhanced inter-cluster dispersion, as outlier removal increased the separation between clusters. Conversely, DB deteriorated unexpectedly, likely because the reduced intra-cluster variance was accompanied by diminished inter-cluster separation. This behavior suggests that SH, XB, or DI are reliable CVIs for datasets with outliers, while DB and CH require careful consideration due to their sensitivity to outlier removal.



(a)



(b)

Figure 3.10: CVIs' response to outlier removal using K-Means on the EL dataset. (**a**). EL dataset without applying dimensionality reduction. (**b**). EL dataset with dimensionality reduction.

- Overlapping; The effect of removing overlapping profiles on clustering performance was evaluated for each algorithm and dataset combination. For the EL dataset without DR, removing overlapping profiles improved clustering performance across algorithms. Specifically, K-Means showed improved CVI metrics except for the XB index, which worsened; K-Medoids demonstrated enhancements in SH, DB, and DI indices, while CH and XB indices remained unchanged; Fuzzy C-Means exhibited minimal impact on CVIs after the removal of one profile; and GMM showed improvements across all CVIs after removing two profiles. For the EL dataset with DR, the removal of overlapping profiles resulted in more distinct clusters and noticeable improvements across all algorithms. K-Means exhibited substantial improvements in SH, DB, DI, and XB indices, with a slight deterioration in CH; K-Medoids and Fuzzy C-Means showed significant improvements across all CVIs; and GMM achieved better-defined clusters with enhanced CVI performance. The results are summarized in Figure 3.11 (without DR) and Figure 3.12 (with DR).

  Removing overlapping data points improved all CVIs across algorithms, both with and without DR, except for XB in k-medoids without DR. This inconsistency in XB's behavior highlights the need for further investigation, as it demonstrated improvements when DR was applied. For overlapping datasets, SH, CH, DI, and DB are recommended, while XB should be used cautiously, particularly in non-DR scenarios.

- Differential Density; The impact of differential density on CVI metrics varied across clustering algorithms and dataset configurations. Without DR, most algorithms showed improvements in SH, CH, and XB indices, while DB and DI generally remained constant. Notably, GMM exhibited slight decreases in DI despite improvements in CH and XB. With DR, CH and XB consistently improved across all algorithms. However, K-Means and Fuzzy C-Means showed unexpected deterioration in SH, and K-Medoids and Fuzzy C-Means experienced decreases in DI. Full details are visualized in Figure 3.13 (without DR) and Figure 3.14 (with DR). Increasing the density of clusters improved most CVIs, except for DI and DB, which remained stable across all algorithms. This stability can be attributed to DI's reliance on minimum inter-cluster distances, which are unaffected by density changes, and DB's focus on the

average similarity between clusters, which does not capture intra-cluster density variations. For datasets with increasing density, SH, XB, and CH are the most reliable CVIs, while DI and DB may be less informative.

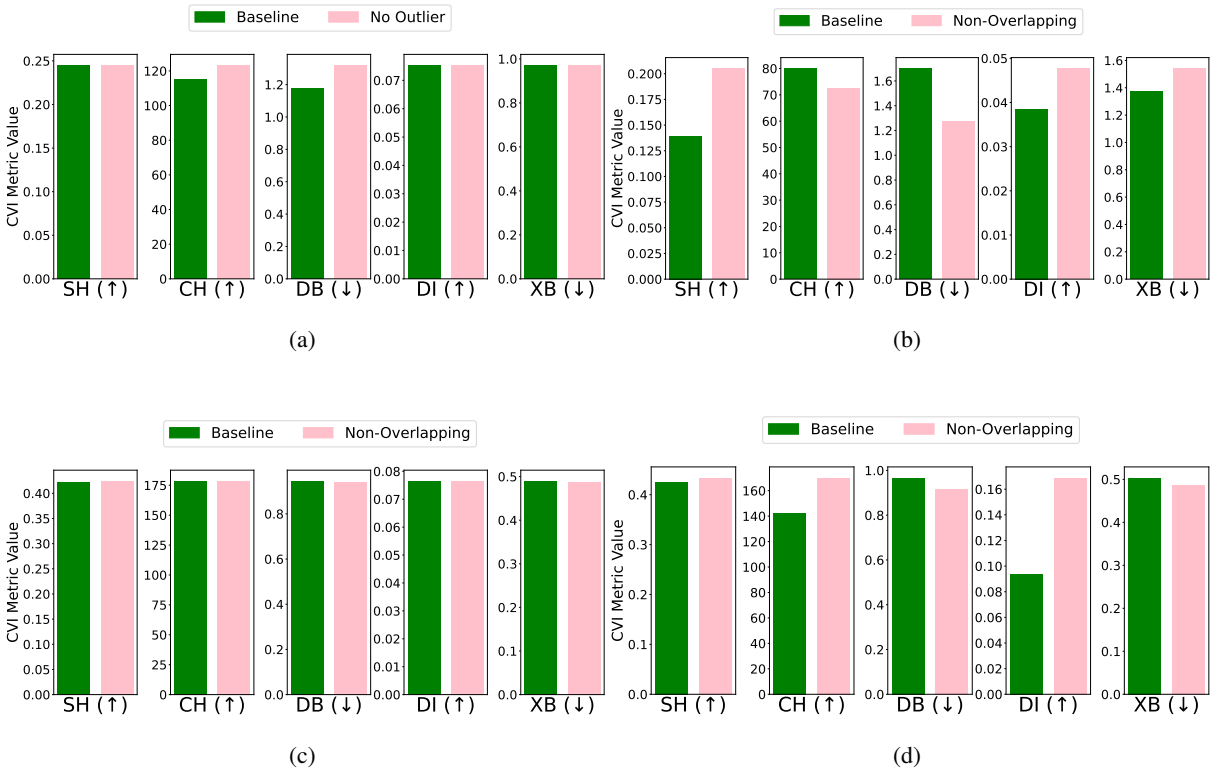


Figure 3.11: CVIs' response to overlapping removal using different algorithms on the EL dataset without dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means, and (**d**) GMM.

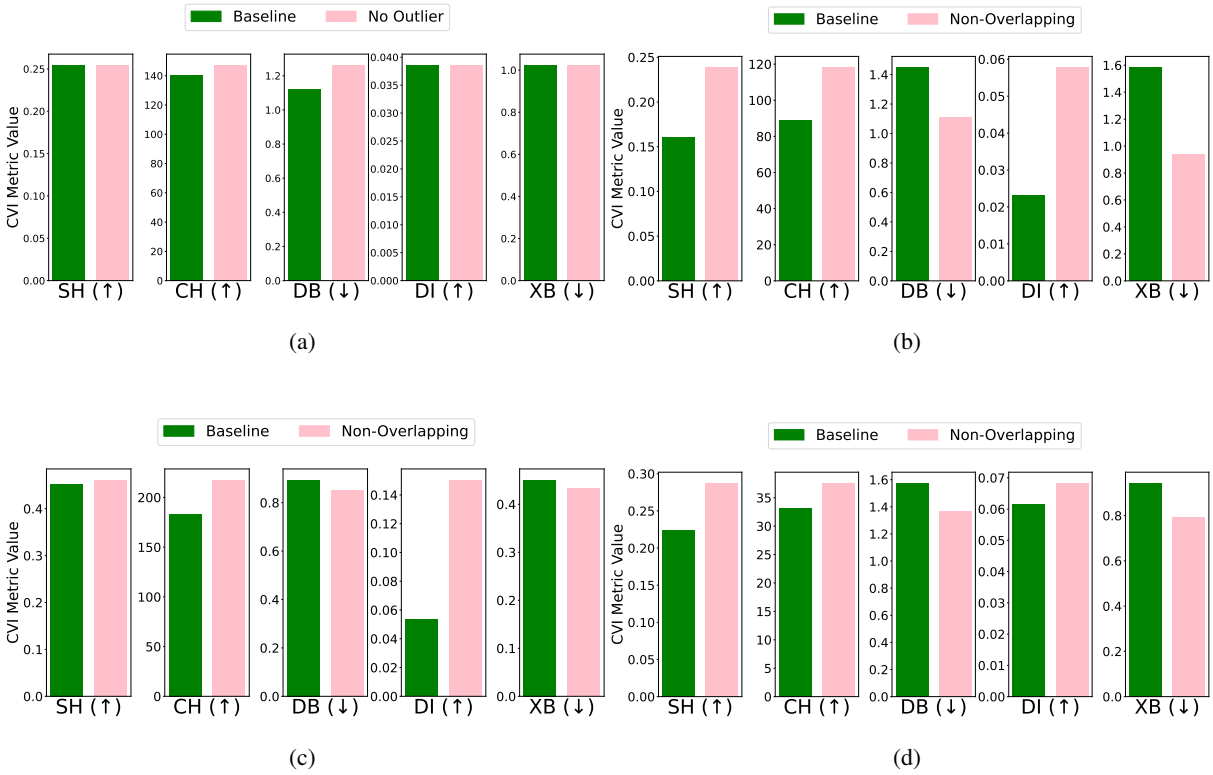


Figure 3.12: CVIs' response to overlapping removal using different algorithms on the EL dataset with dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means, and (**d**) GMM.

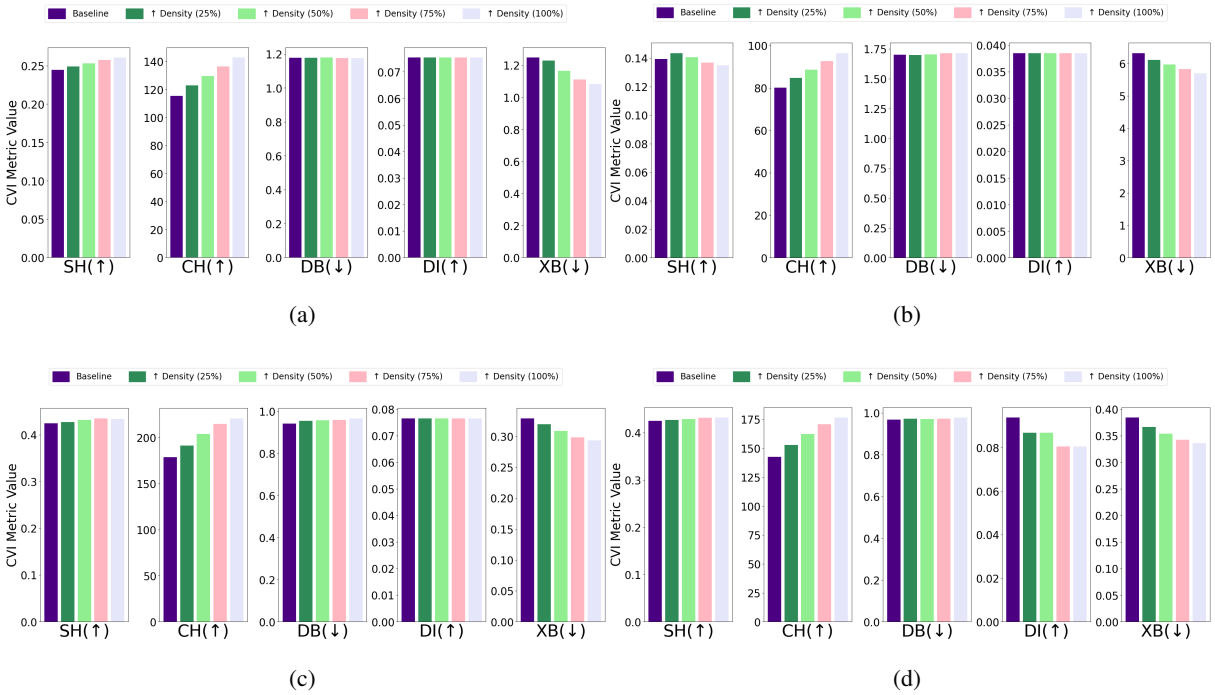


Figure 3.13: The effect of increasing differential density on various CVIs using different algorithms on the EL dataset without dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C Means, and (**d**) GMM.

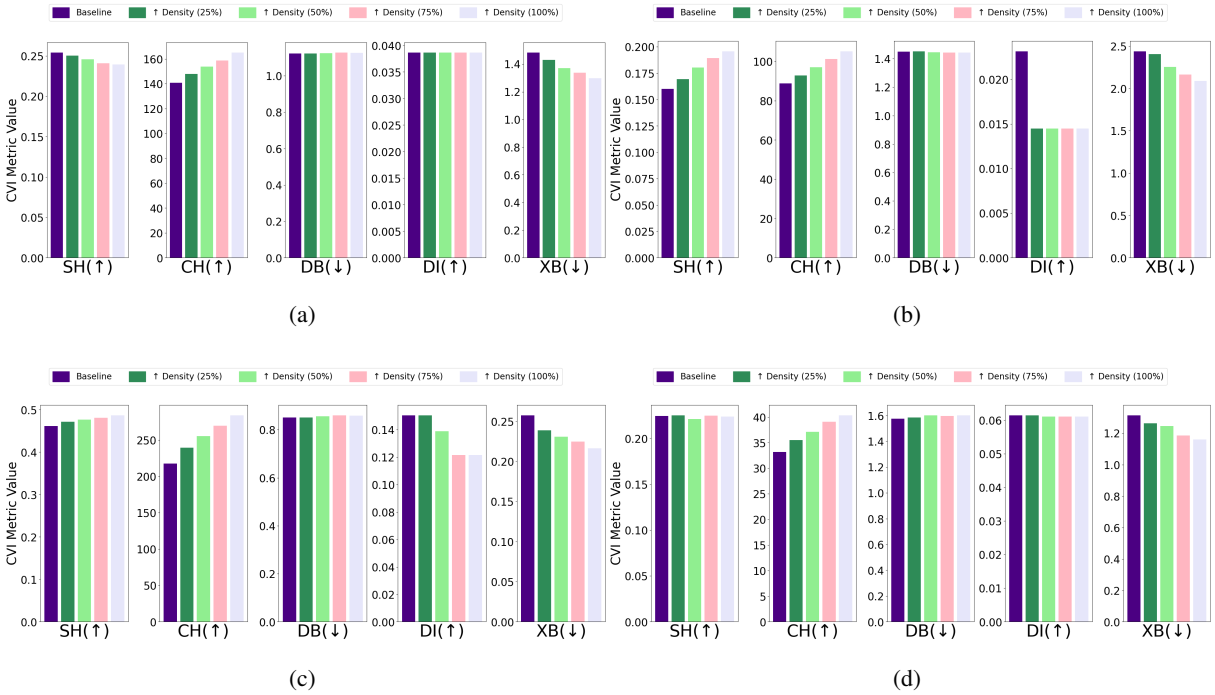


Figure 3.14: The effect of increasing differential density on various CVIs using different algorithms on the EL dataset with dimensionality reduction: (**a**). K-Means, (**b**). K-Medoids, (**c**). Fuzzy C Means, and (**d**). GMM.

Intra-Cluster Characteristics:

- Central Kurtosis; The effect of increasing central kurtosis on clustering performance was evaluated for each algorithm and dataset combination. For the EL dataset without DR, points were adjusted to lie on a 96-dimensional hypersphere with a radius equal to $\frac{1}{8}$ of the cluster size. For the EL dataset with DR, points were shifted to a 9-dimensional hypersphere with a radius equal to $\frac{1}{15}$ of the cluster size. Across all algorithms, increasing central kurtosis led to improvements in CVI metrics for both versions of the EL dataset. The results for CVIs on the EL dataset without DR are displayed in Figure 3.15 while the results for the EL dataset with DR are shown in Figure 3.16. Increasing central kurtosis positively impacted all CVIs, as redistributing points closer to cluster centers enhanced both intra-cluster compactness and inter-cluster separation. This consistent improvement across algorithms suggests that SH, XB, CH, DI, and DB are all effective for datasets with central kurtosis.

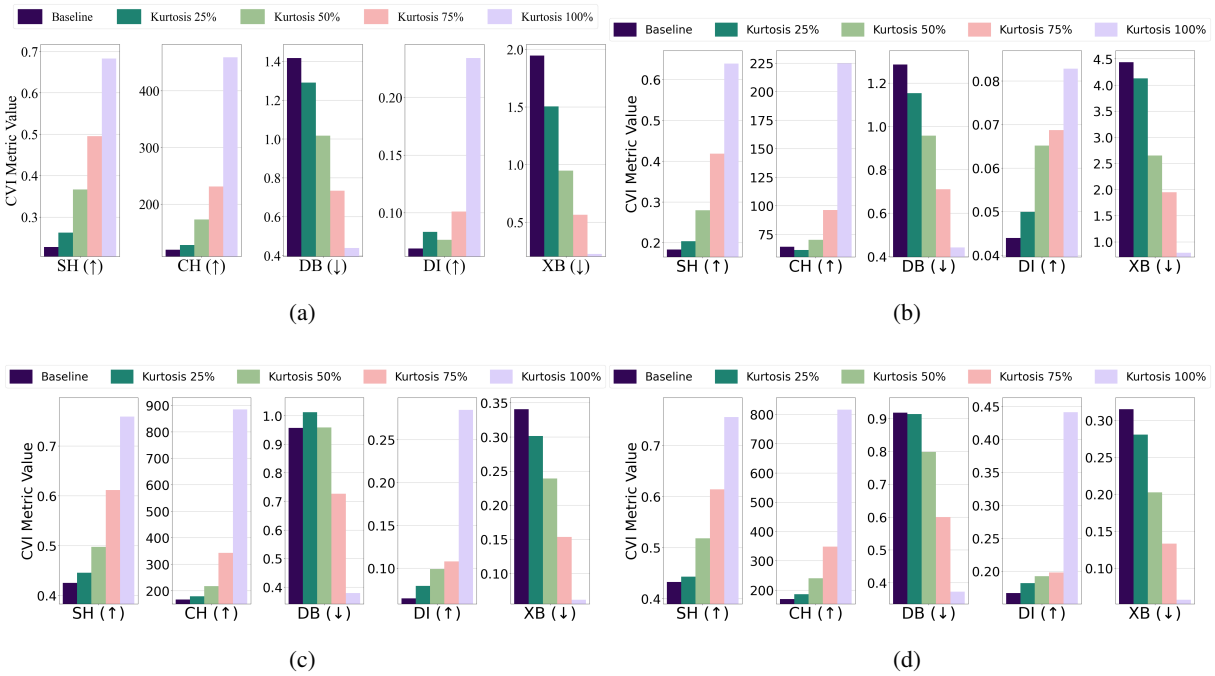


Figure 3.15: The effect of increasing the level of kurtosis close to center on various CVIs using different algorithms on the EL dataset without dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C Means, and (**d**) GMM.

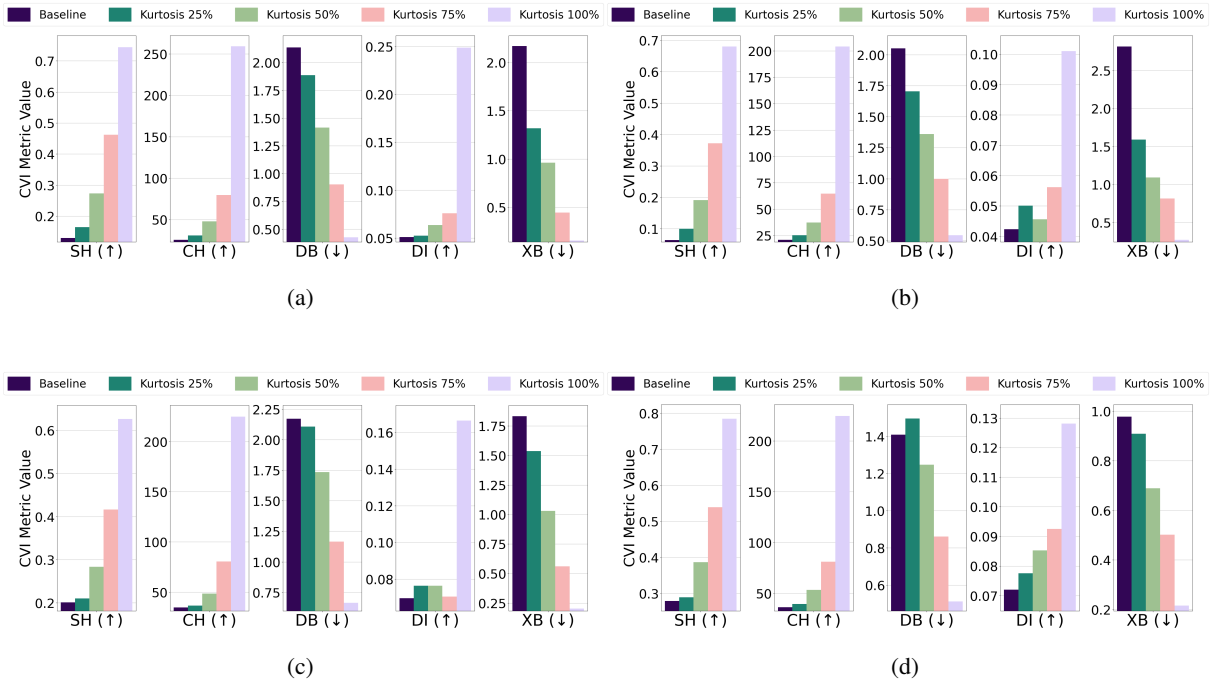


Figure 3.16: The effect of increasing the level of kurtosis close to center on various CVIs using different algorithms on the EL dataset with dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C Means, and (**d**) GMM.

- Skewness; For the EL dataset without DR, points were repositioned on a 96-dimensional hypersphere with a radius equal to $\frac{1}{5}$ of the cluster size. Adjustments led to improvements in CVI metrics across all algorithms. K-Means, K-Medoids, Fuzzy C-Means, and GMM all exhibited enhanced clustering performance with increased skewness. For the EL dataset with DR, points were relocated on a 9-dimensional hypersphere with a radius equal to $\frac{1}{30}$ of the cluster size. Increasing skewness resulted in noticeable improvements in CVI metrics across all algorithms. The results are summarized in Figure 3.17 (without DR) and Figure 3.18 (with DR). Skewness adjustments revealed irregular behavior in DI, likely due to its sensitivity to cluster shape distortions. For skewed datasets, it is recommended to avoid DI and instead use SH, XB, CH, or DB.

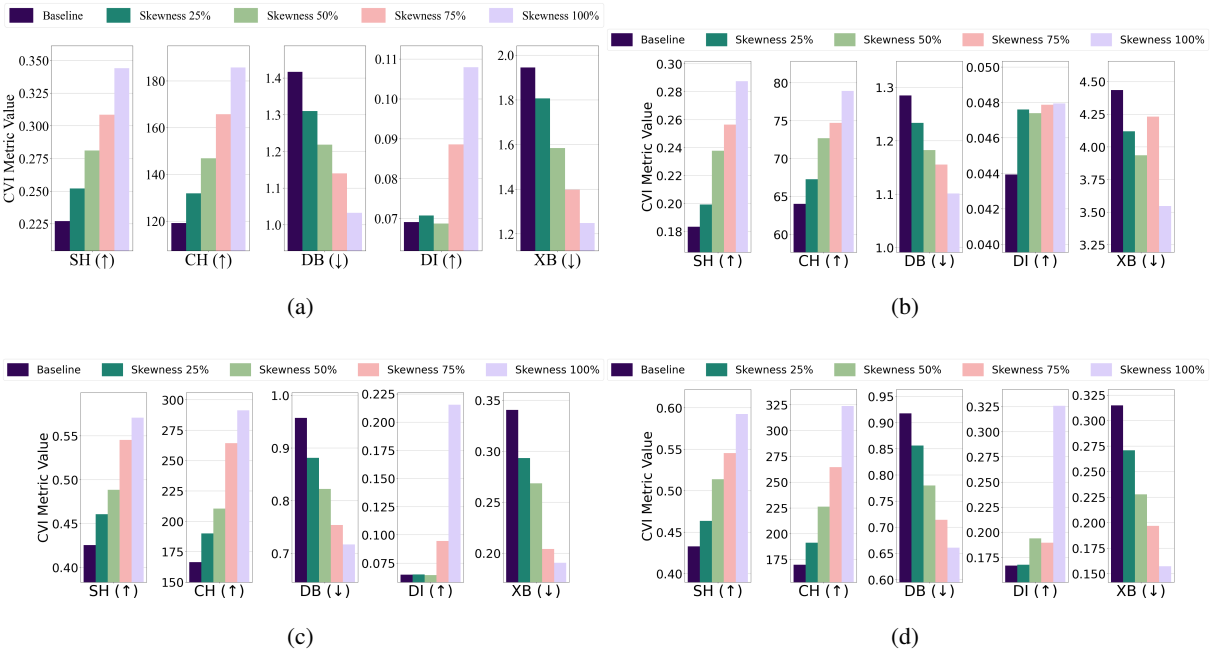


Figure 3.17: The effect of increasing the level of skewness on various CVIs using different algorithms on the EL dataset without dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C Means, and (**d**) GMM.

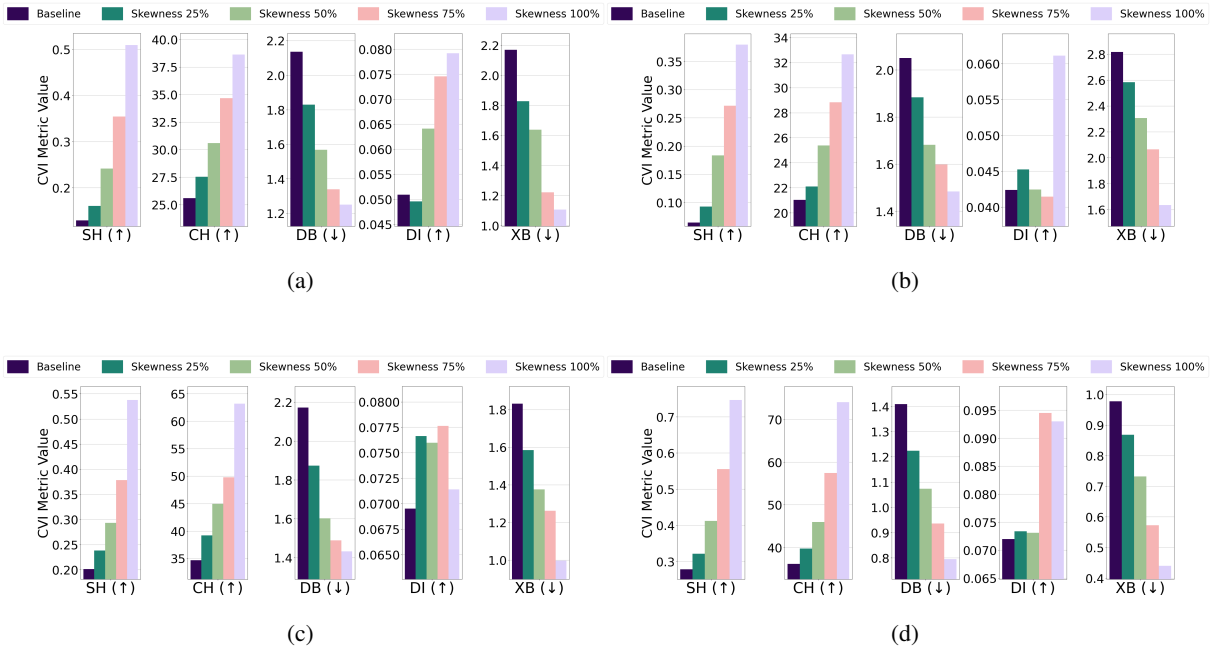


Figure 3.18: The effect of increasing the level of skewness on various CVIs using different algorithms on the EL dataset with dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C Means, and (**d**) GMM.

- Sub Clustering; For the EL dataset without DR, the threshold for all algorithms was set to $\frac{1}{3}$ of the cluster's radius, except for GMM, which used $\frac{1}{4}$. K-Medoids and GMM showed unexpected improvements across all CVIs, while K-Means exhibited stable Dunn's Index initially but a later decline, and Fuzzy C-Means displayed inconsistent CVI behavior. For the EL dataset with DR, thresholds for all algorithms were set to $\frac{1}{4}$ of the cluster's radius. K-Means and GMM demonstrated improvements across all CVIs, while for K-Medoids and Fuzzy C-Means, Dunn's Index deteriorated instead of improving. The results are summarized in Figure 3.19 (without DR) and Figure 3.20 (with DR).

  Sub-clustering affected only DI, which deteriorated as the minimum inter-cluster distance decreased due to the formation of sub-clusters. This makes DI suitable for datasets with sub-clustering.

For outlier removal, prior study [73] recommended DI, while this study highlights CH and DB as more suitable metrics. Similarly, for differential density, prior research endorsed all CVIs except DI, whereas this study finds both DI and DB to be stable and less effective under density changes. However, the results align with previous work regarding the behavior of CVIs for overlapping profiles, kurtosis, skewness, and sub-clustering.

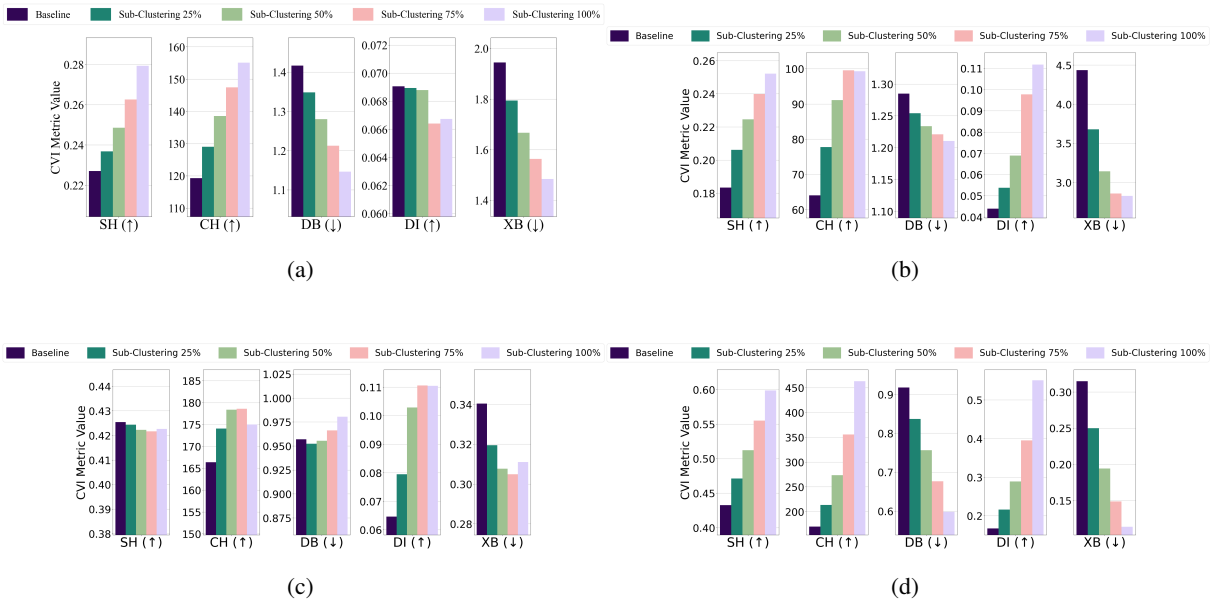


Figure 3.19: The effect of increasing the level of sub-clustering on various CVIs using different algorithms on the EL dataset without dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C Means, and (**d**) GMM.

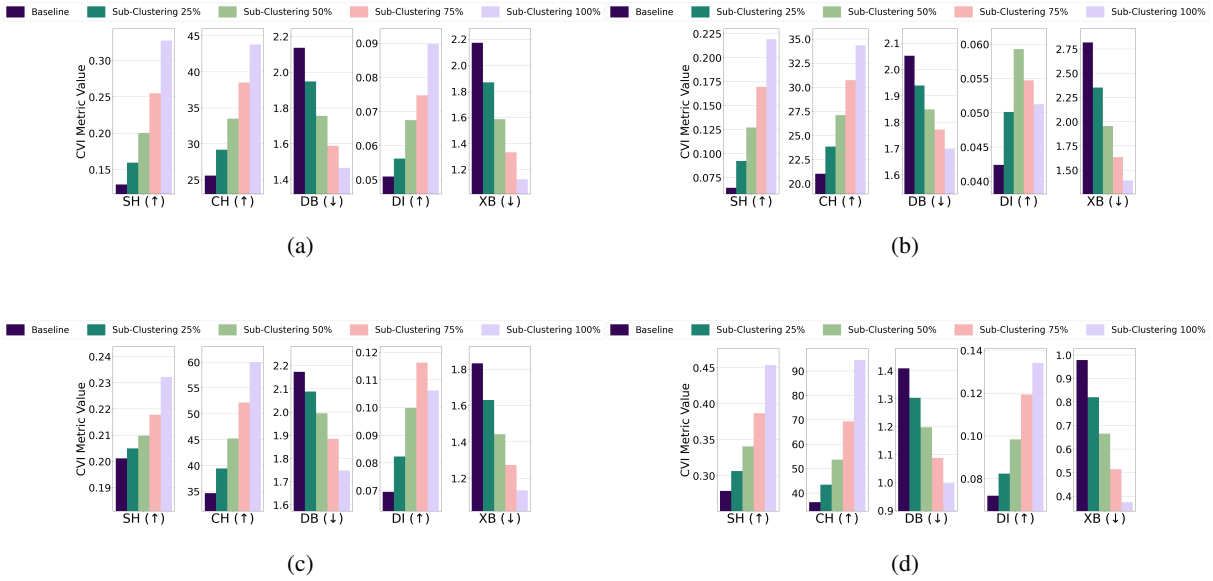


Figure 3.20: The effect of increasing the level of sub-clustering on various CVIs using different algorithms on the EL dataset with dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C Means, and (**d**) GMM.

3.3.4 Explainability Analysis

Clustering algorithms can successfully identify consumption patterns and segment daily load profiles into distinct groups; however, these algorithms do not explain why each data point was assigned to a particular cluster [83–87]. This lack of transparency obscures the criteria used to define the boundaries between clusters, making it unclear why certain data points are grouped together. Consequently, stakeholders and domain experts have difficulty verifying the relevance of discovered patterns, refining the clustering process, or addressing anomalies within individual load profiles. Ultimately, without explainability, clustering results remain opaque "black box" outcomes, limiting their practical utility for making informed decisions about energy management, consumer behavior analysis, or targeted interventions in load demand optimization.To overcome this limitation, two decision tree-based models were employed: an axis-aligned decision tree and a sparse oblique decision tree. These models offer human-readable rules and graphical representations that emphasize the defining characteristics of each cluster.

Axis-aligned decision tree form decision boundaries by splitting on individual 15-minute intervals within the electric load data. Each node in the tree selects a specific time interval (e.g., 06:30–06:45) and checks whether consumption at that interval is above or below a threshold (i.e., IF/ELSE splits). This structure yields straightforward, explainable conditions highlighting key consumption periods—for instance, early morning vs. peak evening usage. In contrast, the sparse oblique decision tree is particularly suited to high-dimensional data, because it constructs decision boundaries using linear combinations of multiple intervals. To ensure explainability, L1 regularization is applied to penalize large coefficients, zeroing out less-relevant time intervals, while a sparsity constraint limits the number of intervals at each split. This keeps the model's complexity manageable, while capturing the most influential consumption patterns driving cluster distinctions.

Despite their differences, both the axis-aligned and sparse oblique trees produce "IF-THEN" rules to explain cluster assignments. The sparse oblique model expresses these rules as:

$$\sum_i \text{Coefficient}_i \times \text{Consumption at Time}_i + \text{Intercept} \leq \text{Threshold} \tag{10}$$

where positive coefficients indicate that higher consumption makes the condition harder to

satisfy, and negative coefficients do the opposite. By contrast, the axis-aligned tree uses simple threshold-based comparisons on individual intervals:

$$\text{Consumption at Time } i \leq \text{Threshold}$$

or

$$\text{Consumption at Time } i > \text{Threshold}$$

From these rules, several key behavioral patterns emerge, such as morning-active households (high usage from 06:30 to 10:30), evening-active households (18:00 to 22:00), overnight appliance use (00:00 to 05:00), and midday activity (around 11:30 and 14:00). Because PCA was used for dimensionality reduction, a feature-mapping step ensured that each principal component could be traced back to its original 15-minute intervals.

To provide additional clarity on these explainability results for residential electricity usage profiles, Figure 3.21 evaluates the explainability of clustering results obtained using the axis-aligned decision tree and the sparse oblique decision tree for GMM clustering on the EL dataset. The analysis focuses on the number of rules generated by each model, their ability to explain clusters, and their role in defining the meaning of each cluster, first without DR and then with DR.

Without DR, the axis-aligned decision tree Figure 3.21b generated 9 distinct rules to explain all four clusters, providing detailed coverage. For instance, one rule classified Cluster 0 by relying on splits for Feature_3 $\leq$ 0.388 (corresponding to consumption at 00:45) and Feature_89 $\leq$ 0.21 (corresponding to consumption at 22:15). This interpretation indicates that Cluster 0 represents consumers with notable electricity usage late at night (00:45) but minimal consumption late in the evening (22:15). This explanation approach generalizes across all trees, allowing for meaningful insights into the specific behaviors that define each cluster. These single-feature thresholds ensured explainability and allowed clusters to be defined in terms of residential electricity demand profiles, such as distinguishing between consumers with early morning vs. late-night energy usage patterns. In contrast, the sparse oblique decision tree in Figure 3.21a explained 3 of the 4 clusters using only 3 rules. While more compact, these oblique splits were less explainable and failed to explain one

cluster. Nevertheless, the rules provided meaningful insights into combined energy usage patterns, highlighting the trade-off between simplicity and coverage. When DR was applied, the axis-aligned tree produced 12 rules to explain all five clusters (see Figure 3.21c). Despite the increased rule count, these rules remained clear, mapping clusters to specific behaviors like weekday midday usage or consistent energy consumption throughout the day. Meanwhile, the sparse oblique tree explained all five clusters with just 5 rules (see Figure 3.21d). These compact rules relied on broader patterns, effectively distinguishing high-consumption users from those with minimal fluctuations but sacrificing granularity.

From Table 3.2, it is evident that in the case of k-means clustering, cluster 4 consistently appeared as an outlier both without DR and with DR. However, the trees differed in their ability to explain this outlier. Without DR, the sparse oblique decision tree failed to explain cluster 4, leaving it unrepresented. Conversely, the axis-aligned decision tree successfully explained this outlier. With DR applied, the results were reversed: the sparse oblique tree succeeded in explaining cluster 4, while the axis-aligned tree failed to do so. This inconsistency in handling outliers indicates that neither tree is entirely reliable in capturing and explaining outliers when they exist in the dataset. These findings highlight a critical limitation of both models and underscore the need for more robust methods to address outliers in clustering explanations. Table 3.2 also shows that the axis-aligned tree often explained all clusters or left only one unexplained, demonstrating its robustness in capturing consumption patterns across different clustering algorithms. While it is true that the axis-aligned tree achieves better F1 scores overall, as highlighted in Table 3.3, this comes at the cost of generating more complex rule sets and larger decision tree graphs, as shown in Appendix A. Conversely, the sparse oblique tree balances its slightly lower F1 performance by providing fewer and simpler rules, which are more concise and explainable.

The axis-aligned tree's reliance on threshold-based splits for individual time intervals often results in an exponential increase in the number of rules as the number of clusters grows. For instance, the sparse oblique tree generated up to 9 rules for k-means, compared to 18 rules by the axis-aligned tree on the EL dataset without DR. This complexity made the axis-aligned tree harder to explain in larger datasets, whereas the sparse oblique tree maintained simplicity by leveraging linear combinations of features. Conversely, the sparse oblique tree, with careful tuning of parameters like

sparsity and lambda, provides a viable alternative, offering concise, explainable rules at the cost of occasionally missing clusters. However, as evidenced by its failure to explain cluster 4 without DR, the sparse oblique tree also suffers from reliability issues when outliers are present. Future research should investigate optimizing these models to balance explainability and cluster coverage, especially in complex datasets with outlier.

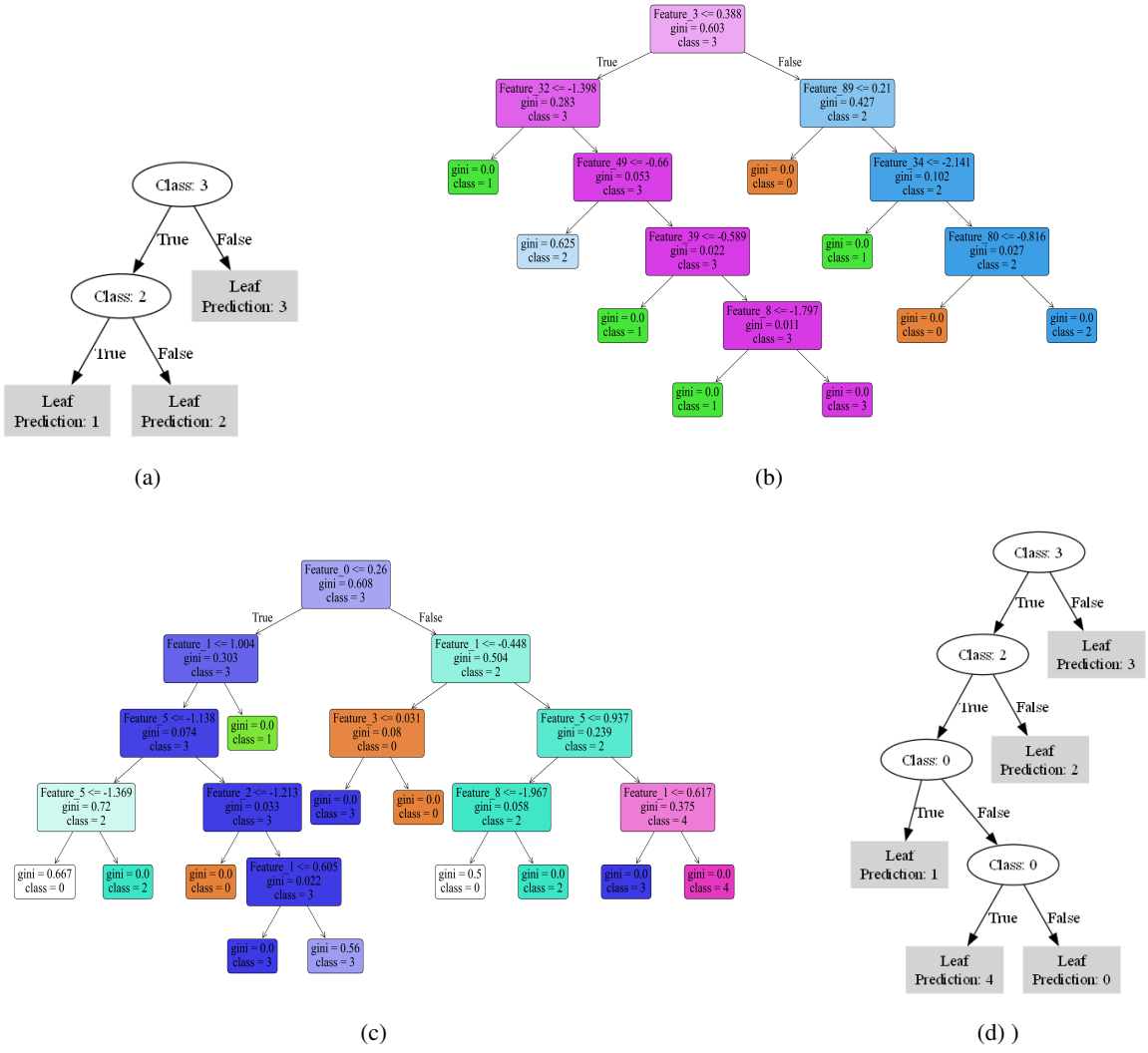


Figure 3.21: Axis aligned tree and sparse oblique decision tree results using GMM on EL dataset without and with dimensionality reduction (**a**) Sparse Oblique on EL without DR, (**b**) Axis Aligned on EL without DR, (**c**) Axis Aligned on EL with DR, and (**d**) Sparse Oblique on EL with DR.

Table 3.2: Explainability results for sparse oblique vs. axis-aligned decision trees on the EL dataset with and without dimensionality reduction for different clustering algorithms.

Algorithm	Model	Clusters Explained	Clusters Unexplained
Without Dimensionality Reduction			
K-Means	Sparse Oblique Tree	6/8	3, 4
	Axis-Aligned Tree	All 8	None
K-Medoids	Sparse Oblique Tree	7/9	0,9
	Axis-Aligned Tree	8/9	7
Fuzzy C-Means	Sparse Oblique Tree	2/3	0
	Axis-Aligned Tree	All 3	None
GMM	Sparse Oblique Tree	3/4	0
	Axis-Aligned Tree	All 4	None
With Dimensionality Reduction			
K-Means	Sparse Oblique Tree	All 8	None
	Axis-Aligned Tree	7/8	4
K-Medoids	Sparse Oblique Tree	7/8	0
	Axis-Aligned Tree	All 8	None
Fuzzy C-Means	Sparse Oblique Tree	2/3	1
	Axis-Aligned Tree	All 3	None
GMM	Sparse Oblique Tree	All 5	None
	Axis-Aligned Tree	All 5	None

Table 3.3: F1 score comparison for sparse oblique vs. axis-aligned decision trees on EL dataset with and without dimensionality reduction for different clustering algorithms.

Algorithm	Sparse Oblique Decision Tree F1 Score	Axis Aligned Decision Tree F1 Score
Without Dimensionality Reduction		
K-Means	0.692	0.913
K-Medoids	0.766	0.739
Fuzzy C-Means	0.613	0.992
GMM	0.681	0.991
With Dimensionality Reduction		
K-Means	0.907	0.846
K-Medoids	0.655	0.902
Fuzzy C-Means	0.616	1.000
GMM	0.951	0.964

To make the interpretation more tangible, a single representative example is selected based on rule simplicity to illustrate how decision tree rules explain clustering assignments. Given that 16 different decision trees (8 Axis-Aligned and 8 Sparse Oblique) were generated across multiple clustering models and dataset configurations, analyzing each one in detail would be impractical.

Below, Table 3.4 presents an excerpt from the Sparse Oblique Decision Tree applied to GMM clustering on the EL dataset without dimensionality reduction, with expanded interpretations that link threshold splits to real-world household behaviors.

Table 3.4: Interpretation of IF/ELSE rules of Axis Aligned decision tree for GMM on EL dataset without dimensionality reduction, linking thresholds to real-world household behaviors.

Rule	**Conditions**	**Predicted Cluster**	**Explanation**
1	00:45 $\leq$ 0.39 08:00 $\leq$ -1.40	Cluster 1	Very low consumption from midnight to early morning and extremely low in the morning. *Real-world implication: Households that barely use electricity overnight, possibly indicating no late-night appliances (e.g., washing machines or dishwashers) and minimal morning routines.*
2	00:45 $\leq$ 0.39 08:00 $>$ -1.40 12:15 $\leq$ -0.66	Cluster 2	Slightly higher morning usage, moderate midday usage. *Real-world implication: Households with moderate breakfast routines but still low midday activity; could be working families with children who leave by mid-morning.*
3	00:45 $\leq$ 0.39 08:00 $>$ -1.40 12:15 $>$ -0.66 09:45 $\leq$ -0.59	Cluster 1	Closer thresholds between morning and midday usage lead back to Cluster 1. *Real-world implication: Small differences in the 09:45 slot can switch a household between two morning-centric clusters.*
4	09:45 $>$ -0.59 02:00 $\leq$ -1.80	Cluster 1	Extremely low consumption at 02:00 but higher mid-morning consumption. *Real-world implication: Possibly households that shut off most devices before bed but wake up with a substantial breakfast or early appliance use.*

Continued on next page

Table 3.4 continued from previous page

Rule	Conditions	Predicted Cluster	Explanation
5	02:00 $>$ -1.80	Cluster 3	Slightly higher overnight usage. *Real-world implication: Households that keep some devices running overnight (e.g., fridge plus charging stations).*
6	00:45 $>$ 0.39 22:15 $\leq$ 0.21	Cluster 0	More consumption after midnight but low late-evening usage. *Real-world implication: Households that shift appliance usage to shortly after midnight but reduce activities earlier in the evening.*
7	22:15 $>$ 0.21 08:30 $\leq$ -2.14	Cluster 1	Late-evening changes trigger very low morning usage. *Real-world implication: Households with moderate to high consumption around 22:15 but extremely low by early morning.*
8	08:30 $>$ -2.14 20:00 $\leq$ -0.82	Cluster 0	Adjusted early-evening consumption leads back to Cluster 0. *Real-world implication: Active in morning but little consumption around 20:00.*
9	20:00 $>$ -0.82	Cluster 2	Slightly higher evening consumption sets Cluster 2. *Real-world implication: Households with pronounced evening activity (e.g., cooking dinner or entertainment).*

This example illustrates how decision tree rules translate clustering assignments into real-world electricity usage behaviors, providing a more transparent and actionable explanation of consumer load profiles. By revealing that small changes in specific time slots can move a household from one cluster to another, the rules highlight nuanced differences in daily routines such as how early residents start using appliances or whether certain devices run overnight. In the context of real-world applicability, these explainable rules can help energy managers, utilities, and policymakers design targeted initiatives. For instance, households showing steady overnight consumption could benefit from demand response programs that encourage shifting appliance use to lower-cost periods, or they might be prime candidates for time-of-use tariffs that encourage energy usage alignment with off-peak hours. Clusters with higher morning usage might receive energy-efficiency advisories promoting more efficient breakfast appliances or thermostats that manage early heating loads. Moreover, dynamic pricing strategies can be tailored to each cluster's characteristic load pattern, ensuring more effective peak load reduction and improved consumer cost savings.

Chapter 4

Explainable Deep Representation Learning for Clustering Building-Energy Time Series

4.1 Introduction

Global floorspace expanded by nearly five billion square metres between 2022 and 2023, raising the total constructed area to more than 260 billion square metres worldwide [88]. Although this 2% annual increase is slightly lower than pre-pandemic trends, it still reflects massive and sustained growth in the built environment. Much of this expansion is concentrated in emerging and developing economies, where of the nearly 51 billion square metres constructed during this period, more than half were built without any applicable building energy codes [88]. Building energy codes are regulatory frameworks that establish minimum requirements for energy performance in buildings, such as insulation standards, heating and cooling system efficiency, lighting, ventilation, and the integration of renewable energy technologies. The absence of such codes means that large portions of the new building stock are highly inefficient, locking in elevated levels of energy consumption and greenhouse gas emissions for decades to come. This situation underscores why the buildings and construction sector remains a critical area for achieving global net-zero carbon and resilience

goals. In 2023, the sector accounted for approximately 34% of energy-related CO_2 emissions and more than 32% of total global energy demand [88]. As one of the largest energy-consuming sectors worldwide , surpassing industry and transportation, buildings therefore represent both a major challenge and a unique opportunity. Improving their efficiency is essential not only for reducing emissions but also for supporting sustainable urbanization, advancing energy security, and meeting international climate targets. At the same time, the rapid deployment of smart meters, IoT devices, and advanced monitoring systems has transformed the way building energy performance can be monitored and analyzed. These technologies now generate unprecedented volumes of time-stamped data, capturing electricity consumption, heating and cooling loads, appliance usage, and even occupant behavior at fine temporal resolutions. Such data provides a valuable foundation for understanding consumption dynamics across diverse building types and user groups.

However, analyzing building energy data is inherently complex due to the diverse and dynamic factors influencing consumption patterns. Building energy time series display multi-scale temporal variability from hourly to seasonal trends driven by variations in occupancy behavior, meteorological conditions, and building operation schedules. These factors introduce non-stationarity and heterogeneity across buildings and temporal scales, making modeling and analysis particularly challenging [89]. These properties make them fundamentally different from other time-series domains and pose unique challenges for pattern discovery and clustering.

To extract actionable insights from such complex energy data, clustering methods have become essential analytical tools in building-energy research [90]. They enable the discovery of consumption archetypes, the identification of abnormal patterns, and the segmentation of buildings or time periods with similar operational behaviors. In particular, time-series clustering provides a powerful framework because it retains the temporal continuity and dependency structure of energy signals, allowing the analysis to capture recurrent load profiles, occupancy-driven variations, and cyclical patterns that would be lost through simple aggregate methods. This temporal perspective allows energy analysts to move beyond static characterizations toward dynamic insights that support energy management, demand forecasting, and policy planning [90].

However, traditional clustering algorithms such as k-means or GMM struggle to capture these dynamics. They often rely on feature extraction, require predefined similarity measures in the case

of k-means or parametric assumptions in the case of GMMs, and cannot easily handle the high dimensionality and noise of time-series data [91–93]. Moreover, they typically assume simple statistical distributions that do not accurately reflect the complex, non-stationary nature of building-energy time series.

To overcome these limitations, deep learning–based clustering methods have emerged as an alternative for analyzing high-dimensional and noisy time-series data [92, 94]. Unlike traditional approaches, deep clustering frameworks jointly learn both latent representations and cluster assignments within a unified optimization process. This integration enables the model to automatically capture complex, nonlinear, and multi-scale dependencies that characterize energy-consumption patterns. Such models are particularly valuable in the energy domain, as they can separate overlapping load behaviors, capture temporal variability, and reveal latent consumption regimes in an unsupervised manner. As a result, deep time-series clustering provides a data-driven pathway toward discovering latent and operationally meaningful patterns in large-scale energy datasets.

Despite their potential, deep clustering models introduce new challenges that are particularly pronounced in the context of unlabeled energy data. First, their performance is highly sensitive to hyperparameter settings which are difficult to tune without labeled validation. Deep neural networks (DNNs) require extensive hyperparameter tuning, which is computationally expensive. Traditional search strategies such as grid search, random search, or Bayesian optimization share well-known drawbacks: they are costly in high dimensions, require many evaluations, are sensitive to noise, and are hard to scale or parallelize, especially when categorical or conditional parameters are involved [70, 95]. To address this, more efficient optimization methods such as genetic algorithms (GAs) have been proposed, offering faster convergence and better scalability in high-dimensional search spaces.

Second, the black-box nature of deep models limits their interpretability, making it difficult to understand why certain energy profiles are grouped together or what physical meaning each cluster represents. While these models frequently achieve higher predictive accuracy and clustering quality, their decision processes remain opaque [96]. This has motivated increasing attention toward

explainable artificial intelligence (XAI), which aims to provide transparency into learned representations and clustering outcomes. Existing XAI approaches range from built-in interpretable architectures to post-hoc explanation methods [97, 98], each with distinct trade-offs between accuracy and interpretability. Integrating such techniques is essential for making deep clustering not only accurate but also explainable and trustworthy.

4.2 Litterature Review and Background

4.2.1 Representation learning–based methods

While traditional clustering methods have contributed valuable insights, they face notable limitations when applied to energy time-series data. A primary drawback lies in the separation of feature extraction from clustering. Handcrafted features or statistical transformations are often used as a pre-processing step, but these may not yield representations that are well suited for clustering tasks [24]. Moreover, identifying which features are truly relevant typically requires strong domain expertise, meaning that the quality of clustering is highly dependent on the adequacy of chosen descriptors. [93]. A second challenge is the sensitivity of clustering results to the similarity metric itself. Numerous measures have been proposed yet studies consistently show that the choice of metric can drastically affect performance, especially in the presence of noise or outliers [99]. Even when robust similarity measures are employed, they may still fall short without appropriate dimensionality reduction, as the high dimensionality of time-series data complicates the search for meaningful structures [99]. These limitations, among others, have motivated the shift toward deep representation learning, where feature extraction and clustering are jointly optimized, allowing latent spaces to be learned directly from data rather than predefined by hand.

Several deep clustering approaches have been proposed in the literature. Xie et al. proposed Deep Embedded Clustering (DEC), which combines a stacked autoencoder with an iterative refinement process that minimizes the Kullback–Leibler divergence between soft cluster assignments and a sharpened target distribution, achieving strong results on image and text benchmarks [100]. Caron et al. introduced DeepCluster, an alternating scheme that repeatedly applies k-means to CNN features and uses the cluster assignments as pseudo-labels to update the network, scaling effectively to

large datasets such as ImageNet and YFCC100M [101] . Yang et al. extended this line where a dual autoencoder is trained to produce robust embeddings that are then optimized jointly with a spectral clustering objective, demonstrating superior performance on image datasets [102]. Paparrizos et al. have focused on the theoretical underpinnings of deep clustering but without applying the methods to real-world data [25]. While these methods highlight the potential of end-to-end deep representation learning, they were primarily validated on images rather than energy time series, leaving open questions about their applicability to temporal data.

This gap is exactly what the paper by Lafabregue et al. set out to address. The authors designed a general pipeline with three components: an architecture, a pretext (representation learning) loss, and a clustering loss [28]. In this framework, the architecture defines the encoder network used to map time series into latent spaces, the pretext loss ensures that the learned features capture meaningful temporal structure, and the clustering loss aligns the latent space with cluster objectives. They systematically tested different combinations across both univariate and multivariate time-series archives, benchmarking against traditional clustering methods. Their work provides a structured foundation for how deep learning can be applied to time-series clustering.

4.2.2 Hyperparameter Tuning

A consistent limitation of many clustering models, such as k-means, is the need to specify the number of clusters k a priori. A choice that strongly affects performance. This challenge extends to deep learning models, where the issue is magnified. In fact, the number of hyperparameters and the size of the search space are far greater than in traditional methods. The performance of deep learning approaches is highly sensitive to these hyperparameters, including architectural settings, learning rates, batch sizes, and latent dimensions as examples. For this reason, hyperparameter optimization has become a crucial step in building effective deep clustering models. Nevertheless, many studies still fix hyperparameters to arbitrary or default values without systematic tuning (e.g., [28]), which can limit model generalization and make reported results suboptimal. This section reviews the main strategies for hyperparameter tuning.

The state of the art methods for hyperparameters tunning are manual search, grid search, random search and bayesian search.

Manual search refers to the process of manually choosing a set of parameters to evaluate. The main problem with this method is its difficulty in reproducing results [70, 103].

The reproducibility problem is addressed by grid search, which systematically explores a predefined set of hyperparameter values by training and evaluating the model for every possible combination in the grid. The main drawback of this method lies in its computational cost. The time complexity of grid search is:

$$\mathcal{O}\left(\left(\prod_{i=1}^{n}|H_i|\right)\ T\right) \tag{11}$$

where n is the number of hyperparameters being tuned, $|H_i|$ is the number of candidate values for the i^{th} hyperparameter, and T is the time required to train and evaluate the model once. The cost therefore grows with the number of parameters and their candidate values. This means grid search can become very expensive for large grids or slow-to-train models. It scales exponentially with the number of hyperparameters therefore it is inefficient for high-dimensional search spaces [103–105].

Random search is an alternative to grid search that samples a fixed number of random hyperparameter combinations from the search space rather than evaluating every possible combination. This approach significantly reduces computational cost, especially in high-dimensional spaces, because its time complexity grows linearly with the number of trials. Despite its simplicity, random search often discovers competitive or even superior solutions compared to grid search because it avoids wasting evaluations on unimportant parameters and explores the space more broadly. However, it suffers from being non-adaptive because the hyperparameter sets selected are not guided by previously observed results. This lack of adaptivity can lead to inefficiency compared to adaptive methods. Furthermore, random search provides no guarantee of covering the entire search space. Its results may vary between runs, and it can still be computationally expensive when model training is slow [105]. Bergstra et al. [103] propose random search as a stronger default than grid search: sample each hyperparameter independently from well-chosen distributions and evaluate. It shows that RS finds good configurations much faster in high-dimensional spaces and argue it should be a standard baseline.

Bayesian optimization (BO) is an adaptive hyperparameter tuning method that iteratively builds

a surrogate probabilistic model of the objective function, commonly using Gaussian Processes to predict which hyperparameter configurations are most promising [106]. Unlike grid or random search, which evaluate points independently, Bayesian optimization leverages previous evaluations to balance exploration of new regions and exploitation of known good areas [105]. This adaptivity allows it to converge to near-optimal hyperparameters with far fewer evaluations, making it particularly well-suited for expensive deep learning models. However, classical Gaussian process–based approaches face scalability issues, as they exhibit cubic computational complexity with respect to the number of evaluated points, and they may perform poorly in high-dimensional search spaces [95]. In addition, they often require specialized kernels for complex configuration spaces and carefully tuned hyper-priors to remain robust [95]. These are the factors that can make Bayesian optimization less straightforward to apply in very large or heterogeneous search spaces.

To address these limits, Vincent & Jidesh [106] improve BO by using evolutionary algorithms to maximize the acquisition, and Falkner et al. [95] combine a Tree-structured Parzen Estimator style surrogate with hyperband's early stopping (BOHB). They start many small budget runs and then allocate more budget to the promising ones, typically outperforming random search and plain hyperband at the same compute.

Given the limitations of the methods above, it is natural to turn to more advanced approaches. Evolutionary computing is inspired by Darwinian evolution and defines a family of evolutionary algorithms [107]. These algorithms search with a population of candidate solutions that compete and improve over generations. They select the best candidates, recombine parents through crossover to create new ones, and then apply mutation to introduce diversity. In hyperparameter optimization, this means evolving a population of hyperparameter configurations, selecting the ones that perform best, and mutating them to create new configurations until the search budget is exhausted.

Slowik and Kwasnicka [108] present a state of the art review of the main evolutionary algorithm families for engineering. They introduce genetic algorithms, genetic programming, differential evolution, evolution strategies, and evolutionary programming with short descriptions and pseudo code, then review real applications and variants. They also highlight open issues for this family of methods.

Vikhar [107] complements this with a concise critical overview of evolutionary algorithms and

their generic workflow, explains advantages over classical optimization, and notes extensions such as memetic and distributed evolutionary variants for harder problems.

Moving from reviews to a concrete system, Young et al. [104] introduce MENNDL, which distributes a genetic algorithm across many compute nodes and evaluates CNNs in Caffe. They evolve kernel sizes and filter counts on CIFAR-10 using a population of 500 over 35 generations and report steady gains. The best models use smaller kernels and more filters than the defaults. The authors argue that evolutionary search leverages past results to focus on promising regions more effectively than random or manual approaches.

Building on this evidence, Jaderberg et al. [20] propose Population Based Training (PBT), an asynchronous exploit-and-explore scheme that periodically "exploits" by copying weights and hyperparameters from stronger performers and then "explores" by perturbing or resampling hyperparameters. This yields adaptive schedules rather than a single fixed setting. With applications shown in deep reinforcement learning, machine translation with Transformers, and generative adversarial networks (GANs), PBT improves stability and final performance. Building on this, the present study operationalizes PBT for unlabeled deep time-series clustering, using internal validity indices as the fitness signal and explicit collapse checks to prevent degeneracy.

4.3 Methodology

This section outlines the methodological framework adopted in this study, detailing the datasets, pre-processing procedures, model architectures, and training strategies employed, followed by hyperparameter tuning and XAI employed technique.

4.3.1 Datasets and Pre-processing

For both datasets, we set the number of clusters k using the elbow heuristic and the gap statistic on the z-scored daily profiles. The elbow heuristic inspects the curve of within-cluster sum of squares versus k and chooses the point of diminishing returns. The gap statistic compares the observed dispersion to that of Monte Carlo reference datasets and selects the smallest k within one standard error of the maximum gap.

Univariate Dataset

The Electricity Load Diagrams 2011–2014 dataset [75], publicaly available from the UCI Machine Learning Repository, contains 15-minute resolution power consumption measurements for 370 clients of an energy distribution company in Portugal collected between 2011 and 2014. The dataset contains no missing values, however, some clients were added after 2011, and their earlier consumption entries were recorded as zeros. Each day includes 96 evenly spaced readings. The raw dataset, initially shaped as 140,160 by 370, was trimmed to include exactly four years of data and reshaped into a three-dimensional tensor of 370 by 1,460 by 96, representing 370 users, 1,460 daily profiles, and 96 quarter-hour intervals per day. Users with fewer than 920 valid days were removed, resulting in 322 retained users with consistent daily records. For each retained user, daily sequences were normalized using z-score normalization to eliminate scale differences and stabilize model convergence. After normalization, all daily sequences were flattened into independent samples, producing a final dataset of shape 470,120 by 96 by 1, corresponding to 470,120 normalized daily load profiles, each containing 96 consecutive measurements. To visually illustrate the diversity in consumption behaviors, Figure 4.1 presents examples of normalized daily electricity load profiles for four randomly selected users from the dataset. Each curve corresponds to one full day of consumption, rescaled using z-score normalization to emphasize intra-day variations rather than absolute magnitude. As seen in the figure, distinct temporal patterns emerge across users. Some profiles exhibit pronounced peaks during specific hours, suggesting concentrated activity or appliance usage, while others maintain flatter consumption trajectories indicative of steady, low-variability demand.

Multivariate Dataset

The ASHRAE Energy Predictor III dataset [109], publicly released as part of the Kaggle competition, contains three years of hourly energy consumption data collected from over one thousand commercial, residual and institutional buildings across multiple sites worldwide. Each record includes a timestamp, building identifier, and four metering channels measuring electricity, chilled

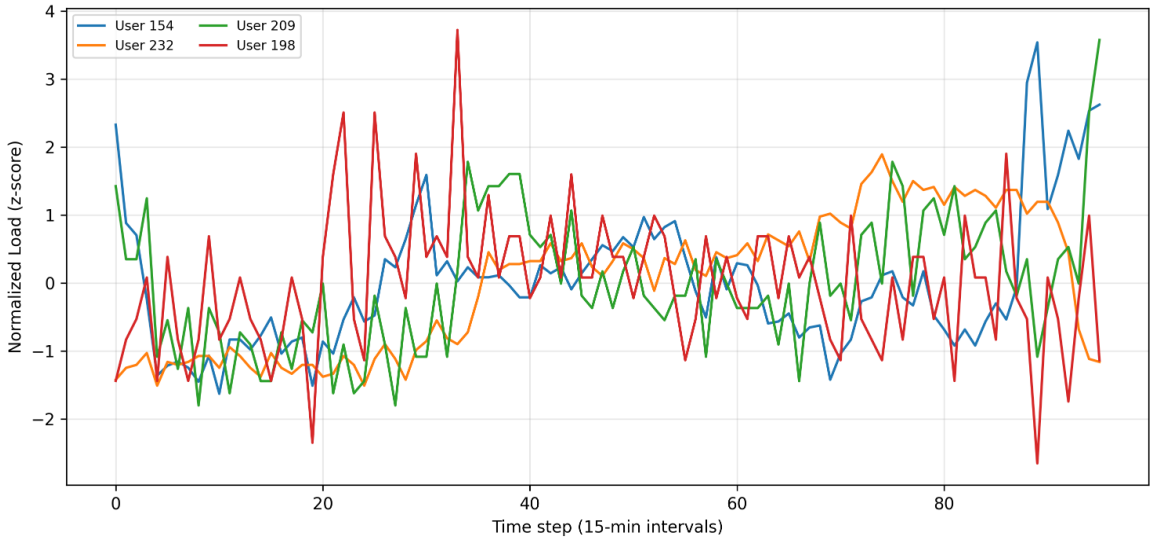


Figure 4.1: Example of daily electricity load profiles from random users

water, steam, and hot water usage. The raw dataset comprised 12,393,999 rows across six variables. To ensure temporal consistency, all entries with missing values were removed, reducing the dataset to 106,842 complete records. Buildings monitored for fewer than 328 days (approximately 11 months) were also excluded to eliminate short or fragmented series that could bias clustering. The resulting data were reshaped into a multivariate time-series format of shape (4,261, 24, 4), corresponding to 4,261 complete daily profiles, each with 24 hourly time steps and 4 energy features. All daily sequences were normalized using z-score normalization to mitigate scale differences between meters and stabilize model convergence.

Figure 4.2 illustrates the log-transformed distribution of the four metering channels after data cleaning and filtering. As shown, electricity readings exhibit a multimodal distribution with clear peaks reflecting distinct consumption regimes and building usage patterns. Chilled water and steam show broader, right-skewed distributions reflecting seasonal and system-level variability in cooling and heating demand. In contrast, hot water consumption is concentrated near lower values, with a sharp initial spike indicating frequent low-load periods typical of irregular demand patterns.

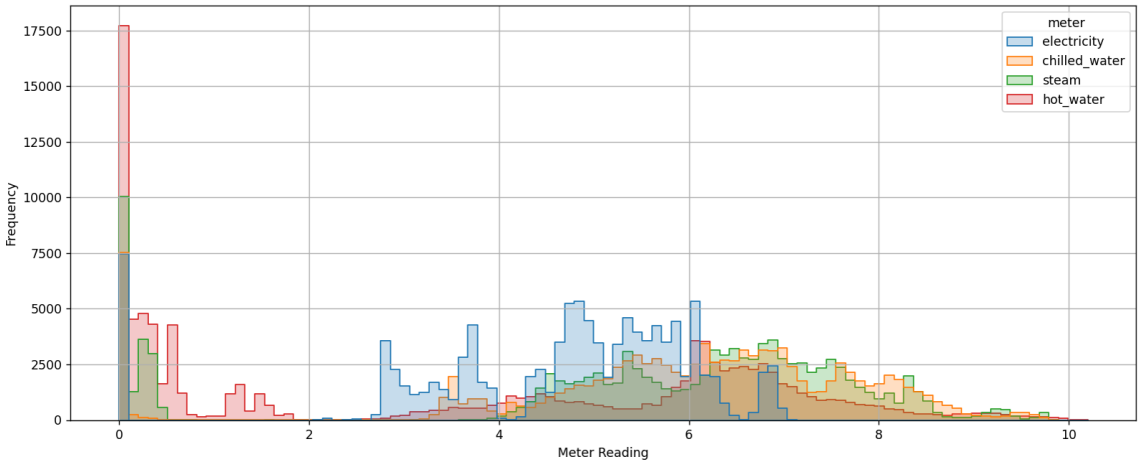


Figure 4.2: Log-transformed distribution of meter readings for multivariate dataset

4.3.2 Evaluation

To assess the quality of the learned representations and clustering performance, five internal evaluation metrics were computed: silhouette score, Davies–Bouldin Index (DB), Calinski-Harabasz Index (CH), Dunn Index, and Xie-Beni Index (XB). The silhouette score measures cohesion–separation and ranges from –1 to 1, with values closer to 1 indicating better-defined clusters [64]. The Davies–Bouldin Index quantifies the average similarity between each cluster and its nearest neighbor, with lower values indicating better structure [66]. The Calinski–Harabasz Index quantifies the balance between cluster separation and within-cluster cohesion, where higher scores correspond to better clustering [65]. The Dunn Index measures the ratio between minimum inter-cluster distance and maximum intra-cluster diameter, with higher values reflecting stronger separation [67]. Finally, the Xie–Beni Index assesses cluster compactness relative to centroid separation, with lower scores indicating higher-quality clusters [68].

Each model was evaluated at two stages: using the latent representations extracted from the encoder, and on the post-clustering results obtained. This two-level evaluation enabled the identification of models that learned meaningful latent structures and those whose clustering stages improved or degraded separability.

For each metric, individual ranks were computed across all models. These ranks were then aggregated into average total rank, computed as the mean of all five metric ranks, representing the

unified performance score for each model. This average total rank served as the final criterion for model comparison, ensuring that configurations demonstrating both high intra-cluster compactness and strong inter-cluster separation were prioritized.

A model was flagged as collapsed if it produced either a single unique cluster label or extreme internal validity scores such as DB < 0.01 or silhouette > 0.98, which in our pipeline indicate vanishing within-cluster dispersion and near-zero centroid separation. The post-clustering evaluation was used as the primary selection criterion, as it reflects the model's ultimate ability to form distinct and meaningful clusters. However, when a model remained stable at the latent stage but collapsed after clustering or vice versa , both evaluations were jointly analyzed to better interpret the consistency of its learned representations.

4.3.3 Baseline Methods

Following the algorithmic distinctions outlined in the literature review, baseline experiments were conducted using classical distance-, shape-, and density-based traditional clustering methods to establish reference performance prior to introducing deep representation learning. These baselines are essential for assessing whether the additional complexity of deep clustering architectures provides benefits over traditional approaches when applied to building energy time-series data.

Specifically, we applied k-means with Euclidean distance, k-means with DTW distance, K-Shape, and DBSCAN on the preprocessed energy consumption profiles. This selection encompasses the main algorithmic families identified in prior studies and ensures that improvements observed in deep models are evaluated against strong baselines commonly used in energy analytics.

To extract cluster structure without fixing the number of clusters k, we employed DBSCAN with a focused parameter refinement strategy. Time-series were first flattened into a normalized feature representation. Based on an initial solution ($\varepsilon \approx 0.1366$, min_samples = 8, cosine distance), we performed a local grid search around these values to stabilize cluster quality. Specifically, we varied min_samples in $\{6,7,8,9,10,11,12\}$ and swept ε within a narrow multiplicative range [0.7,1.3] of the reference value. For each configuration, DBSCAN was applied and clusters were evaluated on inlier points only (excluding noise) using silhouette score, while also tracking the number of clusters and

noise ratio.

To prioritize meaningful structure, we selected only configurations that produced at least three clusters and maintained noise below 15% of samples. If none satisfied these constraints, the highest silhouette configuration among those with 3 clusters was chosen. Final cluster quality was assessed using silhouette, Davies–Bouldin, Calinski–Harabasz, Dunn, and Xie–Beni indices.

Overall, these baselines serve not only as algorithmic references but also as application-grounded controls, allowing us to quantify how much of the inherent structure in building energy consumption behaviors can be captured through traditional methods compared with deep clustering approaches.

4.3.4 Pipeline Design of the Deep Time-Series Clustering Framework

The pipeline follows a modular design that first encodes time-series data into latent representations through deep neural architectures, then refines these embeddings using pretext and clustering losses. It comprises seven architectures ranging from convolutional to recurrent networks, five pretext losses, and seven clustering losses. We fixed the size of embedding layer to D_{ls} = 320.

Architecture

FCNN: The network consists of three fully connected layers with sizes $d \rightarrow 500 \rightarrow 500 \rightarrow 2000 \rightarrow D_{ls}$, where d corresponds to the input dimensionality and D_{ls} denotes the latent space size. The decoder mirrors this structure symmetrically, reconstructing the input from the latent embedding but excluding the final embedding layer.

Residual CNN (Res-CNN): Instead of stacking plain convolutional layers, this design uses residual connections, which create shortcut paths across blocks and help maintain gradient flow in deeper models. The specific encoder we implement contains three residual blocks, where each block applies convolutions of decreasing kernel size (8, 5, and 3) with 64 filters. Batch normalization and ReLU activations follow each convolution, ensuring stable learning. A global average pooling layer is used to compress temporal information before projecting into the latent embedding space of size D_{ls}. By combining residual learning with temporal convolutions, this architecture is well-suited to capture both local and mid-range structures in energy load profiles while remaining robust to vanishing gradients.

Dilated CNN (DCNN): This architecture applies causal padding, which ensures that the output at time step t depends only on the current and past inputs, by padding only at the beginning of the sequence. This preserves the temporal ordering and prevents information from the future leaking into the representation. It is combined with exponentially dilated convolutions, where the dilation factor controls the spacing between kernel elements. A dilation factor of $d = 1$ corresponds to a standard convolution, while exponentially increasing factors $(1, 2, 4, 8, \dots)$ expand the receptive field rapidly, allowing the model to capture long-range dependencies without a proportional increase in parameters. This combination enables the network to model both short- and long-term patterns in energy consumption efficiently. In our implementation, each convolutional layer uses 40 filters with a kernel size of 3. The number of layers and the dilation rates are not fixed arbitrarily, instead, they are computed using the function proposed by [28], which adjusts the depth and dilation schedule to the length of the input sequence. The convolutional stack is followed by a fully connected embedding layer of size D_{ls}, producing the latent representations used for clustering.

Bi-LSTM (Bi-LSTM): Recurrent neural networks (RNNs) are widely used for sequential modeling, but their ability to capture long-range dependencies is limited. Long Short-Term Memory (LSTM) units address this limitation by introducing gated memory cells that regulate the flow of information and mitigate the vanishing gradient problem. The bidirectional variant (Bi-LSTM) extends this idea by processing the input sequence in both forward and backward directions, so that the representation at each time step encodes information from past and future contexts simultaneously. This property is especially relevant in energy load profiles, where consumption at a given time often depends not only on previous hours but also on upcoming periods (e.g., anticipation of morning or evening peaks).

In our implementation, the encoder is composed of two stacked Bi-LSTM layers. The first layer has a fixed hidden size of 50 units, while the second uses a hidden size of $\lfloor D_{ls}/2 \rfloor$. The final hidden state of the second Bi-LSTM layer is used as the latent representation. The decoder mirrors this structure, reconstructing the original input from the latent embedding. This symmetric design ensures that the encoder learns compact yet informative embeddings, while the decoder regularizes the latent space by enforcing reconstruction fidelity.

Bidirectional GRU (Bi-GRU): The Bi-GRU encoder adopts the same structural design as the

Bi-LSTM but substitutes LSTM cells with Gated Recurrent Units (GRUs). Unlike LSTMs, GRUs merge the cell state and hidden state, and reduce the gating mechanism to only two gates: an update gate and a reset gate. This simplification reduces the number of parameters and accelerates training while preserving the ability to capture temporal dependencies.

In our implementation, two stacked Bi-GRU layers are used, with hidden sizes matching those of the Bi-LSTM encoder (50 units in the first layer and $\lfloor D_{ls}/2 \rfloor$ in the second). The final hidden state of the second Bi-GRU layer is taken as the latent representation, and the decoder mirrors the encoder structure. This design offers greater computational efficiency compared to Bi-LSTM, while maintaining sufficient representational capacity. Such a balance between efficiency and expressiveness is particularly advantageous in large-scale energy applications, where both model scalability and temporal accuracy are critical.

Dilated RNN (DRNN): To improve the modeling of long-term temporal dependencies in energy data, we also consider a dilated recurrent design. Unlike standard RNNs that update at every timestep, dilated RNNs introduce skips in the recurrence path, allowing the network to capture patterns over multiple temporal scales while reducing training instabilities.

In our experiments, the encoder is implemented as three stacked bidirectional GRU layers with dilation factors of 1, 4, and 16. We adopt a hidden size of 100–50–50 across the layers. The latent representation is obtained from the final hidden state of the last bidirectional layer ($50 \times 2 = 100$). The decoder is designed as a single GRU layer with 400 units, initialized by concatenating the final hidden states of all encoder layers. It generates reconstructions in an autoregressive way, where each timestep is conditioned on the previous output, starting from a zero vector.

This design allows the model to cover short-term variations (dilation $= 1$), mid-range dependencies (dilation $= 4$), and longer seasonal effects (dilation $= 16$) within a compact architecture.

Attention Mechanism: While recurrent encoders capture temporal dependencies sequentially, they treat all timesteps with equal importance. In energy data, however, specific intervals such as peak demand periods or transition phases may be more informative than others. To address this, we incorporate a temporal attention mechanism on top of a bidirectional GRU encoder. Attention assigns adaptive weights to different timesteps, allowing the model to emphasize the most relevant parts of a load profile when forming latent representations.

This model combines a bidirectional GRU encoder with a temporal attention layer. The Bi-GRU hidden size is set adaptively: 64 units when the training set is small (fewer than 250 samples) and 512 units otherwise. The attention mechanism then aggregates the outputs of the forward and backward directions, producing a latent embedding of size D_{ls}.

For the decoder, we follow the setup proposed by [28], which employs a single Bi-GRU layer with a hidden size of $\lfloor D_{ls}/2 \rfloor$. This design choice reduces complexity and ensures consistency with the other architectures in our framework.

Pretext Loss

Reconstruction Loss: A common pretext task for representation learning is to train the encoder–decoder model to reconstruct the original input sequence. The encoder must extract latent features that are sufficiently informative to allow the decoder to reconstruct the input with minimal error. This pretext task thereby enforces embeddings that capture the essential temporal structure of the data and are suitable for subsequent clustering.

Autoencoders (AEs) consist of two parts which are: an encoder that is a nonlinear mapping $f_\theta : X \to Z$ that projects the data into a latent space, and the decoder that is a nonlinear mapping $g_\phi : Z \to X$ that maps latent variables back to the data space. For a given sample $x_i \in X$, the encoder produces a latent representation $z_i = f_\theta(x_i)$ and the decoder outputs its reconstruction $\hat{x}_i = g_\phi(z_i) = g_\phi(f_\theta(x_i))$. An AE is assessed by how accurately it can reconstruct each input sample x_i as $\hat{x}_i$. To train the AE weights, we minimize the reconstruction loss defined by the mean square error:

$$\mathcal{L}_{\text{rec}} = \frac{1}{n} \sum_{i=1}^{n} \|x_i - g_\phi(f_\theta(x_i))\|_2^2 , \tag{12}$$

where n denotes the number of training samples.

Triplet Loss: The triplet loss offers an alternative to reconstruction-based objectives by training only an encoder, thereby eliminating the need for a decoder. This reduces both computational overhead and the risk of model misspecification due to decoder design. Originally introduced for face recognition [110], the loss encourages embeddings of similar samples to be close while pushing apart those of dissimilar ones.

In the adaptation to time series [111], the notion of similarity is defined through subsequence sampling. From a given series x_i, a random subsequence is selected as the anchor x_{ref}. Another subsequence from the same series serves as the positive sample x_{pos}, under the assumption that segments within the same trajectory share contextual structure. A subsequence from a different series x_j, $j \neq i$ acts as the negative sample x_{neg}. To improve robustness, multiple negatives are typically considered, with the number controlled by a parameter K_{triplet}.

The loss function is given by:

$$\mathcal{L}_{\text{triplet}} = -\log \sigma\big(f(x_{\text{ref}})^\top f(x_{\text{pos}})\big) - \sum_{l=1}^{K_{\text{triplet}}} \log \sigma\big(-f(x_{\text{ref}})^\top f(x_{\text{neg}}^{(l)})\big) \tag{13}$$

where $f(\cdot)$ denotes the encoder and σ the sigmoid activation. The first term draws anchor and positive embeddings together, while the second repels anchors from negative samples.

Variational Autoencoder (VAE): This framework extends the standard autoencoder by introducing a probabilistic latent space that regularizes the encoder's representations. Instead of mapping each input sequence to a single deterministic point in the latent space, the encoder estimates the parameters of a probability distribution typically a Gaussian characterized by a mean vector μ and a covariance Σ. A latent sample z is then drawn from this distribution and passed to the decoder for reconstruction. This stochastic formulation enforces smoother latent manifolds and improves generalization.

The VAE is trained by maximizing the Evidence Lower Bound (ELBO), which balances reconstruction accuracy and latent regularization. The objective function can be expressed as:

$$\mathcal{L}_{\text{VAE}} = E_{q_\theta(z|x)}[\log p_\phi(x|z)] - \text{KL}\big(q_\theta(z|x) \,\|\, p(z)\big), \tag{14}$$

where $q_\theta(z|x)$ denotes the approximate posterior distribution learned by the encoder, $p_\phi(x|z)$ is the likelihood modeled by the decoder, and $p(z)$ represents a prior distribution, typically a standard normal $\mathcal{N}(0, I)$. The first term encourages accurate reconstruction of the input sequence, while the second term minimizes the Kullback–Leibler (KL) divergence between the learned posterior and the prior, thereby constraining the latent representations to follow a well-behaved distribution.

Multi-Reconstruction Loss (Multi-rec): The DEPICT model [112] presented in the next section extends the conventional autoencoder training objective by enforcing reconstruction consistency across all network depths. Instead of computing the mean squared error solely between the input and the final output, the multi-reconstruction loss evaluates reconstruction quality at each encoder–decoder layer pair. This approach encourages the network to preserve information throughout its entire hierarchy rather than concentrating it exclusively in the deepest latent representation.

Formally, for a sample x_i, let $z_i^{(\ell)}$ denote the output of the ℓ-th encoder layer and $\hat{z}_i^{(\ell)}$ its corresponding decoder activation, with L the total number of layers. The loss is defined as:

$$\mathcal{L}_{\text{multi-rec}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{\ell=0}^{L-1} \frac{1}{|z_i^{(\ell)}|} \left\| z_i^{(\ell)} - \hat{z}_i^{(\ell)} \right\|_2^2, \tag{15}$$

where $|z_i^{(\ell)}|$ denotes the size of the ℓ-th layer's output.

This formulation ensures that every level of abstraction contributes to maintaining the integrity of the reconstructed signal, thereby promoting stable and information-rich latent features. However, it requires that the decoder be a strict mirror of the encoder, which restricts its use to feed-forward architectures such as FCNN and CNN, while excluding recurrent designs.

Generative Adversarial Network (GAN): Generative Adversarial Networks (GANs), first introduced by [113], learn data representations through a minimax game between two neural networks: a generator G and a discriminator D. The generator aims to synthesize samples that resemble the true data distribution, while the discriminator seeks to distinguish between genuine and generated examples. Through this adversarial process, the generator progressively captures the underlying structure of the data without relying on explicit reconstruction objectives.

The standard GAN objective is formulated as:

$$\mathcal{L}_{\text{GAN}} = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))], \tag{16}$$

where $p_{\text{data}}(x)$ denotes the real data distribution and $p_z(z)$ represents a prior distribution (typically a Gaussian) from which latent variables are sampled.

In this formulation, GANs do not include an encoder, meaning that no explicit latent mapping

from data to feature space is learned. Consequently, the vanilla GAN cannot be directly employed for clustering tasks, which require encoded representations. Later works, such as [114, 115], introduced encoder components into adversarial frameworks to enable joint representation learning and clustering.

In our framework, we employ the original GAN pretext loss as an auxiliary adversarial objective. However, since it lacks an encoder, it can only be applied within architectures explicitly designed for clustering. Therefore, it is used exclusively in conjunction with the ClusterGAN clustering loss (see next section), which integrates an encoder to bridge the adversarial and clustering components.

Clustering Loss

Deep Embedded Clustering (DEC): This loss proposed by [100] was one of the first models to jointly learn feature representations and cluster assignments within a deep learning architecture. The method proceeds in two stages. In the first phase, an autoencoder is trained to initialize the encoder parameters and produce a stable latent representation. The decoder is then discarded, and the encoder is retained as a feature extractor. An initial set of cluster centroids $\{\mu_j\}_{j=1}^{k}$ is obtained by performing k-means on the latent embeddings.

In the second phase, the clustering layer maps each latent embedding $z_i = f_\theta(x_i)$ to a soft cluster assignment using the Student's t-distribution as a similarity kernel:

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2/\alpha)^{-(\alpha+1)/2}}{\sum_{j'}(1 + \|z_i - \mu_{j'}\|^2/\alpha)^{-(\alpha+1)/2}}, \tag{17}$$

where α controls the degrees of freedom and is typically fixed to $\alpha = 1$. Each q_{ij} represents the probability, or degree of belief, that the sample x_i belongs to cluster j.

To refine the clustering structure, DEC constructs a target distribution $P = [p_{ij}]$ that sharpens confident assignments and balances cluster frequencies:

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'} q_{ij'}^2/f_{j'}}, \qquad \text{with } f_j = \sum_i q_{ij}. \tag{18}$$

The encoder parameters θ and the centroids $\{\mu_j\}$ are jointly optimized by minimizing the

Kullback–Leibler (KL) divergence between the target and predicted distributions:

$$\mathcal{L}_{\text{DEC}} = \text{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{19}$$

This self-training process progressively refines the latent space, pulling similar representations closer together and separating dissimilar ones. In its original implementation, DEC employed a simple feed-forward architecture with fully connected layers, designed as a generic framework adaptable to various data modalities.

Improved Deep Embedded Clustering (IDEC): IDEC method [116] extends DEC by preserving the feature information acquired during the autoencoder pretraining phase. Unlike DEC, which discards the decoder before clustering, IDEC retains it and jointly optimizes both the clustering and reconstruction objectives. The total loss function is defined as a weighted combination of the clustering loss $\mathcal{L}_c$ (from DEC) and the reconstruction loss $\mathcal{L}_r$:

$$\mathcal{L}_{\text{IDEC}} = (1-\gamma)\,\mathcal{L}_c + \gamma\,\mathcal{L}_r, \tag{20}$$

where $\gamma \in [0, 1]$ controls the trade-off between cluster compactness and reconstruction fidelity. DEC can be viewed as a special case of IDEC obtained when $\gamma = 0$, in which only the clustering loss is optimized. Following the configuration used in the authors' original implementation, we adopt a γ value of 0.1.

The inclusion of the reconstruction term helps maintain the local neighborhood structure of the latent space, mitigating the risk of representation drift caused by the clustering objective alone.

Structural Deep Clustering Network (SDCN): SDCN [117] integrates autoencoder-based representation learning with graph convolutional networks (GCNs) to simultaneously capture feature and structural relationships within the data. While the autoencoder focuses on learning nonlinear latent representations, the GCN propagates information across samples connected in a similarity graph.

In our implementation, the graph structure was constructed using a k-nearest neighbors (k-NN) approach with $k = 3$. This setup allows each time series to interact with its closest neighbors in the

latent space.

The combined training objective optimizes three terms — the clustering loss $\mathcal{L}_c$, the reconstruction loss $\mathcal{L}_{\text{rec}}$, and the graph regularization loss $\mathcal{L}_g$:

$$\mathcal{L}_{\text{SDCN}} = \mathcal{L}_c + \alpha\,\mathcal{L}_{\text{rec}} + \beta\,\mathcal{L}_g, \qquad \text{with} \qquad \mathcal{L}_g = \frac{1}{2N}\,\|Z_g - Z_e\|_F^2\,. \tag{21}$$

Here, Z_e and Z_g denote the latent representations obtained from the autoencoder and the graph convolutional network, respectively. This term enforces consistency between both representations, ensuring that the learned features respect the similarity structure defined by the graph.

Because of the high computational complexity of the GCN, we applied SDCN only to the multivariate dataset. The univariate dataset was excluded due to resource constraints and memory limitations. Despite this, on the multivariate data, the model successfully leveraged the k-NN graph structure to identify meaningful clusters.

Deep Embedded Regularized Clustering (DEPICT): DEPICT [112] extends the IDEC framework by introducing additional regularization and denoising mechanisms to improve cluster stability and representation robustness. Unlike DEC and IDEC, DEPICT replaces the Student's t-distribution with a softmax-based clustering layer, composed of a dense layer of dimension k (number of clusters) followed by a softmax activation. This produces the predicted distribution Q:

$$q_{ij} = \frac{\exp(\theta_{\text{soft},j}^T z_i)}{\sum_{j'=1}^{K} \exp(\theta_{\text{soft},j'}^T z_i)}, \tag{22}$$

where $\Theta_{\text{soft}} = [\theta_{\text{soft},1}, \ldots, \theta_{\text{soft},K}]$ represents the clustering layer parameters.

The model minimizes the sum of two Kullback–Leibler (KL) divergence terms: one between the sharpened target distribution P and the predicted assignments Q, and another between the empirical cluster frequencies $f_j = \frac{1}{N}\sum_i q_{ij}$ and a uniform prior $u_j = \frac{1}{K}$:

$$\mathcal{L}_{\text{DEPICT}} = \frac{1}{N}\sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} + \sum_j f_j \log \frac{f_j}{u_j}. \tag{23}$$

This dual objective simultaneously encourages accurate cluster assignment and balanced cluster proportions, avoiding degenerate solutions dominated by a few large clusters.

Deep Temporal Clustering Representation (DTCR): DTCR [24] was designed to jointly learn temporal representations and clustering assignments within a single recurrent model. Its core objective is to capture both the sequential dependencies inherent in time-series data and the latent grouping structure that emerges from them, without decoupling representation learning from clustering.

The encoder is composed of multiple bidirectional recurrent layers with exponentially increasing dilation rates, allowing the network to model dependencies across multiple temporal scales. The decoder consists of a single recurrent layer whose initial hidden state is formed by concatenating the final states of all encoder layers, ensuring that temporal information captured at different depths contributes to sequence reconstruction.

The training objective integrates three complementary components into a unified optimization scheme. The first term ensures reconstruction fidelity, encouraging the model to preserve the temporal characteristics and overall structure of the original input sequences. The second term introduces a real/fake discrimination mechanism, in which an auxiliary branch predicts whether an input sequence is authentic or artificially perturbed. These "fake" samples are created by randomly shuffling a portion of the time steps, forcing the encoder to learn temporally consistent and realistic representations. The final term promotes cluster separability in the latent space through a differentiable relaxation of the k-means objective, aligning encoded features with evolving cluster indicators during training.

The overall objective can be formulated as:

$$\mathcal{L}_{\text{DTCR}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{disc}} + \lambda \mathcal{L}_{\text{cluster}}, \tag{24}$$

where $\mathcal{L}_{\text{rec}}$ represents the reconstruction loss, $\mathcal{L}_{\text{disc}}$ the discrimination loss distinguishing real from perturbed sequences, and λ a regularization coefficient controlling the relative weight of the clustering component. Following the configuration of the reference implementation by [24], we set $\lambda = 0.5$.

By integrating temporal modeling, discrimination, and clustering into a single objective, DTCR produces latent spaces that are both temporally coherent and structurally organized.

Variational Deep Embedding (VADE): VADE [118] integrates clustering directly into the probabilistic structure of a variational autoencoder (VAE). Unlike standard VAEs, which assume a single isotropic Gaussian prior over the latent space, VADE models the latent distribution as a GMM, allowing each component to correspond to a distinct cluster. This approach enables the model to simultaneously perform representation learning, clustering, and data generation within a unified probabilistic framework.

It is built upon the ELBO formulation and is therefore only compatible with VAE-based architectures. The encoder maps each input x_i to a joint latent distribution $q(z, c \mid x_i)$, where z denotes the latent variable and $c \in \{1, \ldots, K\}$ represents the cluster assignment. The decoder reconstructs the input from samples drawn from this distribution, while the prior is defined as a mixture of Gaussians:

$$p(z, c) = p(c)\, p(z \mid c), \tag{25}$$

where $p(c)$ is a categorical distribution over clusters and $p(z \mid c)$ denotes a Gaussian component corresponding to cluster c.

The optimization objective extends the standard VAE loss by incorporating this mixture structure:

$$\mathcal{L}_{\text{VADE}} = -E_{q(z,c|x)}[\log p(x|z, c)] + \mathrm{KL}\big(q(z, c|x) \,\|\, p(z, c)\big), \tag{26}$$

where the reconstruction term encourages accurate decoding of the input, and the KL divergence regularizes the latent distribution toward the mixture prior.

Through this formulation, VADE jointly optimizes the latent representation, the clustering assignments, and the mixture parameters, producing a smooth and well-structured latent space. Unlike deterministic approaches such as DEC or IDEC, VADE provides a probabilistic clustering interpretation, allowing both soft assignments and sample generation from specific clusters.

ClusterGAN: The ClusterGAN [114] builds upon the standard GAN framework and is therefore only compatible with GAN-based architectures. While conventional GANs lack an encoder and do not preserve structured latent representations, ClusterGAN introduces an additional encoder trained to reconstruct the latent code from generated samples. This modification enables the learned latent space to exhibit discrete, cluster-specific organization rather than being purely continuous.

The encoder is jointly optimized with the generator and discriminator so that the reconstructed latent features remain consistent with the inputs sampled from the prior distribution. This bidirectional training encourages a one-to-one correspondence between latent variables and generated outputs, effectively combining generative modeling and clustering within a single adversarial framework. In our implementation, this loss follows the configuration provided in the authors' publicly available code.

4.3.5 Hyperparameter Optimization :Population-Based Training Configuration

To automatically tune hyperparameters and model configurations, we adopt PBT introduced by Jaderberg et al. [20]. PBT combines principles from evolutionary algorithms and online hyperparameter adaptation, enabling the simultaneous optimization of model parameters and hyperparameters during training rather than relying on fixed, manually tuned configurations. The approach maintains a population of models that are trained in parallel, each with its own hyperparameter configuration. Training proceeds over several rounds,, where each corresponds to a full training–evaluation–selection cycle. At the end of every round, models are ranked according to a chosen performance metric, and an explore–exploit procedure is applied to evolve the population toward better configurations.

During the exploitation phase, top-performing models are retained, and in some settings, their weights are transferred to replace lower-performing models. In the exploration phase, new derived configurations are generated by applying stochastic mutations to selected hyperparameters of the best-performing models, thereby introducing diversity into the population. Formally, each perturbed hyperparameter θ is updated as:

$$\theta' = \theta \times (1 + \epsilon), \quad \epsilon \sim \mathcal{U}(-\alpha, \alpha) \tag{27}$$

where α defines the mutation amplitude, also known as the mutation factor. This iterative process balances exploitation of high-performing configurations and exploration of new regions of the search space, promoting adaptive convergence toward stable optima.

In this study, the hyperparameter search space included: batch size, number of filters, depth,

number of steps, number of pretrain steps, kernel size, learning rate, number of random samples, negative penalty, latent dimension, and reduced size. Common parameters such as batch size, number of filters, latent dimension, and learning rate follow standard conventions. The number of steps and number of pretrain steps represent iteration counts for clustering and encoder pretraining, respectively. The parameters negative penalty and number of random samples are specific to the triplet loss formulation, balancing positive–negative pair contributions, while reduced size is exclusive to the dilated causal CNN architecture, defining the dimensionality reduction applied before latent-space projection.

Finally, the models selected for hyperparameter tuning were chosen based on the evaluation presented in the previous section. Specifically, the top three models from each dataset and each number of clusters were tuned, resulting in a total of twelve tuned models. This targeted tuning strategy ensured computational feasibility while still providing a representative exploration of the most promising model configurations across both datasets.

Univariate Dataset Configuration

For the univariate electricity-load dataset, given its large sample size and relatively low feature dimensionality, hyperparameters were randomly initialized from predefined discrete sets rather than reusing those employed during the initial model training phase preceding evaluation. The initialization included batch size, number of filters, learning rate, number of pretraining steps, and number of clustering steps. Each round evaluated the silhouette coefficient, computed from both latent representations and predicted cluster assignments. Collapsed or degenerate solutions that are single-cluster outcomes or vanishing latent variance were automatically detected and assigned zero fitness.

To maintain exploration diversity on the large univariate dataset and avoid propagating early biases, we did not using weight inheritance, all models were retrained from scratch at each round. The top half survived unchanged each round, while the bottom half were replaced by children of winners. We applied a perturbation to the learning rate $\alpha = \pm 0.5$ and resampled all other hyperparameters from predefined discrete sets. This keeps the search bounded and diverse without initializing from prior checkpoints.

The process was repeated for five evolutionary rounds with a population size of six, ensuring a robust exploration–exploitation balance while maintaining computational feasibility.

Multivariate Dataset Configuration

In contrast to the univariate case, the multivariate dataset uses 24 hourly steps per day and multiple channels. Although the temporal resolution is lower, the multi-channel structure and smaller sample size made training more sensitive, so we adjusted PBT to improve stability and preserve learned representations across generations. A population of ten models was trained over eight evolutionary rounds higher than the univariate case to compensate for the smaller dataset size and collapsed or degenerate runs received zero fitness. Unlike the univariate setup, which restarted models from scratch, the multivariate configuration employed weight inheritance. At the end of each training round, encoder weights from top-performing models were transferred as initialization checkpoints for the next round, allowing subsequent models to refine rather than relearn temporal dependencies.

This strategy maintained continuity in both architecture and latent representations, reducing the risk of unstable convergence that often arises when smaller datasets are retrained from random initializations.

During the mutation phase, the learning rate, batch size, number of filters, pretraining steps, and clustering steps were continuously perturbed by a factor of ±0.1, while all other parameters were re-sampled from predefined discrete sets. A smaller mutation factor ($\alpha = 0.1$) was selected to maintain training stability around well-performing regions and to prevent divergence, particularly given the small dataset size and the high sensitivity of multivariate temporal dependencies to large hyperparameter shifts. This configuration combined elitism, where the top two models were directly carried over without modification, and controlled exploration, where hyperparameters were slightly adjusted within a small perturbation range.

4.3.6 XAI

To enhance the explainability of clustering outcomes, we adopted the prototype–criticism framework introduced by Kim et al. [63], known as MMD-Critic. This approach complements quantitative cluster-evaluation metrics by providing a human-explainable summary of the latent representations learned by the encoder. It identifies two complementary sets of representative daily profiles: prototypes, which capture the dominant energy-use patterns within each cluster, and criticisms, which highlight atypical or under-represented consumption behaviors that prototypes fail to explain. Together, these examples illustrate both the central tendencies such as typical weekday or peak-load shapes and the exceptions, including irregular, low-activity, or outlier consumption days.

The method is grounded in the Maximum Mean Discrepancy (MMD) criterion, which measures the dissimilarity between two sample distributions P and Q in a reproducing-kernel Hilbert space (RKHS). Each data point x_i is mapped to a high-dimensional feature vector $\phi(x_i)$ through a kernel function:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle, \tag{28}$$

using a Gaussian RBF kernel. Within this space, the mean embedding of a distribution summarizes its overall statistical characteristics, and the squared MMD quantifies how far the mean embeddings of P and Q are separated. In this context, P corresponds to the latent distribution of all time-series embeddings belonging to a given energy cluster, while Q represents the subset of selected prototypes. Minimizing the MMD aligns the prototypes with the overall distribution of the cluster, allowing them to represent its dominant consumption trends with minimal redundancy.

Once prototypes are identified, criticisms are extracted using the witness function:

$$f(x) = E_{x' \sim P}[k(x, x')] - E_{y' \sim Q}[k(x, y')], \tag{29}$$

which measures how well each energy-consumption profile is represented by the prototypes. Data points with large positive $f(x)$ values correspond to under-represented or anomalous load patterns and are thus selected as criticisms. This approach provides an explainable framework for distinguishing representative consumption patterns from atypical or divergent behaviors within the latent

energy-consumption space. In this paper, MMD-Critic is applied independently per cluster to the encoder's latent embeddings. We extract five prototypes and three criticisms per cluster (when sample size permits). To calibrate the kernel, bandwidths are set from the latent distances within each cluster. For prototypes we use the 30th percentile of pairwise distances (scaled by 0.5), and for criticisms we use the median distance (scaled by 0.5). Kernels are computed in chunks to handle large N.

To avoid redundant selections, prototype choice uses a greedy MMD gain with a diversity constraint: among the top-150 gain candidates, we pick the point farthest from already chosen prototypes and enforce a minimum separation of 0.15 times the median within-cluster latent distance (with backoffs if needed). Criticisms are drawn from the top-70 witness candidates and diversified analogously. Clusters with fewer than five samples are skipped.

From the twelve PBT–tuned configurations, we aggregated post-clustering silhouette scores across runs and z-normalized them within each (dataset, k) group to remove dependence on k. For the prototype–criticism analysis, we selected the top-ranked multivariate model at its optimal k as the prototype model to explain. We prioritized the multivariate setting because it is more informative and demanding than the univariate case, capturing cross-channel interactions and richer load semantics.

Rather than relying on reconstruction accuracy which reflects decoder fidelity but not cluster representativeness, we base explainability on kernel MMD in latent space, selecting prototypes that best cover each cluster's distribution and criticisms that expose under-represented cases. Displaying these selected days in the original signal space links latent structure to concrete load patterns, improving transparency and trust in the unsupervised analysis.

4.4 Results and Discussion

In this section, we present pre- and post-clustering results , excluding collapsed runs, and rank models via average total rank over five internal indices. Results are presented in the following order: encoder architectures, pretext objectives, and clustering losses. We then show Population-Based Training (PBT) heatmaps and runtime, and conclude with an MMD-Critic analysis of the

best multivariate model.

4.4.1 Traditional Clustering as Baseline

We estimated k using the elbow heuristic and the gap statistic. For the univariate dataset, elbow suggested k=8 and gap suggested k=4. For the multivariate dataset, elbow suggested k=8 and gap suggested k=6. Accordingly, we report baseline results at both k values for each dataset.

Table 4.1 presents the results for univariate and multivariate traditional clustering techniques under different numbers of clusters. On the univariate dataset, at $k = 8$, Euclidean *k*-means performs best across all indices, achieving a silhouette score of approximately 0.214 compared with 0.147 for DTW and 0.048 for K-Shape. At $k = 4$, DTW *k*-means becomes most competitive in terms of separation and compactness (silhouette ≈ 0.353, XB ≈ 1.19), while Euclidean distance retains the best DB, CH, and Dunn scores. K-Shape remains the weakest method at both cluster counts. Additionally, univariate DBSCAN using cosine distance with min_samples = 9 and $\varepsilon \approx 0.1413$ yields 23 clusters with approximately 19% noise, revealing multiple dense load-shape regimes that are not constrained by a predefined number of clusters and highlighting DBSCAN's ability to uncover fine-grained consumption patterns overlooked by *k*-based methods.

For the multivariate dataset ($k = 8$ and $k = 6$), Euclidean *k*-means consistently achieves the highest performance across all five internal metrics, with DTW ranking second and improving slightly at $k = 6$, while K-Shape again performs worst. A parameter exploration of DBSCAN's density settings using cosine distance identified $\varepsilon \approx 0.1366$ and min_samples = 8, yielding three clusters with approximately 10% noise. On inliers, the indices are strong (Silhouette ≈ 0.264, DB ≈ 0.85, Dunn ≈ 0.65, XB ≈ 0.48), indicating compact, dense regimes, although direct comparison with *k*-based methods is not possible.

Overall, Euclidean *k*-means emerges as the most reliable baseline. DTW is advantageous when using fewer and broader regimes, whereas K-Shape remains consistently weak. Multivariate DBSCAN reveals dense structures with manageable noise. These results serve as baseline references for subsequent comparison with deep clustering models.

Table 4.1: Baseline results for univariate and multivariate datasets under different numbers of clusters

Method	**Number of Clusters = 8**					**Number of Clusters = 4**				
Univariate Dataset	**Silhouette**	**DB**	**CH**	**Dunn**	**XB**	**Silhouette**	**DB**	**CH**	**Dunn**	**XB**
K-Means (Euclidean)	0.214	1.933	85738.560	0.213	1.679	0.232	1.649	650.320	0.233	1.406
K-Means (DTW)	0.147	3.193	58571.430	0.090	2.120	0.353	2.470	505.260	0.161	1.190
K-Shape	0.048	2.846	33931.400	0.126	5.070	0.168	2.564	358.390	0.127	4.860
DBSCAN (Cosine):	Sil = 0.366	DB = 0.714	CH = 180.680	Dunn = 0.339	XB = 0.871					

Method	**Number of Clusters = 8**					**Number of Clusters = 6**				
Multivariate Dataset	**Silhouette**	**DB**	**CH**	**Dunn**	**XB**	**Silhouette**	**DB**	**CH**	**Dunn**	**XB**
K-Means (Euclidean)	0.282	1.298	1701.850	0.355	0.729	0.261	1.316	1816.840	0.406	0.784
K-Means (DTW)	0.203	1.510	1282.280	0.261	1.546	0.232	1.601	1592.860	0.290	1.214
K-Shape	0.045	2.354	719.230	0.144	19.474	0.175	1.988	758.740	0.237	4.036
DBSCAN (Cosine):	Sil = 0.264	DB = 0.847	CH = 101.160	Dunn = 0.649	XB = 0.476					

4.4.2 Deep Clustering vs. Traditional Baselines

Based on Table 4.1 and Table 4.3, deep models clearly dominate traditional clustering approaches for both $k = 8$ and $k = 4$. At $k = 8$, the top deep configuration achieves substantially higher compactness and separation than Euclidean or DTW-based k-means, which show weak cohesion and high intra-cluster dispersion. Even at $k = 4$, deep models maintain a strong advantage, with all high- and mid-rank variants outperforming traditional methods across all internal validity indices. While DBSCAN (cosine) uncovers shape-dense regimes, its overall performance remains below the deep architectures.

As shown in Table 4.1 and Table 4.2, deep models again surpass traditional baselines at both $k = 8$ and $k = 6$. They produce clusters that are markedly more compact and well-separated, indicating a stronger ability to capture complex cross-channel relationships in multivariate energy profiles. Even mid-performing deep models exceed all traditional baselines, confirming the robustness of learned representations. DBSCAN (cosine) performs comparably to k-means in certain compactness metrics but fails to reach the consistency and quality achieved by deep clustering.

Across both univariate and multivariate settings, deep clustering offers a large and consistent

Table 4.2: Multivariate deep models at k=8 and k=6 (1st / 15th / 30th)

k	Model	Silouette	DB	CH	Dunn	XB
8	DCNN + REC + SDCN (1st)	0.739	0.382	21711.829	0.787	0.074
	DCNN + REC + None (15th)	0.512	0.648	5886.669	0.626	0.195
	DCNN + Multi_Rec + IDEC (30th)	0.487	0.691	7474.508	0.424	0.286
6	DCNN + REC + SDCN (1st)	0.750	0.326	20909.294	0.893	0.062
	Bi_GRU + VAE + None (15th)	0.501	0.702	7163.076	0.693	0.151
	Bi_GRU + REC + DEC (30th)	0.465	0.767	7873.042	0.580	0.228

Table 4.3: Univariate deep models at k=4 and k=8 (1st / 15th / 30th)

k	Model	Silouette	DB	CH	Dunn	XB
4	DCNN + VAE + DEC (1st)	0.681	0.420	1923423.288	0.606	0.127
	FCNN + Triplet + IDEC (15th)	0.606	0.546	1026475.062	0.222	0.177
	FCNN + VAE + DTCR (30th)	0.558	0.569	1108072.676	0.079	0.379
8	DCNN + VAE + DEC (1st)	0.621	0.506	1938773.266	0.340	0.186
	FCNN + Multi_Rec + DEC (15th)	0.314	1.249	311803.930	0.363	0.618
	DCNN + REC + DEC (30th)	0.285	1.453	235918.023	0.318	0.968

improvement over all traditional baselines. Density-based methods such as DBSCAN provide complementary insights into local data structure but do not approach the overall performance of deep models in terms of stability, compactness, and separation.

4.4.3 Pipeline Results

Architecture

Results from Figure 4.3 and Figure 4.4 demonstrate that architectures with convolutional encoders, particularly the residual CNN and dilated CNN, consistently dominated the top rankings across all dataset configurations. Their strong performance confirms their superior ability to capture short- and mid-term temporal dependencies in building load profiles such as how demand rises, stabilizes, and declines throughout the day. In building energy-consumption data, these temporal dynamics correspond to operational transitions like morning start-ups, occupancy-driven peaks, and evening shutdowns. The convolutional receptive fields of DCNN and Rest-CNN are particularly well suited to extracting these localized temporal variations, allowing the models to learn both the intensity and duration of recurring load events. The DCNN architecture emerged as the overall

top performer across both the univariate and multivariate datasets, while the Rest-CNN closely followed, both demonstrating strong generalization from latent representation learning to final cluster formation. Their hierarchical feature extraction enabled them to detect recurring consumption patterns and temporal transitions between operational states, resulting in compact energy clusters.

In contrast, FCNN, despite its simpler feedforward structure, maintained competitive performance, particularly in the univariate datasets. Its architecture, which processes entire sequences, is likely better suited for distinguishing broad consumption archetypes such as base-load, morning-peak, or evening-peak profiles, rather than fine-grained temporal fluctuations. Unlike convolutional models that emphasize localized temporal dynamics, the FCNN tends to capture overall load-shape characteristics, which may explain its stable performance across datasets with consistent daily consumption patterns. However, its performance declined in the multivariate setting, suggesting limited scalability when multiple meters or end-use channels must be modeled jointly. The Bi-GRU architecture showed moderate but stable behavior across all configurations, confirming its robustness in tracking short-term temporal variations, while the Bi-LSTM achieved higher presence in the multivariate datasets, where its longer memory window helped capture cross-channel dependencies. Attention-based models displayed similar trends in both stages, excelling in identifying short-duration spikes but struggling to maintain stability under multi-pattern consumption behaviors. Interestingly, the Deep Recurrent Neural Network, which was scarcely represented among top pre-clustering models, appeared more frequently after clustering refinement, suggesting that its sequential modeling becomes more beneficial once the latent representation is stabilized by the clustering phase.

The relatively stable performance of DCNN and Res-CNN across datasets indicates that convolutional filters are well adapted to the recurring temporal structures of building energy profiles. By contrast, FCNNs performed better on low-dimensional aggregate loads, likely due to their holistic treatment of input sequences. Recurrent models provided complementary strengths in representing short-term variability, although their latent spaces showed lower structural consistency compared to convolutional architectures.

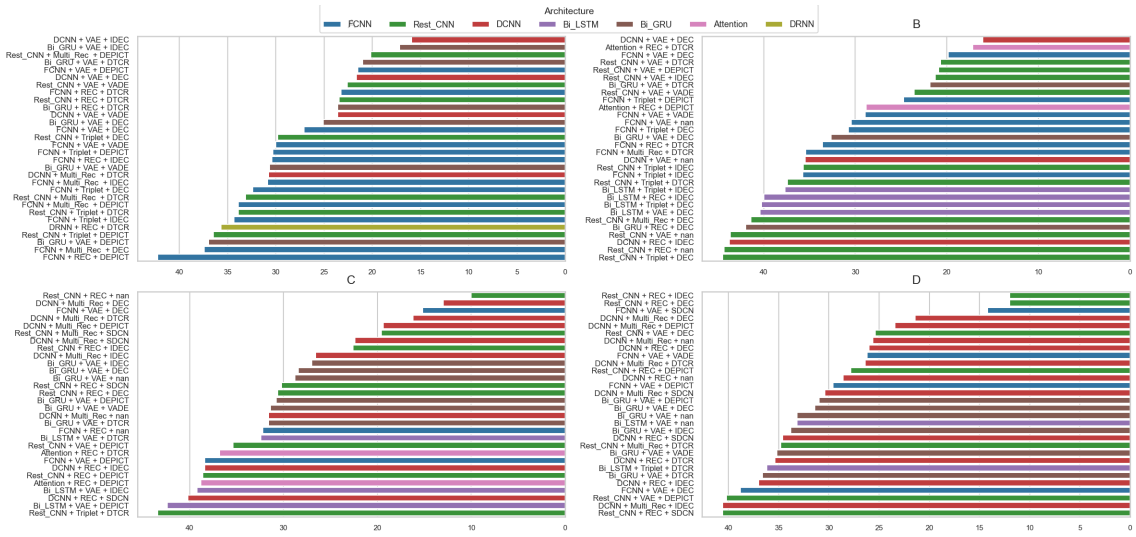


Figure 4.3: Architecture for pre-clustering step. (A) Univariate dataset with k=8 (B) Univariate dataset with k= 4 (C) Multivariate dataset with k=6 (D) Multivariate dataset with k=8

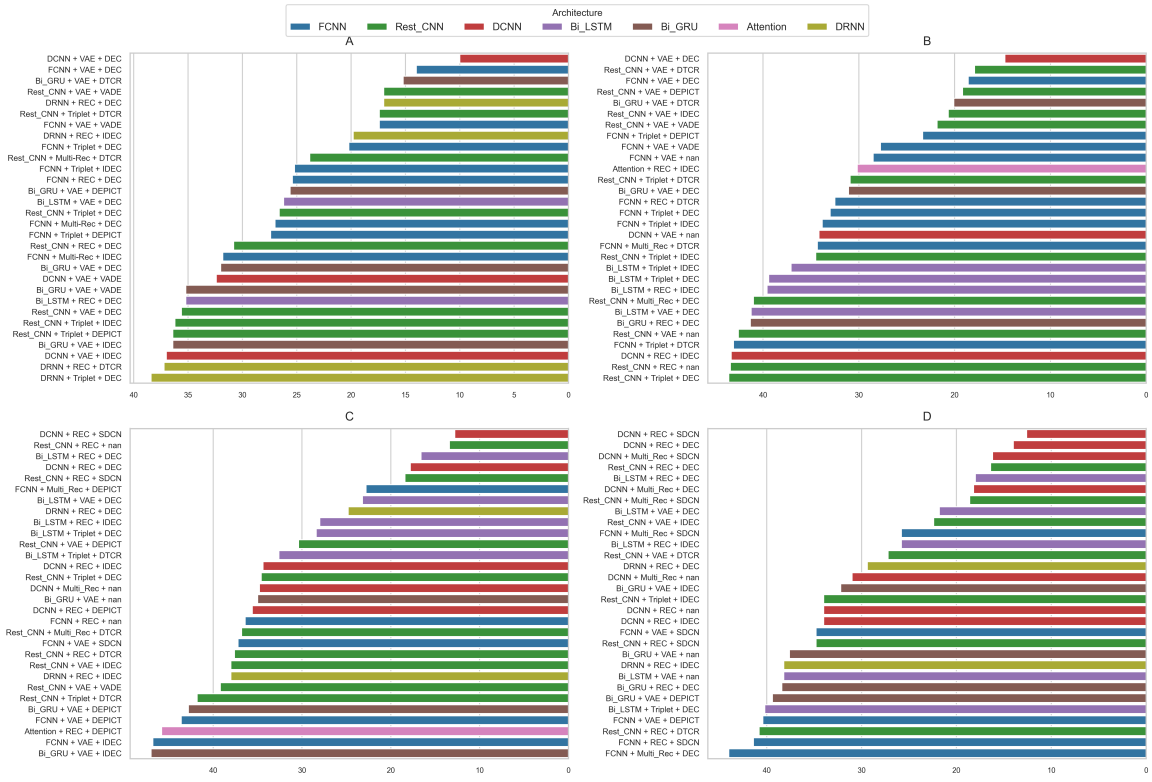


Figure 4.4: Architecture for post-clustering step. (A) Univariate dataset with k=8 (B) Univariate dataset with k= 4 (C) Multivariate dataset with k=6 (D) Multivariate dataset with k=8

Pretext Loss

Across both the pre- and post-clustering phases, the results in Figure 4.5 and 4.4 reveal consistent patterns regarding how pretext objectives influence the quality of learned representations and the separability of latent clusters in energy consumption data. Among all objectives, the VAE loss exhibited the most stable and dominant behavior, particularly on univariate load profiles, where it consistently achieved first-rank performance across clustering configurations (k = 4 and k = 8). This performance can be attributed to its probabilistic regularization, which leads to smooth, low-variance latent manifolds capable of capturing gradual transitions in daily electricity demand. This resulted in well-separated clusters within the latent space, indicating that VAE-based representations capture meaningful variations in energy-use behavior, potentially corresponding to different operational modes such as base-load or occupancy-driven periods.

The Triplet loss appeared frequently among univariate configurations before clustering but became more effective in the multivariate energy datasets after clustering refinement. This suggests that distance-based objectives become more useful once the relationships between multiple variables are stabilized in the latent space.

Among all objectives, the reconstruction loss was the most reliable and frequently dominant across both univariate and multivariate datasets. Unlike multi-reconstruction, which reconstructs several variations of the input sequence, the single reconstruction objective achieved better results, showing that increased reconstruction complexity does not necessarily improve the learned representations.

Finally, the GAN-based objective was absent from the top-performing configurations in all experiments, indicating that adversarial learning did not provide stable or meaningful representations for energy consumption patterns. Overall, these results show that VAE loss is particularly suitable for low-dimensional univariate energy signals, while single-target reconstruction offers the most robust and general performance across all datasets. Together, they provide strong and explainable latent representations that capture the natural variability and operational modes of building energy use.

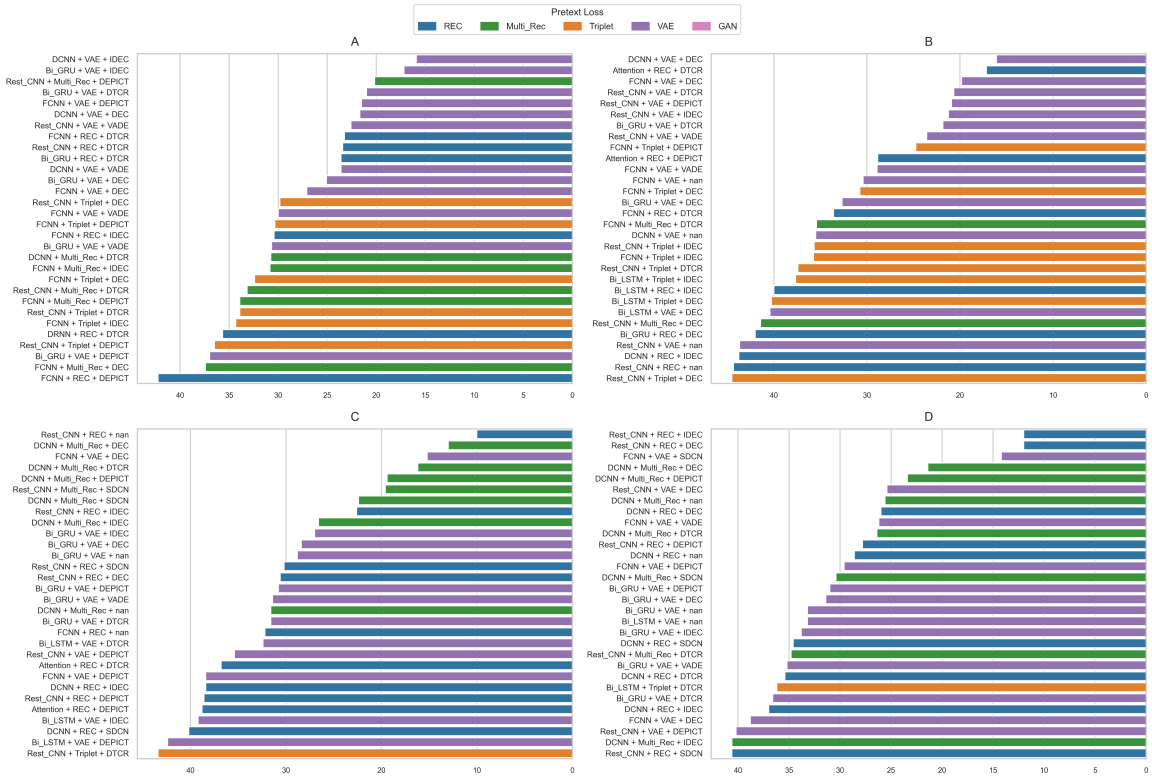


Figure 4.5: Pretext loss for pre-clustering step. (A) Univariate dataset with k=8 (B) Univariate dataset with k= 4 (C) Multivariate dataset with k=6 (D) Multivariate dataset with k=8

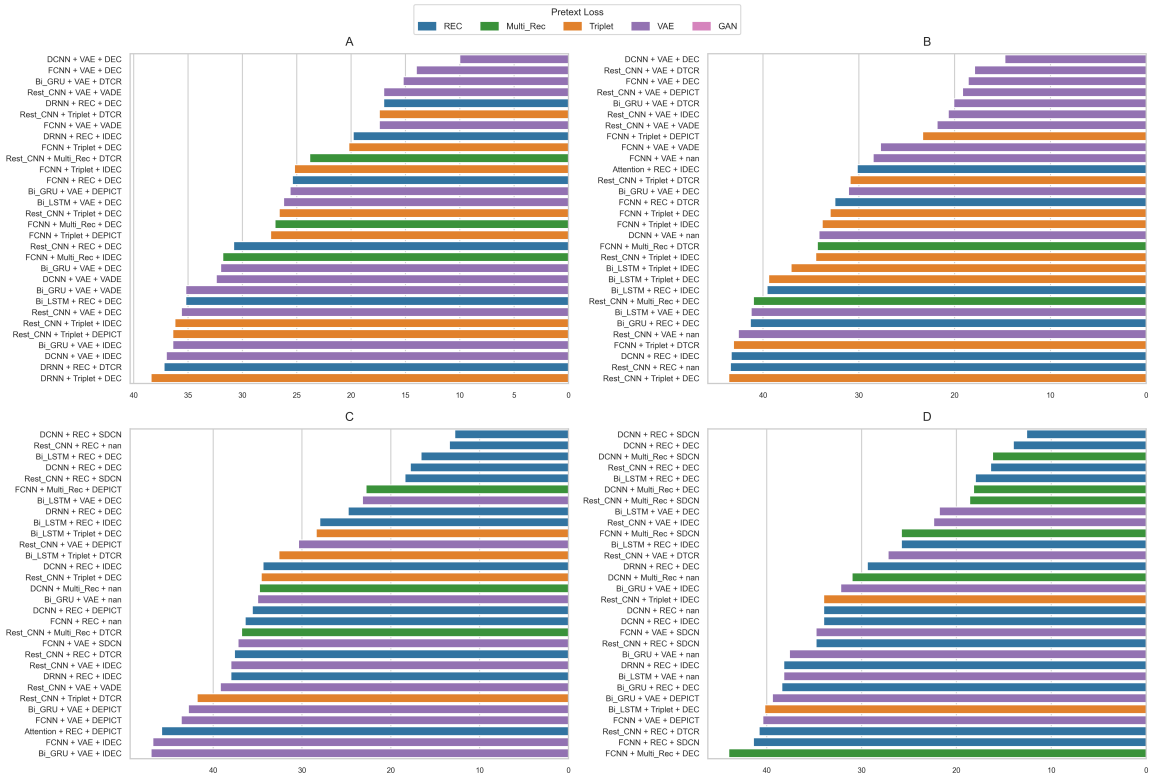


Figure 4.6: Pretext loss for post-clustering step. (A) Univariate dataset with k=8 (B) Univariate dataset with k= 4 (C) Multivariate dataset with k=6 (D) Multivariate dataset with k=8

Clustering Loss

Based on the pre- and post-clustering stages, the results in Figure 4.7 and Figure 4.8 showed that no single clustering loss consistently dominated across all datasets, indicating that clustering performance in building energy time series is highly dependent on data dimensionality and complexity. Nonetheless, several methods recurrently appeared among the top configurations. DEC loss exhibited the most stable and consistent behavior, becoming increasingly dominant in post-clustering and reaching the first rank in the univariate energy datasets. Together with the DTCR and DEPICT losses, DEC formed the most competitive loss, with these three methods frequently alternating among the top positions across both univariate and multivariate energy datasets. IDEC also achieved strong results in the pre-clustering phase, confirming that the integration of reconstruction and clustering regularization helps preserve the smooth evolution of load curves and prevents sharp variations between energy-use regimes.

VAE appeared less consistently, showing moderate performance in the univariate datasets but less effectively on multivariate ones. This result indicates that its probabilistic formulation was less capable of representing the complex and non-Gaussian characteristics typical of building energy consumption, where correlated loads such as electricity, chilled water, steam, and hot water often exhibit nonlinear dependencies

SDCN achieved excellent performance on the multivariate datasets, often surpassing DEC. However, since it was not evaluated on univariate data, its generalizability cannot be confirmed, though its success highlights the value of graph-based structural regularization for multi-channel energy profiles.

Interestingly, models trained with no clustering loss (None) continued to appear among the top results for the multivariate datasets occasionally even reaching the first rank demonstrating that, under certain high-dimensional conditions, the learned latent representations alone can yield meaningful separability without explicit clustering optimization. Finally, ClusterGAN was absent from all top-performing configurations in both stages, indicating that adversarial clustering objectives did not provide stable or explainable organization of the latent space.

Overall, these findings emphasize that DEC, DTCR, and DEPICT constitute the most reliable

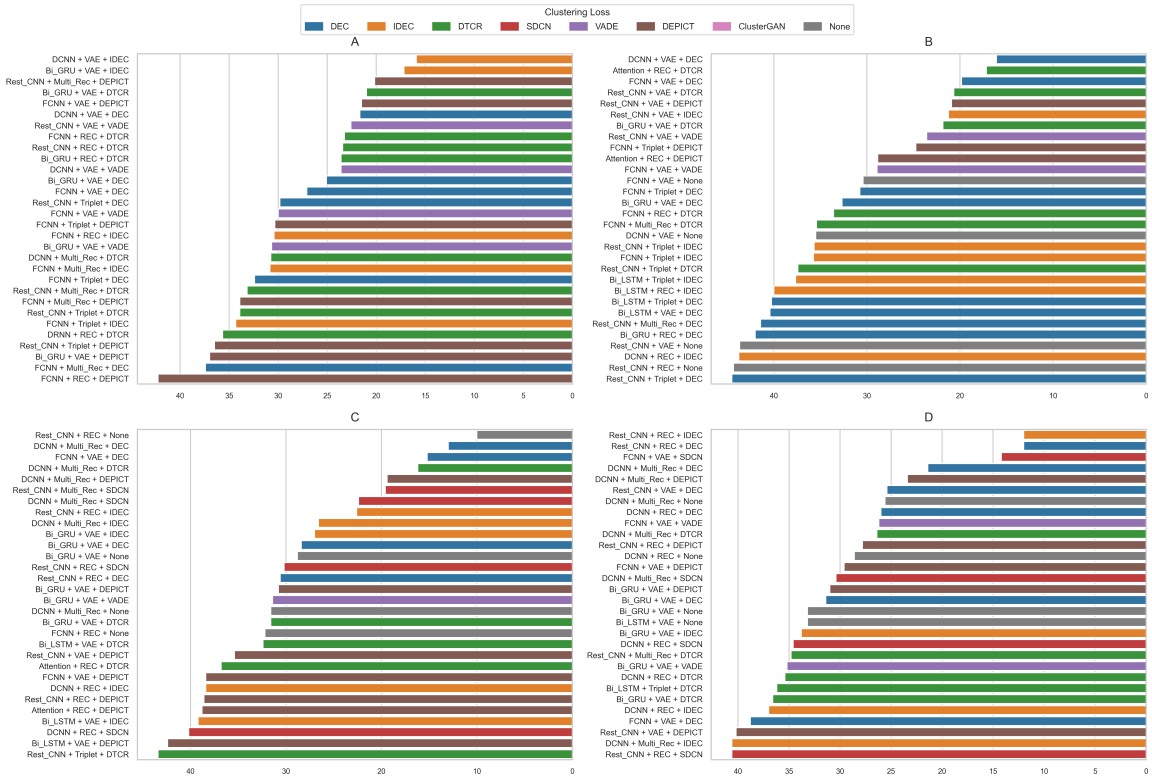


Figure 4.7: Clustering loss for pre-clustering step. (A) Univariate dataset with k=8 (B) Univariate dataset with k= 4 (C) Multivariate dataset with k=6 (D) Multivariate dataset with k=8

and consistently energy-relevant high-performing clustering losses, while probabilistic and adversarial formulations such as VAE and ClusterGAN show limited robustness in the context of building energy time-series data.

4.4.4 Hyperparameter Tunning

Multivariate Results

Across eight evolutionary rounds, the multivariate heatmap 4.9 shows sustained high silhouette performance in the 0.80–0.90 range for most rows, with consistent gains over the baseline column. All models remain remarkably stable across populations, indicating that the PBT schedule combining weight inheritance and small perturbations successfully refined promising regions of the hyperparameter space rather than drifting randomly. The only visible instability appears for the dilated_cnn_multi_rec_SDCN at rounds 6 and 7, where three collapsed solutions are detected by our

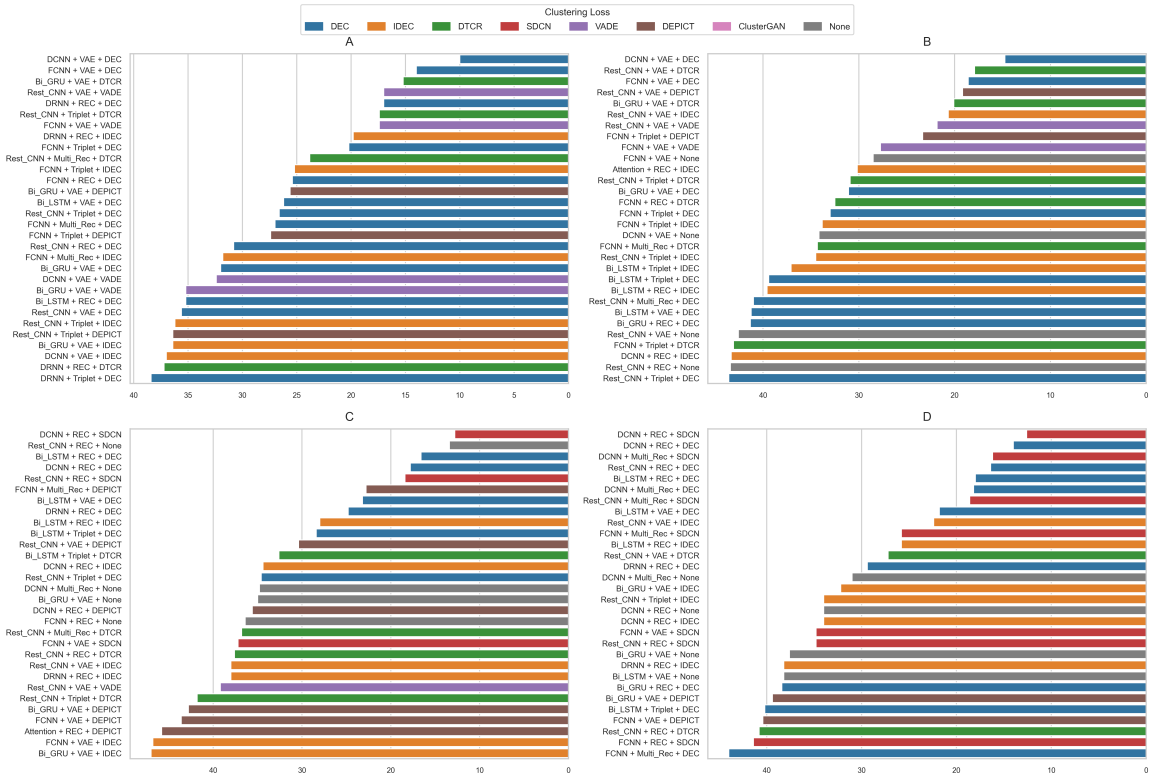


Figure 4.8: Clustering loss for post-clustering step. (A) Univariate dataset with k=8 (B) Univariate dataset with k= 4 (C) Multivariate dataset with k=6 (D) Multivariate dataset with k=8

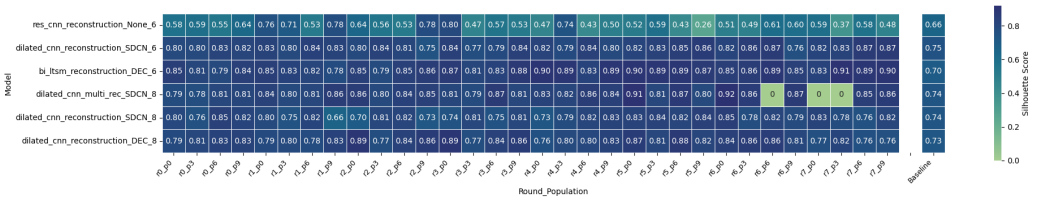


Figure 4.9: Multivariate silhouette heatmap (models × round–population) with baseline and $k \in \{6, 8\}$.

degeneracy filter. Two additional contrasts are evident. First, the configuration with no clustering loss ("None") fluctuates widely (≈0.26–0.80) and only occasionally surpasses its baseline, highlighting the necessity of a dedicated clustering objective. Second, several rows at $k = 8$ maintain plateau-like performance around 0.8 to 0.9 across successive populations, showing that the search converges not to isolated optima but to extended high-performing regions of the hyperparameter search space. Overall, the multivariate setting exhibits strong population-level convergence and robust improvement relative to baselines, with only a few isolated late-stage runs showing transient instability or collapse.

In terms of time complexity, Table 4.4 presents the run time for each model. Multivariate configurations with SDCN clustering require multi-day executions on the order of ≈60–70 hours per configuration, whereas centroid-based DEC updates complete in only ≈3–4 hours. Because SDCN repeatedly builds and updates a k-NN graph and runs graph propagation alongside encoder training, each epoch is far heavier than simple centroid updates. Encoder-only baselines are similarly fast but, as the heatmap confirms, fail to achieve comparable quality or stability. Despite their higher cost, the long SDCN runs remain computationally efficient within the PBT framework. By inheriting weights and applying small perturbations, a single extended trajectory often produces multiple stable, high-performing configurations across populations, rather than a few isolated peaks. This demonstrates that the additional runtime investment translates directly into a more reliable convergence, rather than redundant computation.

Univariate Results

The univariate heatmap 4.10 exhibits a more heterogeneous landscape, a direct consequence of the restart-from-scratch protocol and resampling most hyperparameters each round. Several rows

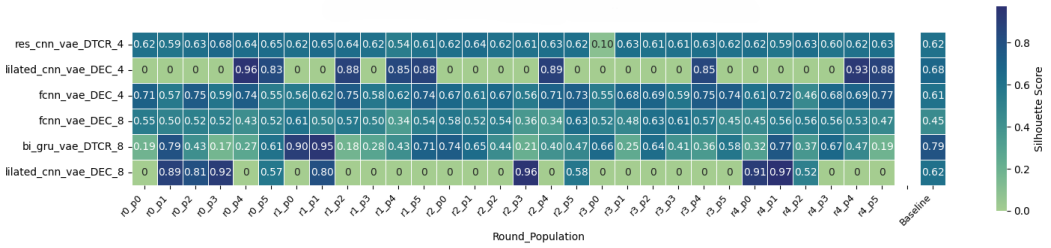


Figure 4.10: Univariate silhouette heatmap (models × round–population) with baseline and $k \in \{4, 8\}$.

show an alternating pattern of excellent peaks interleaved with collapsed solutions, confirming that the search space contains high-quality optima but that, without weight inheritance, these solutions are unstable and hard to reproduce consistently across populations. In contrast, models like residual or fully convolutional encoders with a clustering loss deliver modest but steady gains over their baselines (typically +0.05 to +0.15) and rarely collapse, illustrating a clear stability–performance trade-off induced by the exploration design. A second robust pattern is the effect of cluster count. Configurations that are stable at $k = 4$ often see silhouette drop by ≈0.10–0.15 at k = 8, reflecting the expected fragmentation cost when splitting coherent daily profiles into more regimes. Taken together, these results show that PBT can uncover top-tier solutions, but because it emphasizes broad exploration rather than refining the best settings, outcomes are more variable and less uniformly superior to baseline than in the multivariate case.

As summarized in Table 4.4, each univariate pass is computationally expensive both due to the dataset scale and because PBT reinitializes models every round yielding high runtime variance. Centroid-based DEC updates typically finish in ≈8–30 hours, whereas more involved objectives can extend to ≈3 days. Moreover, runtime is not strictly monotonic in k. With resampled step counts and occasional early collapses, some $k = 4$ trajectories run longer than their $k = 8$ counterparts.

4.4.5 XAI

We focus explainability analysis on the multivariate setting, as it captures cross-channel couplings (electricity, chilled water, steam, hot water). To avoid cluster-count bias, we normalized the last-round silhouette scores within each k and selected the top model. The best was Dilated-CNN +

Table 4.4: Top deep clustering configurations and runtimes by dataset and cluster count.

Dataset	k	Architecture	Pretext loss	Clustering loss	Runtime
Multivariate	6	dilated_cnn	reconstruction	SDCN	4 days, 03:12:52
		res_cnn	reconstruction	None	02:15:33
		Bi_LSTM	reconstruction	DEC	03:06:09
Multivariate	8	dilated_cnn	reconstruction	SDCN	2 days, 14:54:52
		dilated_cnn	reconstruction	DEC	04:02:12
		dilated_cnn	multi_rec	SDCN	2 days, 22:12:30
Univariate	8	dilated_cnn	vae	DEC	1 day, 05:30:15
		fcnn	vae	DEC	08:07:10
		bi_gru	vae	DTCR	3 days, 05:13:47
Univariate	4	dilated_cnn	vae	DEC	1 day, 18:01:09
		res_cnn	vae	DTCR	3 days, 01:50:39
		fcnn	vae	DEC	13:00:22

multi-reconstruction + SDCN (k=8), which we analyze below.

Figure 4.11 illustrates the t-SNE projection of the latent space learned by the dilated_cnn with multi_reconstruction and SDCN model on the multivariate energy-consumption dataset ($k = 8$). For each cluster, five prototypes and three criticisms are highlighted. The elongated clusters exhibit criticisms primarily along their tails, suggesting the presence of borderline or transitional energy-use patterns, while the more compact clusters display dense prototype cores with limited overlap with neighboring groups. A small, isolated cluster appears distinctly separated from the others, indicating a highly specific consumption regime that will be further analyzed in the subsequent section. Overall, the well-separated formations across the two-dimensional projection demonstrate that the model effectively captured distinct latent representations for the eight behavioral patterns of energy use. The coherent placement of prototypes within dense regions and the peripheral positioning of criticisms confirm a strong balance between intra-cluster compactness and inter-cluster separation.

Quantitatively, Table 4.5 confirms these observations. The five selected prototypes achieve high coverage of within-cluster variability ranging from 0.81 to 0.91, indicating that the set of prototypes effectively summarizes most member profiles. The most compact regime is Cluster 7 with coverage 0.91 and redundancy 0.55, where prototypes are diverse and criticisms show negligible divergence

Table 4.5: MMD-Critic explainability metrics coverage, redundancy, and criticism severity by cluster on the multivariate dataset ($k = 8$)

Cluster Number	Number of datapoints	Coverage	Redundancy	Criticism Severity
0	502	0.86	0.68	0.67
1	624	0.86	0.78	1.07
2	726	0.83	0.82	0.99
3	105	0	0	0
4	358	0.88	0.71	1.21
5	1031	0.83	0.91	0.59
6	551	0.81	0.83	0.88
7	364	0.91	0.55	0.09

from the main behavioral trend (critic_severity = 0.09) reflecting the cluster's internal uniformity and absence of outlier behaviors.. These results are consistent with a homogeneous energy-use pattern. In contrast, Cluster 5, which covers the largest group, exhibits the highest redundancy (0.91) with only moderate coverage (0.83), suggesting that its prototypes over-represent a single subregion of the cluster. Allocating more prototypes or enforcing stronger separation could maybe improve its representativeness. Clusters 1, 2, 4, and 6 combine fair coverage with elevated critic severity (1.07, 0.99, 1.21, and 0.88, respectively), indicating the presence of peripheral or irregular load behaviors at the cluster boundaries that are not represented by the central prototypes. Among them, Cluster 4 shows the strongest deviations, pointing to internal heterogeneity. Cluster 0 displays intermediate characteristics, whereas Cluster 3 presents zero values across all explainability metrics, suggesting either an outlier regime or highly uniform patterns that diverge from the variability observed in the other clusters. The absence of measurable coverage, redundancy, and critic severity indicates that the prototypes and criticisms could not establish meaningful similarity relations within this group, likely due to its small sample size or limited internal diversity. This behavior points to an exceptional or marginal energy-use regime that will be examined qualitatively in the following section.

To complement the latent-space view, the profile grid in Figure 4.12 clarifies the temporal semantics of each regime. For each cluster, we compute channel-averaged daily profiles across electricity, chilled water, steam, and hot water for the five prototypes and three criticisms, aggregating features to summarize overall load dynamics rather than channel-specific variation. We partition the day into morning (6AM–9AM), midday (10AM–5PM), evening (6PM–8PM), night (9PM–12AM),

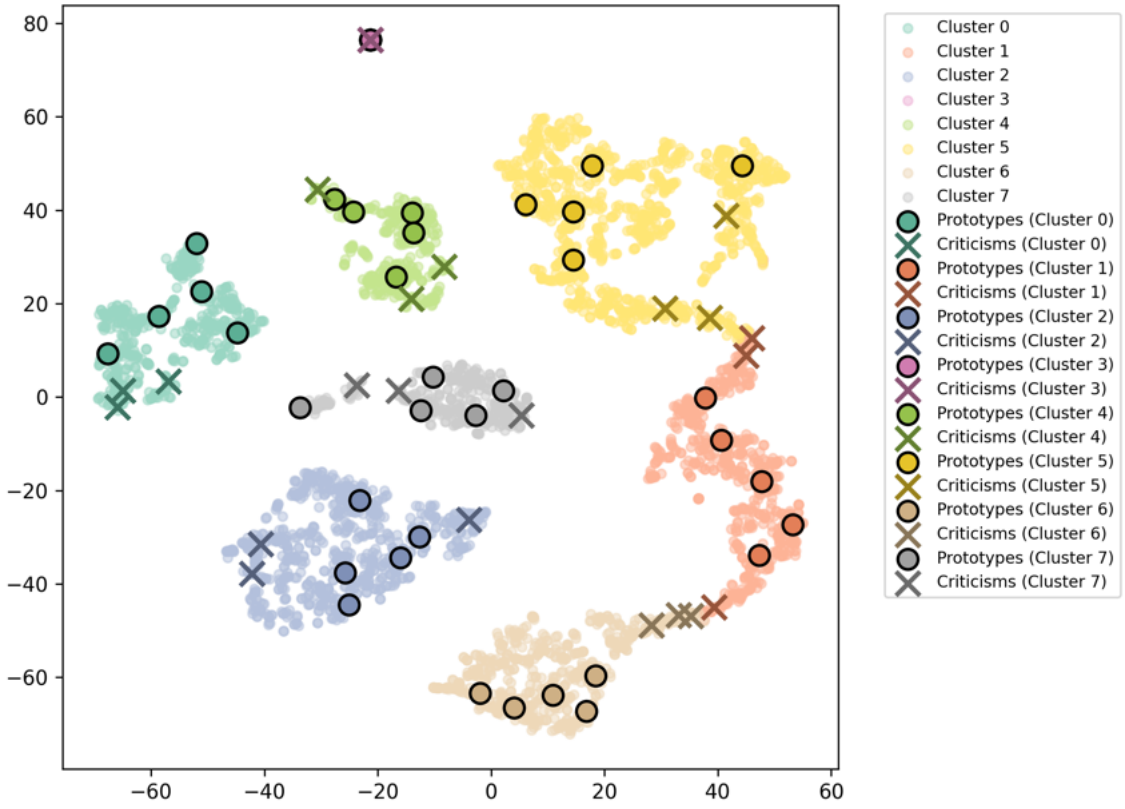


Figure 4.11: t-SNE projection of the latent space learned by the dilated CNN with multi_reconstruction and SDCN model on the multivariate energy-consumption dataset with k=8 presenting the 5 prototypes and 3 criticisms for each cluster.

and overnight (1AM–5AM).

In k=0, prototypes show a sharp midday plateau followed by an evening trough. Criticisms deepen that trough and reveal late night to overnight rebounds. In k=1, prototypes capture a suppressed day with a steep evening to night ramp persisting into overnight, whereas criticisms push this rise later and steeper. k=2 prototypes encode a late-day step and broad evening plateau with gradual night decay. Criticisms highlight higher and later plateaus or slower post 11PM declines. Cluster 3 corresponds to a zero-load regime, where all daily profiles exhibit a perfectly flat consumption of 0 kW throughout the 24-hour period. Such profiles typically arise from inactive or disconnected meters, vacant units, or potential data-recording errors. Because the cluster contains no internal variability, the prototype–criticism framework cannot form meaningful similarity relations, resulting in zero values across coverage, redundancy, and critic-severity metrics. Consequently, this group represents a structural class rather than a behavioral consumption pattern. Cluster 3 reflects a zero-consumption regime, where all profiles record 0 kWh throughout the day. This pattern is most consistent with inactive or temporarily disconnected meters rather than an operational load shape. Because the cluster contains no internal variability, the prototype–criticism metrics (coverage, redundancy, critic severity) all return zero, indicating the absence of meaningful structure to explain. k=4 prototypes exhibit a two-hump day, while criticisms exaggerate hump amplitudes, deepen the inter-hump dip, or delay the evening drop, consistent with the cluster's higher critic-severity score. k=5 prototypes remain quiet through daytime before a two-stage night to overnight surge. Criticisms emphasize earlier and later step times, deeper pre-spike dips, or a sharper midnight rebound, aligning with its very high prototype redundancy. In k=6, prototypes show a step at 6 PM followed by a monotonic night-to-pre-dawn climb. Criticisms extend or advance this rise and occasionally introduce small daytime blips. Finally, k=7 prototypes represent a day-active, bursty regime with a strong evening plateau and limited overnight activity. Criticisms represent sporadic daytime spikes or slightly prolonged evening levels, matching the cluster's high coverage, low redundancy, and very low critic-severity. Overall, prototypes anchor the dense cores of each cluster, while criticisms systematically surface the edge cases for example later ramps, deeper troughs, or prolonged plateaus that define within-cluster heterogeneity.

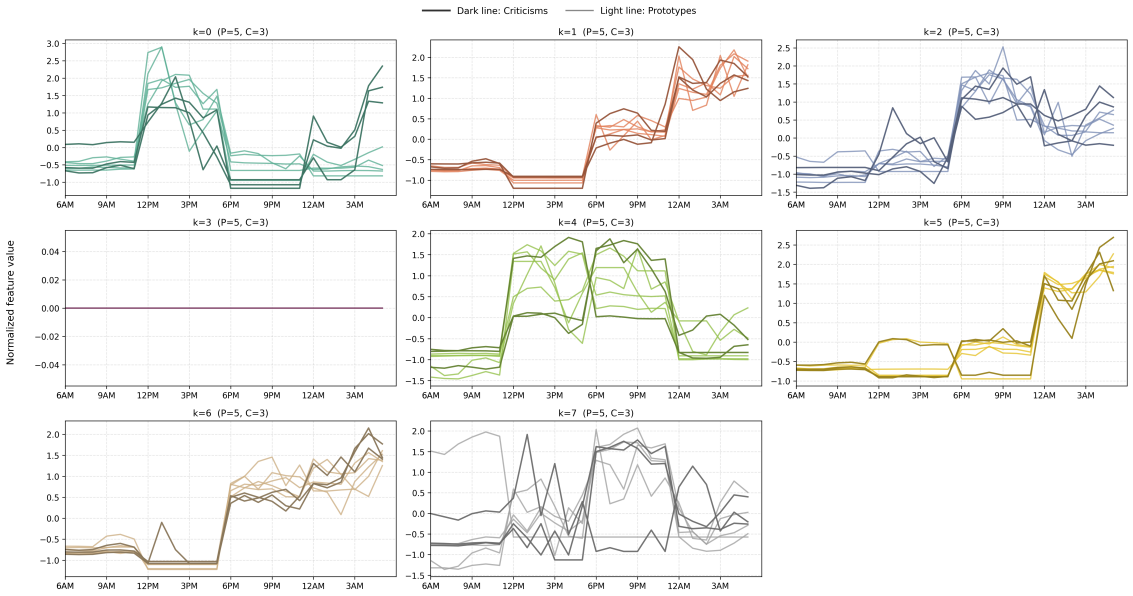


Figure 4.12: Channel-averaged multivariate daily profiles (6 AM $\rightarrow$ 6 AM) for prototypes (light) and criticisms (dark) across all clusters. At each time step values are averaged across electricity, chilled water, steam, and hot water

Chapter 5

Conclusion

Using univariate and multivariate energy consumption datasets, this thesis develops a unified methodological line that starts from traditional clustering with rule-based explanations and extends to deep time-series clustering with evolutionary tuning and prototype–criticism analysis, all evaluated under a common internal validation framework.

The first part of the thesis addressed two key challenges: the stability and validity of traditional clustering results under realistic data conditions, and the explainability of clustering results for operational and policy-oriented decision-making. It evaluated how traditional hard and soft clustering algorithms cluster residential daily electricity demand profiles and how their performance depends on intra- and inter-cluster characteristics. By comparing K-Means and K-Medoids with Fuzzy C-Means and Gaussian Mixture Models, the study showed that soft clustering methods, particularly GMM, are better suited to capturing overlapping regimes and subtle variations in consumption, while hard methods tend to struggle with boundary definition and sensitivity to outliers. Controlled variations in data characteristics such as outliers, overlap, density, kurtosis, skewness, and sub-clustering revealed distinct behaviors among five internal cluster validity indices. Silhouette, Calinski–Harabasz, and Xie–Beni generally behaved robustly across most conditions, whereas Dunn was overly sensitive to minimum inter-cluster distances and skewed shapes, and Davies–Bouldin often deteriorated in the presence of complex separation patterns. Dimensionality reduction had limited impact on the fundamental clustering structure, but occasionally improved separability in scenarios with overlapping profiles or differential density.

To address the frequent criticism that clustering is a black box, the thesis incorporated axis-aligned and sparse oblique decision trees to explain cluster assignments. The results highlighted a clear trade-off: axis-aligned trees provided full coverage of clusters but produced increasingly complex rule sets as the number of clusters grew, while sparse oblique trees yielded more compact and interpretable rules at the cost of occasionally leaving some clusters underrepresented. These rule-based explanations connected clusters to time-of-day thresholds, peak and off-peak patterns, and typical daily routines, offering an explainable bridge between abstract cluster labels and household behavior.

Building on this foundation, the thesis then benchmarked four traditional algorithms and deep clustering pipelines on univariate and multivariate building-energy datasets, using the same internal validity indices. The traditional algorithms were Euclidean K-Means, DTW K-Means, K-Shape, and cosine-DBSCAN. Among these baselines, Euclidean K-Means emerged as the most reliable method overall, with DTW-based K-Means performing competitively in scenarios with fewer, broader regimes and K-Shape consistently underperforming. Cosine-DBSCAN provided complementary density-based insights by identifying compact, shape-dense regimes with manageable levels of noise, but its overall performance remained below that of the best deep models.

Deep clustering pipelines consistently outperformed traditional approaches. Across both univariate and multivariate settings, convolutional architectures dominated, with dilated CNNs offering the strongest overall performance and residual CNNs closely following, particularly in multivariate configurations. Fully convolutional networks remained competitive on univariate data, while recurrent architectures such as Bi-GRU and Bi-LSTM delivered moderate but stable performance. Among pretext losses, reconstruction-based objectives proved to be the most reliable choice, producing robust latent spaces, with VAE losses performing especially well for univariate profiles but less consistently on multivariate data. Triplet loss offered additional gains in some multivariate configurations after clustering refinement, while more complex reconstruction schemes and GAN-based losses failed to deliver systematic improvements and often introduced instability.

Among clustering losses, DEC emerged as the most stable and generally best-performing option, especially for univariate datasets, whereas SDCN excelled in multivariate cases by leveraging graph-based structural regularization to capture cross-channel coupling. IDEC was effective during

representation learning with clustering regularization but less dominant after full clustering optimization, and methods like DTCR and DEPICT formed a competitive second tier. In some multivariate settings, models without explicit clustering loss reached strong internal scores but lacked the stability of DEC- or SDCN-based configurations. Population-Based Training for hyperparameter tuning further improved performance and stability, particularly for multivariate pipelines, by steering the search toward broad high-performing regions in hyperparameter space, even though univariate searches retained more heterogeneity due to their more exploratory restart dynamics.

To ensure that the superior performance of deep clustering did not come at the expense of transparency, the thesis integrated prototype–criticism explainability on top of the learned latent spaces. For each cluster, a small set of prototypes and criticisms summarized the core and boundary behaviors of the cluster. The proposed metrics, coverage, redundancy, and critic severity, quantified how well these selected examples represented typical consumption patterns, avoided redundant representations, and highlighted under-represented or transitional regimes. Applied to multivariate energy profiles, this analysis confirmed the semantic coherence of the learned clusters: prototypes captured dense operational cores, while criticisms revealed patterns such as shifted ramps, extended plateaus, or mixed-load interactions that can be operationally important. This example-based explanation complements the rule-based decision-tree view from the first part of the thesis, offering practitioners two complementary lenses for understanding and acting on clustering results.

Overall, this thesis shows that clustering workflows for building-energy time series can move beyond black-box analysis and become transparent tools that directly support operational decision-making. By unifying traditional and deep clustering, internal validation, and explainable AI within a single methodological framework, the work provides both theoretical insights and practical tools for turning raw smart-meter and multi-energy data into interpretable behavioral archetypes that can support demand-side management, portfolio analytics, and the development of more efficient and adaptive energy systems.

Appendix A

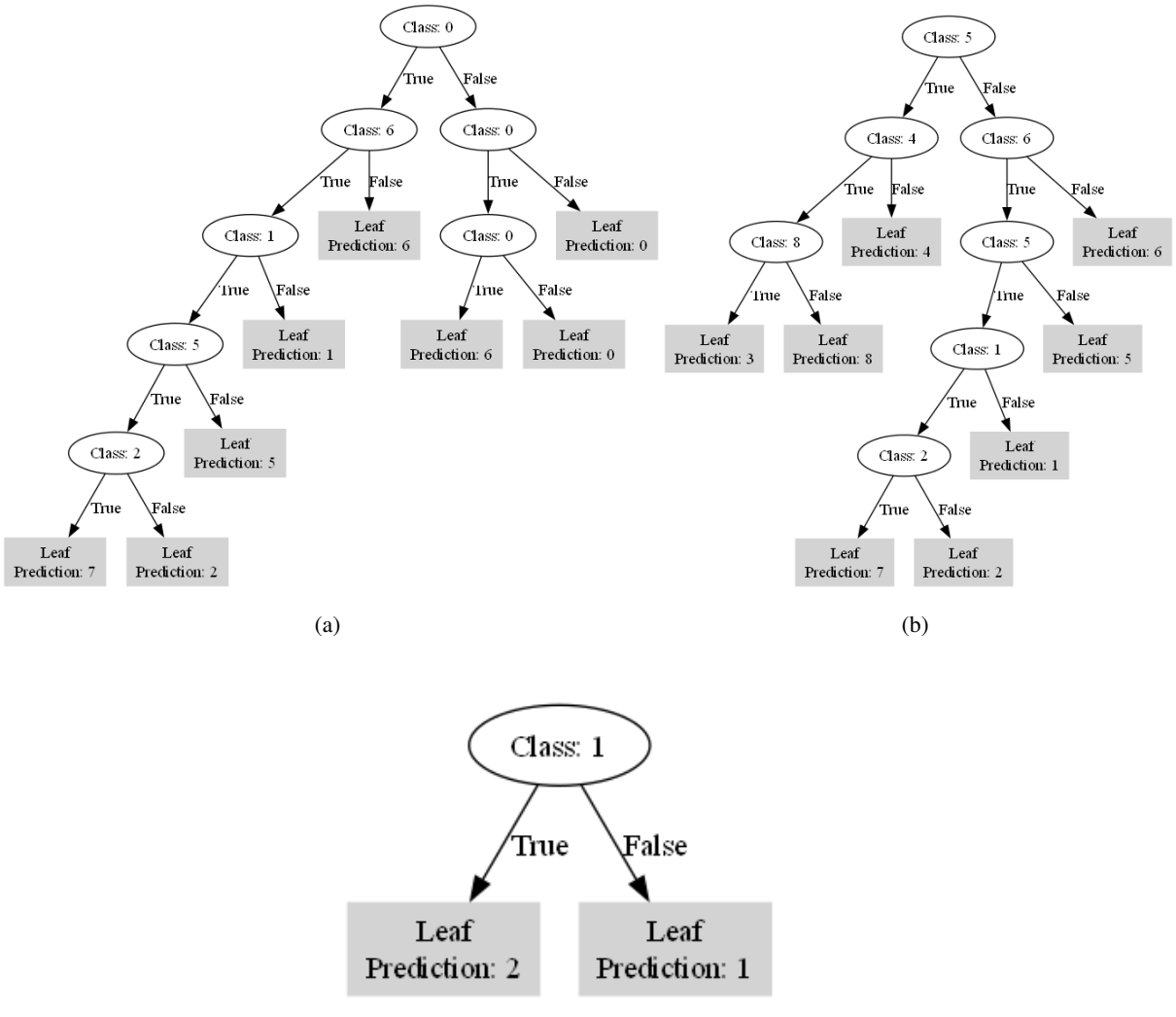
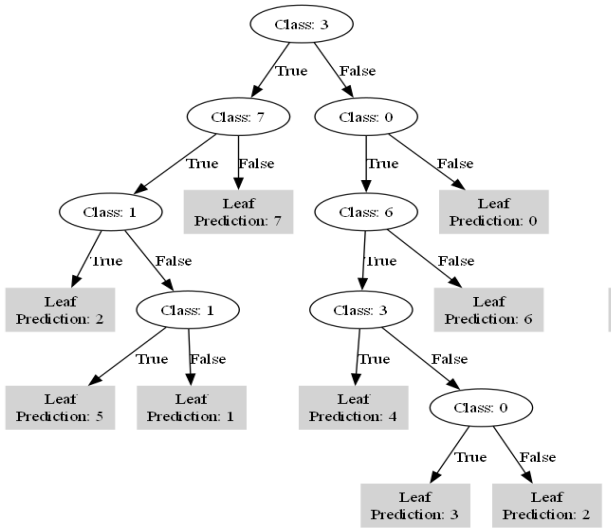
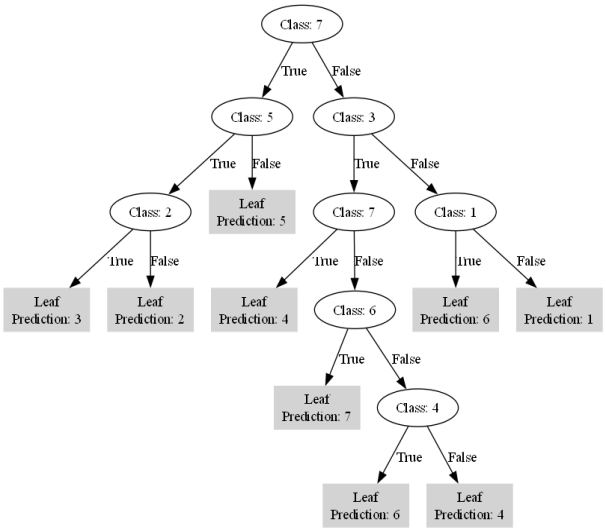


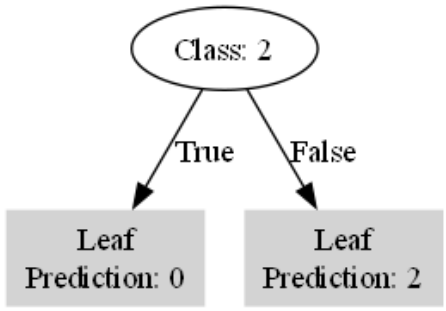
Figure A.1: Sparse oblique decision tree using different algorithms on the EL dataset without dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means.



(a)

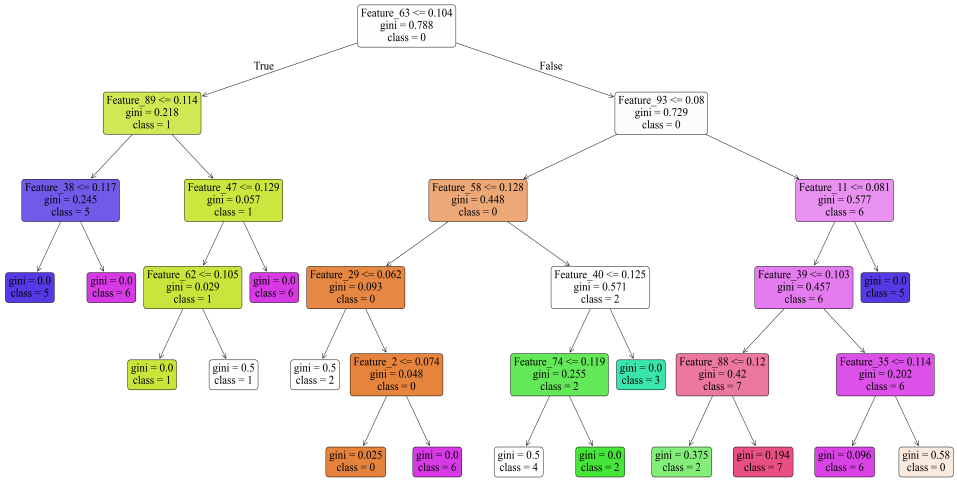


(b)

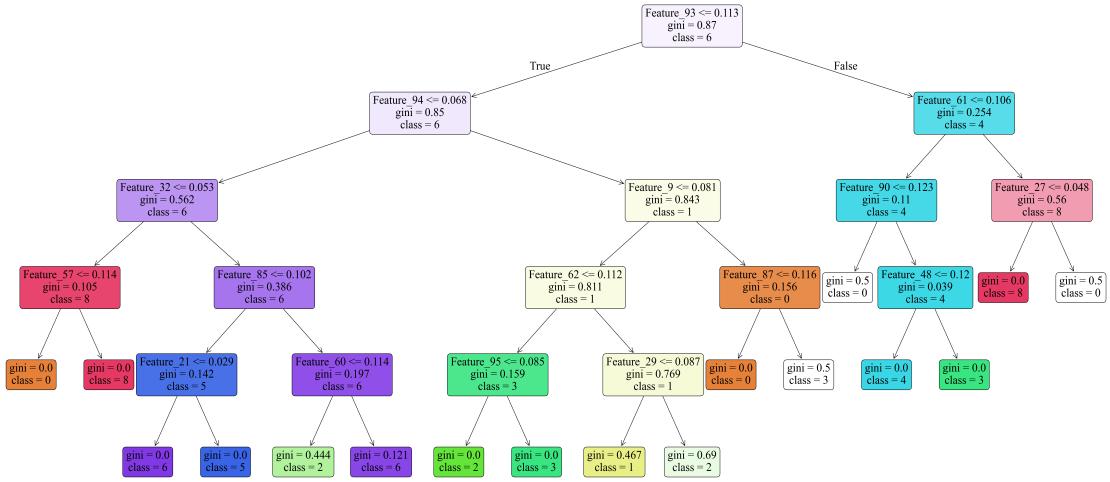


(c)

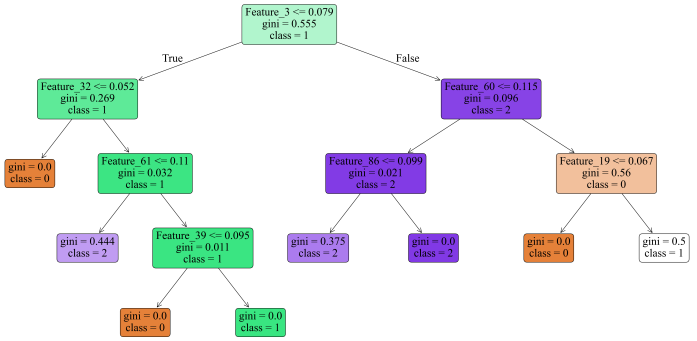
Figure A.2: Sparse oblique decision tree using different algorithms on the EL dataset with dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means.



(a) K-means without DR

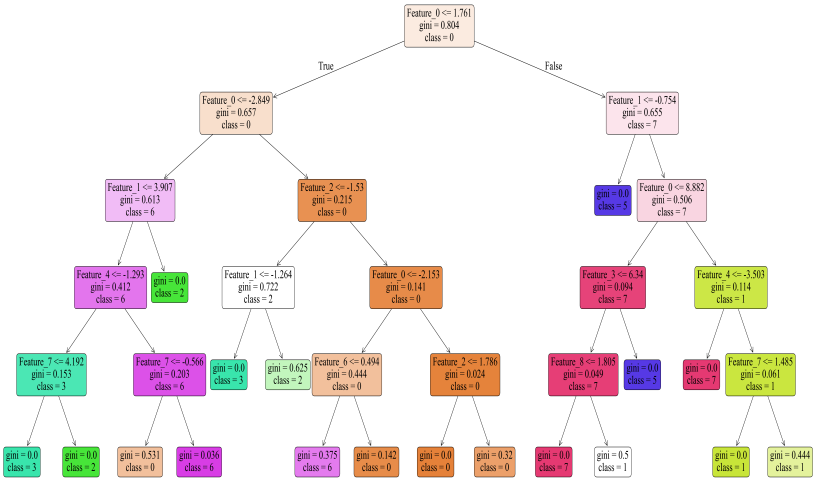


(b) K-medoids without DR

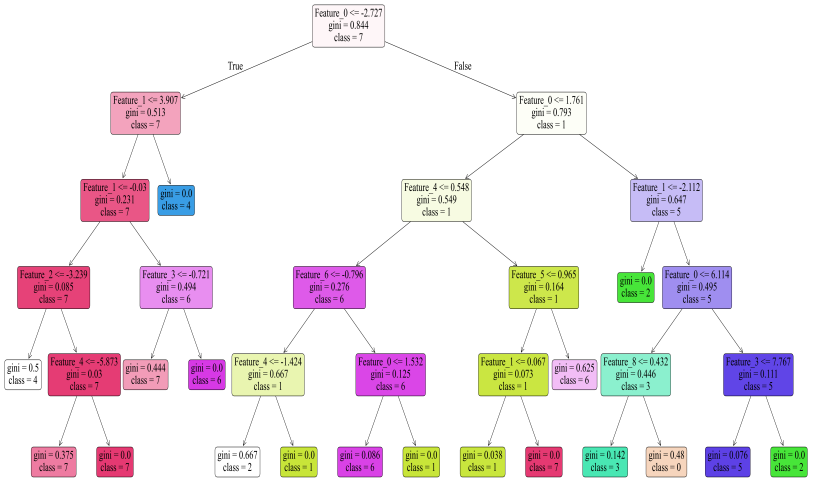


(c) Fuzzy C-means without DR

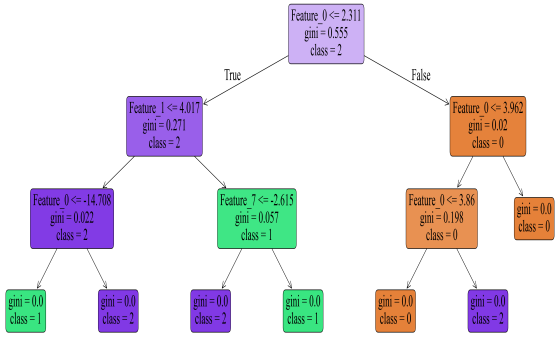
Figure A.3: Axis aligned tree using different algorithms on the EL dataset without dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means.



(a) K-means with DR



(b) K-medoids with DR



(c) Fuzzy C-means with DR

Figure A.4: Axis aligned tree using different algorithms on the EL dataset with dimensionality reduction: (**a**) K-Means, (**b**) K-Medoids, (**c**) Fuzzy C-Means.

Bibliography

[1] International Energy Agency (IEA). Global energy review 2025: Electricity. https://www.iea.org/reports/global-energy-review-2025/electricity, 2025. Accessed: 2025-11-13.

[2] McKinsey & Company. Global energy perspective 2025. https://www.mckinsey.com/industries/energy-and-materials/our-insights/global-energy-perspective, 2025. Accessed: 2025-11-13.

[3] United Nations Environment Programme (UNEP). Global status report for buildings and construction 2024/2025. https://www.unep.org/resources/report/global-status-report-buildings-and-construction-2024-2025, 2025. Accessed: 2025-11-13.

[4] Mwoya Byaro, Juvenal Nkonoki, and Gemma Mafwolo. Exploring the nexus between natural resource depletion, renewable energy use, and environmental degradation in sub-saharan africa. *Environmental Science and Pollution Research*, 30(8):19931–19945, 2023.

[5] Frederica Perera. Pollution from fossil-fuel combustion is the leading environmental threat to global pediatric health and equity: Solutions exist. *International journal of environmental research and public health*, 15(1):16, 2018.

[6] Alena Lohrmann, Javier Farfan, Upeksha Caldera, Christoph Lohrmann, and Christian Breyer. Global scenarios for significant water use reduction in thermal power plants based on cooling water demand estimation using satellite imagery. *Nature Energy*, 4(12):1040–1048, 2019.

[7] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on smart Grid*, 10(3):3125–3148, 2018.

[8] Bing Dong, Vishnu Prakash, Fan Feng, and Zheng O'Neill. A review of smart building sensing system for better indoor environment control. *Energy and Buildings*, 199:29–46, 2019.

[9] Muhammad Waseem Ahmad, Monjur Mourshed, David Mundow, Mario Sisinni, and Yacine Rezgui. Building energy metering and environmental monitoring–a state-of-the-art review and directions for future research. *Energy and Buildings*, 120:85–102, 2016.

[10] Mengda Jia, Ali Komeily, Yueren Wang, and Ravi S Srinivasan. Adopting internet of things for the development of smart buildings: A review of enabling technologies and applications. *Automation in construction*, 101:111–126, 2019.

[11] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. Smart grid—the new and improved power grid: A survey. *IEEE communications surveys & tutorials*, 14(4):944–980, 2011.

[12] Raluca Laura Portase, Ramona Tolas, and Rodica Potolea. From sensors to insights: An original method for consumer behavior identification in appliance usage. *Electronics*, 13(7):1364, 2024.

[13] Gianfranco Chicco, Roberto Napoli, and Federico Piglione. Comparisons among clustering techniques for electricity customer classification. *IEEE Transactions on power systems*, 21(2):933–940, 2006.

[14] Christoph Flath, David Nicolay, Tobias Conte, Clemens Van Dinther, and Lilia Filipova-Neumann. Cluster analysis of smart metering data: An implementation in practice. *Business & Information Systems Engineering*, 4(1):31–39, 2012.

[15] Eng L Ofetotse, Emmanuel A Essah, and Runming Yao. Evaluating the determinants of

household electricity consumption using cluster analysis. *Journal of Building Engineering*, 43:102487, 2021.

[16] Ian Ayres, Sophie Raseman, and Alice Shih. Evidence from two large field experiments that peer comparison feedback can reduce residential energy usage. *The Journal of Law, Economics, & Organization*, 29(5):992–1022, 2013.

[17] Yassine Bouabdallaoui, Zoubeir Lafhaj, Pascal Yim, Laure Ducoulombier, and Belkacem Bennadji. Predictive maintenance in building facilities: A machine learning-based approach. *Sensors*, 21(4):1044, 2021.

[18] Fanlin Meng, Qian Ma, Zixu Liu, and Xiao-Jun Zeng. Multiple dynamic pricing for demand response with adaptive clustering-based customer segmentation in smart grids. *Applied Energy*, 333:120626, 2023.

[19] Sarra Kallel, Manar Amayri, and Nizar Bouguila. Clustering and interpretability of residential electricity demand profiles. *Sensors*, 25(7):2026, 2025.

[20] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[21] Sarra Kallel, Manar Amayri, and Nizar Bouguila. Explainable deep representation learning for clustering building-energy time series. Manuscript submitted for publication, 2025.

[22] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. *Information systems*, 53:16–38, 2015.

[23] Sangeeta Rani and Geeta Sikka. Recent techniques of clustering of time series data: a survey. *International Journal of Computer Applications*, 52(15), 2012.

[24] Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. Learning representations for time series clustering. *Advances in neural information processing systems*, 32, 2019.

[25] John Paparrizos, Fan Yang, and Haojun Li. Bridging the gap: A decade review of time-series clustering methods. *arXiv preprint arXiv:2412.20582*, 2024.

[26] Federico Reppucci. A clustering-based approach for city electricity demand forecasting. Master's thesis, Politecnico di Milano, Milan, Italy, 2018.

[27] Eamonn Keogh and Shruti Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 102–111, 2002.

[28] Baptiste Lafabregue, Jonathan Weber, Pierre Gançarski, and Germain Forestier. End-to-end deep representation learning for time series clustering: a comparative study. *Data mining and knowledge discovery*, 36(1):29–81, 2022.

[29] Vit Niennattrakul and Chotirat Ann Ratanamahatana. On clustering multimedia time series data using k-means and dynamic time warping. In *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, pages 733–738. IEEE, 2007.

[30] J. V. De Oliveira and W. Pedrycz. *Advances in Fuzzy Clustering and Its Applications*. John Wiley & Sons, 2007.

[31] N. H. M. M. Shrifan, M. F. Akbar, and N. A. M. Isa. An adaptive outlier removal aided k-means clustering algorithm. *Journal of King Saud University–Computer and Information Sciences*, 34:6365–6376, 2022.

[32] P. Arora and S. Varshney. Analysis of k-means and k-medoids algorithm for big data. In *Procedia Computer Science*, volume 78, pages 507–512, 2016.

[33] B. Mirkin. *Clustering for Data Mining: A Data Recovery Approach*. Chapman and Hall/CRC, 2005.

[34] S. Ghosh and S. K. Dubey. Comparative analysis of k-means and fuzzy c-means algorithms. *International Journal of Advanced Computer Science and Applications*, 4:1–7, 2013.

[35] E. Patel and D. S. Kushwaha. Clustering cloud workloads: K-means vs gaussian mixture model. *Procedia Computer Science*, 171:158–167, 2020.

[36] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[37] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[38] Paige Wenbin Tien, Shuangyu Wei, Jo Darkwa, Christopher Wood, and John Kaiser Calautit. Machine learning and deep learning methods for enhancing building energy efficiency and indoor environmental quality–a review. *Energy and AI*, 10:100198, 2022.

[39] Xuefeng Gao and Ali Malkawi. A new methodology for building energy performance benchmarking: An approach based on intelligent clustering algorithm. *Energy and Buildings*, 84:607–616, 2014.

[40] Pandarasamy Arjunan, Kameshwar Poolla, and Clayton Miller. Beem: Data-driven building energy benchmarking for singapore. *Energy and Buildings*, 260:111869, 2022.

[41] Félix Iglesias and Wolfgang Kastner. Analysis of similarity measures in times series clustering for the discovery of building energy patterns. *Energies*, 6(2):579–597, 2013.

[42] R. Damayanti, A. G. Abdullah, W. Purnama, and A. B. D. Nandiyanto. Electrical load profile analysis using clustering techniques. In *IOP Conference Series: Materials Science and Engineering*, volume 180, page 012081, 2017.

[43] Wiebke Toussaint and Deshendran Moodley. Clustering residential electricity consumption data to create archetypes that capture household behaviour in south africa. *South African Computer Journal*, 32(2):1–34, 2020.

[44] Yixiu Guo, Yong Li, Sisi Zhou, Zhenyu Zhang, Zuyi Li, and Mohammad Shahidehpour. A data-driven three-stage adaptive pattern mining approach for multi-energy loads. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[45] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham

Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[46] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[47] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE transactions on visualization and computer graphics*, 25(8):2674–2693, 2018.

[48] O. Loyola-Gonzalez, A. E. Gutierrez-Rodríguez, M. A. Medina-Pérez, R. Monroy, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and M. Garcia-Borroto. An explainable artificial intelligence model for clustering numerical databases. *IEEE Access*, 8:52370–52384, 2020.

[49] L. Hu, M. Jiang, J. Dong, X. Liu, and Z. He. Interpretable clustering: A survey. *arXiv preprint*, 2024.

[50] S. Bandyapadhyay, F. V. Fomin, P. A. Golovach, W. Lochet, N. Purohit, and K. Simonov. How to find a good explanation for clustering? *Artificial Intelligence*, 322:103948, 2023.

[51] D. Bertsimas, A. Orfanoudaki, and H. Wiberg. Interpretable clustering: An optimization approach. *Machine Learning*, 110:89–138, 2021.

[52] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[53] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[54] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[55] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[56] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[57] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[58] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[59] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018.

[60] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[61] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31, 2018.

[62] Javier Antorán, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a clue: A method for explaining uncertainty estimates. *arXiv preprint arXiv:2006.06848*, 2020.

[63] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

[64] S. A. L. Mary, A. N. Sivagami, and M. U. Rani. Cluster validity measures dynamic clustering algorithms. *ARPN Journal of Engineering and Applied Sciences*, 10:4009–4012, 2015.

[65] G. Liu. A new index for clustering evaluation based on density estimation. *arXiv preprint arXiv:2207.01294*, 2022.

[66] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17:107–145, 2001.

[67] J.-C. Lamirel, N. Dugué, and P. Cuxac. New efficient clustering quality indexes. In *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3649–3657. IEEE, 2016.

[68] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 911–916. IEEE, 2010.

[69] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao. Data mining techniques and applications: A decade review from 2000 to 2011. *Expert Systems with Applications*, 39:11303–11311, 2012.

[70] Steven R Young, Derek C Rose, Thomas P Karnowski, Seung-Hwan Lim, and Robert M Patton. Optimizing deep learning hyper-parameters through an evolutionary algorithm. In *Proceedings of the workshop on machine learning in high-performance computing environments*, pages 1–5, 2015.

[71] R. Mikut and M. Reischl. Data mining tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1:431–443, 2011.

[72] W. Toussaint and D. Moodley. Clustering residential electricity consumption data to create archetypes that capture household behaviour in south africa. *South African Computer Journal*, 32:1–34, 2020.

[73] M. Jain, M. Jain, T. AlSkaif, and S. Dev. Which internal validation indices to use while clustering electric load demand profiles? *Sustainable Energy, Grids and Networks*, 32:100849, 2022.

[74] K. Zhou, N. Peng, D. Hu, and Z. Shao. Industrial park electric power load pattern recognition: An ensemble clustering-based framework. *Energy and Buildings*, 279:112687, 2023.

[75] A. Trindade. Electricity load diagrams 2011–2014 dataset. https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014, 2014. Accessed on 4 January 2025.

[76] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In *International Work-Conference on Artificial Neural Networks*, pages 758–770. Springer, 2005.

[77] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2:433–459, 2010.

[78] T. Raykov and G. A. Marcoulides. Population proportion of explained variance in principal component analysis: A note on its evaluation via a large-sample approach. *Structural Equation Modeling: A Multidisciplinary Journal*, 21:588–595, 2014.

[79] J. Paparrizos and L. Gravano. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)*, 42(2):1–49, 2017.

[80] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto. Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In *IOP Conference Series: Materials Science and Engineering*, volume 336, page 012017. IOP Publishing, 2018.

[81] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63:411–423, 2001.

[82] S. Saitta, B. Raphael, and I. F. C. Smith. A comprehensive validity index for clustering. *Intelligent Data Analysis*, 12:529–548, 2008.

[83] H. Al-Bazzaz, M. Azam, M. Amayri, and N. Bouguila. Explainable finite mixture of mixtures of bounded asymmetric generalized gaussian and uniform distributions learning for energy demand management. *ACM Transactions on Intelligent Systems and Technology*, 15:1–64, 2024.

[84] H. Al-Bazzaz, M. Azam, M. Amayri, and N. Bouguila. Enhanced energy characterization and feature selection using explainable non-parametric aggmm. In *Recent Challenges in Intelligent Information and Database Systems – 15th Asian Conference (ACIIDS 2023)*, pages 145–156. Springer, 2023.

[85] K. Prabhakaran, J. Dridi, M. Amayri, and N. Bouguila. Explainable k-means clustering for occupancy estimation. In *17th International Conference on Future Networks and Communications / 19th International Conference on Mobile Systems and Pervasive Computing / 12th International Conference on Sustainable Energy Information Technology (FNC/MobiSPC/SEIT 2022)*, pages 326–333, 2022.

[86] H. Al-Bazzaz, M. Azam, M. Amayri, and N. Bouguila. Explainable robust smart meter data clustering for improved energy management. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 5194–5199. IEEE, 2023.

[87] H. Al-Bazzaz, K. S. Prabhakaran, M. Amayri, and N. Bouguila. Refining nonparametric mixture models with explainability for smart building applications. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 5212–5217. IEEE, 2023.

[88] Global Alliance for Buildings and Construction (GlobalABC) and United Nations Environment Programme (UNEP). 2021 global status report for buildings and construction, 2021.

[89] Ying Sun, Fariborz Haghighat, and Benjamin CM Fung. A review of the-state-of-the-art in data-driven approaches for building energy prediction. *Energy and Buildings*, 221:110022, 2020.

[90] Jungsuk Kwac, June Flora, and Ram Rajagopal. Household energy consumption segmentation using hourly data. *IEEE Transactions on Smart Grid*, 5(1):420–430, 2014.

[91] Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of clustering high dimensional data. In *New directions in statistical physics: econophysics, bioinformatics, and pattern recognition*, pages 273–309. Springer, 2004.

[92] Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Computing Surveys*, 57(3):1–38, 2024.

[93] Ali Alqahtani, Mohammed Ali, Xianghua Xie, and Mark W Jones. Deep time-series clustering: A review. *Electronics*, 10(23):3001, 2021.

[94] Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE access*, 6:39501–39514, 2018.

[95] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning*, pages 1437–1446. PMLR, 2018.

[96] Lianyu Hu, Mudi Jiang, Junjie Dong, Xinying Liu, and Zengyou He. Interpretable clustering: A survey. *arXiv preprint arXiv:2409.00743*, 2024.

[97] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[98] Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel,

Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. Interpreting black-box models: a review on explainable artificial intelligence. *Cognitive Computation*, 16(1):45–74, 2024.

[99] Naveen Sai Madiraju. Deep temporal clustering: Fully unsupervised learning of time-domain features. Master's thesis, Arizona State University, 2018.

[100] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[101] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[102] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4066–4075, 2019.

[103] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(1):281–305, 2012.

[104] Steven R Young, Derek C Rose, Thomas P Karnowski, Seung-Hwan Lim, and Robert M Patton. Optimizing deep learning hyper-parameters through an evolutionary algorithm. In *Proceedings of the workshop on machine learning in high-performance computing environments*, pages 1–5, 2015.

[105] Ismail Damilola Raji, Habeeb Bello-Salau, Ime Jarlath Umoh, Adeiza James Onumanyi, Mutiu Adesina Adegboye, and Ahmed Tijani Salawudeen. Simple deterministic selection-based genetic algorithm for hyperparameter tuning of machine learning models. *Applied Sciences*, 12(3):1186, 2022.

[106] Amala Mary Vincent and P Jidesh. An improved hyperparameter optimization framework for automl systems using evolutionary algorithms. *Scientific Reports*, 13(1):4737, 2023.

[107] Pradnya A Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*, pages 261–265. IEEE, 2016.

[108] Adam Slowik and Halina Kwasnicka. Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, 32(16):12363–12379, 2020.

[109] ASHRAE. Ashrae – great energy predictor iii: Competition data. Kaggle dataset, 2020. Accessed: 2025-11-06.

[110] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[111] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

[112] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, pages 5736–5745, 2017.

[113] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[114] Kamran Ghasedi, Xiaoqian Wang, Cheng Deng, and Heng Huang. Balanced self-paced learning for generative adversarial clustering network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4391–4400, 2019.

[115] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4610–4617, 2019.

[116] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Ijcai*, volume 17, pages 1753–1759, 2017.

[117] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of the web conference 2020*, pages 1400–1410, 2020.

[118] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: A generative approach to clustering. *CoRR, abs/1611.05148*, 1, 2016.