universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Multi-View Clustering of Smart Meter Data

verfasst von | submitted by

Zoheir El Houari

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	Ass.-Prof. Dott.ssa Dott.ssa.mag. Yllka Velaj PhD

Acknowledgments

I would like to express my gratitude to my supervisor, **Ass.-Prof. Dott.ssa Dott.ssa.mag PhD Yllka Velaj**, for her guidance, expertise, and insightful feedback throughout the course of this research. Her mentorship and support have been central to the successful completion of this work.

I would also like to extend my sincere thanks to **Pascal Weber, B.Sc. M.Sc.**, and **Peter Salah, M.Eng. MSc**, for their valuable suggestions, helpful discussions, and continuous assistance during the preparation of this thesis. Although not formally appointed as supervisors, their engagement with this work contributed substantially to its development and refinement. Finally, I wish to thank my family and friends for their patience and support throughout the course of this research.

Abstract

Smart meter infrastructures generate large-scale, high-dimensional time-series data that enable detailed analysis of household electricity consumption. However, clustering such data in isolation often leads to unstable or less interpretable results due to noise, high temporal variability, and absence of contextual information. At the same time, external factors such as temperature and solar radiation play a significant role in energy demand, making their integration essential for a more comprehensive analysis of consumption behavior. Although multi-view clustering provides a framework for integrating multiple heterogeneous data sources, most existing methods are not explicitly designed nor evaluated on large-scale multi-view time-series datasets, and often face scalability or representation challenges.

In this thesis, we address the problem of scalable multi-view clustering for high-dimensional smart meter time-series data. To tackle this challenge, the FastMICE framework, a scalable ensemble-based multi-view clustering method, is adopted as the core methodology. To improve its applicability to multi-view time-series data, several extensions are introduced, including time-aware feature sampling strategies, feature normalization procedures, and symbolic time-series representations using Symbolic Aggregate Approximation (SAX).

The framework is evaluated on a real-world smart meter dataset containing approximately 195,000 households with daily electricity consumption over one year, complemented with temperature and solar radiation views. Since ground-truth labels are not available for this dataset, additional labeled benchmark datasets (CMAPSS, HAR, SCRM, and ALOI) are used for quantitative evaluation. The method is compared against representative multi-view clustering baseline approaches using standard evaluation metrics.

Kurzfassung

Intelligente Zählerinfrastrukturen generieren groß angelegte, hochdimensionale Zeitreihendaten, die eine detaillierte Analyse des Stromverbrauchs von Haushalten ermöglichen. Die isolierte Clusterung solcher Daten führt jedoch aufgrund von Rauschen, hoher zeitlicher Variabilität und fehlenden Kontextinformationen oft zu instabilen oder weniger interpretierbaren Ergebnissen. Gleichzeitig spielen externe Faktoren wie Temperatur und Sonneneinstrahlung eine wichtige Rolle für den Energiebedarf, sodass ihre Einbeziehung für eine umfassendere Analyse des Verbrauchsverhaltens unerlässlich ist. Obwohl Multi-View-Clustering einen Rahmen für die Integration mehrerer heterogener Datenquellen bietet, sind die meisten bestehenden Methoden nicht explizit für große Multi-View-Zeitreihendatensätze konzipiert oder evaluiert und stehen oft vor Herausforderungen hinsichtlich Skalierbarkeit oder Darstellung.

In dieser Arbeit befassen wir uns mit dem Problem des skalierbaren Multi-View-Clusterings für hochdimensionale Zeitreihendaten von intelligenten Stromzählern. Um diese Herausforderung zu bewältigen, wird das FastMICE-Framework, eine skalierbare ensemblebasierte Multi-View-Clustering-Methode, als Kernmethodik verwendet. Um die Anwendbarkeit auf Multi-View-Zeitreihendaten zu verbessern, werden mehrere Erweiterungen eingeführt, darunter zeitbewusste Merkmals-Sampling-Strategien, Merkmalsnormalisierungsverfahren und symbolische Zeitreihendarstellungen unter Verwendung von Symbolic Aggregate Approximation (SAX).

Das Framework wird anhand eines realen Smart-Meter-Datensatzes evaluiert, der etwa 195.000 Haushalte mit ihrem täglichen Stromverbrauch über ein Jahr hinweg enthält und durch Temperatur- und Sonneneinstrahlungsdaten ergänzt wird. Da für diesen Datensatz keine Ground-Truth-Labels verfügbar sind, werden zusätzliche beschriftete Benchmark-Datensätze (CMAPSS, HAR, SCRM und ALOI) für die quantitative Bewertung herangezogen. Die Methode wird anhand von Standard-Bewertungsmetriken mit repräsentativen Multi-View-Clustering-Baseline-Ansätzen verglichen.

Contents

1 Introduction

"Clustering is a data mining technique where similar data are placed into related or homogeneous groups without advanced knowledge of the groups' definitions. In detail, clusters are formed by grouping objects that have maximum similarity with other objects within the group, and minimum similarity with objects in other groups. It is a useful approach for exploratory data analysis as it identifies structure(s) in an unlabeled dataset by objectively organizing data into similar groups. Moreover, clustering is used for exploratory data analysis for summary generation and as a pre-processing step for other data mining tasks or as a part of a complex system."[1].

When the data objects are temporally ordered observations, this idea naturally extends to time-series clustering, where similar sequences are grouped to reveal recurring temporal patterns and characteristic behaviors [2, 3]. Time series data consist of measurements collected chronologically (often at regular intervals) and are widely used across various industries, including finance, healthcare, retail, and energy [4, 5, 6, 7]. In such settings, extracting and analyzing trend components helps summarize long-term evolution, while forecasting leverages historical observations to anticipate future values in order to support planning and decision-making [8, 6].

1.1 Problem Setting

In the energy sector, data collected from grid sensors and smart meters is increasingly being used to develop various applications within the distribution system [9]. These devices record electricity usage at fine intervals—daily, hourly, or even by the minute. The recorded data enables analyses that reveal consumption patterns such as daily peaks, seasonal trends, and anomalies in consumption.

These insights help understand consumption dynamics at the individual or household level, enabling energy providers to optimize distribution, improve grid stability, and develop targeted demand response strategies. However, smart meter readings do not capture external factors such as weather conditions, geographical location, and population density, which also influence electricity usage. Among these, weather plays a crucial role in shaping electricity demand, especially with the increasing use of air conditioning systems. This relationship has been acknowledged since the 1940s, with early studies such as [10] highlighting the link between weather and energy consumption.

More recent research, including a 2022 study [17], which analyzed data from over 3,800 Irish households, confirmed that weather factors like temperature, rainfall, and sunlight duration significantly influence electricity usage. To address these complexities, multi-view clustering has emerged as a promising approach for integrating external factors

into electricity consumption analysis. By combining multiple data sources, or "views" such as weather information, geographical details, and population demographics, multi-view clustering offers a more comprehensive view of energy consumption behaviors. Despite its potential, this approach presents several challenges. Variations in the format, scale, and completeness of different data sources can complicate the clustering process. Moreover, selecting appropriate similarity measures and algorithms is crucial, as each data view contributes differently to the clustering task. For time-series data, the choice of similarity metrics significantly affects clustering performance.

The study by [12] examined the impact of various similarity measures on performance and concluded that no similarity measure is consistently more effective than others for multi-view clustering. For instance, correlation-based metrics such as Pearson and Spearman effectively capture temporal dependencies and rank-order relationships, while distance-based metrics such as Euclidean and Manhattan are more sensitive to noise but excel at identifying well-separated clusters. High dimensionality further complicates the clustering process, as larger datasets and multiple views require scalable methods. Additionally, irrelevant or noisy data from external sources can degrade clustering quality, leading to less meaningful insights.

1.2 Objectives and Contributions

This research aims to tackle these challenges by utilizing advanced multi-view clustering techniques that are be adapted to multi-view time series data. Through the integration of external data sources and the enhancement of clustering techniques, it aims to improve the accuracy, scalability, and interpretability of clustering results. These improvements will enable energy providers to understand consumption patterns better, predict demand, allocate resources efficiently, and implement targeted energy-saving strategies, addressing the growing need for effective solutions to handle large-scale energy data.

1.3 Thesis Structure

This thesis is organized into six chapters, each addressing a specific aspect of the multi-view clustering problem for smart meter time-series data.

Chapter 1 introduces the research context and motivation, highlighting the challenges of analyzing electricity consumption data and the importance of integrating external information such as weather data. It also outlines the specific objectives and methodological contributions of this work.

Chapter 2 reviews existing literature on multi-view clustering. It discusses theoretical foundations, representative methodological families, and recent scalable approaches, with a particular focus on their applicability to large-scale and heterogeneous datasets. The chapter concludes by identifying the research gap addressed in this thesis.

Chapter 3 formally defines the multi-view clustering problem considered in this work. It describes the key characteristics of smart meter data, including their temporal nature,

high dimensionality, and heterogeneity across views, and introduces the notation used throughout the thesis.

Chapter 4 presents the datasets used in the experiments and details the FastMICE framework adopted in this thesis. It explains the main components of the methodology and describes the modifications and extensions introduced to better handle multi-view time-series data, including time-aware sampling strategies and symbolic representations.

Chapter 5 describes the experimental setup and evaluation procedure. It reports quantitative and qualitative results, compares the proposed approach to baseline methods, and analyzes the impact of different views and scalability aspects.

Chapter 6 concludes the thesis by summarizing the main findings and contributions of the thesis and discusses potential directions for future research.

2 Related Work

In recent years, a wide range of multi-view clustering methods has been proposed to address the challenge of integrating information from multiple data sources or *views*. Despite their methodological differences, these approaches share the common objective of exploiting complementary information across views in order to obtain cluster assignments that are more accurate, more stable, and more informative than those obtained from any single view alone. A straightforward baseline consists in concatenating features from all available views and applying a conventional single-view clustering algorithm on the resulting representation. While simple, this strategy completely ignores the structure that is specific to each view, which may amplify the effects of heterogeneous feature scales across views and view-dependent noise. In smart meter analysis, consumption patterns show clear temporal variability and are often combined with contextual information such as weather data. In this setting, the limitations of direct feature concatenation become more apparent. A suitable multi-view approach should preserve the specific information provided by each data source while remaining robust to noise and scalable to large numbers of customers and long observation periods.

This chapter reviews representative lines of work in multi-view clustering, discusses their relevance for the problem considered in this thesis, and highlights the methodological gaps that motivate the adopted approach.

2.1 Theoretical Foundations and Taxonomies of Multi-View Clustering

Several foundational studies have contributed to the structuring of the multi-view clustering literature by introducing taxonomies that clarify the design space of existing methods. [13] provides a survey that categorizes multi-view clustering approaches into two broad families, namely generative and discriminative methods, as shown in Figure 2.1. Generative approaches, including mixture-model formulations [28] and EM-based estimation [15], aim to capture a probabilistic model of the data-generating process and can be relevant when the assumptions of the model match the empirical distribution of the views. In contrast, discriminative approaches directly optimize clustering-related objectives and cover a wide range of techniques, including shared indicator matrix formulations (e.g., multi-view NMF), shared coefficient matrix models (e.g., subspace-based methods [41]), shared eigenvector formulations (e.g., spectral clustering variants), and unsupervised deep learning approaches. While this survey does not propose new methods, it provides a conceptual map that helps relate different multi-view clustering approaches, clarify their assumptions, and anticipate practical trade-offs. For large-scale datasets, this perspective

is particularly relevant as it highlights that multi-view integration can be performed either at the level of latent representations or similarity graphs, and that each choice implies different requirements in terms of distance measures, stability, and computational cost.

[16] further refine the taxonomy by distinguishing heuristic-based methods, as shown in Figure 2.1, which are largely built upon classical machine learning techniques such as graph learning and matrix factorization, alongside neural network-based methods that rely on representation learning with architectures such as autoencoders. In particular, representation learning is appealing when raw views are high-dimensional or contain complex dependencies, since it may produce compact embeddings that facilitate the clustering process. At the same time, the applicability of deep multi-view clustering to large-scale smart meter datasets requires careful attention to scalability, training stability, and the risk of learning representations that are sensitive to variations or view imbalance. Moreover, many benchmark datasets used in the survey literature, such as Caltech7 and NUS-WIDE, focus on vision and multimedia tasks and do not reflect the temporal structure and measurement noise typically present in smart meter data. As a result, while these surveys provide a useful conceptual foundation, they also highlight that applying multi-view clustering to the energy domain requires reconsidering assumptions about data structure, temporal variability, and computational feasibility.

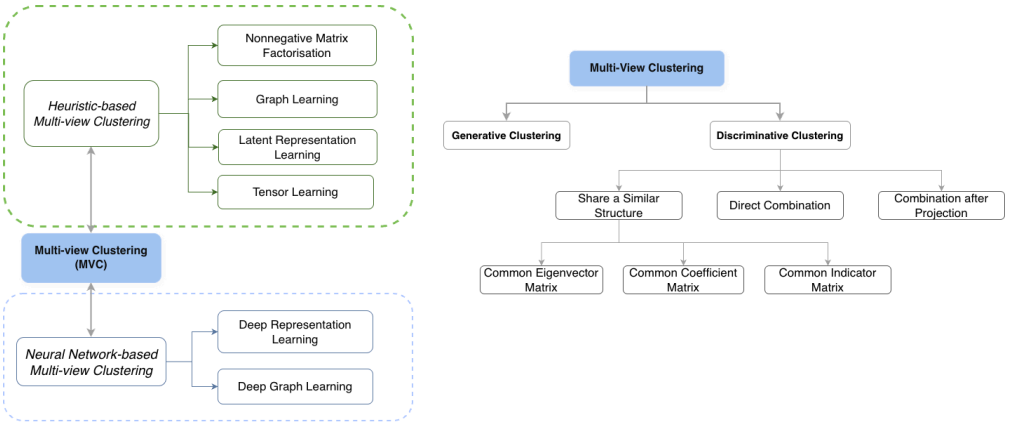


Figure 2.1: Taxonomies of multi-view clustering [13, 16].

2.2 Graph-Based, Spectral, and Matrix Factorization Approaches

One common approach in multi-view clustering uses graph constructions to represent relationships within each view and to combine them into a unified clustering solution. Graph-based methods are appealing to use because they can encode local neighborhood structure and capture non-linear manifolds, which is particularly relevant when view-specific measurements are not well separated in the original feature space. [17] introduced the Graph-based Multi-View Clustering (GMC) framework, which iteratively constructs

view-specific graphs and refines a unified graph representation. The method was evaluated on synthetic datasets and widely available multi-view datasets such as BBC, Handwritten Digit 2, and Mfeat. In principle, such a framework can be considered for energy datasets by defining similarity graphs for consumption data and contextual variables and then fusing them into a consistent structure. However, the effectiveness of graph-based approaches depends strongly on the quality of the underlying similarity graphs. Smart meter data are often noisy and may exhibit customer-specific shifts, seasonal effects, and irregularities, which can lead to unstable neighborhoods under simple similarity measures. In addition, graph construction and iterative refinement can become computationally demanding at large scale, especially when the number of customers is high and when temporal sequences are represented in high-dimensional spaces.

To reduce the reliance on manual hyperparameter tuning and improve robustness, [18] proposed a parameter-free auto-weighted multiple graph learning framework for multi-view clustering and semi-supervised classification. They eliminated the need for manual hyperparameter tuning by automatically learning optimal graph weights through an iterative optimization process. Building on this, [19] further developed the idea with a self-weighted multi-view clustering method based on Laplacian rank constraints, encouraging the method to prioritize high-quality views while down-weighting less informative ones. These approaches are conceptually appealing for heterogeneous multi-view settings, since they explicitly acknowledge that not all views contribute equally to clustering quality. However, these methods were evaluated primarily on image-based benchmarks such as Caltech101-7 and MSRCv1, and their performance on multi-view time-series datasets was not investigated. Furthermore, although parameter-free weighting reduces manual intervention, the iterative nature of multiple-graph learning can still present computational challenges when scaling to large datasets.

Recently, [20] proposed a multi-graph fusion framework that integrates graph learning, graph fusion, and spectral clustering in a unified procedure. A central contribution of this method lies in its dynamic graph fusion strategy, which assigns weights to individual graphs based on their proximity to a consensus graph. It explicitly incorporates clustering structure by ensuring the fused graph has the exact number of connected components corresponding to the clusters. Tested on datasets like BBC, Reuters, and Caltech20, this method demonstrated robustness against noisy views. In the context of energy datasets, this approach could be adapted to integrate graphs representing smart meter data and weather patterns, potentially improving clustering accuracy. However, the iterative updates required for graph learning may present scalability challenges when applied to large datasets.

Multi-view spectral clustering also constitutes a widely used baseline family due to its strong empirical performance over single-view spectral clustering. In practice, multi-view spectral clustering variants differ in how they combine view-wise affinities or Laplacians, and they provide a useful reference point when evaluating more specialized multi-view methods. In this thesis, two spectral baselines are considered via the `mvlearn` Python library [37]: Multi-View Spectral Clustering, which aggregates information across view-specific similarity structures to derive a common embedding, and Multi-View Co-

Regularized Spectral Clustering, which encourages agreement between view-specific spectral embeddings through co-regularization. These methods can offer an explicit mechanism for aligning views while preserving spectral structure. However, their computational profile is closely tied to spectral decomposition, which can become a bottleneck for large-scale datasets. In addition, as with other graph-based techniques, performance is sensitive to the construction of view-wise similarities, an aspect that is particularly delicate for time-series, where naive distances can be dominated by temporal misalignment or scale variations.

Matrix factorization and subspace-based approaches provide an alternative perspective by seeking low-dimensional latent structures shared across views. [21] introduced the Consistent and Specific Multi-View Clustering (CSMVC) method, which combines non-negative matrix factorization with subspace learning to capture both global shared structure and view-specific characteristics. The method further incorporates graph regularization to improve robustness to noise. For smart meter data, CSMVC offers a promising framework for combining weather and time-series views while addressing challenges like heterogeneity. The study uses 13 benchmark datasets covering diverse domains, including physical activities (KP, CMU, Auslan), handwriting (CT) and robot execution failures (LP1, LP4, LP5), to show the significantly improved clustering performance of the proposed method against four state-of-the-art methods. However, CSMVC remains sensitive to parameter choices, and the need for careful tuning can be limiting in practical settings where large-scale experiments and deployment require stable runtime defaults.

2.3 Deep Multi-View Clustering and Representation Learning Approaches

Deep multi-view clustering methods have received increasing attention because of their ability to learn view-specific and shared representations jointly with clustering objectives. These methods are particularly useful when the input views are high-dimensional or when the relationships between them are complex.

In this thesis, two deep multi-view clustering baselines are included because they represent recent advances that address view imbalance and robustness in modern settings. [39] proposed Deep Safe Multi-View Clustering (DSMVC), which is motivated by the observation that incorporating additional views may sometimes degrade clustering performance when certain views are noisy or misaligned. DSMVC introduces mechanisms that limit the influence of unreliable views by controlling how information is integrated across sources. This idea is relevant for energy datasets, where contextual data such as weather variables may vary in quality and reliability. Nevertheless, deep models typically introduce additional training complexity and sensitivity to architecture and optimization settings. In large-scale smart meter settings, these practical aspects become critical. Model training must remain stable and computationally feasible, and the learned embeddings should remain consistent across different customer groups or subsets.

More recently, [38] introduced Self-weighted Contrastive Fusion (SCMVC) for deep multi-view clustering, using contrastive objectives and self-weighting strategies to enhance

fusion quality and reduce the influence of less informative views. Contrastive learning has the advantage of shaping representations based on agreement and disagreement across views, which can improve robustness when views provide complementary signals. At the same time, the application of such methods to smart meter time-series raises two practical considerations. First, temporal data typically requires either specialized encoders or careful preprocessing to ensure that learned embeddings capture relevant temporal characteristics rather than superficial correlations. Second, contrastive learning can be computationally demanding because it requires many pairwise comparisons. Without careful implementation, this can limit scalability. For these reasons, deep baselines serve an important role in experimental evaluation, but their direct adoption in large-scale energy analytics must account for the additional complexity introduced by training dynamics and representation choices.

2.4 Scalable and Efficient Multi-View Clustering for Large Datasets

Although graph-based, spectral-based, and deep learning clustering methods offer strong modeling capabilities, they often struggle to scale to large datasets due to dense similarity matrices or computationally expensive optimization procedures. To overcome this limitation, a substantial body of work has focused on scalable multi-view clustering methods, often by approximating full graphs through anchor points or bipartite structures. [22] proposed a large-scale multi-view spectral clustering approach based on bipartite graphs, reducing computational and memory requirements by avoiding the construction of full pairwise similarity matrices. This strategy is particularly relevant to smart meter datasets where the number of customers can be large, and where direct spectral clustering on dense graphs becomes infeasible. Nevertheless, the quality of the approximation depends on the selection of anchor points and on the similarity measure used to link samples to anchors, which again becomes non-trivial when temporal sequences must be represented in a way that preserves meaningful similarities between data points.

[14] introduced an anchor-based strategy for large-scale multi-view subspace clustering with linear time complexity, demonstrating that carefully designed anchor constructions can make multi-view methods tractable at scale. [23] further developed this direction and improved scalability by automatically learning optimal consensus anchor guidance in a parameter-free framework, tested on large datasets with more than 30,000 samples (including NUSWIDEIBJ, AWA, MNIST, and YoutubeFace). [24] proposed unified anchors for scalable multi-view subspace clustering with dynamic fusion of complementary views. Together, these methods show that anchor-based approximations can make multi-view clustering computationally feasible at scale. In the energy domain, they suggest that large-scale integration of consumption and contextual views can be achieved if temporal data are transformed into compact representations that preserve meaningful information. However, without a time-series aware representation, anchor construction may be driven by superficial similarity caused by noise or scale, thereby weakening the stability of the resulting clusters.

Among scalable approaches, [25] introduced the FastMICE framework, which combines random view grouping, bipartite graph construction, and a hybrid early-late fusion strategy to achieve scalable and robust multi-view clustering without the burden of hyperparameter tuning. The framework is evaluated on 22 datasets, including CIFAR-10, VGGFace2-50, and NUS-WIDE, where it shows strong empirical performance and good scalability. At the same time, its evaluation largely focuses on conventional multi-view benchmarks, and the framework does not explicitly target multi-view time-series data. For smart meter analysis, this distinction is important because time series introduce additional structure and variability that may be missed by standard similarity measures. Nonetheless, FastMICE provides a practical foundation for large-scale multi-view clustering in energy datasets. Its success, however, depends on encoding temporal characteristics in a way that preserves meaningful similarities while remaining computationally efficient.

Summary: The literature reviewed in this chapter shows that multi-view clustering has developed into a broad and methodologically rich field. Graph-based and spectral methods provide principled approaches for integrating multiple views and capturing structural relationships in the data. However, they rely on similarity graphs and spectral decompositions that are often expensive to construct and difficult to scale, particularly for time-series data. Matrix factorization and subspace methods learn structured latent representations across views, but they typically require careful parameter tuning and remain sensitive to how temporal sequences are represented.

Deep multi-view clustering methods increase representational flexibility and introduce mechanisms to reduce the influence of noisy views. At the same time, they increase computational cost and add sensitivity to architectural and optimization choices. In large-scale smart meter settings, these limitations are amplified by the combination of high-dimensional temporal measurements, heterogeneous contextual views, and the requirement for scalable and stable algorithms. Scalable anchor- and bipartite-based approaches address many computational limitations and provide promising directions for large datasets. However, their effectiveness ultimately depends on representations that properly capture temporal similarity.

Despite substantial progress in multi-view clustering, there remains a gap in practical and scalable frameworks designed specifically for multi-view time-series data. This thesis addresses this gap by focusing on methods that preserve temporal structure while remaining robust and computationally efficient at scale.

3 Problem Definition

The rapid expansion of sensor-based technologies has resulted in the widespread availability of large-scale, high-frequency, and multi-view data, particularly in the energy sector. One of the most prominent examples is the deployment of smart meters, which have transformed traditional electricity consumption monitoring into a continuous and data-intensive process. These devices automatically measure electricity usage and transmit detailed consumption data to utility providers at fine-grained intervals, ranging from daily readings to hourly or minute-level measurements.

The availability of such detailed temporal data creates opportunities to identify consumption patterns, support demand-response strategies, and improve energy planning. However, it also introduces significant computational challenges. To make effective use of smart meter data, scalable unsupervised learning frameworks are required. These methods must cluster consumers based on multi-view time-series data while accounting for the diversity and variability of real-world data.

This chapter formally defines the problem of clustering smart meter data from a multi-view perspective, grounding the motivation in the characteristics of the data and the technical challenges they present.

3.1 Characteristics of Smart Meter Data

Smart meter datasets possess several distinct characteristics that make them analytically valuable but also computationally challenging. These characteristics are central to the formulation of this thesis and have a direct impact on the selection of algorithms and design choices.

- **High Dimensionality:** Smart meters record electricity consumption at high temporal resolution. A single time series can consist of hundreds or thousands of readings, depending on the length and frequency of measurement. For instance, daily recordings across an entire year yield 365 data points per customer, while hourly measurements yield over 8,700. When combined across thousands of households, the resulting dataset becomes extremely high-dimensional, creating both memory and processing bottlenecks for traditional clustering methods.

- **Temporal Complexity:** Unlike simple tabular records, electricity usage profiles are inherently temporal. Consumption patterns may exhibit daily, weekly, and seasonal variations, as well as irregular spikes due to user behavior or external factors such as holidays and weather. Capturing these patterns requires clustering algorithms to move beyond Euclidean distance and adopt temporal distance measures (e.g.,

Dynamic Time Warping [30]), or dimensionality reduction techniques (e.g., SAX or iSAX [29]) that can approximate the structure of time-series data.

- **Heterogeneity Across Views:** In real-world applications, electricity consumption is rarely analyzed in isolation. A more comprehensive understanding of user behavior can be obtained by integrating multiple data views, such as geolocation, building metadata (e.g., household size or appliance ownership), and weather information. Each view provides complementary information but may differ in scale, format, and semantic meaning. This heterogeneity requires clustering models that can integrate diverse modalities while preserving their individual contributions.

- **Non-Stationarity and Concept Drift:** Consumption behavior may evolve over time, either due to changes in user routines or external interventions (e.g., energy-saving policies or tariff shifts). This non-stationarity adds further complexity to the clustering process, as static assumptions about feature distributions may no longer hold. Adaptive or ensemble-based approaches may be required to account for these evolving patterns.

- **Large Scale and Real-Time Generation:** Smart meters are deployed at scale across entire cities or regions, and they produce data continuously. This means that clustering algorithms must not only be accurate and interpretable, but also scalable and efficient, capable of handling millions of time series without requiring exhaustive pairwise distance computations or extensive hyperparameter tuning.

  These characteristics define the core problem addressed in this thesis:

  How can high-dimensional, multi-view time-series data from smart meters be clustered efficiently and robustly, in a way that scales to millions of consumers, handles heterogeneity across views, and accounts for the temporal structure of the data?

3.2 Notations

Let $X = \{x_1, x_2, \ldots, x_N\}$ be a dataset with N data samples, where x_i is the i-th sample. For a multi-view dataset, each data sample can be represented by features from different views. Thus, the multi-view dataset is denoted as $X = \{X^1, X^2, \ldots, X^V\}$, where $X^v \in \mathbb{R}^{N \times d_v}$ is the data matrix of the v-th view, V is the total number of views, and d_v is the dimension of the v-th view.

For convenience of view group formation in later sections, we define the set of V views as $\mathcal{V} = \{\text{View}_1, \text{View}_2, \ldots, \text{View}_V\}$, where View_v corresponds to the v-th view. Note again that X^v is the data matrix associated with View_v.

The goal of multi-view clustering is to partition the dataset X into k disjoint clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ such that:

- Each cluster groups households with similar electricity consumption behavior, taking into account multiple data views.

- The clustering model can handle view-specific characteristics such as differing scales, temporal dependencies, and noise.
- The solution scales to large datasets, avoiding computational bottlenecks such as pairwise time-series distance calculations.

Formally, we aim to learn a clustering function $\mathcal{F}$ that maps the multi-view dataset into k clusters:

$$\mathcal{F} : \left(X^1, X^2, \ldots, X^V\right) \to \mathcal{C}$$

Below, we define the main notations used throughout the FastMICE [25] framework:

N	Total number of data samples.
V	Total number of views.
$X^v \in \mathbb{R}^{N \times d_v}$	Data matrix of the v-th view.
d_v	Feature dimensionality of the v-th view.
M	Number of random view groups / base clusterings (ensemble size).
$\mathsf{VG}^{(m)}$	The m-th randomly selected view group.
$V^{(m)}$	Number of views in view group $\mathrm{VG}^{(m)}$.
$X_v^{(m)}$	Sampled data matrix of view v within group m, possibly after feature subsampling.
$\tau_v^{(m)}$	Feature sampling ratio applied to view v in group m.
p	Total number of anchors used per view group.
$\bar{p}^{(m)} = \left\lceil \frac{p}{V^{(m)}} \right\rceil$	Number of anchors assigned to each view in group m.
K	Number of nearest neighbors (anchors) used when constructing bipartite affinity graphs.
$\bar{K}^{(m)} = \left\lceil \frac{K}{V^{(m)}} \right\rceil$	Number of neighbors used per view in group m.
$A_v^{(m)}$	Set of anchors selected for view v in group m.
$B_v^{(m)} \in \mathbb{R}^{N \times \bar{p}^{(m)}}$	Sparse affinity matrix linking data samples to anchors in view v of group m.
$B^{(m)}$	Combined cross-affinity matrix obtained by concatenating $\{B_v^{(m)}\}$ across all views in $\mathrm{VG}^{(m)}$.
$G^{(m)}$	View-sharing bipartite graph constructed from $B^{(m)}$.

$k^{(m)}$	Number of clusters for the m-th base clustering, randomly selected from the range $[k, 2k]$.
$\pi^{(m)}$	Base clustering result obtained from spectral partitioning of $G^{(m)}$.
Π	The set of all base clusterings: $\Pi = \{\pi^{(1)}, \ldots, \pi^{(M)}\}$.
C_j	A cluster obtained from one of the base clusterings.
$k_c = \sum_{m=1}^{M} k^{(m)}$	Total number of clusters generated across all base clusterings.
$B \in \mathbb{R}^{N \times k_c}$	Binary matrix linking each data sample to the base clusters it belongs to.
G	Unified bipartite graph constructed between data samples and cluster nodes based on B.
$\mathcal{C}$	Final clustering result obtained from spectral clustering on G into k clusters.

This notation provides a foundation for understanding the subsequent methodological components, including view group formation, bipartite graph construction, ensemble generation, and final consensus clustering.

4 Methodology and Data

This chapter presents the methodological foundations and data preparation procedures underlying the proposed multi-view clustering framework. It first describes the datasets used throughout the thesis, including the real-world smart meter data and several benchmark datasets used for quantitative evaluation. Particular emphasis is placed on the construction of consistent multi-view representations and the preprocessing steps required to ensure temporal alignment and data integrity. The chapter then introduces the FastMICE framework as the core clustering methodology, detailing its ensemble-based design and graph-based fusion strategy. Finally, the modifications and extensions developed in this work are presented, highlighting how the original framework is adapted to handle large-scale multi-view time-series data efficiently and effectively.

4.1 Dataset Descriptions

This research is grounded in real-world, high-resolution electricity consumption data collected through smart metering infrastructure and enriched with external weather information. The primary objective is to construct a clean, consistent, and reliable multi-view dataset that captures both household-level electricity usage patterns and the surrounding environmental context over a full annual cycle.

Since the smart meter dataset does not provide ground truth labels, quantitative validation of the proposed multi-view clustering framework cannot be performed directly on this data. Instead, validation experiments are conducted using additional labeled datasets, namely the CMAPSS jet engine simulation dataset, the Human Activity Recognition Using Smartphones dataset, and synthetically generated time Series in Risk Assessment. These datasets allow quantitative evaluation while the smart meter data serves as the main real-world application scenario.

Smart Meter Dataset: The core electricity consumption data originates from a large-scale smart metering deployment, comprising several related datasets that differ in temporal resolution and coverage:

- **smartmeter_daily:** This dataset contains daily electricity consumption values for 304,940 unique household identifiers. Each row corresponds to a single household, while each column represents electricity usage on a specific day of the year.
- **smartmeter_daily_fullyear:** A filtered subset of the daily dataset, consisting of 209,301 households with complete observations for an entire year (365 non-missing

daily measurements). Due to its completeness and consistency, this dataset is selected as the primary input for the clustering experiments.

- **smartmeter_ims:** This dataset includes daily electricity readings obtained from a simplified smart meter configuration with a reduced set of sampling features.
- **smartmeter_ime:** A high-resolution dataset capturing electricity consumption at 15-minute intervals for 2,147 households. Although this configuration enables fine-grained temporal analysis, it is not used in the core multi-view clustering experiments due to its limited population coverage.

To complement the consumption data with environmental context, two weather-related datasets are incorporated:

- **days_temperature:** Provides daily weather statistics for each customer, including minimum, average, and maximum temperatures, as well as total and peak solar radiation values.
- **std_temperature:** Contains hourly temperature and solar radiation measurements. While this dataset offers higher temporal granularity, the daily-aggregated version is primarily used to maintain consistency with the daily electricity consumption data.

All weather datasets are aligned with the smart meter data using customer identifiers, enabling the construction of contextual views that capture external factors influencing household electricity consumption behavior.

In order to use the dataset in the experiments, several data cleaning steps were applied to ensure the integrity and reliability of the final datasets.

- **Completeness Filtering:** Only households with full-year data (365 days) were retained, using the `smartmeter_daily_fullyear` subset. This ensures temporal consistency and avoids issues arising from imputation.
- **Cross-View Matching:** Smart meter records were matched with corresponding weather data based on timestamps and customers IDs. Entries with unmatched or ambiguous identifiers were removed to ensure view consistency.
- **Data Type Conversion:** All time-related variables were converted to numerical indices or timestamps for efficient indexing and alignment between views.

Following preprocessing, the multi-view representation required for the FastMICE framework was constructed. In the experiments, three complementary views were defined to capture both consumption behavior and environmental context. The first view corresponds to daily electricity consumption per household, represented as a univariate time series over 365 days. The second view captures the maximum daily solar radiation, reflecting the influence of sunlight on energy usage patterns. The third view consists of the average daily temperature.

Each view is represented as a matrix where rows correspond to households and columns represent daily time indices. All views share a common temporal structure and are perfectly aligned across households. These preprocessing steps resulted in a clean, temporally aligned, and multi-view-ready dataset.

Jet Engine Simulated Data (CMAPSS): The CMAPSS dataset [33] consists of multiple multivariate time series generated from simulated aircraft engine run-to-failure experiments. Each time series corresponds to a different engine, and the dataset can be interpreted as observations collected from a fleet of engines of the same type.

Each engine begins operation with an unknown degree of initial wear and manufacturing variability, which is considered normal behavior rather than a fault condition. During operation, the engine gradually develops a fault that grows in severity over time. In the training subset, each engine is observed until system failure, whereas in the test subset, the time series terminates prior to failure. A vector of ground truth Remaining Useful Life (RUL) values is provided for the test data, indicating the number of operational cycles remaining after the final recorded observation.

The dataset includes three operational settings that affect engine performance, along with multiple sensor measurements. Sensor readings are contaminated with noise to reflect realistic operating conditions. Each data file is provided as a space-separated text file with 26 columns, where each row represents a single operational cycle. The columns correspond to the unit identifier, cycle index, three operational settings, and twenty-one sensor measurements.

The CMAPSS dataset is organized into four subsets with varying operating conditions and fault modes:

- **FD001:** 100 training trajectories and 100 test trajectories, operating under a single condition (sea level) with one fault mode (HPC degradation).
- **FD002:** 260 training trajectories and 259 test trajectories, operating under six conditions with one fault mode (HPC degradation).
- **FD003:** 100 training trajectories and 100 test trajectories, operating under a single condition with two fault modes (HPC degradation and fan degradation).
- **FD004:** 248 training trajectories and 249 test trajectories, operating under six conditions with two fault modes (HPC degradation and fan degradation).

Both subsets (FD001 and FD003) are used to validate the extended FastMICE framework. Since the data are inherently multivariate, the construction of multiple views is performed by treating each sensor measurement as an independent view. Consequently, sensor variants capturing different physical quantities—such as temperature, pressure, and rotational speed- are separated into distinct views. This results in a total of 20 views for each subset, with each view represented as a matrix containing the temporal evolution of a single sensor across all engines.

Before constructing the multiple views, a preprocessing step is applied to ensure consistency across time series. As engines exhibit different time cycles, the resulting trajectories vary in length. To obtain view matrices with uniform temporal dimensions, shorter time series are padded with zeros at missing timestamps. This padding allows all trajectories to be aligned to a common length while preserving the original structure. These preprocessing steps ensure a clean and aligned multi-view-ready dataset.

Each view is represented as a matrix where rows correspond to detectors and columns correspond to consecutive 5-minute time intervals. Since the dataset is complete and uniformly sampled, no additional preprocessing steps such as completeness filtering or temporal alignment are required. All dataset subsets provides ground-truth Remaining Useful Life (RUL) values that quantify the number of cycles remaining until failure for each engine. Since RUL is a continuous variable rather than a categorical class indicator, it cannot be directly interpreted as true cluster labels. To enable clustering-based evaluation, the RUL values are therefore transformed into discrete categories by partitioning their range into a predefined number of equally spaced intervals, corresponding to the assumed number of clusters. Each engine is assigned to a category based on the interval into which its RUL value falls.

The optimal number of partitions is determined empirically using an elbow method, which is applied independently to the FD001 and FD003 subsets. The results of this analysis are illustrated in Figures 5.2 and 5.3. It is important to note that these discretized RUL categories do not represent true underlying clusters in the data but rather serve as pseudo labels that approximate different stages of engine degradation. These pseudo labels are used solely as a reference for quantitative evaluation of clustering performance and should not be interpreted as definitive ground truth.

Human Activity Recognition Using Smartphones (HAR): The Human Activity Recognition Using Smartphones dataset [34] is a real-world multivariate time-series dataset collected from experiments involving 30 volunteers performing daily physical activities while carrying a smartphone positioned at the waist. The participants, aged between 19 and 48 years, executed six predefined activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying) while the embedded accelerometer and gyroscope of a Samsung Galaxy S II recorded tri-axial linear acceleration and angular velocity signals at a constant sampling rate of 50 Hz. The experiments were video-recorded to enable accurate manual annotation of activity labels.

Raw sensor signals were preprocessed by applying noise filtering and were segmented into fixed-length sliding windows of 2.56 seconds with 50% overlap, resulting in windows of 128 consecutive readings per axis. The acceleration signal was further decomposed into body acceleration and gravitational components using a Butterworth low-pass filter with a cutoff frequency of 0.3 Hz. From each window, a set of handcrafted features was extracted in both the time and frequency domains, yielding a compact statistical representation of human motion. The dataset was partitioned at the subject level, with 70% of the participants assigned to the training set and the remaining 30% to the test set.

In this study, each segmented time window is treated as an entity and is associated

with a categorical ground-truth label corresponding to the performed activity. These labels, encoded as integers from 1 to 6, represent distinct human activities and are used for clustering evaluation. No additional preprocessing or transformation is applied to the labels.

To construct a multi-view representation suitable for clustering, multiple complementary views are derived from the available sensor data. One view is formed by concatenating the tri-axial body acceleration signals over each window, resulting in a high-dimensional representation capturing fine-grained motion dynamics. A second view is constructed analogously from the tri-axial body gyroscope signals, emphasizing rotational motion patterns. A third view represents the total acceleration signal across all three axes, capturing combined body and gravitational effects. Finally, a fourth view consists of the precomputed statistical feature vectors, which summarize the underlying time-series behavior using time- and frequency-domain descriptors. Each view provides a distinct representation of the same set of entities, and the training and test subsets are concatenated to form a unified dataset. This results in a clean, well-aligned multi-view-ready dataset.

Synthetic Time Series in Risk Assessment (SCRM): The Supply Chain Risk Management dataset used in this thesis is a synthetically generated multivariate time-series dataset obtained from a probabilistic risk assessment framework coupled with Linear Programming optimization models implemented in MATLAB Simulink [35]. The simulation models the dynamic behavior of a multi-supplier supply chain network, where supplier-level risk indices and the total supply chain cost are evaluated over time under varying operational conditions.

The original data are generated at a temporal resolution of 5 minutes and are then aggregated to 30-minute intervals to reduce dimensionality and smooth high-frequency fluctuations. Feature variables are aggregated using the mean within each interval, while the categorical supply chain stability label, denoted as `SCMstability_category`, is aggregated using the mode to preserve the dominant system state. Time windows containing missing values are discarded, resulting in a uniformly sampled time series.

Each aggregated time window is treated as an entity and is described by a set of continuous features, including supplier-specific risk indicators and global cost-related variables. The associated stability label represents the overall operational condition of the supply chain at the corresponding time and serves as the ground truth for clustering evaluation. No additional transformations are applied to the labels beyond temporal aggregation.

To construct a multi-view representation, multiple complementary views are derived from the aggregated data. The first view consists of normalized raw feature values and captures the instantaneous operational state of the supply chain. The second view models short-term temporal dynamics by extracting statistical descriptors, including the mean, standard deviation, minimum, and maximum, over rolling time windows. The third view isolates supplier-level risk metrics, providing a focused representation of risk propagation across individual suppliers, while the fourth view captures cost dynamics through statistical descriptors such as the mean and standard deviation. All numerical

views are standardized by removing the mean and scaling to unit variance, to ensure consistency of scale across representations. This construction yields a clean, aligned four-view dataset suitable for multi-view clustering.

Amsterdam Library of Object Images (ALOI): The Amsterdam Library of Object Images (ALOI) is a large collection of color images of everyday objects captured under systematically varied imaging conditions. Originally introduced by Geusebroek et al [36]. to study object appearance under changes in viewing angle, illumination angle, and illumination color, the ALOI dataset contains images of 1,000 distinct small objects, each recorded from multiple viewpoints and lighting configurations to capture a wide range of sensory variation in object appearance. The full dataset comprises more than 110,000 images, making it a widely used benchmark for object recognition, retrieval, and representation learning in computer vision research.

In this thesis, a curated multi-view variant of ALOI, which was present in the original FastMICE MATLAB repository and has been used in prior multi-view clustering research, is employed to validate the Python implementation of the FastMICE framework. In this variant, each object is represented not as raw images but as a set of precomputed feature vectors that encode complementary aspects of visual content. These features are organized into four distinct views, each with a specific interpretative focus.

The first view, referred to as the Similarity view, consists of a 77-dimensional representation capturing similarity measures to a set of reference colors or prototypes; this view emphasizes global color-based relationships between objects. The second view is based on Haralick texture features, providing a 13-dimensional description of local texture patterns and co-occurrence statistics that reflect surface and structural properties. The third view encodes HSV color histogram features with 64 dimensions, representing hue, saturation, and value distributions that are robust to certain photometric changes. The fourth view consists of 125-dimensional RGB histogram features, capturing fine-grained color distributions in the red, green, and blue channels. Together, these views form a multi-faceted representation of each object, with all views aligned across the same set of 10,800 instances. The differing dimensions of the views—77, 13, 64, and 125 reflect the distinct feature extraction strategies applied to the underlying image data.

Table 4.1: Overview of relevant statistics of the used datasets

Statistic	**Smart Meter**	**FD001**	**FD003**	**SCRM**	**HAR**	**ALOI**
Number of samples	194,849	100	100	43,364	7,352	10,800
Number of clusters	6	6	5	5	6	100
Number of views	3	20	20	4	4	4

4.2 The FastMICE Framework

In this section, I present the methodological foundation behind my approach to scalable and robust multi-view clustering. As the dimensionality and complexity continue to grow, especially in applications involving time series from multiple heterogeneous sources, traditional clustering algorithms often face limitations in scalability, effective information integration, and adaptability to diverse view structures. To address these challenges, I adopt and extend FastMICE, an advanced multi-view clustering via ensembles framework [25] that combines the strengths of ensemble learning and hybrid fusion strategies. The key idea is to leverage the complementary information contained in different views by organizing them into diverse combinations, while ensuring computational efficiency through representative data abstractions and graph-based partitioning techniques.

This methodology section first outlines the conceptual foundations and design rationale behind FastMICE, then systematically describes its main components. Finally, I highlight the specific modifications and extensions I have incorporated into the original framework, including additional preprocessing steps to assess the impact of different data preparation methods on clustering performance and to enhance its applicability for large-scale multi-view time series datasets.

In the following subsections, I will explain each core component of the FastMICE framework while using the same notations as the original paper [25], starting with the Early-stage view group formation, followed by the construction of view-sharing bipartite graphs, base clusterings generation, and the final consensus process that produces the final clusters.

4.2.1 Early-Stage View Group Formation

An essential goal of multi-view clustering (MVC) is to effectively combine information from multiple views in order to achieve robust and accurate clustering results. While many existing MVC algorithms differ in how and when they fuse this information, they typically share two common traits. First, they usually perform the fusion process in a single stage, either early or late. Second, they tend to follow either the *single pattern*, where they process each view entirely on its own or the *all pattern*, where they combine all the views in one step.

Unlike the different MVC methods that rely on single or all patterns, the FastMICE framework introduces the concept of *random view groups*, which allows for more flexible and diversified combinations of views. Each view group contains a random subset of views and serves as the basic unit for view-wise diversification and the hybrid early-late fusion strategy.

Formally, let $\mathcal{V} = \{\text{View}_1, \text{View}_2, \ldots, \text{View}_V\}$ denote the set of V available views. A single view group $VG^{(m)}$ is defined as a random subset of views:

$$VG^{(m)} = \{\text{View}_1^{(m)}, \text{View}_2^{(m)}, \ldots, \text{View}_{V^{(m)}}^{(m)}\},$$

Each view member $\text{View}_v^{(m)}$ in a view group $VG^{(m)}$ is associated with a data matrix

$X_v^{(m)} \in \mathbb{R}^{N \times d_v^{(m)}}$, where N is the number of samples and $d_v^{(m)}$ is the dimensionality of that view.

where $V^{(m)}$ denotes the number of views selected in the m-th view group and satisfies $1 \leq V^{(m)} \leq V$. By repeating this random selection process M times, a set of M random view groups is generated:

$$\mathcal{VG} = \{VG^{(1)}, VG^{(2)}, \ldots, VG^{(M)}\}.$$

This random grouping strategy effectively breaks the single-or-all limitation, allowing the clustering process to capture more versatile view-wise relationships and to benefit from the complementary and diverse information across different views.

Each view group will subsequently be used to construct a view-sharing bipartite graph, enabling the hybrid early-stage fusion within each group. The results from all groups will later be integrated in the late-stage consensus step, as described in subsequent sections.

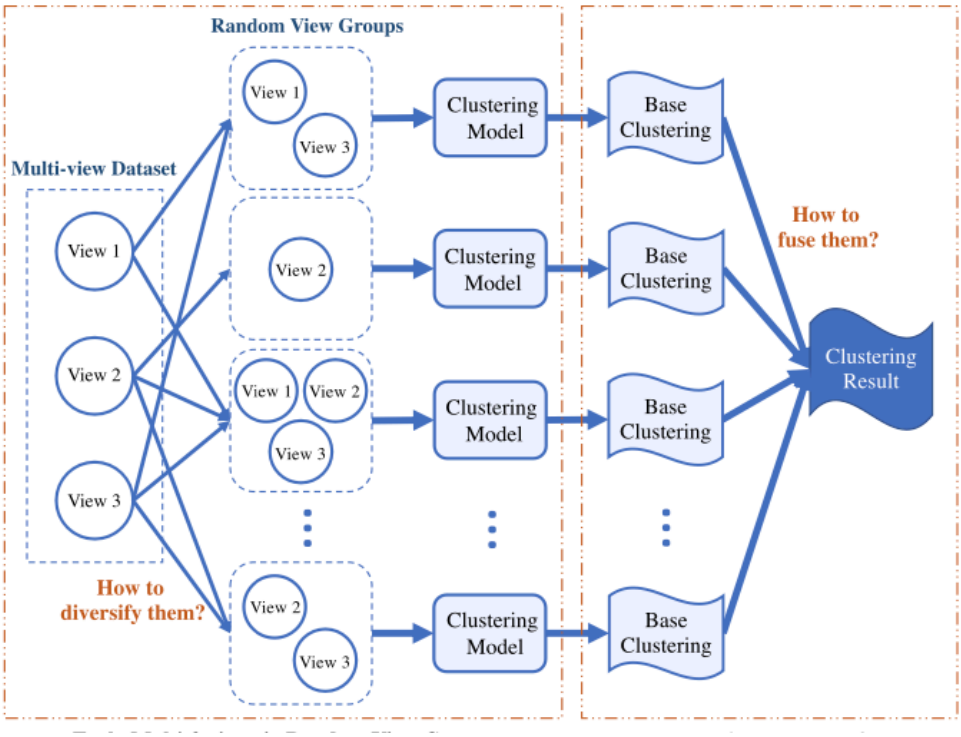


Figure 4.1: Hybrid early-late fusion strategy [25]

4.2.2 View-Sharing Bipartite Graph Construction

Once the view group formation is complete, I obtain M view groups, each containing different combinations of views. The next goal is to perform the early-stage fusion within these view groups. It is important to note that the early-stage fusion does not aim to

find a single optimal solution in one step; rather, it performs fusion across multiple view groups by constructing multiple view-sharing bipartite graphs that incorporate diversity at three levels: feature-level, anchor-level, and neighborhood-level.

In particular, for each view group, I use the bipartite graph structure to organize the information from its view members in a way that is both efficient and flexible. A bipartite graph is a special type of graph consisting of two disjoint sets of nodes: one set L of N data samples and one set R of p anchor points (also called representatives). The edges represent the similarity relationships between data samples and anchors. Formally, the adjacency matrix of a bipartite graph can be written as:

$$G = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}, \quad L = \{1, 2, \ldots, N\}, \quad R = \{1, 2, \ldots, p\}, \quad B \in \mathbb{R}^{N \times p}.$$

Here, B is the cross-affinity matrix that encodes the pairwise similarities between the N data points and the p anchors.

To introduce and maintain diversity in the bipartite graph construction, I apply the following three levels of diversification: feature-level, anchor-level, and neighborhood-level.

Feature-Level Diversification

At the feature level, I perform random feature sampling on each view member within a view group. Formally, let $\tau_v^{(m)}$ denote the sampling ratio for the v-th view member in the m-th view group. To introduce randomness and encourage diversity, the sampling ratio for each view member is randomly chosen within a predefined range:

$$\tau_v^{(m)} \in [\tau_{\min}^{(m)}, \tau_{\max}^{(m)}], \quad 0 < \tau_{\min}^{(m)} \leq \tau_{\max}^{(m)} \leq 1.$$

After random feature sampling, I obtain a subset of features for each view member. For the v-th view member $\text{View}_v^{(m)}$, its data matrix after feature sampling can be denoted as:

$$\widetilde{X}_v^{(m)} \in \mathbb{R}^{N \times \widetilde{d}_v^{(m)}}, \quad \text{where} \quad \widetilde{d}_v^{(m)} = \lceil \tau_v^{(m)} \cdot d_v^{(m)} \rceil.$$

This process ensures that each view member contributes differently across different view groups, enhancing the model's ability to capture varied patterns and local structures.

Anchor-Level Diversification

After introducing the feature-level diversification, the next step is to construct a bipartite sub-graph for each view member within a view group. These bipartite sub-graphs are then combined to form a unified view-sharing bipartite graph for each view group. To achieve this, a set of p anchor points is required, along with neighborhood connections based on K nearest neighbors (K-NN).

Unlike many existing MVC algorithms that select anchors for all views jointly, often by applying k-means clustering on the concatenated features of all views, the FastMICE framework distributes the anchor selection task among the multiple view members in

each view group. Specifically, each view member is expected to contribute equally to the total set of anchors. Formally, the expected number of anchors for each view member is

$$\bar{p}^{(m)} = \left\lceil \frac{p}{V^{(m)}} \right\rceil,$$

Similarly, the expected number of nearest neighbors contributed by each view member is

$$\bar{K}^{(m)} = \left\lceil \frac{K}{V^{(m)}} \right\rceil.$$

Accordingly, for each view member in $\mathrm{VG}^{(m)}$, The aim is to construct a bipartite sub-graph between the N data samples and $\bar{p}^{(m)}$ anchors, with each sample connected to its $\bar{K}^{(m)}$ nearest anchors.

For the m-th view group $\mathrm{VG}^{(m)}$, which is associated with the feature-sampled data matrix $\widetilde{X}_v^{(m)}$, the anchor selection for the v-th view member $\mathrm{View}_v^{(m)}$ is performed using a hybrid representative selection strategy [31], which efficiently identifies a set of $\bar{p}^{(m)}$ anchors:

$$A_v^{(m)} = \{a_{v,1}^{(m)}, a_{v,2}^{(m)}, \ldots, a_{v,\bar{p}^{(m)}}^{(m)}\},$$

where $a_{v,i}^{(m)}$ is the i-th anchor selected from $\widetilde{X}_v^{(m)}$.

This leads to the bipartite sub-graph for the view member $\mathrm{View}_v^{(m)}$, defined as:

$$G_v^{(m)} = \{L_v^{(m)}, R_v^{(m)}, B_v^{(m)}\}, \quad L_v^{(m)} = X, \quad R_v^{(m)} = A_v^{(m)}, \quad B_v^{(m)} \in \mathbb{R}^{N \times \bar{p}^{(m)}}.$$

An edge between two nodes exists if and only if one node is a data sample, the other is an anchor, and the anchor is among the sample's $\bar{K}^{(m)}$ nearest anchors. Formally, the (i,j)-th entry of the cross-affinity matrix $B_v^{(m)}$ is defined as:

$$b_{i,j}^{(m)} = \begin{cases} \mathrm{Sim}(\widetilde{x}_{v,i}^{(m)}, a_{v,j}^{(m)}), & \text{if } a_{v,j}^{(m)} \in \mathcal{N}_{\bar{K}^{(m)}}(\widetilde{x}_{v,i}^{(m)}), \\ 0, & \text{otherwise,} \end{cases}$$

where $\widetilde{x}_{v,i}^{(m)}$ denotes the i-th sample in the v-th view member with reduced dimensionality, $\mathrm{Sim}(\cdot)$ is a similarity function (e.g., Euclidean distance, DTW, or others), and $\mathcal{N}_K(x)$ denotes the set of K nearest anchors to sample x.

Since each sample is linked to only $\bar{K}^{(m)}$ anchors, the cross-affinity matrix $B_v^{(m)}$ is sparse, containing only $N \cdot \bar{K}^{(m)}$ non-zero entries. This sparsity significantly reduces storage and computation requirements, which helps maintain the scalability of the framework.

Because there are $V^{(m)}$ view members in $\mathrm{VG}^{(m)}$, each data sample is ultimately represented by a total of $\bar{p}^{(m)} \cdot V^{(m)}$ anchors, forming the view-sharing bipartite graph for the entire view group:

$$G^{(m)} = \{L^{(m)}, R^{(m)}, B^{(m)}\}, \quad L^{(m)} = X, \quad R^{(m)} = \bigcup_{v=1}^{V^{(m)}} A_v^{(m)}, \quad B^{(m)} = [\, B_1^{(m)}, B_2^{(m)}, \ldots, B_{V^{(m)}}^{(m)} \,].$$

Finally, each $B_v^{(m)}$ is row-normalized to unit norm so that the multiple bipartite sub-graphs are scaled consistently within the unified representation.

4.2.3 Ensemble Generation in View Groups

With a view-sharing bipartite graph constructed for each view group, we now have M such graphs, denoted as $G^{(1)}, G^{(2)}, \ldots, G^{(M)}$, corresponding to the M randomly generated view groups.

To generate base clusterings from these graphs, I perform fast graph partitioning on each $G^{(m)}$ independently. For each view group m, the number of clusters, denoted by $k^{(m)}$, is randomly selected from a predefined range:

$$k^{(m)} \in [k_{\min},\, k_{\max}]\,,$$

where $k_{\min}$ and $k_{\max}$ are the lower and upper bounds, respectively. This random variation in the number of clusters across base clusterings increases ensemble diversity and reduces the risk of structural bias introduced by a fixed cluster count.

Recall that each view-sharing bipartite graph $G^{(m)}$ is constructed using a cross-affinity matrix $B^{(m)} \in \mathbb{R}^{N\times p}$, where p is the total number of anchors in that group. To avoid computational overhead caused by large affinity matrices, I exploit the imbalanced size of the node sets: $N \gg p$. This makes it feasible to reduce the eigen-decomposition problem from the full graph to a much smaller graph.

Specifically, I treat $G^{(m)}$ as a general graph with an affinity matrix defined as:

$$E^{(m)} = \begin{bmatrix} 0 & B^{(m)} \\ \left(B^{(m)}\right)^{\top} & 0 \end{bmatrix},$$

which would normally require solving a generalized eigenproblem of size $(N+p)\times(N+p)$ for spectral clustering [32]. To avoid this, I instead compute the Laplacian of the smaller graph:

$$E_s^{(m)} = \left(B^{(m)}\right)^{\top} \left(D^{(m)}\right)^{-1} B^{(m)},$$

where $D^{(m)}$ is a diagonal matrix with entries equal to the row sums of $B^{(m)}$. This matrix $E_s^{(m)}$ represents a similarity graph between the anchors and has size $p \times p$, which makes spectral decomposition computationally feasible.

Next, I solve the eigen-decomposition problem:

$$L_s^{(m)} u_i^{(m)} = \lambda_i^{(m)} D_s^{(m)} u_i^{(m)}, \quad i = 1, \ldots, k^{(m)},$$

where $L_s^{(m)} = D_s^{(m)} - E_s^{(m)}$ is the Laplacian matrix and $D_s^{(m)}$ is the degree matrix of $E_s^{(m)}$. The first $k^{(m)}$ eigenvectors are then stacked row-wise into a feature matrix $U^{(m)} \in \mathbb{R}^{p\times k^{(m)}}$.

Each anchor is now embedded in a $k^{(m)}$-dimensional space. To propagate this information to the original data samples, I compute:

$$H^{(m)} = \left(D^{(m)}\right)^{-1} B^{(m)} U^{(m)} \in \mathbb{R}^{N\times k^{(m)}},$$

so that each sample is now represented in the same low-dimensional space as the anchors.

Finally, k-means clustering is performed on the rows of $H^{(m)}$ to obtain the base clustering $\pi^{(m)}$. Repeating this process for all M view groups results in an ensemble of M base clusterings:

$$\Pi = \left\{ \pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(M)} \right\}.$$

This ensemble not only incorporates diverse views and features but also varies in clustering granularity and structure, making it a strong foundation for the final consensus step.

4.2.4 Late-Stage Consensus Function

Once the M base clusterings are generated, the next step is to combine them into a single, unified clustering result. This is accomplished through a late-stage fusion process, also known as consensus clustering.

To achieve this, I treat each base cluster in the ensemble Π as a new high-level feature that can be used to describe each data sample. Specifically, let $\mathcal{C} = \{C_1, C_2, \ldots, C_{k_c}\}$ denote the set of all clusters across all base clusterings, where $k_c = \sum_{m=1}^{M} k^{(m)}$ is the total number of clusters.

A new bipartite graph is then constructed where the left node set consists of the N original data samples and the right node set consists of the k_c clusters. A sample is linked to a cluster if it belongs to that cluster in any of the base clusterings.

This results in a binary cross-affinity matrix $B \in \mathbb{R}^{N \times k_c}$ defined as:

$$B_{ij} = \begin{cases} 1, & \text{if sample } x_i \in C_j, \\ 0, & \text{otherwise.} \end{cases}$$

Since each sample appears in exactly one cluster in each of the M base clusterings, each row of B contains exactly M non-zero entries. This structure makes B extremely sparse, which again improves computational efficiency.

Following a similar approach to the ensemble generation step, I construct an anchor-only graph:

$$E_s = B^\top D^{-1} B,$$

where D is a diagonal matrix with row sums of B. I then solve the generalized eigenproblem:

$$L_s u_i = \lambda_i D_s u_i, \quad i = 1, \ldots, k,$$

where $L_s = D_s - E_s$ is the Laplacian of the consensus graph and k is the desired number of final clusters.

Once the k leading eigenvectors are obtained and stacked into a matrix $U \in \mathbb{R}^{k_c \times k}$, I propagate the cluster structure back to the data samples using:

$$H = D^{-1} B U \in \mathbb{R}^{N \times k}.$$

Finally, the k-means clustering algorithm is applied to the rows of H to produce the final clustering result C.

This late-stage consensus process leverages the structural diversity captured in the ensemble of base clusterings and fuses them into a coherent global clustering, without requiring explicit access to the original views. Moreover, the use of bipartite graph structures and sparse matrix operations ensures that the method remains computationally scalable, even on very large datasets. Here is the pseudo-code for the FastMice framework:

Input: Multi-view dataset $X = \{X^1, X^2, \ldots, X^V\}$
Number of base clusterings M
Number of anchors p, number of neighbors K
Number of final clusters k
Output: Final clustering result $\mathcal{C}$

Step 1: Generate Random View Groups
for $m = 1$ **to** M **do**
  Randomly sample $V^{(m)} \in [1, V]$ views to form view group $\mathrm{VG}^{(m)}$
end

Step 2: Construct View-Sharing Bipartite Graphs
for *each view group* $VG^{(m)}$ **do**
  for *each view* $View_v^{(m)} \in VG^{(m)}$ **do**
    Perform random feature sampling with ratio $\tau_v^{(m)}$
    Select $\bar{p}^{(m)}$ anchors using hybrid strategy
    Compute affinity matrix $B_v^{(m)}$ via K-NN between samples and anchors
  end
  Concatenate all $B_v^{(m)}$ to form $B^{(m)}$
  Form view-sharing bipartite graph $G^{(m)}$ from $B^{(m)}$
end

Step 3: Generate Base Clusterings
for $m = 1$ **to** M **do**
  Randomly choose $k^{(m)} \in [k, 2k]$
  Perform bipartite spectral clustering on $G^{(m)}$ to get clustering $\pi^{(m)}$
end

Step 4: Fuse Base Clusterings
Construct final bipartite graph G with samples and clusters $\{C_1, C_2, \ldots, C_{k_c}\}$
Compute sparse binary matrix $B \in \mathbb{R}^{N \times k_c}$ where $B_{ij} = 1$ if $x_i \in C_j$
Perform bipartite spectral clustering on G to obtain final clustering $\mathcal{C}$
return $\mathcal{C}$

Algorithm 1: Pseudo-code FastMICE

4.3 Modifications and Extensions

This section presents the methodological enhancements introduced to adapt the FastMICE framework for the clustering of smart meter data. The original FastMICE algorithm

[25] provides a scalable and ensemble-based multi-view clustering solution. However, applying it to multi-view time-series data such as daily electricity consumption, daily average temperature, and maximum daily solar radiation requires several structural extensions. These modifications aim to improve scalability, temporal representation, and interpretability while maintaining the linear-time complexity of the original framework.

4.3.1 Time Aware Sampling Strategies

- **Random Feature Selection (Modified):** The original FastMICE relies on random feature sampling to introduce diversity across view groups. However, since time series data is inherently sequential, a naive random selection would break temporal continuity. Therefore, I introduce a time-aware modification.

  Given a time series with T values, we sample a random interval $[s, s + l]$ such that:

  $$s \sim \mathcal{U}(0, T - l), \quad l \sim \mathcal{U}(l_{\min}, l_{\max})$$

  where $l_{\min}$ and $l_{\max}$ define the bounds on sample lengths (e.g., from index 30 to index 120). The selected window $x_i^v[s : s+l]$ ensures temporal structure is preserved within the sampled features.

- **Full-Length Feature Sampling:** In addition to the modified random interval sampling strategy, a full-length feature sampling approach is considered as a baseline. In this setting, the entire time series of length is retained without any temporal truncation.

  This strategy preserves the complete temporal evolution of each view and allows the clustering algorithm to exploit global behavioral patterns. By comparing full-length sampling with localized window-based approaches, this experiment investigates whether incorporating the entire temporal context improves clustering performance.

All sampling strategies were applied independently across different views and view groups to encourage feature diversity while maintaining the temporal structure of the data.

4.3.2 Normalization and Extracted Features:

Each view $X^v \in \mathbb{R}^{N \times T}$ was normalized using a z-score transformation across the time axis:

$$\tilde{x}_i^v = \frac{x_i^v - \mu(x_i^v)}{\sigma(x_i^v)}, \quad \forall i \in \{1, \ldots, N\}$$

where $\mu(x_i^v)$ and $\sigma(x_i^v)$ are the mean and standard deviation of the i-th sample in view v.

Following normalization, a set of statistical features was extracted for each time series to capture global trends and frequency characteristics. These include:

- Minimum and maximum values: $\min(x_i^v)$, $\max(x_i^v)$
- Mean and standard deviation: $\mu(x_i^v)$, $\sigma(x_i^v)$
- First k FFT coefficients: $\text{FFT}_k(x_i^v)$

The extracted features were concatenated with the original multi-view input as an additional view to form an augmented representation used in the clustering pipeline.

4.3.3 Symbolic Representations for Time Series: SAX

One of the most influential approaches for time series dimensionality reduction and discretization is the Symbolic Aggregate approXimation (SAX) proposed by Lin et al [29]. SAX converts real-valued time series into symbolic strings by first applying Piecewise Aggregate Approximation (PAA) to reduce dimensionality, and then discretizing the PAA segments using a Gaussian-based breakpoint strategy. This transformation enables time series to be represented as fixed-length words drawn from a discrete alphabet, making them amenable to traditional text indexing techniques and symbolic processing.

A key strength of SAX is its ability to provide a lower-bounding distance measure (MINDIST) on the original Euclidean distance, which allows SAX-transformed sequences to be indexed and searched efficiently without false dismissals. The authors also demonstrated that SAX is robust to noise and scales well to massive datasets, making it suitable for a wide range of tasks, including similarity search, classification, anomaly detection, and clustering.

The symbolic nature of SAX supports extensions to more scalable variants such as iSAX, which introduces variable cardinality per segment to enable hierarchical indexing. In this thesis, SAX representations are employed to improve clustering scalability and mitigate the computational cost of traditional time-series distances like Dynamic Time Warping [30].

The transformation begins by normalizing a time series $x = [x_1, x_2, \ldots, x_T]$ to have zero mean and unit variance. The normalized series is then divided into w equal-sized segments, and the mean of each segment is computed using Piecewise Aggregate Approximation (PAA):

$$\text{PAA}_i = \frac{w}{T} \sum_{j=(i-1)\cdot\frac{T}{w}+1}^{i\cdot\frac{T}{w}} x_j, \quad i = 1, \ldots, w$$

These w mean values are then discretized into symbols from a fixed-size alphabet $\mathcal{A}$ using breakpoints derived from the standard normal distribution. The result is a symbolic representation $S = [s_1, s_2, \ldots, s_w]$, where $s_i \in \mathcal{A}$.

Practical Advantages of SAX:

SAX has several practical benefits that make it highly compatible with scalable clustering:

- **Computational Efficiency:** Converting time series into symbolic strings significantly reduces storage and computation overhead. Instead of comparing full-length series, clustering can be performed on short symbolic sequences.
- **Avoiding DTW:** Unlike many time-series clustering methods that rely on Dynamic Time Warping (DTW) [30] to compute similarity, SAX enables the use of simple and fast Euclidean distance in the symbolic space. This is particularly valuable when clustering hundreds of thousands of time series, where DTW is computationally prohibitive.
- **Lower-Bounding Guarantee:** SAX satisfies a theoretical guarantee through the *MINDIST* function, which lower-bounds the true Euclidean distance between the original time series. That is, for any two time series x and y and their corresponding SAX representations S_x and S_y, the following inequality holds:

  $$\text{MINDIST}(S_x, S_y) \leq \|x - y\|_2$$

  This property ensures that no false dismissals occur when pruning or comparing series, making SAX especially effective for clustering or indexing large datasets.

In this thesis, SAX representations were used as a preprocessing step to transform time-series views before clustering. This transformation reduced the reliance on computationally expensive distance metrics such as (DTW [30]) and enabled the use of ensemble clustering on a scale without compromising accuracy or structure. By combining SAX with the FastMICE framework, the system should benefit from both symbolic abstraction and multi-view fusion.

The modified pipeline integrates SAX and the sampling process into the early-stage bipartite graph construction of FastMICE. Each view—electricity, temperature, and sun radiation is processed individually to obtain symbolic representations and anchor sets. These anchors serve as representatives in the view-specific bipartite graphs, which are then fused through the FastMICE ensemble mechanism. Together, these extensions enable the framework to process multi-view smart-meter data efficiently while aiming to preserve the clustering performance.

Table 4.2: Summary of Modifications and Their Impact on the Extended FastMICE Framework

Modification	**Purpose**	**Impact on Clustering**
SAX representation	Compress time-series data while preserving temporal shape	Enables efficient distance computation and improves scalability
Feature normalization	Align feature scales across views	Ensures balanced contribution of each view in the fusion process
Time-window sampling	Capture seasonal and localized temporal patterns	Improves robustness and interpretability of the resulting clusters
Integration into FastMICE	Combine symbolic representations with ensemble-based fusion	Preserves near-linear runtime while incorporating temporal structure

5 Experiments

Extensive experiments were conducted to assess the performance of the extended Fast-MICE framework. This chapter describes the experimental setup, the parameter choice, and discusses the obtained results.

The objectives of this chapter are as follows:

- Performance assessment of the original FastMICE framework
- Evaluate the effectiveness of the extended FastMice in comparison to the original framework and other Multi-view clustering methods
- Investigate the impact of the view change on clustering performance
- Compare the runtime of the multi-view clustering methods on the real-world smart meter dataset (large datasets)

5.1 Evaluation Metrics

The clustering performance is evaluated using the Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI), which measure the agreement between the predicted cluster assignments and a set of reference labels. These reference labels may correspond to ground-truth labels or pseudo labels, depending on the dataset.

The Normalized Mutual Information is defined as

$$\mathrm{NMI}(y, c) = \frac{I(y, c)}{0.5\,(H(y) + H(c))}, \tag{5.1}$$

where y denotes the reference labels, c denotes the cluster assignments produced by the model, $I(y, c)$ is the mutual information between y and c, and $H(\cdot)$ denotes entropy. Mutual information measures the reduction in uncertainty of one variable given knowledge of the other and is defined as

$$I(y, c) = H(y) - H(y \mid c). \tag{5.2}$$

The entropy $H(y)$ measures the uncertainty associated with the label distribution and is given by

$$H(y) = -\sum_{i=1}^{|K|} p(y_i) \log p(y_i), \tag{5.3}$$

where $|K|$ is the total number of samples and $p(y_i)$ denotes the probability of observing label y_i. The NMI score takes values in the range $[0, 1]$, where higher values indicate stronger agreement between the clustering and the reference labels.

In addition to NMI, the Adjusted Rand Index is used to evaluate clustering performance from a pairwise perspective while correcting for chance agreement. ARI is defined as

$$\text{ARI} = \frac{\text{Index} - \mathbb{E}[\text{Index}]}{\max(\text{Index}) - \mathbb{E}[\text{Index}]}, \tag{5.4}$$

where the index counts the number of pairs of samples that are assigned consistently in both y and c, and $\mathbb{E}[\text{Index}]$ denotes its expected value under random labeling. The ARI score lies in the interval $[-1, 1]$, where a value of 1 indicates perfect agreement, values close to 0 correspond to random clustering, and negative values indicate agreement worse than random.

Together, NMI and ARI provide complementary perspectives on clustering quality. While NMI emphasizes the information-theoretic similarity between partitions, ARI focuses on pairwise consistency with statistical correction for chance. Higher values of both metrics indicate better alignment between the obtained clustering structure and the reference labeling.

5.2 Parameter Settings

The key FastMICE parameters used in all the experiments were set as follows:

- Number of anchors p = 1000
- Number of nearest neighbors K = 5
- Ensemble size M = 20
- sampling ratio will vary based on the experiment.

These parameters are the same as in the original paper. The authors of the *FastMICE* framework Huang et al. (2023) [25], deliberately designed the algorithm to be tuning-free and scalable, avoiding any dataset-specific hyperparameter tuning. To achieve this, they fixed or randomized key parameters within general ranges that proved robust across all 22 benchmark datasets. Specifically, they set the number of anchors to $p = \min\{1000, N\}$ to balance computational efficiency and representation accuracy, and the ensemble size to $M = 20$ to ensure sufficient diversity in the random view groups without increasing runtime significantly. Other parameters, such as the feature sampling ratio (randomized in [0.2, 0.8]), the number of nearest anchors ($K = 5$), and the number of clusters per base clustering (randomized in $[k, 2k]$), were chosen to maintain multi-level diversity and robustness. This parameter design enables *FastMICE* to generalize well across datasets while preserving its near-linear time complexity.

For the SAX representations, the parameters were fixed across all datasets. Before fixing these values, an empirical evaluation was conducted on the smart meter datasets

to examine the effect of different parameter choices on the resulting cluster centroids. Several word length configurations were considered to capture different levels of temporal aggregation, ranging from quarterly patterns (*word length = 4*) to monthly patterns (*word length = 12*), intermediate resolutions (*word length = 36*), and finer weekly patterns (*word length = 52*). In addition, alphabet sizes in the range of 6 to 8 symbols were evaluated to allow for finer discretization of amplitude variations. Based on visual inspection of the resulting cluster centroids, a word length of 36 segments and an alphabet size of 8 symbols generally produced centroid profiles that were more distinct and better separated, while preserving the dominant temporal consumption characteristics observed in the raw time series. These parameters were subsequently fixed and applied uniformly across all datasets, in order to remain consistent with the tuning-free and scalable design principles of the *FastMICE* framework.

Lastly, due to the absence of ground-truth labels, determining the optimal number of clusters k is essentially challenging. To address this, the elbow method was employed using k-means clustering algorithm. For each dataset, all views were independently normalized, and k-means was applied separately to each view for candidate values of k in the range 2 to 20. For a given k, the within-cluster sum of squares (inertia) was calculated for each view and then summed across all views, producing a global measure of cluster compactness. To objectively identify the elbow point, a curvature-based criterion was adopted: the aggregated inertia values and corresponding k values were first normalized, and the optimal number of clusters was selected as the point with the maximum perpendicular distance to the line connecting the endpoints of the inertia curve. Using this procedure, the elbow analysis identified $k = 6$ for the real-world smart meter dataset (Figure 5.1), and $k = 6$ and $k = 5$ for the CMAPSS FD001 and FD003 subsets, respectively (Figures 5.2 and 5.3). Subsequently, these values were fixed and used consistently in all reported experiments.

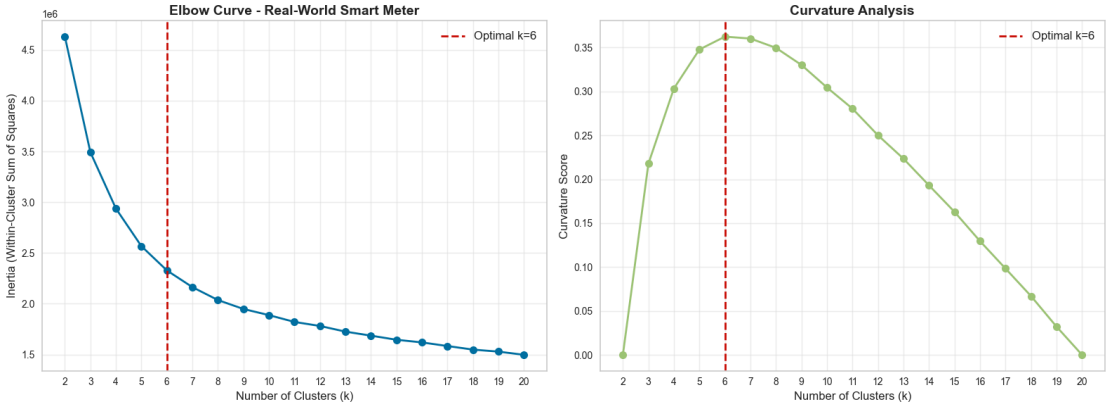


Figure 5.1: Elbow curve and curvature analysis for the real-world smart meter dataset. The optimal number of clusters is identified at $k = 6$, corresponding to the maximum curvature score. The analysis is performed on a 10% subset of the data to reduce computational complexity.

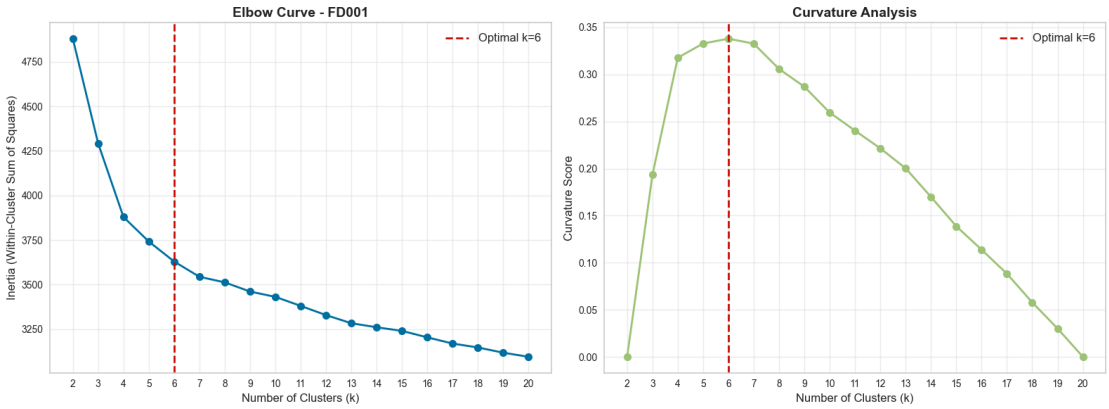


Figure 5.2: Elbow curve and curvature analysis for the FD001 dataset. The inertia curve exhibits a clear elbow at $k = 6$, which is further confirmed by the peak in the curvature score, indicating the optimal number of clusters.

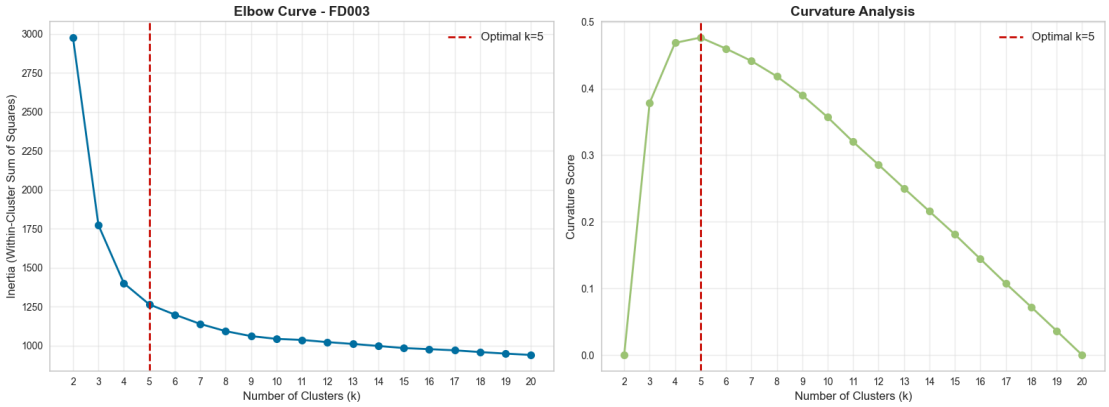


Figure 5.3: Elbow curve and curvature analysis for the FD003 dataset. The inertia curve exhibits a clear elbow at $k = 5$, which is further confirmed by the peak in the curvature score, indicating the optimal number of clusters.

5.3 Performance Evaluation

Since the FastMICE framework constitutes the methodological foundation of this thesis, the first experimental objective is to reproduce the results reported in the original paper. As the original implementation of FastMICE is provided in MATLAB, a full reimplementation of the framework was developed in Python to enable further experimentation and extension. The source code of the Python implementation is publicly available at: `https://github.com/ZoheirELhouari/MultiView-Clustering-smart-meters`.

To assess the correctness and fidelity of the Python implementation, an evaluation is conducted on the Amsterdam Library of Object Images (ALOI) dataset [36], which was

also used as a benchmark in the original FastMICE paper. In the subsequent experiments, the FastMICE framework is evaluated only on multi-view time series datasets.

The performance of the reimplemented Python version is compared against the results reported in the original FastMICE paper. The values for the original FastMICE framework, shown in the first row of Table 5.1, are taken directly from [25], where the authors report the average performance over 20 runs. These results were not recomputed using the original MATLAB implementation. The Python implementation is evaluated using the same experimental feature sampling ratio (randomized in [0.2, 0.8]), and clustering performance is measured in terms of average Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) over 5 independent runs.

Method	**NMI (%)**	**ARI (%)**
Original FastMICE [25]	**83.39**	**66.40**
FastMICE (Python, sampling ratio [0.2, 0.8])	78.84	55.44

Table 5.1: Evaluation on the ALOI dataset: comparison of average NMI (%) and ARI (%) scores. The results for the original FastMICE framework are taken directly from [25], where they are reported as averages over 20 runs. The Python implementation results correspond to averages over 5 runs.

Table 5.1 shows that the primary objective of reproducing the original FastMICE results is largely achieved. While the Python implementation exhibits slightly lower NMI and ARI scores compared to the reported MATLAB results, the values remain relatively close. This indicates that the reimplementation captures the essential behavior of the FastMICE framework. Minor discrepancies may be attributed to implementation-level differences, numerical precision, differences in random initialization, or variations in optimization routines between MATLAB and Python environments.

Having established the validity of the Python implementation, the next step will focus on evaluating the effectiveness of the proposed modifications and extensions relative to the original FastMICE framework. As discussed in Section 4.3, these extensions are designed to improve scalability and enhance temporal representations when clustering multi-view time-series data. Accordingly, the experimental evaluation is conducted on the Human Activity Recognition (HAR) Supply Chain Risk Management (SCRM) dataset, and the CMAPSS subsets (FD001 and FD003).

The analysis proceeds in an incremental manner. First, the impact of different feature sampling strategies is examined, including the original randomized feature sampling ratio in the range [0.2, 0.8], the modified random sampling, and full-length feature sampling. Next, the influence of symbolic SAX representations and feature normalization as preprocessing steps is investigated. The word length and the alphabet size were set to 36 and 8, respectively. For the feature normalization, each view was normalized using a z-score transformation across the time axis.

Finally, the best-performing variant is selected and compared against the baseline multi-view clustering methods to assess their relative effectiveness.

Table 5.2 shows the clustering performance of the different FastMICE modifications

Method	HAR		SCRM		FD001		FD003	
	NMI	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**
FastMICE with sampling ratio [0.2, 0.8]	**62.41**	**53.88**	69.06	64.41	**44.55**	39.37	39.05	**28.66**
FastMICE with modified random sampling	60.76	52.16	58.41	48.76	40.55	21.68	42.67	27.65
FastMICE with full-length	61.46	53.17	64.71	63.06	41.62	23.08	**43.38**	26.41
FastMICE + feature normalization	53.59	41.42	69.99	65.45	42.90	**40.09**	34.85	23.87
FastMICE + SAX representation	60.34	45.68	**70.01**	**65.82**	44.36	39.98	34.79	22.95

Table 5.2: Evaluation of the proposed FastMICE modifications and extensions across multiple datasets: comparison of the best percentages of NMI (%) and ARI (%) scores over 5 runs.

and extensions across the HAR, SCRM, FD001, and FD003 datasets. Overall, the results indicate that the proposed variations yield comparable NMI and ARI scores to the original FastMICE configuration, with only marginal differences across datasets. In particular, no single modification consistently outperforms the baseline sampling strategy with sampling ratio [0.2, 0.8] across all datasets and evaluation metrics.

For the HAR dataset, the original randomized sampling strategy achieves the highest NMI and ARI values, while the remaining variants exhibit slightly lower but comparable performance. This suggests that, for relatively short and well-structured sensor-based time series such as HAR, the original FastMICE sampling mechanism is already sufficient to capture the dominant temporal patterns, and additional modifications do not lead to significant gains. Similarly, for the SCRM dataset, all variants achieve high clustering performance, with feature normalization and SAX-based representations yielding marginal improvements in NMI and ARI. However, these improvements are small in magnitude and do not represent a substantial gain from the baseline performance.

The results on the FD001 and FD003 datasets further support this observation. While certain variants achieve locally higher scores for specific metrics—such as improved ARI for FD001 under feature normalization or slightly higher NMI for FD003 using full-length sampling—these gains are not consistently observed across both metrics or datasets. This variability indicates that the effectiveness of individual modifications is dataset-dependent and does not generalize uniformly across different time-series characteristics and degradation patterns.

Taken together, these findings suggest that the initial hypothesis—that the proposed modifications and extensions would improve temporal representations and thereby enhance clustering performance—does not hold in a consistent or statistically meaningful manner. Instead, the results imply that the original FastMICE framework already provides a robust mechanism for integrating temporal information in multi-view time-series clustering. While certain preprocessing steps, such as normalization or symbolic representations, may offer modest benefits in specific scenarios, their overall impact on clustering performance remains limited.

These observations highlight an important insight: improvements in temporal representation do not necessarily translate into better clustering performance when the underlying framework is already capable of effectively capturing cross-view dependencies.

Consequently, further performance gains may require more fundamental changes to the clustering objective or representation learning mechanism, rather than modifications to feature sampling or preprocessing strategies. In light of this finding, the next subsection shifts focus from internal modifications to a comparative analysis, in which the original FastMICE framework with the random feature sampling strategy is evaluated against baseline multi-view clustering methods in order to assess its relative effectiveness.

5.3.1 Comparison to Baseline Methods

In this subsection, the clustering performance of the FastMICE framework is compared against several established baseline multi-view clustering methods. The evaluation is conducted across multiple datasets using all available views, with particular emphasis on the ALOI dataset, a commonly used benchmark in the multi-view clustering literature and employed in the original FastMICE study. The selected baselines include representative spectral-based and deep-learning-based multi-view clustering approaches, providing a broad comparison against different methodological paradigms.

The considered methods include FastMICE using feature sampling ratio (randomized in [0.2, 0.8]), SCMVC [38], DSMVC [39], Multi-View Spectral Clustering [37], and Multi-View Co-Regularized Spectral Clustering [37]. Together, these methods cover a wide range of design choices, from traditional spectral clustering formulations to modern deep multi-view learning approaches. The quantitative comparison is summarized in Table 5.3, which reports the average Normalized Mutual Information (NMI (%)) and Adjusted Rand Index (ARI (%)) percentages over five independent runs.

Table 5.3: Comparison of average NMI (%) and ARI (%) scores over 5 runs across different multi-view clustering methods and datasets.

Method	**HAR**		**SCRM**		**FD001**		**FD003**		**ALOI**	
	NMI	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**
FastMICE	**62.41**	**53.88**	**69.06**	**64.81**	41.91	19.17	39.05	28.66	**78.84**	**55.44**
SCMVC [38]	54.76	38.30	62.86	48.60	39.79	20.24	41.55	28.09	20.00	1.12
DSMVC [39]	43.99	34.67	64.02	54.68	**43.70**	**30.49**	40.71	**32.52**	40.71	8.60
MVSC [37]	N/A	N/A	N/A	N/A	33.81	14.31	**46.61**	27.95	N/A	N/A
MVCRSC [37]	N/A	N/A	N/A	N/A	43.40	29.64	39.65	26.82	N/A	N/A

Note: N/A indicates that the experiment could not be completed due to out-of-memory errors.

As shown in Table 5.3, FastMICE consistently achieves strong clustering performance across the evaluated datasets. In particular, it attains the highest NMI and ARI scores on the HAR, SCRM, and ALOI datasets, indicating its effectiveness in capturing complementary information across multiple views. These results suggest that the anchor-based bipartite graph formulation and ensemble clustering employed by FastMICE provide a robust mechanism for integrating heterogeneous views without overfitting to a specific data modality.

On the CMAPSS datasets (FD001 and FD003), the performance differences between methods are more nuanced. While DSMVC achieves competitive or slightly higher scores on certain metrics, FastMICE remains comparable and demonstrates stable performance across both subsets. This behavior highlights the sensitivity of deep multi-view methods to dataset size and structure, whereas FastMICE exhibits more consistent behavior across datasets with varying temporal characteristics and noise levels.

It is also noteworthy that the spectral-based baseline methods could not be evaluated on all datasets due to memory limitations, as indicated by the N/A entries in Table 5.3. These limitations further emphasize the scalability advantages of FastMICE, particularly when dealing with high-dimensional or large-scale multi-view datasets. A more detailed discussion on the runtime and scalability is available on Section 5.5

Overall, the results demonstrate that FastMICE outperforms the baseline methods on three out of the five evaluated datasets and remains competitive on the remaining ones. This outcome is consistent with the findings reported in the original FastMICE paper and provides additional confidence in the correctness and robustness of the Python reimplementation developed in this thesis. Building on these observations, the next set of experiments shifts focus from method-level comparison to an analysis of how the number of available views influences clustering performance. In particular, the following subsection investigates the impact of view change on clustering quality, aiming to better understand the role of complementary information across multiple views.

5.3.2 Impact of View Change

This subsection investigates the impact of varying the number of views on clustering performance. Experiments are conducted on the HAR and ALOI datasets to examine whether the effect of view increase is consistent across datasets with different data modalities. HAR represents a multi-view time-series dataset derived from wearable sensor data, while ALOI is a multi-view image dataset composed of heterogeneous visual features, allowing the analysis to cover both temporal and visual settings.

The experiments are performed using the FastMICE framework, the SCMVC and DSMCV baseline methods, Multi-View Spectral Clustering [37], and Multi-View Co-Regularized Spectral Clustering [37] are excluded from this analysis due to out-of-memory errors encountered during execution. The goal is to evaluate whether increasing the number of views enables the clustering algorithms to exploit complementary information and improve clustering performance.

For each dataset, clustering is carried out by gradually increasing the number of views, starting from two and adding one view at a time until all available views are included. The performance is evaluated at each step, enabling the assessment of how view cardinality influences clustering quality across datasets of different characteristics.

As shown in Table 5.4, the impact of increasing the number of views on clustering performance differs across methods and datasets, but several consistent trends can be observed. Overall, increasing the number of views generally leads to improved clustering performance, indicating that additional views provide complementary information that can be effectively exploited by multi-view clustering algorithms.

Table 5.4: Impact of view change on clustering performance: comparison of average NMI (%) and ARI (%) percentage over 5 runs.

Method	**HAR**						**ALOI**					
	2 Views		**3 Views**		**4 Views**		**2 Views**		**3 Views**		**4 Views**	
	NMI	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**	**NMI**	**ARI**
FastMICE	43.32	29.15	53.47	46.26	**62.41**	**53.88**	64.68	30.90	77.10	54.05	**78.84**	**55.44**
SCMVC [38]	28.63	14.28	**56.06**	**40.35**	54.76	38.30	19.42	1.08	19.73	1.12	**20.18**	**1.25**
DSMVC [39]	37.41	20.97	**44.93**	**38.24**	43.99	34.67	40.53	8.45	40.02	8.46	**40.71**	**8.60**

For the HAR dataset, which represents a multi-view time-series setting derived from wearable sensor data, FastMICE exhibits a clear and consistent improvement as the number of views increases. Starting from two views, both NMI and ARI values steadily rise as additional views are incorporated, reaching their highest values when all four views are used. This behavior suggests that the different sensor-derived views capture complementary temporal dynamics, and that FastMICE is able to integrate this information effectively to form more coherent clusters. In contrast, SCMVC and DSMVC show less consistent trends. While both methods benefit from an increase from two to three views, their performance either stagnates or slightly decreases when all four views are included, indicating a reduced ability to robustly integrate additional temporal information.

A similar but more pronounced pattern is observed on the ALOI dataset, which represents a multi-view image-based scenario with heterogeneous visual feature representations. For FastMICE, clustering performance improves consistently with the addition of more views, achieving the highest NMI and ARI scores when all four views are utilized. This result highlights the strength of FastMICE in leveraging complementary visual descriptors, such as color and texture features, to enhance clustering quality. The steady improvement suggests that the additional views contribute meaningful and non-redundant information in the image domain.

In contrast, SCMVC and DSMVC show limited sensitivity to view increase on the ALOI dataset. Although slight improvements are observed as more views are added, the overall clustering performance remains significantly lower compared to FastMICE. This behavior may indicate that these methods are more sensitive to noise or redundancy when dealing with high-dimensional visual features, or that their fusion mechanisms are less effective at exploiting complementary information across multiple image-based views.

Comparing the two datasets, it becomes evident that the effect of view increase is more stable and pronounced for FastMICE across both temporal and visual modalities. While HAR benefits from additional views through improved temporal representation, ALOI demonstrates how diverse visual features can substantially enhance clustering performance when appropriately integrated. These results support the hypothesis that increasing the number of views can positively impact clustering performance, provided that the clustering framework is capable of effectively modeling cross-view relationships.

Overall, this analysis confirms that FastMICE benefits consistently from view increase across datasets of different nature, whereas the baseline methods exhibit more dataset-

dependent and less stable behavior. The observation of the NMI and ARI improvements on both HAR and ALOI datasets confirm that FastMICE is particularly effective at exploiting complementary information when additional views are introduced. Given its superior and more consistent performance compared to the other multi-view clustering methods, the FastMICE framework is therefore selected for the subsequent exploratory analysis on the real-world smart meter dataset.

5.4 Exploratory Analysis on the Real-World Smart Meter Dataset

Since the real-world smart meter dataset does not provide ground truth labels, a direct quantitative evaluation of clustering accuracy is not feasible. Instead, this section adopts an exploratory analysis approach to visually assess the clustering solutions obtained from different multi-view clustering methods. Within the broader context of this study on multi-view clustering, the objective of this analysis is to examine whether the discovered clusters exhibit meaningful structure, interpretability, and consistency across methods when applied to real-world smart meter data.

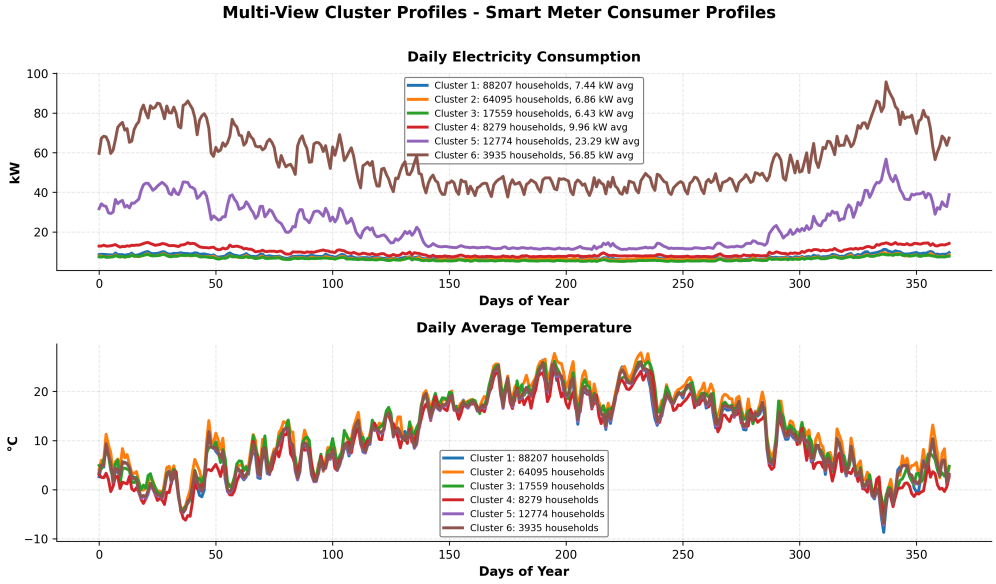
The exploratory analysis focuses on a visual and qualitative comparison of the clustering solutions produced by FastMICE using feature sampling ratio (randomized in [0.2, 0.8]) and multi-view clustering baselines, namely DSMVC and SCMVC. MVSC and MVCRSC were excluded from this analysis due to out-of-memory errors encountered during experiments on smaller datasets, as reported in Table 5.3. In particular, the shape and smoothness of the electricity consumption cluster centroids are examined to assess cluster separability and noise-handling behavior across methods. Intra-cluster variability and average consumption magnitudes are analyzed to evaluate cluster stability and physical plausibility, while cluster size distributions are examined to determine whether the different methods uncover comparable population structures or converge to fundamentally different clustering solutions. Line chart visualizations of electricity consumption centroids and bar chart visualizations of cluster distributions are used throughout this section to support the analysis.

As discussed in Section 5.2, the elbow method was used to determine the optimal number of clusters. Figure 5.1 indicates that $k = 6$ provides a suitable trade-off between model complexity and representation quality, and this value is therefore adopted consistently across all methods evaluated in this section.

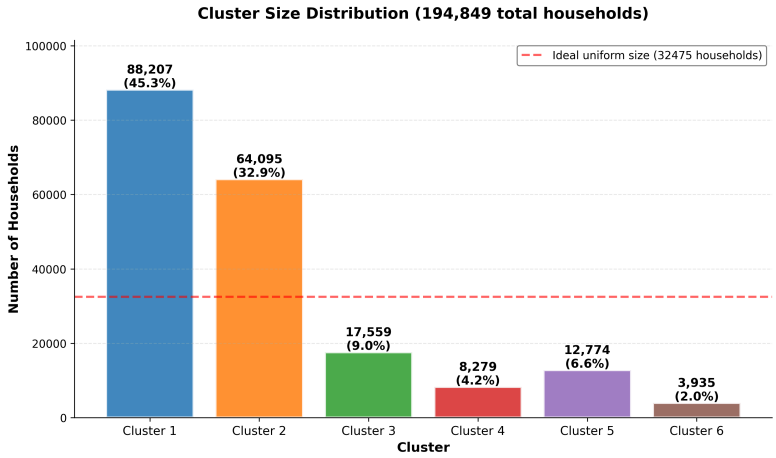
5.4.1 FastMICE

Figure 5.4 presents the clustering results obtained using the FastMICE framework, including representative electricity consumption profiles and the corresponding cluster size distribution. The distribution reveals a pronounced imbalance, with clusters 1 and 2 dominating the population. Given that the dataset represents real-world Austrian household electricity consumption, this imbalance is consistent with expectations, as the majority of buildings correspond to residential apartments and houses with moderate

and relatively stable electricity consumption. Notably, the smallest cluster identified by FastMICE corresponds to the highest consumption cluster, indicating that high-demand users constitute a relatively small fraction of the overall population.



(a) Representative average electricity consumption and temperature profiles identified by FastMICE.

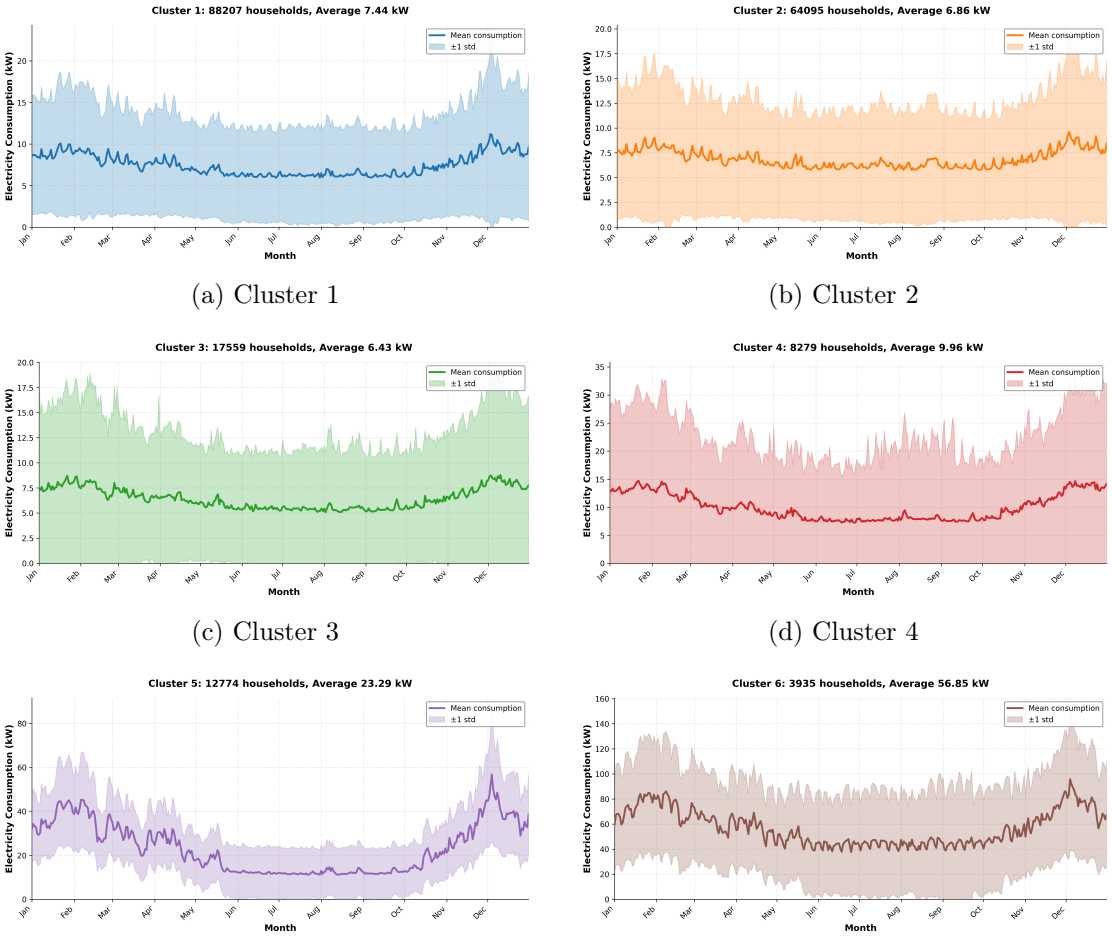


(b) Cluster size distribution obtained by FastMICE.

Figure 5.4: Clustering results of FastMICE on the real-world smart meter dataset, showing representative consumption profiles and the corresponding cluster size distribution.

The electricity consumption centroids produced by FastMICE, shown in Figure 5.5, exhibit clear seasonal patterns and relatively smooth temporal behavior. Clusters 5 and 6

represent high-consumption users and display distinct winter peaks occurring around the same period of the year, while clusters 1, 2, and 3 correspond to low-consumption users with stable consumption throughout the year and only modest increases during winter. The variability bands indicate that clusters 1, 2, and 3 exhibit relatively tight intra-cluster variability of approximately ±15 kW, whereas clusters 4 and 5 show moderate variability around ±25 kW. The highest consumption cluster (cluster 6) displays substantially larger variability in comparison to the other clusters, reaching approximately ±80 kW, reflecting the heterogeneous nature of high-demand electricity usage.



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

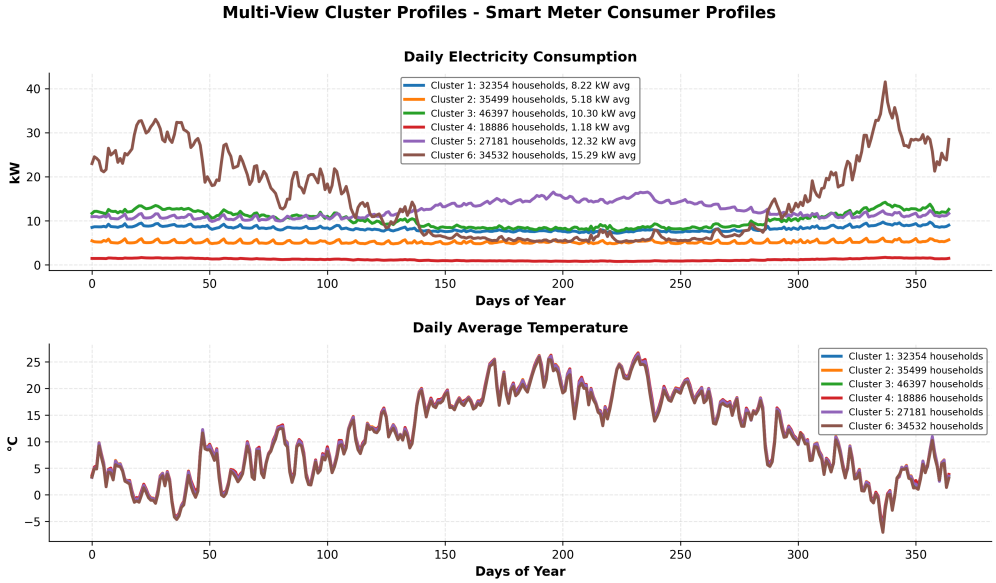
(e) Cluster 5

(f) Cluster 6

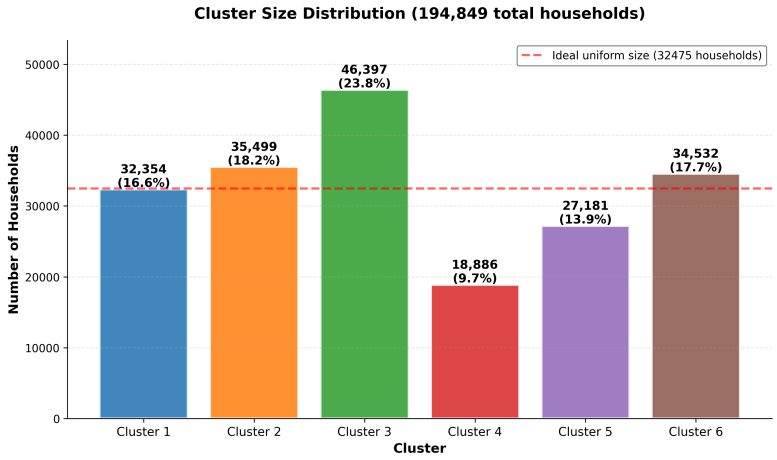
Figure 5.5: Electricity consumption profiles of the six clusters obtained from the real-world smart meter dataset using FastMICE. Each subplot shows the cluster centroid (mean daily consumption profile) together with the variability band corresponding to ±1 standard deviation.

5.4.2 DSMVC

The clustering solution obtained using DSMVC is illustrated in Figures 5.6 and 5.7. DSMVC produces the smoothest electricity consumption centroids among the evaluated methods, with minimal short-term fluctuations and strong separation between clusters.



(a) Representative average electricity consumption and temperature profiles identified by DSMVC.

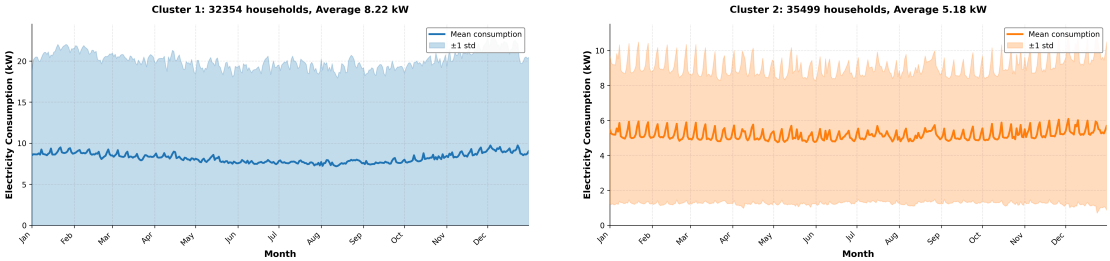


(b) Cluster size distribution obtained by DSMVC.

Figure 5.6: Clustering results of DSMVC on the real-world smart meter dataset, showing representative consumption profiles and the corresponding cluster size distribution.

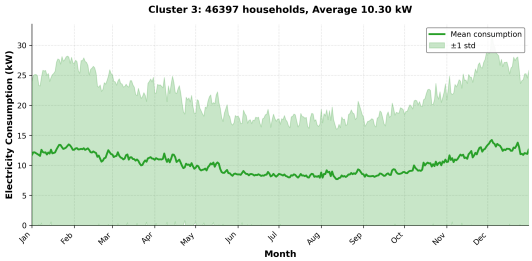
DSMVC produces a relatively uniform cluster size distribution, with no strongly

dominant clusters. Although this balance suggests structural stability, it differs from the natural population imbalance typically observed in real-world household electricity consumption, where low- and moderate-consumption users form the majority. The variability bands are notably narrow, particularly for low-consumption clusters. For example, the lowest consumption cluster exhibits a variability range of approximately ± 4 kW, while the highest consumption cluster reaches approximately ± 40 kW. These characteristics suggest strong internal cohesion and limited mixing of heterogeneous behaviors within clusters. However, the absolute magnitude of the electricity consumption centroids produced by DSMVC appears compressed, with the lowest average cluster consumption of approximately 1.18 kW. Such values are unrealistically low for daily household electricity usage and reduce the physical interpretability of the clustering solution.

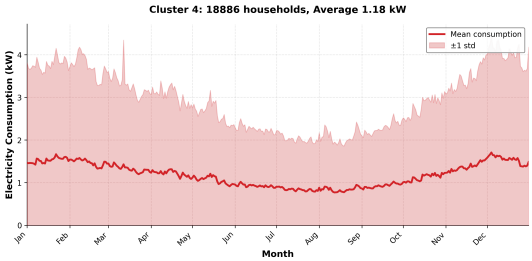


(a) Cluster 1

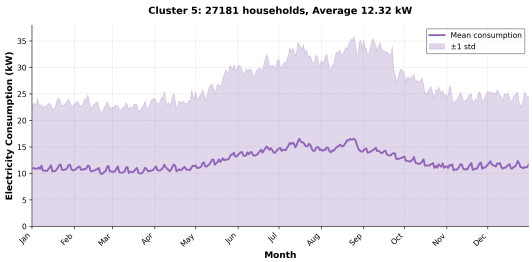
(b) Cluster 2



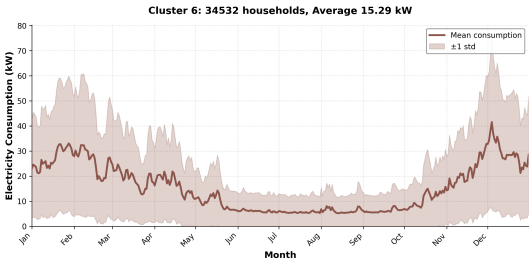
(c) Cluster 3



(d) Cluster 4



(e) Cluster 5

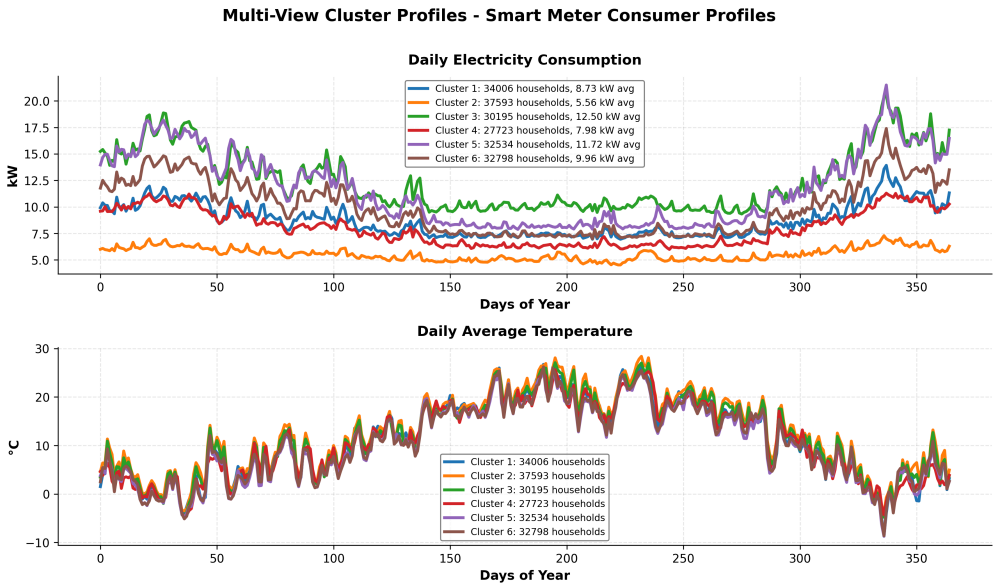


(f) Cluster 6

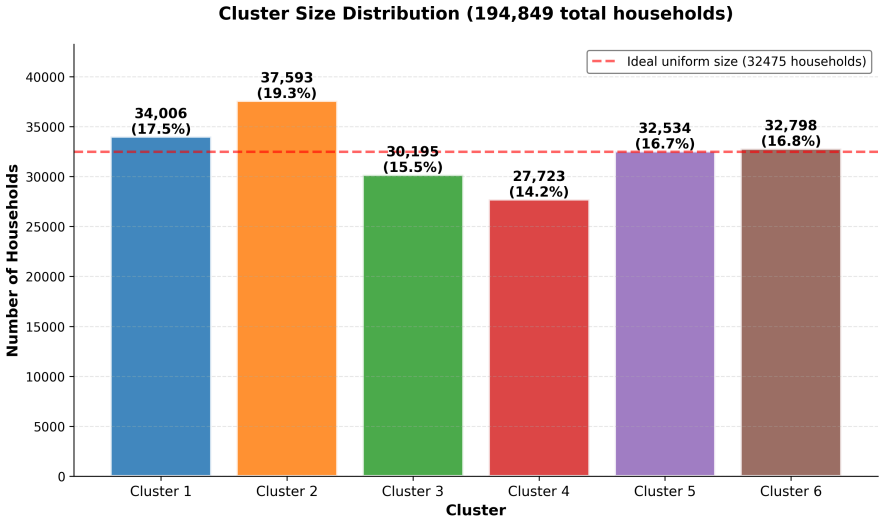
Figure 5.7: Electricity consumption profiles of the six clusters obtained from the real-world smart meter dataset using DSMVC. Each subplot shows the cluster centroid (mean daily consumption profile) together with the variability band corresponding to ±1 standard deviation.

5.4.3 SCMVC

Figures 5.8 and 5.9 present the clustering results obtained using SCMVC. Compared to FastMICE and DSMVC, SCMVC produces the least smooth electricity consumption centroids, with pronounced fluctuations observed across all clusters.



(a) Representative average electricity consumption and temperature profiles identified by SCMVC.



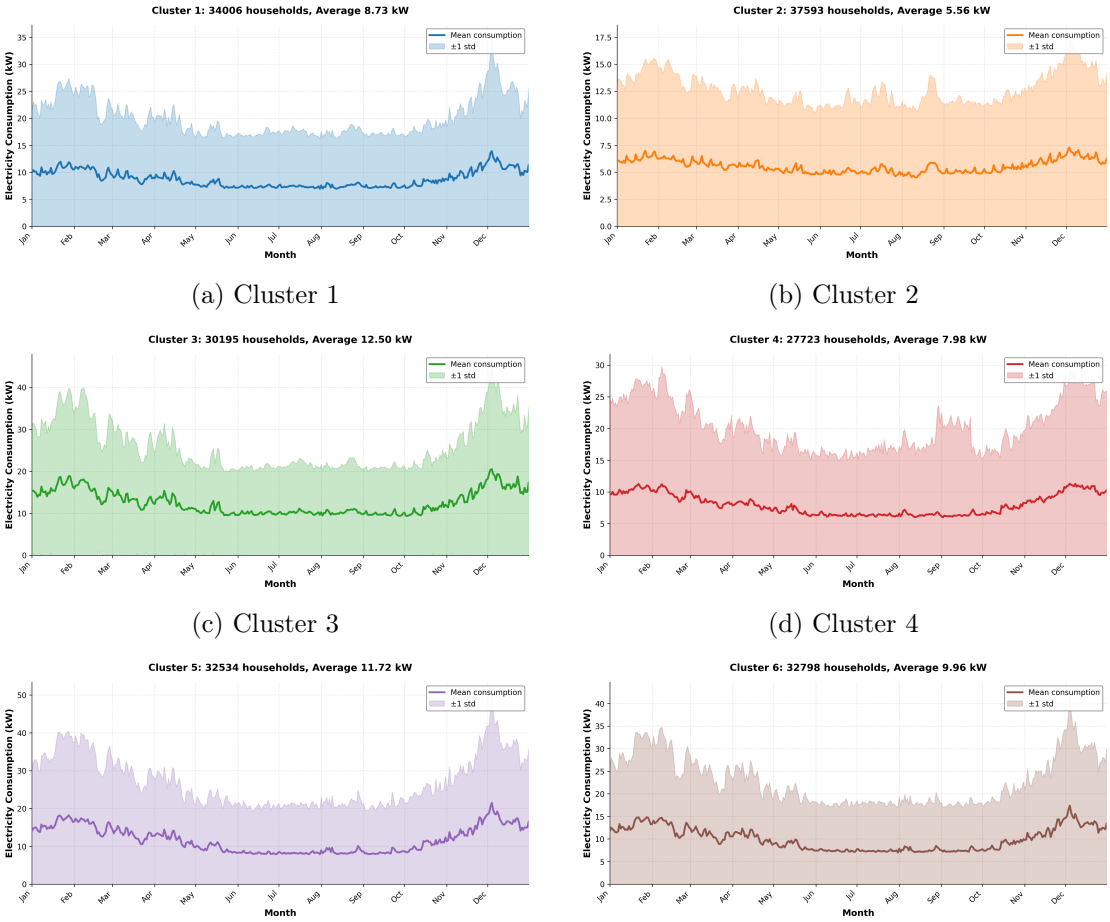
(b) Cluster size distribution obtained by SCMVC.

Figure 5.8: Clustering results of SCMVC on the real-world smart meter dataset, showing representative consumption profiles and the corresponding cluster size distribution.

Similarly to DSMVC, SCMVC also produces a relatively uniform cluster size distribution, with no strongly dominant clusters. Although this balance suggests structural stability, it differs from the natural population imbalance typically observed in real-world household electricity consumption, where low and moderate-consumption users form the majority.

The variability bands are relatively uniform, with an average range of approximately

±25 kW. While SCMVC captures the dominant seasonal structure of electricity demand, including winter peaks in high-consumption clusters, the increased centroid irregularity and noise reduce interpretability. Similar to DSMVC, SCMVC also exhibits compressed centroid magnitudes, with the lowest average cluster consumption around 5.56 kW.



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

(e) Cluster 5

(f) Cluster 6

Figure 5.9: Electricity consumption profiles of the six clusters obtained from the real-world smart meter dataset using SCMVC. Each subplot shows the cluster centroid (mean daily consumption profile) together with the variability band corresponding to ±1 standard deviation, illustrating both typical average consumption patterns and intra-cluster variability.

Cross-Method Comparison and Cross-View Dependencies: A comparison of cluster size distributions across methods further highlights structural differences. As shown in Figures 5.4b, 5.6b, and 5.8b, DSMVC and SCMVC produce relatively balanced and near-uniform cluster distributions. In contrast, FastMICE yields a distribution dominated by low-consumption clusters, which aligns more closely with the expected population

structure of real-world household electricity consumers. Moreover, the smallest clusters identified by DSMVC and SCMVC differ substantially in both size and consumption magnitude, indicating limited semantic consistency between these methods. This suggests that DSMVC and SCMVC converge to fundamentally different clustering solutions with low mutual agreement, whereas FastMICE maintains a more interpretable mapping between cluster size and consumption behavior.

Beyond the electricity consumption view, the multi-view nature of the dataset enables an examination of cross-view dependencies between electricity usage and weather-related variables. When inspecting Figures 5.4a, 5.6a, and 5.8a, which depict representative clusters in the average daily temperature view, an overlap between clusters can be observed. This overlap is not necessarily indicative of poor clustering performance, but rather reflects the relatively limited spatial and temporal variation in temperature across Austrian regions, leading to naturally similar temperature profiles among customers.

For FastMICE, Figure 5.4a reveals a clear inverse alignment between average daily electricity consumption and average daily temperature in clusters 5 and 6, where consumption peaks coincide with periods of lowest temperature. This behavior suggests heating-driven energy usage and indicates that FastMICE effectively integrates temperature information as a complementary view. In contrast, clusters 1 through 4 exhibit weaker cross-view coupling, maintaining relatively stable electricity usage throughout the year, which is consistent with baseline residential consumption patterns.

DSMVC also captures cross-view dependencies, as shown in Figure 5.6a, where high-consumption clusters display seasonal consumption peaks that broadly align with temperature minima. However, despite this alignment, the compressed magnitude of the electricity consumption centroids limits the physical interpretability of the cross-view relationships. For SCMVC, the cross-view alignment is weaker and less consistent, as illustrated in Figure 5.8a. Although seasonal trends are present, the correspondence between temperature variations and electricity consumption is less clearly defined than in FastMICE or DSMVC.

Summary: Taken together, the exploratory analysis highlights distinct strengths and limitations across the evaluated methods. DSMVC achieves superior centroid smoothness and compact variability, SCMVC captures seasonal trends but exhibits higher noise levels, and FastMICE provides a balanced compromise by preserving realistic consumption magnitudes, meaningful cluster structure, and interpretable cross-view dependencies. Overall, FastMICE offers the most coherent trade-off between interpretability, physical plausibility, and multi-view integration on the real-world smart meter dataset. The practical implications of these findings are further examined in the following section, which focuses on the runtime and scalability of the evaluated methods and their suitability for real-world applications.

5.5 Runtime and Scalability

As discussed in Section 2, scalability challenges are present in many existing multi-view clustering methods. Consequently, evaluating the computational efficiency and scalability of the proposed FastMICE framework is an important aspect of this thesis, serving as further evidence of the method's practical usefulness and its superior performance compared to baseline multi-view clustering methods in real-world settings.

All experiments were conducted on a MacBook Pro equipped with an Apple M1 Pro chip and 16 GB of RAM. The evaluation was performed using two general-scale datasets, ALOI and SCRM, as well as the large-scale real-world smart meter dataset utilizing all available views. A summary of the datasets is provided in Table 4.1. In addition, to assess the impact of the SAX-based representations introduced in Section 4.3 on runtime performance, an additional experiment was conducted on the smart meter dataset using the same parameter configuration described in Section 5.2.

Table 5.5: Runtime (in seconds) of multi-view clustering methods. Best runtime per dataset is highlighted in bold.

Method	**ALOI**	**SCRM**	**Smart-Meter**
FastMICE	**47.26**	**11.18**	**62.80**
FastMICE + SAX	-	-	**31.53**
SCMVC	1153.21	6824.90	59260.00
DCMVC	188.97	548.42	4221.43
MVSC	N/A	N/A	N/A
MVCRSC	N/A	N/A	N/A

Note: "–" indicates that the experiment is not reported. N/A indicates that the experiment could not be completed due to out-of-memory errors.

The runtime results of the evaluated multi-view clustering methods are summarized in Table 5.5. It can be observed that Multi-View Spectral Clustering and Multi-View Co-Regularized Spectral Clustering were unable to scale beyond the small CMAPSS (FD001–FD003) subsets consisting of approximately 100 samples. This behavior highlights the inherent computational limitations of traditional spectral-based multi-view clustering approaches when applied to larger datasets.

Across all evaluated datasets, FastMICE consistently outperforms both DCMVC and SCMVC by a substantial margin. This performance gap becomes particularly pronounced on the large-scale smart meter dataset, where FastMICE completes in only 62.80 seconds, compared to 4221.43 seconds for DCMVC and 59260.00 seconds for SCMVC. Furthermore, incorporating SAX representations reduces the runtime of FastMICE by approximately a factor of two, achieving a runtime of 31.53 seconds. This result highlights the positive impact of dimensionality reduction on computational efficiency by reducing the complexity of the input views prior to clustering.

These runtime characteristics can be explained by the computational design of the FastMICE framework. Specifically, the most computationally expensive operations—anchor

selection and bipartite ensemble fusion—scale with respect to the number of anchors p and the neighborhood size K, rather than the total number of data points N. Since $p \ll N$ in all experiments, the overall computational complexity grows near-linearly with N when p and K are held fixed. In contrast to methods that rely on full affinity matrix construction or repeated spectral decompositions, FastMICE avoids quadratic or cubic scaling behavior, enabling efficient processing of large datasets.

To contextualize these runtimes, modern energy providers typically operate on-premise or cloud-based infrastructures equipped with 32–64 CPU cores and 128–512 GB of RAM. Such systems can be expected to outperform a consumer-grade laptop by at least one order of magnitude, implying that the same clustering workload could complete in only a few seconds even for datasets containing millions of customers. Moreover, because FastMICE naturally parallelizes over ensemble components and view groups, the framework can be distributed across compute nodes or deployed as containerized microservices without requiring algorithmic modification.

In practical terms, these properties indicate that the proposed multi-view clustering approach is well suited for daily or weekly segmentation updates in real-world energy analytics pipelines. Its lightweight computational footprint enables seamless integration into production environments for applications such as demand-response targeting, tariff optimization, and load-profile monitoring. Rather than serving solely as an offline analytical tool, FastMICE can operate as a continuous analytics layer, dynamically re-clustering customers as new smart meter data becomes available.

Overall, the empirical runtime analysis demonstrates that the FastMICE framework constitutes a scalable and efficient clustering solution for large-scale multi-view datasets. The results further suggest that, with modest computational resources, real-time or near real-time clustering is achievable, reinforcing the practical relevance of the proposed framework for real-world deployments.

6 Conclusion

This thesis investigated multi-view clustering for time-series smart meter data. The work began with a systematic review of existing multi-view clustering methods, in which different techniques were analyzed and compared. Through this review, the strengths and limitations of current methods were discussed, with particular emphasis on their ability to handle large-scale datasets. The analysis also examined the impact of view selection and view changes on clustering performance, identifying them as important factors influencing the results.

The thesis emphasized the need for scalable multi-view clustering methods tailored to time-series data, an area that remains relatively underexplored despite its practical relevance. Motivated by the challenges associated with large-scale multi-view time-series clustering, the FastMICE framework was adopted as a scalable ensemble-based approach originally designed for large datasets. The framework was implemented in Python and extended with additional preprocessing strategies and modifications aimed at improving its suitability for multi- view time-series data. Extensive experiments were conducted to evaluate the framework and its extensions, and the results were compared with several multi-view clustering baselines.

The results show that the proposed FastMICE outperformed the baselines on three of the five datasets, while achieving comparable performance on the FD001 and FD003 noisy datasets. These findings indicate some sensitivity to noise; however, its impact on overall clustering quality remains limited. Notably, these results were obtained using a fixed set of hyperparameters across all datasets. This suggests that the framework generalizes well across datasets with varying characteristics, domains, and structures. The analysis of the proposed modifications shows that the preprocessing techniques and structural extensions do not substantially improve clustering performance. Instead, they tend to yield only marginal gains or losses of clustering quality, suggesting that the original FastMICE framework is already well suited for multi-view time-series data. In this context, the SAX-based representations extension primarily improves computational efficiency rather than fundamentally changing the clustering behavior.

In the context of the real-world smart meter dataset, the multi-view clustering results revealed temperature-sensitive consumption patterns that were not observable from electricity consumption data alone. These findings support the inclusion of contextual information, such as weather variables, in energy consumption analysis. From a practical perspective, the observed runtime and scalability characteristics indicate that FastMICE is a viable and efficient component for real-world energy analytics pipelines. Potential applications include periodic customer segmentation on daily or weekly timescales for demand-response initiatives, tariff optimization, or load-profile monitoring.

Furthermore, the analysis of view change effects confirms that multi-view clustering

generally benefits from the incorporation of additional views, although the performance gains decrease as additional views are incorporated. This observation indicates that not all views contribute equally to the clustering process. Given the observed variation in the impact of individual views, adaptive view weighting and automated assessment of view quality therefore represent promising directions for future research. However, these aspects fall outside the scope of this thesis.

In conclusion, this thesis shows that FastMICE provides a practical foundation for large-scale multi-view time-series clustering. The proposed SAX-based extension further improves computational efficiency without compromising clustering quality, reinforcing the suitability of the framework for real-world smart meter data analysis.

Bibliography

[1] Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering – A decade review. *Information Systems*, 53, 16–38.

[2] Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition*, 38(11), 1857–1874.

[3] Fu, T. C. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1), 164–181.

[4] Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications*, 36(3), 5932–5941.

[5] Harutyunyan, H., Khachatrian, H., Kale, D. C., & Galstyan, A. (2019). Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6, 96.

[6] Fildes, R., Ma, S., & Kolassa, S. (2022). Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4), 1283–1318.

[7] Tureczek, A., Nielsen, P. S., & Madsen, H. (2018). Electricity consumption clustering using smart meter data. *Energies*, 11(4), 859.

[8] Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on Loess. *Journal of Official Statistics*, 6(1), 3.

[9] Wang, Y., Chen, Q., Hong, T., & Kang, C. (2018). Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, 10(3), 3125–3148.

[10] Dryar, H. A. (1944). The effect of weather on the system load. *Electrical Engineering*, 63(12), 1006-1013.

[11] Kang, J., & Reiner, D. M. (2022). What is the effect of weather on household electricity consumption? Empirical evidence from Ireland. *Energy Economics*, 111, 106023.

[12] Serra, A., Greco, D., & Tagliaferri, R. (2015). Impact of different metrics on multi-view clustering. In *2015 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

[13] Chao, G., Sun, S., & Bi, J. (2021). A survey on multiview clustering. *IEEE Transactions on Artificial Intelligence*, 2(2), 146-168.

[14] Kang, Z., Zhou, W., Zhao, Z., Shao, J., Han, M., & Xu, Z. (2020, April). Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4), 4412-4419.

[15] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1-22.

[16] Fang, U., Li, M., Li, J., Gao, L., Jia, T., & Zhang, Y. (2023). A comprehensive survey on multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(12), 12350–12368.

[17] Wang, H., Yang, Y., & Liu, B. (2019). GMC: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6), 1116-1129.

[18] Nie, F., Li, J., & Li, X. (2016, July). Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *IJCAI*, 9, 1881-1887.

[19] Nie, F., Li, J., & Li, X. (2017, August). Self-weighted multiview clustering with multiple graphs. In *IJCAI* (pp. 2564-2570).

[20] Kang, Z., Shi, G., Huang, S., Chen, W., Pu, X., Zhou, J. T., & Xu, Z. (2020). Multi-graph fusion for multi-view spectral clustering. *Knowledge-Based Systems*, 189, 105102.

[21] He, G., Wang, H., Liu, S., & Zhang, B. (2022). CSMVC: A multiview method for multivariate time-series clustering. *IEEE Transactions on Cybernetics*, 52(12), 13425-13437.

[22] Li, Y., Nie, F., Huang, H., & Huang, J. (2015). Large-scale multi-view spectral clustering via bipartite graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

[23] Wang, S., Liu, X., Zhu, X., Zhang, P., Zhang, Y., Gao, F., & Zhu, E. (2021). Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31, 556-568.

[24] Sun, M., Xu, Z., Zhao, Z., & Han, M. (2021). Scalable multi-view subspace clustering with unified anchors. In *Proceedings of the ACM International Conference on Multimedia*, 3528-3536.

[25] Huang, D., Wang, C.-D., & Lai, J.-H. (2023). Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. *IEEE Transactions on Knowledge and Data Engineering*, 35(11), 11388-11402.

[26] Müller, M. (2007). Dynamic time warping. In *Information Retrieval for Music and Motion* (pp. 69-84). Springer.

[27] Tao, Z., Liu, H., Li, S., Ding, Z., & Fu, Y. (2017). From ensemble clustering to multi-view clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2745-2751.

[28] Bickel, S., & Scheffer, T. (2004, November). Multi-view clustering. In *ICDM* (Vol. 4, No. 2004, pp. 19-26).

[29] Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2007). Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2), 107–144.

[30] Senin, Pavel. (2008). Dynamic time warping algorithm review. *Information and Computer Science Department, University of Hawaii at Manoa, Honolulu, USA*, 855.1-23, 1–40.

[31] Huang, D., Wang, C.-D., Wu, J.-S., Lai, J.-H., & Kwoh, C.-K. (2020). Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6), 1212–1226.

[32] Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 849–856.

[33] Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Proceedings of the 1st International Conference on Prognostics and Health Management (PHM08)*, Denver, CO, October.

[34] Reyes-Ortiz, J., Anguita, D., Ghio, A., Oneto, L., & Parra, X. (2013). Human activity recognition using smartphones. *UCI Machine Learning Repository.* doi: 10.24432/C54S4K

[35] Banerjee, H., Saparia, G., Ganapathy, V., Garg, P., & Shenbagaraman, V. M. (2019). Time series dataset for risk assessment in supply chain networks. *Mendeley Data*, V2. doi: 10.17632/gystn6d3r4.2

[36] Geusebroek, J.-M., Burghouts, G. J., & Smeulders, A. W. M. (2005). The Amsterdam library of object images. *International Journal of Computer Vision*, 61(1), 103–112.

[37] Perry, R., Mischler, G., Guo, R., Lee, T., Chang, A., Koul, A., Franz, C., Richard, H., Carmichael, I., Ablin, P., Gramfort, A., & Vogelstein, J. T. (2021). mvlearn: Multiview machine learning in Python. *Journal of Machine Learning Research*, 22(109), 1–7.

[38] Wu, S., Zheng, Y., Ren, Y., He, J., Pu, X., Huang, S., Hao, Z., & He, L. (2024). Self-weighted contrastive fusion for deep multi-view clustering. *IEEE Transactions on Multimedia*, 26, 9150–9162. doi: 10.1109/TMM.2024.3387298

[39] Tang, H., & Liu, Y. (2022). Deep safe multi-view clustering: Reducing the risk of clustering performance degradation caused by view increase. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 202–211. doi: 10.1109/CVPR52688.2022.00030

[40] Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., Payne, M., Yurchak, R., Rußwurm, M., Kolar, K., & Woods, E. (2020). Tslearn: A machine learning toolkit for time series data. *Journal of Machine Learning Research*, 21(118), 1–6.

[41] Yin, Q., Wu, S., He, R., & Wang, L. (2015). Multi-view clustering via pairwise sparse subspace representation. *Neurocomputing*, 156, 12–21.