Advanced Clustering and AI-Driven Decision Support Systems for Smart Energy Management

Loup-Noé Lévy

université PARIS-SACLAY

Advanced clustering and AI-driven decision support systems for smart energy management

Systèmes avancés de regroupement et d'aide à la décision pilotés par l'IA pour la gestion intelligente de l'énergie

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n°580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Graduate school: Informatique et sciences du numérique
Référent: Université de Versailles-Saint-Quentin-en-Yvelines

Thèse préparée au sein de l'unité de recherche LI-PARAD (Université Paris Saclay, UVSQ),
sous la direction de **Soufian BEN AMOR**, MCF, HDR,
la co-direction de **Guillaume GUERARD**, Docteur, Associate Professor
et la co-supervision de **Haï TRAN**, Docteur, CTO a Energisme.

Thèse soutenue à Guyancourt, le 04/06/2024, par

LOUP-NOÉ LEVY

COMPOSITION DU JURY

Membres du jury avec voix délibérative

Nahid EMAD PU, Université de Versailles Saint-Quentin-en-Yvelines	Président
Olivier FLAUZAC PU, Université de Reims Champagne-Ardenne	Rapporteur et examinateur
Dritan NACE PU, Université de Technologie de Compiègne	Rapporteur et examinateur
Marc BUI PU, École Pratique des Hautes Études	Examinateur
Isis TRUCK PU, Université Paris 8	Examinateur

THESE DE DOCTORAT

NNT : 2024UPASG027

université PARIS-SACLAY | ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC)

Titre: Systèmes avancés de regroupement et d'aide à la décision pilotés par l'IA pour la gestion intelligente de l'énergie

Mots clés: Intelligence Artificielle, Clustering, Énergie, Aide à la decision, Gouvernance, Systèmes complexes

Résumé: Cette thèse aborde le clustering de systèmes énergétiques complexes et hétérogènes au sein d'un système d'aide à la décision (SAD).
Dans le chapitre 1, nous explorons d'abord la théorie des systèmes complexes et leur modélisation, reconnaissant les bâtiments comme des Systèmes Sociotechniques Complexes. Nous examinons l'état de l'art des acteurs impliqués dans la performance énergétique, identifiant notre cas d'étude comme le Tiers de Confiance pour la Mesure et la Performance Énergétique (TCMPE). Face à nos contraintes, nous nous focalisons sur le besoin d'un système d'aide à la décision pour fournir des recommandations énergétiques, le comparant aux systèmes de supervision et de recommandation et soulignant l'importance de l'explicabilité dans la prise de décision assistée par IA (XAI). Reconnaissant la complexité et l'hétérogénéité des bâtiments gérés par le TCMPE, nous argumentons que le clustering est une étape initiale cruciale pour développer un SAD, permettant des recommandations sur mesure pour des sous-groupes homogènes de bâtiments.
Dans le Chapitre 2, nous explorons l'état de l'art des systèmes semi-automatisés pour la prise de décisions à haut risque, mettant l'accent sur la nécessité de gouvernance dans les SAD. Nous analysons les régulations européennes, mettant en lumière le besoin d'exactitude, de fiabilité, et d'équité de notre SAD, et identifions des méthodologies pour adresser ces besoins, telles que la méthodologie DevOps et le data lineage. Nous proposons une architecture distribuée du SAD qui répond à ces exigences et aux défis posés par le Big Data, intégrant un datalake pour la manipulation des données hétérogènes et massive, des datamarts pour la sélection et le traitement spécifiques des données, et une ML-Factory pour peupler une bibliothèque de modèles. Différentes méthodes de Machine Learning sont sélectionnées pour les différents besoins spécifiques du SAD.
Le Chapitre 3 se concentre sur le clustering comme méthode d'apprentissage automatique primaire dans notre cas d'étude, il est essentiel pour identifier des groupes homogènes de bâtiments. Face à la nature plurielle - numérique, catégorique, séries temporelles - des données décrivant les bâtiments, nous proposons le concept de clustering complexe. Après avoir examiné l'état de l'art, nous identifions la nécessité d'introduire des techniques de réduction de dimensionnalité, associé à des méthodes de clustering numérique et mixte état de l'art. La Prétopologie est proposée comme approche novatrice pour le clustering de données mixtes et complexes. Nous soutenons qu'elle permet une plus grande explicabilité et interactivité, en permettant un clustering hiérarchique construit sur de règles logiques et de notions de proximité adaptées au contexte. Les défis de l'évaluation du clustering complexe sont abordés, et des adaptations de l'évaluation des jeux de donnée numérique sont proposées.
Dans le chapitre 4, nous analysons les performances computationnelles des algorithmes et la qualité des clusters obtenus sur différents jeux de données variant en taille, nombre de clusters, distribution et nombre de dimensions. Ces jeux de donnée sont publique, privées ou généré pour les tests. La Prétopologie et l'utilisation de la réduction de dimensionnalité montrent des résultats prometteurs comparés aux méthodes de clustering de données mixtes de l'état de l'art.
En conclusion, nous discutons des limitations de notre système, y compris les limites d'automatisation du SAD à chaque étape du flux de données. Nous mettons l'accent sur le rôle crucial de la qualité des données et les défis de prédire le comportement des systèmes complexes au fil du temps. L'objectivité de nos méthodes d'évaluation de clustering est questionnée en raison de l'absence de vérité terrain. Nous envisageons des travaux futurs, tels que l'automatisation de l'hyperparamètrisation et la continuation du développement du SAD.

université PARIS-SACLAY | ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC)

Title: Advanced clustering and AI-driven decision support systems for smart energy management

Keywords: Arificial Intelligence, Clustering, Energy, Decision Support , Governance, Complex systems

Abstract: This thesis addresses the clustering of complex and heterogeneous energy systems within a Decision Support System (DSS).

In chapter 1, we delve into the theory of complex systems and their modeling, recognizing buildings as complex systems, specifically as Sociotechnical Complex Systems. We examine the state of the art of the different agents involved in energy performance within the energy sector, identifying our case study as the Trusted Third Party for Energy Measurement and Performance (TTPEMP.) Given our constraints, we opt to concentrate on the need for a DSS to provide energy recommendations. We compare this system to supervision and recommender systems, highlighting their differences and complementarities and introduce the necessity for explainability in AI-aided decision-making (XAI). Acknowledging the complexity, numerosity, and heterogeneity of buildings managed by the TTPEMP, we argue that clustering serves as a pivotal first step in developing a DSS, enabling tailored recommendations and diagnostics for homogeneous subgroups of buildings. This is presented in Chapter 1.

In Chapter 2, we explore DSSs' state of the art, emphasizing the need for governance in semi-automated systems for high-stakes decision-making. We investigate European regulations, highlighting the need for accuracy, reliability, and fairness in our decision system, and identify methodologies to address these needs, such as DevOps methodology and Data Lineage. We propose a DSS architecture that addresses these requirements and the challenges posed by big data, featuring a distributed architecture comprising a data lake for heterogeneous data handling, datamarts for specific data selection and processing, and an ML-Factory populating a model library. Different types of methods are selected for different needs based on the specificities of the data and of the question needing answering.

Chapter 3 focuses on clustering as a primary machine learning method in our architecture, essential for identifying homogeneous groups of buildings. Given the combination of numerical, categorical and time series nature of the data describing buildings, we coin the term complex clustering to address this combination of data types. After reviewing the state-of-the-art, we identify the need for dimensionality reduction techniques and the most relevant mixed clustering methods. We also introduce Pretopology as an innovative approach for mixed and complex data clustering. We argue that it allows for greater explainability and interactability in the clustering as it enables Hierarchical clustering and the implementation of logical rules and custom proximity notions. The challenges of evaluating clustering are addressed, and adaptations of numerical clustering to mixed and complex clustering are proposed, taking into account the explainability of the methods.

In the datasets and results chapter, we present the public, private, and generated datasets used for experimentation and discuss the clustering results. We analyze the computational performances of algorithms and the quality of clusters obtained on different datasets varying in size, number of clusters, distribution, and number of categorical and numerical parameters. Pretopology and Dimensionality Reduction show promising results compared to state-of-the-art mixed data clustering methods.

Finally, we discuss our system's limitations, including the automation limits of the DSS at each step of the data flow. We focus on the critical role of data quality and the challenges in predicting the behavior of complex systems over time. The objectivity of our clustering evaluation methods is challenged due to the absence of ground truth and the reliance on dimensionality reduction to adapt state-of-the-art metrics to complex data. We discuss possible issues regarding the chosen elbow method and future work, such as automation of hyperparameter tuning and continuing the development of the DSS.

Composition du Jury Complet

Incluant les membres du jury sans voix délibérative

Nahid EMAD PU, Université de Versailles Saint-Quentin-en-Yvelines	Président
Olivier FLAUZAC PU, Université de Reims Champagne-Ardenne	Rapporteur et examinateur
Dritan NACE PU, Université de Technologie de Compiègne	Rapporteur et examinateur
Marc BUI PU, École Pratique des Hautes Études	Examinateur
Isis TRUCK PU, Université Paris 8	Examinateur
Soufian BEN AMOR MCF, HDR	Invité
Guillaume GUERARD Docteur, Associate Professor	Invité
Haï TRAN Docteur, CTO a Energisme	Invité

Cette thèse a été effectué sous un contrat CIFRE et en partenariat avec l'entreprise Energisme.

Remerciements

Une thèse ne se réalise jamais seul. Elle est le fruit de la participation de nombreuses personnes sans lesquelles ces travaux n'auraient pu voir le jour.

Je souhaite tout d'abord exprimer ma profonde gratitude aux membres du jury : Nahid EMAD, Professeur des Universités à l'Université de Versailles Saint-Quentin-en-Yvelines, Président du jury ; Olivier FLAUZAC, Professeur des Universités à l'Université de Reims Champagne-Ardenne, Rapporteur et examinateur ; Dritan NACE, Professeur des Universités à l'Université de Technologie de Compiègne, Rapporteur et examinateur ; Marc BUI, Professeur des Universités à l'École Pratique des Hautes Études, Examinateur ; et Isis TRUCK, Professeur des Universités à l'Université Paris 8, Examinateur.

Je tiens à exprimer mes remerciements les plus sincères à mon directeur de thèse, Soufian Ben Amor, Maître de Conférences HDR à l'Université Paris-Saclay, UVSQ. Merci de m'avoir guidé tout au long de cette thèse face aux défis scientifiques, académiques et administratifs. Vous avez su me donner du recul sur mes travaux et m'indiquer la direction à suivre sans me perdre dans les méandres de la recherche scientifique. Merci de m'avoir aidé à développer mes qualités de chercheur en me poussant, par exemple, à identifier les limites des approches et méthodes développées dans un contexte donné, tout en reconnaissant les principes fondamentaux généralisables à d'autres domaines. Vous avez toujours été disponible, à l'écoute de mes nombreuses questions, et constamment intéressé à l'avancée de mes travaux et de ma carrière.

Je remercie également mon co-directeur de thèse, le Docteur Guillaume Guerard, Associate Professor au De Vinci Research Center, pour avoir vu en moi un chercheur avant que je le devienne, pour m'avoir encouragé dans cette voie et avoir trouvé l'entreprise qui finirait par financer cette thèse. Merci pour vos conversations toujours fascinantes, la confiance que vous m'avez apportée et votre aide précieuse pour la rédaction des articles, souvent en une soirée, annihilant ainsi toute angoisse de la page blanche. Votre aide dans la structuration et la relecture des articles et du manuscrit final a été inestimable.

Je souhaite également remercier mon superviseur de thèse à l'entreprise Energisme, Hai Tran, pour m'avoir fait confiance depuis le début de cette thèse, m'avoir donné le temps et les moyens d'effectuer mes recherches de manière approfon-

die et pertinente. Merci à l'entreprise Energisme pour avoir financé cette thèse.

Un grand merci à mon encadrant de thèse, Jérémie Bosom, pour son investissement total dans mon projet de thèse, ses retours d'expérience, son soutien inestimable tant sur le plan technique que sur la rigueur et le recul scientifique nécessaires aux travaux de recherche. Merci pour avoir su me donner foi en mon projet (et en moi-même) durant les inévitables périodes de doute, tout en ayant à cœur de me préparer aux embûches qui m'attendaient. Tes relectures et aides à la rédaction de ce manuscrit ont été précieuses.

Merci à Maxence Chouffa et Clément Cornet, étudiants en parcours recherche à l'École Supérieure d'Ingénieur Léonard de Vinci, pour leur contribution à ces travaux de thèse, leur sérieux et leur rigueur. Ils ont démontré à mes yeux toutes les qualités pour devenir chercheurs à leur tour.

Je tiens à remercier les membres d'Energisme pour leur accueil, les discussions souvent riches et fructueuses qui ont permis d'ancrer cette thèse dans des problématiques réelles et de ne pas être « hors sol ». Merci tout particulièrement pour leurs contributions et leur aide à Jérémie Uhlrich Meunier, Laure Cadec, Ronan Bebin, Gaspard Leblanc, Arnaud Lossi, Léandre Jouanneau et Soline Aury.

Merci aux data scientists d'Energisme et N'Gage pour leur soutien, leurs conseils et leur riche expertise, tout particulièrement à Trang Tran.

Je remercie ma famille, mes amis et ma compagne pour leur soutien sans faille, leur écoute, leurs conseils et leur patience. Merci pour leurs réguliers « quand est-ce que tu soutiens ta thèse ? » qui n'ont pas manqué de faire monter mon taux de cortisol. Merci également à tous ceux qui sont venus partager un coin de table ou de mur à « Climbing District » pendant que j'avançais sur mon manuscrit ou sur un article.

Merci donc à tous ceux qui, de près ou de loin, m'ont soutenu depuis le début de mes études jusqu'à l'achèvement de cette thèse, qui est l'aboutissement d'un parcours et un tremplin vers la suite de ma carrière. La suite, je la dois aussi à vous.

Table of Contents

List of Figures

List of Tables

Introduction

In this pivotal era of human history, we find ourselves at multiple crossroads. The ongoing collapse of our planetary ecosystem and escalating global warming are not distant concerns but immediate realities. Concurrently, topics once relegated to science fiction, like Artificial General Intelligence and Smart Cities, are increasingly discussed in the present tense. Yet, it appears that the decisions we make now in these domains will shape the decades to come. Among these topics, leveraging Big Data and Machine Learning (ML) to mitigate the human impact on our ecosystem emerges as a potentially very impactful research area. Specifically, in Europe, where the building sector contributes over 35% of greenhouse gas emissions, harnessing vast datasets from already existing smart sensors in energy grids presents a great opportunity. By applying ML to the ever-increasing data at our disposal, we can develop actionable strategies to enhance energy efficiency, particularly through the identification and analysis of homogeneous building groups. This approach enables us to discern consumption patterns and associate them with other building characteristics, fostering novel categorizations and specific performance scales. Such targeted insights are vital for optimizing resource allocation in energy efficiency renovations and suggesting relevant changes in energy usage.

To face the urgency of climate change mitigation and to enforce effective actions, the European Climate Law, enacted on 29 July 2021, mandates a legally binding target of net-zero greenhouse gas emissions by 2050 [2]. This ambitious goal, necessitating concerted efforts at both EU and national levels, underscores the need to address the significant energy consumption in the building sector. According to the European Commission, as of 2023, approximately 35% of the EU's buildings are over 50 years old , and almost 75% are energy inefficient, yet only about 1% undergo renovations annually [34]. Enhancing the energy efficiency of existing buildings could substantially reduce the EU's total energy consumption by 5-6% and decrease CO2 emissions by roughly the same percentage [33]. Beyond meeting our climate objectives, such investments in energy efficiency stimulate the economy. The construction industry, for instance, accounts for about 9% of Europe's GDP and provides 18 million jobs. Small and medium-sized enterprises, in particular, stand to gain from a revitalized renovation market, contributing significantly to the EU's building sector's added value [1].

However, identifying the buildings that require renovation and making relevant recommendations demands a lot of time, money, and qualified personnel. I have personally heard during meetings how money allocated to energy renovation in buildings in territorial collectivities was sometimes never spent because of a lack of knowledge of how to use it best. Indeed, when dealing with a large number of heterogeneous buildings, the solutions to apply are not easily identi-

fiable. If the buildings vary in usage, size, meteorological environment, and year of construction, identifying those with abnormal consumption is not easy, and conducting an energy audit on all of them can be extremely costly and time-consuming. Hence, the needs for the clustering of buildings according to the building characteristics. Those characteristics are of heterogeneous natures; some are categorical, others are numerical, and others are time series.

Based on this case study, this thesis will describe scientific theories, methodologies, and the associated technical tools for decision support in the Trusted Third Party for Energy Measurement and Performances (TTPEMP), focusing on clustering as the first step in making energy performance diagnostic and recommendation. This contribution is not restricted to energy systems, it can be applied in numerous other contexts dealing with heterogeneous systems clustering and recommendation systems regarding Complex Socio-Technical System (CSTS). Applications to other systems will be discussed throughout this thesis.

The main contributions of this thesis are the following. The identification of challenges in our case study. A comprehensive state-of-the-art analysis of the various topics studied in this thesis, some theoretical some more practical, including the modeling of CSTS, smart energy systems, the governance of decision support systems, the advanced clustering of mixed and complex data as well as cloud architecture tools. During this thesis we introduce the architecture of a Decision Support System (DSS) and present innovative clustering methods and tools specifically designed for the clustering of mixed and complex data. Additionally, we introduce a library for mixed data clustering and analysis, adapting state-of-the-art clustering methods to accommodate mixed data types. The thesis further explores and proposes complex clustering methodologies using a developed clustering and analysis platform, demonstrating the practical application and utility of the research findings in real-world scenarios.

Chapter 1

Context : Complex Energy Systems

This thesis studies buildings and their energy management systems. Buildings and their stakeholders within energy systems are quintessential examples of Complex Socio-Technical System (CSTS)[23]. These systems demonstrate properties that significantly influence not only the theories, methods, and tools employed for analysis and recommendation generation but also the design and conception of the recommendation systems themselves. This dual influence is crucial for ensuring both the relevance and long-term usability of these systems.

CSTSs are characterized by intricate interactions between human, technological, and organizational components. The complexity of these systems arises not only from the individual elements but, more importantly, from their interconnections and the emergent behaviors that result [32]. Therefore, understanding these systems requires a holistic approach that considers both social and technical dimensions. Moreover, designing recommendation systems for these contexts is not a straightforward task. It necessitates a deep understanding of the system's properties to ensure that the solutions proposed are not only technically sound but also socially acceptable and practically implementable. The long-term success of these systems hinges on their ability to adapt to evolving conditions and to be sensitive to the needs and constraints of the various stakeholders involved.

In this chapter, we present a comprehensive overview of the foundational principles of complex systems and their modeling, followed by a state-of-the-art about building and smart energy system modeling. We then introduce our case study as a Trusted Third Party for Energy Measurement and Performances (TTPEMP). The chapter further delves into the specifics of modeling the TTPEMP, focusing on its nature as a Decision Support System (DSS). Finally, we discuss the integration of Explainable Artificial Intelligence (XAI) to enhance the DSS, leading to an exploration of pretopology for clustering mixed data. This approach effectively bridges the gap between theoretical complexity and practical application in energy management systems.

1.1 Definition of Complex Systems

According to Johnson [74], Complexity Science can be seen as the study of the phenomena that emerge from a collection of interacting objects.

However, there is no simple definition of a Complex System (CS) on which all scientists agree, as it varies from one domain to another, and arriving at a defini-

tion of complexity through necessary and sufficient conditions seems difficult if not impossible. Therefore in this section, without giving a final definition, we will present several properties that are associated with complexity in the literature.

First, we can briefly differentiate between complex and complicated. A complicated system can be subdivided into simpler problems using traditional analytical methods. On the contrary, the simplification of a complex problem is often considered a source of more complexity because it reduces the intelligibility of the system [89]. Most of CSs are defined as follows in the literature :

Définition 1.1. A Complex System is a system with three complexity factors :
- Multiplicity of space and time scales ;
- Heterogeneity of the components ;
- Emergence properties observed during sudden and unpredictable phase transitions.

From these factors, Ladyman et al. [87] identify that CSs are associated to the following concepts :

Définition 1.2. A Complex System is a system with the following properties : irreducibility, non-linearity, feedback, self-organization, emergence, distributed control, hierarchical organization, and numerosity.

To better understand the concept of CS, let us introduce all these properties.

Irreducibility [86] : To give an overview of the subject, we note that the study of systems has led scientists to extract common characteristics of systems described as complex. They are non-deterministic and sensitive to initial conditions, are often of large scales and reduced scale analysis does not take into account all the aspects of these systems. They are therefore irreducible.

Nonlinearity [50] : Nonlinearity is often considered to be essential for complexity. Many CSs respond non-linearly to stimuli. Nonetheless, being subject to non-linear dynamics is not a necessary condition for a CS. For example, there are structures involving linear matrices that describe networks, and there are CSs subject to game-theoretic and quantum dynamics all of which are subject to linear dynamics.

Feedback [115] : Consider the dynamic interaction within a flock of birds. Each bird adjusts its flight based on the position and movement of its nearby companions. However, this adjustment then influences the behavior of these neighbors in return. Consequently, when the bird plans its subsequent move, the current state of its neighbors is partially a response to its own previous actions. This illustrates a continuous loop of action and reaction, showcasing the concept of feedback within the group.

Self-organization and emergence [32] : Emergence is the way in which complex structures and behaviors are born from simple elements interacting with one another [39]. The system may converge to different organizations with the emergence of various patterns and behaviors. Considering the flock of birds, its size and dynamics depend on the intrinsic specificities of the studied species.

Distributed control [141] : The order observed in the way a flock of birds stays together despite the individual and erratic motions of its members is stable in the

sense that the buffeting of the system by the wind or the random elimination of some of the members of the flock does not destroy it.

Hierarchical organisation [142] : The best example of such a system is an ecosystem or the whole system of life on Earth. Other systems that display such organization include individual organisms, the brain, the cells of complex organisms, and so on. A non-living example of such organization is the cosmos itself with its complex structure of atoms, molecules, gases, liquids, chemical kinds, and geological kinds, and ultimately stars and galaxies, and clusters and superclusters.

Numerosity [120] : Many more than a handful of individual elements need to interact to generate CSs. CSs often include heterogeneous elements in a very large quantity. There is no clear-cut threshold for this property.

The great variety of CS observed legitimizes the abstraction and an interdisciplinary approach [118]. This approach seeks to describe the characteristics of a system while accounting for the intertwining and entanglement between the elements. The systemic analysis consists of describing the different parts of the system as subsystems and describing the interaction of these subsystems with other subsystems at different time and space scales. These interactions are often feedback loops.

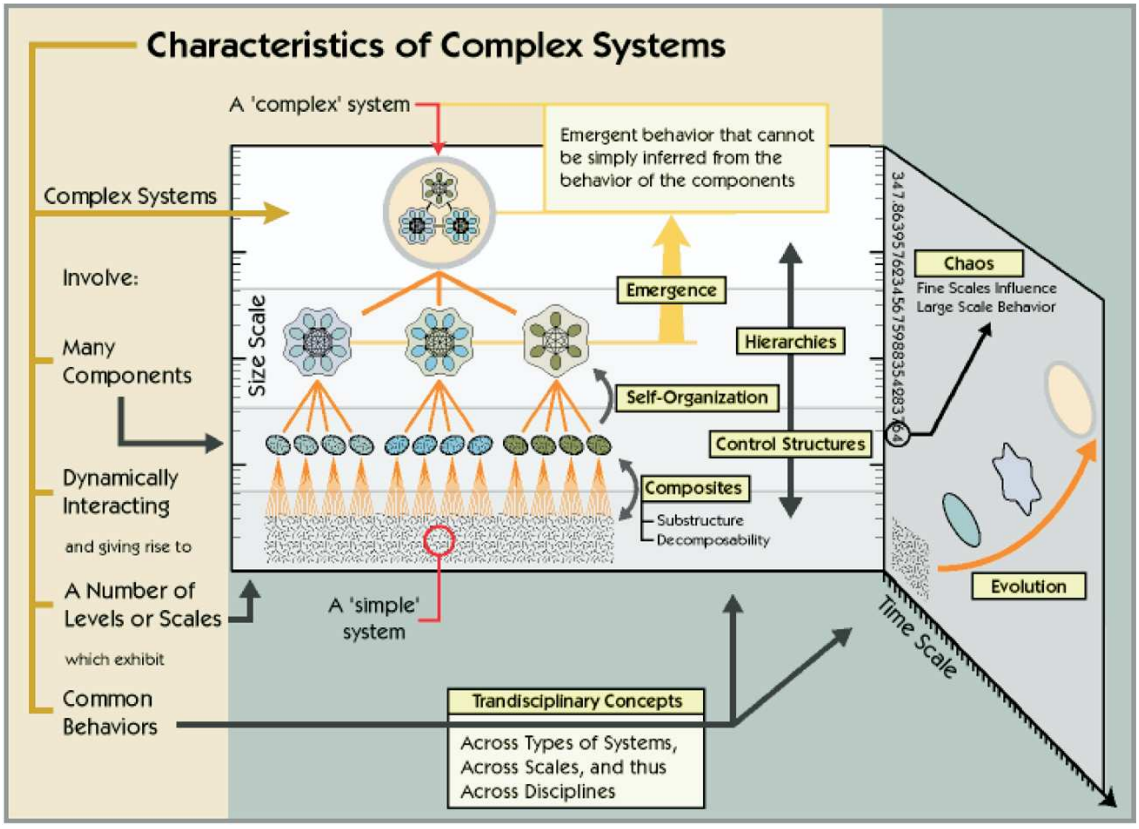


Figure 1.1 – The complex adaptive systems models characteristic of complex adaptive systems, a model by Marshall Clemens [31].

1.2 Modeling of Complex Systems

To describe the systemic analysis, we must go into greater detail on the structure in space and the evolution in time of CSs. We will refer to the diagram of Figure 1.1 from Marshall Clemens' works [31].

The systemic analysis allows us to highlight the various obstacles the system faces and to provide a modeling methodology.

This methodology takes into account the three factors of a CS discussed in Section 1.1 : Multiplicity of space and time scales, Heterogeneity of the components, and emergence dynamics.

Distributed intelligence : A CS is composed of interacting entities whose feedback, behavior, or evolution cannot be determined via a direct (immediate and macroscopic) calculation. The entities must have a reactive or cognitive behavior of their own that allows them to adapt to their environment and to learn the impact of their actions on it.

Self-organization and evolution : The interacting entities have specific and common characteristics. The association of entities forms subsystems with their specific and common characteristics. Its subsystems can again form new larger subsystems until the system is formed in its entirety. Entities and subsystems are subject to internal constraints due to the entities and external constraints from other entities or the environment. The constraints and interactions trigger an evolution of the entities' characteristics in time and space at any scale of the system.

Attractors : It is easy to imagine CSs as a living organism subject to internal and external constraints at different scales, and varying in time and space. It should be noted that these systems are subject to Chaos Theory, therefore they have stable states (attractors) and transition states in time and space. A CS tends to converge to a stable state if it is in a transition state.

A CS is characterized by the fact that the behavior of the system cannot be deduced from that of its components. Therefore, a CS must be modeled and simulated adequately to observe the emergence of behaviors. The simulations allow us to subsequently rectify the modeling.

Let us now define the concept of CSTS and give an overview of some of the most important CSTSs in the energy ecosystem. We will define a CSTS as follows :

Définition 1.3. Complex Systems that are subject to the interactive influences of socio-organizational and technical factors are Complex Socio-Technical System.

They include the influences related to human activities and the ones related to their organization. We can mention the organizational structure of the workplace, the formal and informal relationships of command and control, or the number and types of functions of the employees [64]. The technical tools and systems that support a work-related activity, as well as the processes and techniques used to perform the work, are called technical factors.

1.3 Related work

Our research is focused on the domain of building energy efficiency and energy usage. To provide a context for this area of study, we will briefly outline the key concepts and current trends in this field. The discussion in this section draws upon comprehensive surveys, including the state-of-the-art review by Belussi et al. [19], which offers an insightful overview of recent advancements and methodologies in energy efficiency.

1.3.1 Modeling of Building

We define buildings as any structure intended to provide shelter or insulation. We chose such a broad definition because we want to encompass the wide range of building types that are present in our case study. Yoshino et al. [160] identify six socio-technical factors that influence energy consumption in buildings.

Définition 1.4. Factors influencing energy consumption in buildings are climate, building envelope and other physical characteristics, equipment, indoor environmental conditions, operations and maintenance, and user behavior [160].

Buildings are CSTS in part because of user behaviors. The way in which the inhabitants of a building respond to technology in the context of energy efficiency is particularly difficult to model and remains a scientific challenge [157]. However, certain characteristics of building users can allow us to better predict consumption. For example, household income has been determined to be correlated to energy use [121].

1.3.2 Grey box modeling

The methods to study one building cannot all be applied to a great number of buildings[78]. Conventional mechanistic or heuristic approaches quickly reach their limits when generalizing or extending to a larger scale. More statistical approaches, such as Machine Learning (ML) techniques, are better suited to reveal typical behaviors on large sets of buildings. White Box, Black Box, or Grey Box model can be used to model the buildings [98]

White Box modeling, also known as forward modeling, employs physics-based equations to simulate the behavior of building components, sub-systems, and entire systems. This approach aims to accurately predict aspects like energy consumption and indoor comfort within buildings, focusing on detailed, component-level interactions. Due to the detailed dynamic equations in White Box models, they have the potential to capture precisely the building dynamics, but they are time-consuming to develop and to solve [99]. Even though these elaborate simulation tools are effective and accurate, they require detailed information and parameters of buildings, energy systems, and outside weather conditions. These parameters, however, are always difficult to obtain, and even sometimes are not available. What's even more challenging, creating these White Box models normally requires expert work, and the calculation is extremely time-consuming, which is the major

barrier for White Box building models to be used in on-line model-based control and operation [99].

Black Box models, on the other hand, are simple to build and demand less computation. They are therefore very relevant to issues regarding a large park of buildings. However, such models often require long training periods and a large forecasting range for the model to encompass all the possible situations. The development of IoT makes this last condition less and less constraining [98].

However, in many situations, we wish to extract understandable knowledge from our model rather than just predictions. Hence the need for Grey Box Models. Grey Box Models are hybrid models that use simplified physical descriptions to simulate the behavior of the systems. Using the simplified physical models reduces the requirement of training data sets and calculation time. Model coefficients are identified based on the operation data using statistics or parameter identification methods.

In order to model a building in our context, it is necessary to build a Grey Box Model.

1.3.3 Modeling of Smart Energy Systems

We will now present some energy systems that are transformed by or born from the convergence of global environmental challenges, data massification, and innovation in energy and information technologies. Here, the notion of hierarchical organization of CSs presented in Section 1.1 is important. Indeed, the presented systems are of different scales and are in interaction with each other, some entirely encompassing others.

To determine the most relevant approach to model smart energy systems, we found in the currently published literature the following concepts :

- The **Smart Grid (SG)** [6] : an intelligent, robust, and flexible energy grid that includes communication between each element of the grid. It integrates various technologies like advanced metering infrastructure, renewable energy sources, and energy-efficient resources. SGs play a crucial role in modernizing the energy system, enhancing energy efficiency, and ensuring sustainable energy management.
- The **Internet of Energy (IoE)**, or **Energy Internet** [75] : refers to an advanced networking infrastructure that facilitates the integration and management of distributed energy resources. It considers energy management as a network packet management, similar to the internet process. IoE encompasses the convergence of energy and information technology, enabling enhanced control, efficiency, and reliability in energy distribution and consumption. It supports the dynamic balancing of energy supply and demand, advanced energy analytics, and the integration of renewable energy sources.
- **Corporate Real Estate Management Systems** [63] : this SG model offers advice on issues such as sustainability, workplace productivity, real estate performance measurement, change management, and customer focus.
- **Bottom-up Building Stock Models** [84] : dynamic bottom-up simulation tool designed to assess the impact of economic and regulatory incentives on buildings' energy demands, energy carrier mix, CO2 reductions, and as-

sociated costs related to heating, cooling, hot water, and lighting. It allows for the simulation of various scenarios, including changes in energy prices, renovation packages, and consumer behaviors, to project future energy demand and the mix between renewable and conventional energy sources at both national and regional levels.

- **Multi-agent Based Building Energy Management Systems** [83] : Optimize behavior for energy performance and comfort optimization using a Multi Agent model of the users and of the devices of the building. Necessitates a detailed models and comprehensive sensors and actuators.
- The **Trusted Third Party for Energy Measurement and Performances (TTPEMP)** [23, 92] is an actor in the IoE. Born from the need to certify the energy savings achieved within the framework of Energy Performance Contracts, the TTPEMP aims to be neutral in the process of implementing energy-saving solutions. This neutrality allows it to position itself as a trusted third party. It validates or questions the measurements made by energy producers or the parties who have deployed energy reduction solutions. The TTPEMP bridges the gap between major energy producers, typical consumer-producers, and service providers implementing energy performance improvement solutions. It constitutes an external actor considered reliable and having no interests either in the sale of energy resources (or similar) or in the sale of renovation or installation contracts (or similar).

Property	Complex Systems	Classic Systems
Multiplicity of scales	Operate across multiple space and time scales.	Typically operate within a single scale or have limited scale variation.
Heterogeneity	Composed of heterogeneous components.	Components are usually homogeneous or less varied.
Emergence	Exhibit emergent properties through sudden and unpredictable phase transitions.	Lack of emergent properties; system behavior is directly related to component behavior.
Irreducibility	Cannot be simplified without losing the intelligibility of the system.	Can be subdivided into simpler problems using traditional analytical methods.
Nonlinearity	Respond non-linearly to stimuli; not all complex systems are subject to nonlinear dynamics.	Responses and behaviors are often linear or predictable.
Feedback	Dynamic interactions and feedback loops are prevalent.	Feedback loops, if present, are simpler and more predictable.
Distributed control	Control is distributed; no single element dictates the behavior of the system.	Control is often centralized or hierarchical with clear control elements.
Hierarchical organization	Display hierarchical organization, but with complex interdependencies.	May display hierarchical organization, but with simpler and more linear relationships.
Numerosity	Composed of a large number of interacting elements.	The number of elements is limited or interactions are simpler.
Distributed intelligence	Entities exhibit autonomous behavior, adapting and learning from their environment.	Behavior is often centrally controlled or directed, with limited autonomy or learning capacity in entities.
Self-organization and evolution	Entities and subsystems evolve over time, self-organizing into new structures in response to internal and external constraints.	Tend to maintain initial organization and structure over time, with minimal evolution or self-organization.
Attractors and Chaos Theory	Systems have stable and transition states, influenced by Chaos Theory, converging to stable states from transitions.	Typically do not exhibit behavior influenced by Chaos Theory, with more predictable and stable states.

Table 1.1 – Comparison between Complex Systems and Classic Systems

Table 1.2 – Comparison of Smart Energy System Approaches

System	Scale	Key Concepts	Technology Used	Input / Output	Real-time actuation
SG	Grid-wide	Advanced metering, Renewable integration, Energy efficiency	Advanced Energy Grid Infrastructure interfacing with IoE	Energy production, transportation, storage, consumption	Yes
IoE	International	Network packet management, Enhanced control, Renewable integration	IoT infrastructure for Energy	All information regarding energy	Yes
Corporate Real Estate Management Systems	Building / Corporate level	Sustainability, Workplace productivity, Real estate performance	Real estate management software	User behavior	No
Bottom-up Building Stock Models	National to international	Energy efficiency, Consumer behavior	Modeling tools, Bottom Up models, Economic simulation	Energy policy, incentives and recommendations	No
Multi-agent Based Building Energy Management Systems	Building	Digital twins	Multi-agent systems models	User and systems behaviors	Yes
TTPEMP	Building	Energy savings recommendation and validation	Data analytics, Decision Support, Machine Learning	User behaviors, Buildings renovations	No

By studying all the systems cited above we realize that they are all encompassed in one larger system : the IoE. There is no agreed upon definition of IoE, however it is often described as the continuation and the extension of the SG [161, 162, 76]. Therefore, IoE is sometimes also called the SG 2.0. Indeed the IoE is wider than the SG. For instance, in most definitions the SG is restrained to electricity whereas the IoE is not. It is also described as the IoT principles applied to the energy world [76, 66, 69].

IoE is a concept first explored systematically in *The third industrial revolution : how lateral power is transforming energy, the economy, and the world* published in 2011 [134]. However, there is still no consensus on the definition of IoE [152]. Rifkin present IoE as a new system of energy use that, through the Internet, integrates renewable energy, distributed generation, hydrogen, energy storage technologies and electric vehicles.

1.3.4 Monitoring of Smart Energy Systems

The monitoring of SG and microgrid can be a source of inspiration for our case study. Most monitoring systems include Internet of Things (IoT) and an analysis platform. We present the main subjects of these subsystems :

- **Building energy management systems (BEMSs)** are used to monitor and control a building's energy requirements. They can lead to a net energy building also known as zero-energy building through the management of devices, local production and batteries [61].
- **Microgrids** are Demand-Response systems : a combination of sensors monitoring with forecasting and peer-to-peer exchange are used in real time to manage the production and consumption. Blockchain can be relevant in this situation [100]. The goal of these system is to manage the demand in energy according to the production. They can add a dashboard where the customer may change their needs or behaviors and can be extended to multiple microgrids to enhance the flexibility of the grid. The tools used for the monitoring of microgrids are often cloud based tools used for IoT. As the open source data visualization software Grafana [67, 53], or the cloud-based IoT analytics software *Thingspeak* [146]. These microgrids can be isolated from the grid such as NRLab [43] or isolated smart houses such as Al Summarmad's model [146], or connected to the network [53]. Microgrids are based on Wireless Sensor Networks [102] which can be monitored using Binary Logistic Regression.
- Some models act as **digital twins** of real systems. Public transport such as trains or a fleet of electric vehicles need to be monitored according to customers and the Demand-Response system. These systems are huge consumers but include various strategies to manage and smooth their impact to the grid. For example, Crotti et al. [36] present a monitoring system with braking systems, reversible substations and on-board storage systems to limit the impact of a railway system . Khan and Wang propose a multi-agent simulation, as a digital double, to monitor and schedule a microgrid with electric vehicles aggregators [81] ;
- A **Smart Grid Architecture Model (SGAM)** is a conceptual framework for SG design and deployment. It was proposed by the European Committee

for Electro-technical Standardization and the European Telecommunications Standards Institute [29]to manage and pilot smart energy systems. They include the Demand-Response system and can simulate large scale energy networks such as a whole island in the H2020 MAESHA project [148] or the InteGrid project [129]. We refer to the following article for an in-depth review of the existing SGAMs [123].

Our system refers to the BEMs. As services, our system aims to propose various strategies to enhance the use of energy in a building according to its specificity such as building renovation, devices organization, human-centered management, integration of artificial intelligence for Demand-Response, etc. Our model must include a feedback system to understand how a strategy impacts a building and how the strategy could impact another building as a prediction.

1.4 The Trusted Third Party For Energy Measurement And Performances (TTPEMP) as a Complex Socio-Technical System

In this section, we will present our case study, a TTPEMP, we will explain why and how it can benefit from the complex approach.

The TTPEMP acts as a link between large energy producers, typical consumer-producers, and providers of energy efficiency solutions [92].

Définition 1.5. The TTPEMP is an external actor that is considered trustworthy and has no interest in the sale of energy resources (or similar) or in the sale of retrofit or installation contracts (or similar). However, the TTPEMP acts as a social and economic actor in the evolution of the system [23, 92].

We also introduce the TTPEMP that provided funding for this work, including its specificities and needs. This TTPEMP is named *Energisme* [45]. *Energisme* is a company whose field of activity is historically the measurement of energy performance. It describes itself as a new actor of the IoE [45].

TTPEMPs model buildings for several reasons :

- Energy management, and energy performance policy ;
- Personnel management and activities policy ;
- Security policy ;
- Partnership with other organizations (subcontracting, etc.).

The aspect of energy management and performance policy are adressed. While the TTPEMP functions as a monitoring system, its primary purpose is to facilitate decision-making, we will therefore focus on his nature as a DSS. To meet these challenges, we implement an Information Technology (IT) solution that not only models but also simulates building operations, effectively functioning as a comprehensive virtual platform. It ensures the collection and monitoring of data as well as the supervision of the simulation of the elements as follows :

- The diversity of the types of sensors allows the measurement of all the energy consumptions of the buildings (electricity, water, gas, heating and cooling networks, etc.) ;

- Energy billing data ;
- Meteorological data and other data whose sources are external to the buildings studied ;
- Meta-data of the buildings such as their location, the buildings materials, the plan of the buildings ;
- Buildings' usage such as their usage category, peak and off-peak times or frequency, metadata provided by personnel management or Customer Relationship Management software ;
- Operations and maintenance that have been performed on the buildings ;
- Relationships maintained by the buildings installations, for example the relationship between the consumption of the air conditioning system and the sensors of room temperature, external temperature and human frequentation of the building.

The study of the energy performance of the buildings opens the possibility of extracting the common characteristics of groups of buildings. This notion of groups of buildings will be used for :

- The study of building stocks over time and space ;
- The study of the similarities between buildings grouped by the same geographical location or by the same typologies of usage or energy consumption ;
- The establishment of a profile map allowing the extrapolation of new buildings without establishing a systemic modeling of these buildings. For example, it is possible for an entity to be managing only a part of a building, such as a story. In this case, it is necessary to estimate the missing parts using a similar known system to be able to estimate the evolution of the consumption and production of the whole building.

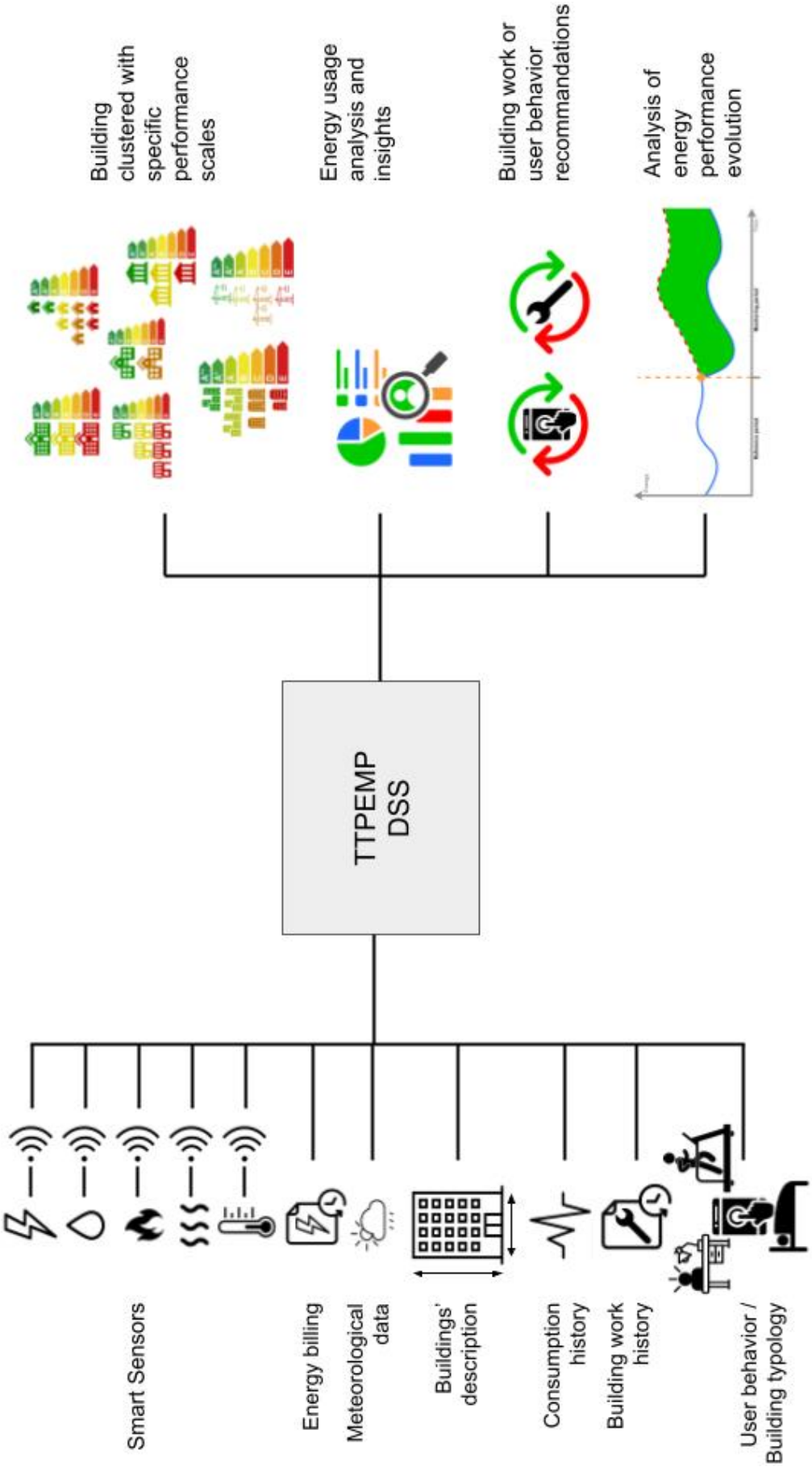


Figure 1.2 – The TTPEMP inputs and outputs

All these data describe the material reality of a building at a « human » scale. The TTPEMP uses these data to put intelligence into the studied system, abstracting level by level the different components of its environment to form the systemic modeling of the problem.

The TTPEMP has to face the complexity of developing a virtual platform for intelligent energy management supervision. The platform *e2-Diagnoser* introduced by [128] is a good example of a possible architecture but does not sufficiently dive in the needs of the new IoE actor that is TTPEMP. The virtual platform must highlight the complex and socio-technical aspects of the studied system, taking into consideration the socio-technical factors to answer the issues related to the maintainability and scalability of the platform. They must also take into account the TTPEMP actors who will maintain the information system. These are the developers, operators and managers who are in charge of development and production. In the context of our complex approach, it seems relevant to consider the interaction between these people and the supervision platform.

By broadening the context, it becomes clear that the supervision system produced for intelligent energy management is itself a CSTS composed of the platform and the people working to maintain it. In this context, the solutions that would be applied to guarantee the robustness and stability of the supervision system should integrate the hazards resulting from human factors. These problems can be partly solved thanks to a theoretical approach and an adapted design presented in the chapter 2, especially in section 2.1. In the next section, we present related works on the modeling and monitoring of building and smart energy systems and on microservices systems.

1.5 Modeling the TTPEMP

1.5.1 Hybrid Modeling Approach

In this thesis, we aim to define a modeling approach tailored for our case study. To achieve this, we adopt an approach akin to that employed by Bosom in their doctoral work. Our research is a continuation and further development of the investigative field initiated by Bosom, building upon and expanding their foundational concepts.

Bosom [23] presents three different approaches for the modeling of intelligent energy systems in the TTPEMP context. The analytical approach focuses on the elements that make up the system but faces issues regarding biases arising from the complexity of this system. The systemic approach consists of studying the system in a bottom-up way with appropriate theoretical tools. In this specific context, each actor pursues their own goal. Using systemic approach, one should identifies three generic subsystems from a regulated and identifiable active phenomenon : operating, information, and decision subsystems.

The third approach is an hybrid of the analytical and systemic ones. It makes it possible to combine the benefits of the two methods, one operational, consisting in abstracting the components identifiable thanks to the data, the other systemic, arranged in the subsystems participating in the complexity of the TTPEMP projects. We therefore adapt a hybrid approach for the same reasons. We therefore present the specificities of each approach useful for our work.

1.5.1.1 Analytical Approach

The analytical approach is also referred to as data-driven in the literature [10]. This approach focuses on functions and seeks to identify the most influential parameters. Thus, the input data of the system is seen as a source of values. It is based on a data processing pipeline that collects, cleans, formats, and automatically or semi-automatically extracts significant characteristics. The latter are then exploited thanks to data mining techniques, ML methods, knowledge extractions, and discovery methods. The intelligence obtained by this process must be associated with an adequate representation and visualization system to give meaning to the studied system.

The analytical approach is subject to biases that prevent us from considering it as the sole approach for the modeling of a CSTS [132, 40] :

- It is very effective when the interactions are linear but it is less suitable for the study of systems that reach a certain level of complexity, uncertainty, or emergent logic;
- The causes of dysfunction are often wrongly attributed to human factors rather than the more diffuse influences of socio-technical factors [23];
- It only favors the use of knowledge already available and thus limits the search for alternative solutions.

Before presenting the hybrid approach, let us examine the contributions of a systemic approach.

1.5.1.2 Systemic Approach

The systemic theory provides tools for modeling any CS. Indeed, contrary to the previous approach, each actor pursues its own goal. Therefore, the system cannot be summarized with a single purpose. It must be seen as a list of subsystems and associated actors.

For a CS, subsystems must be identified and articulated. Each subsystem is seen as a functional level of the system, i.e., a stable intermediate system that we call a project [89]. Each of these projects is represented in a relatively autonomous way by its network and can be increasingly complex (via modeling). By iteration between projects and their representation, we seek to compose or aggregate these levels by taking into account possible feedback.

The decision system can be partitioned more finely to highlight its ability to coordinate the many decisions about actions, develop and evaluate new strategies, implement the chosen strategy through the operating system. When the subsystems of a CS are identified, we can articulate projects and actions in networks with possible levels as a graph, multi-graph, or more generally a lattice. However, determining the subsystems is a complicated task that requires expertise in the CS and a modeling and simulation system to continuously improve the system to get closer to the reality of the CS.

1.5.1.3 Hybrid Approach

While maintaining an analytical approach, we decide to also adopt a systemic approach to organize the application of data-driven methods. In other words, our modeling allows us to find a compromise between two approaches, one highligh-

ting the subsystems involved in the complexity of the considered projects, the other, more operational, consisting in abstracting the identifiable components using the data. The aim of the model is to be able to take into account emergent phenomena, interactions, and non-productive modifications brought about in particular by human behavior.

In the context of building energy efficiency, the TTPEMP faces many constraints and objectives that coexist and cannot all be satisfied (competing multi-objectives). The model we build must, therefore, allow us to sort and prioritize them to build a supervision policy with measurable effects. Modeling these influencing factors and, more generally, the buildings supervised by the system, requires the exploitation of the data made available by the actors. Moreover, to identify *leverage points*, i.e., the places where the energy actor can intervene on the system, it is also necessary to rely on the data.

Our model is itself structured as a CSTS, taking into account both the objectives of the model, but also the actors having to use the model.

Table 1.3 – Comparison of Analytical, Systemic, and Hybrid Approaches

Approach	Description
Analytical	Focuses on functions and the most influential parameters, employing a data processing pipeline for collecting, cleaning, formatting, and extracting significant characteristics. It is effective for linear interactions but less suitable for complex, uncertain systems or emergent logic.
Systemic	Utilizes a bottom-up analysis with theoretical tools to view the system as a network of subsystems and actors, each with their own goals. It emphasizes the importance of identifying and articulating subsystems, considering feedback, and dealing with the complexity through a holistic perspective.
Hybrid	Combines the operational efficiency of the analytical approach with the comprehensive insight of the systemic approach. It allows for the abstraction of components identifiable through data and arranges them in subsystems to address the complexity of projects, accommodating emergent phenomena and interactions.

1.5.2 Hybrid Approach Applied to TTPEMP

The first step consists of building an abstract model thanks to the systemic approach to make the proof of concept and to integrate all the actors. Then, the data-driven approach enables the analysis of the existing data structures and thus the validation and completion of the designed model. Each of the elements is structured in the form of a graph. It contains the relations between the subsystems or the relation to a data source operated by an actor. These two approaches enrich one another. This process of mutual enrichment constitutes a feedback loop allowing the models to converge towards a common and stable representation.

The modeling of the data model (in GraphQL) allows the mutual validation of the hybrid approach. The methods of Development and Operations (DevOps) help to set up iterative design and evolution. They also meet a strong need for scalability and contribute to the resilience of the platform. They will be presented in Section 2.2.1. The GraphQL model highlights the APIs required to meet the needs of data insertions or transformations and exchanges between agents or links with the outside world. Most of the time, a microservice will respond to an API.

The organization of the concurrent execution of these numerous services raises the question of the orchestration of services which arises for the distributed computing. It is taken into account by a Microservices Architecture (MSA) for implementation. MSAs reduce the interdependence of services and thus facilitate the integration of new IT tools (new database technologies, infrastructure changes, etc.). The development of these new implementations then allows to meet three main categories of functionality :

- The ones expressed by the TTPEMP (see Section 1.4) which include, notably, business algorithms for the prediction of energy behaviors and IT tools for monitoring, alerting, and personalization ;
- The ones raised by the modeling, to respond to the APIs and that are declined in microservices ;
- The ones born of the need for coherence and supervision of the IT platform itself, especially orchestration.

The hybrid modeling approach presented highlighted an iterative method for the evolution of our system when facing new elements, making the platform resilient.

1.6 Decision Support System for the TTPEMP

1.6.1 Supervision System

Bosom [23] highlights the necessity for the TTPEMP to have an effective supervision system. Supervision, as commonly understood in the context of system management, refers to the ongoing monitoring and control of a system's operations [11, 30]. Our focus, however, is on the decision support subsystem, which is crucial for providing stakeholders with information tailored to their specific needs for understanding. Such a system is commonly known in the literature as a DSS.

In the context of the TTPEMP, distinguishing between a supervision system and a DSS is crucial, as they serve different but complementary functions in the management of energy systems. A supervision system is essentially the cornerstone of operational management in energy systems. It is designed to continuously monitor and oversee the entire energy system, ensuring that all components operate within their specified parameters and performance thresholds. This system is responsible for real-time surveillance, control, and immediate response to any operational anomalies or emergencies. It ensures the system's stability and reliability by constantly checking for deviations, faults, or inefficiencies and can trigger automatic corrective actions or alert human operators to intervene. In the TTPEMP framework, the supervision system plays a critical role in maintaining the integrity and smooth functioning of operations, focusing predominantly on the operational and technical aspects.

1.6.2 Decision Support Systems

On the other hand, a DSS in the TTPEMP setup serves a higher-level strategic function. Unlike the supervision system, which focuses on immediate operational issues, the DSS is designed to assist in making informed, data-driven decisions. It integrates and analyzes complex datasets collected from various sources, including the supervision system, to provide actionable insights and recommendations. A DSS aids stakeholders in interpreting data, understanding trends, evaluating potential impacts of different decisions, and planning future strategies. In the realm of energy management, this could involve optimizing energy distribution, forecasting future energy needs, or developing new energy-saving initiatives. The DSS is more about strategic analysis and planning, providing a broader perspective to guide decision-making processes.

The concept of Decision Support Systems has evolved significantly over time. Early references in the 70s [56] lay the groundwork for understanding what DSS is.

Définition 1.6. *A Decision Support System can be defined as an interactive computer based systems, which help decision makers utilize data and models to solve unstructured problems* [144].

Subsequent works, like Power's definition [130], further refine this understanding by emphasizing DSS as systems aiding in making decisions based on accessible data, analytical tools, and models. In the literature, a DSS is widely accepted as a computerized system that aids in decision-making processes. It is a knowledge-based system that provides comprehensive information and tools for analysis to improve the quality of decisions [79]. A DSS typically includes an interactive software-based system that compiles useful information from a combination of raw data, documents, personal knowledge, or business models to identify and solve problems and make decisions.

The integration of a DSS within the TTPEMP aligns seamlessly with the hybrid approach discussed in Chapter 2. This approach combines systemic thinking with data-driven methods, where the DSS plays a pivotal role in synthesizing information from various subsystems and data sources. The ability of the DSS to process complex data sets and provide actionable insights is crucial in the context of energy management and performance monitoring. The hybrid approach enhances the DSS's capabilities by allowing for a more nuanced understanding of both the systemic and data-specific aspects of energy management. This synergy between the systemic and analytical components within the DSS framework ensures a more holistic and effective decision-making process in the context of TTPEMP.

The DSS for the TTPEMP, as presented in Chapter 2, represents a crucial component in the overarching management and operational strategy of the TTPEMP. By leveraging both systemic and data-driven approaches, the DSS offers a robust platform for informed decision-making, essential for the effective and efficient functioning of the TTPEMP.

The DSS allows decision makers to navigate situation in which the amount of information, their complexity and the number of possible choices makes deci-

sion making too difficult a task. They offer comprehensive data analysis, leading to more informed and effective decision-making, they reduce the time taken to gather and process information. DSS also must present Flexibility by adapting to the changing needs of the organization. However they also present issues such as Complexity and Cost in their development and maintenance. They can lead to over reliance and lack of critical thinking by the users. Finally, they can be misleading if they are fed with poor quality data. These points will be addressed in Chapter 2 and 5.

1.6.3 Recommender System

The study of recommender systems can be an inspiration for the design of certain aspects of our DSS. Recommendation systems are primarily used to recommend products or content to online consumers. The goal of these systems is to assist in selecting the product most likely to be chosen based on the characteristics of the consumer and/or based on choices made by similar users.

Recommendation systems are broadly classified into the following categories based on the underlying technique used for making recommendations [4, 73, 124] :

- Collaborative filtering : It assesses the relevance of an item for a user based on the opinion of members of a community (or cluster) [117]. It is based on the principle that users with similar interests tend to prefer similar items.
- Content-based filtering : Such systems are developed on the principle that items with similar characteristics will be evaluated similarly by users [156]. That is, they recommend items similar to those liked by the user in the past.
- Knowledge-based recommendation systems : Such a system recommends products based on specific domain knowledge about how certain item features satisfy the needs and specifications of users [133].
- Hybrid recommendation systems : Any combination of the above techniques can be classified as a hybrid recommendation system [73].

Similar methods can be applied by the TTPEMP, making a recommendation for a specific type of energy performance action is similar, as we can also make a recommendation based on knowledge of the building and/or based on recommendations made for similar buildings.

However, the choice of an energy performance decision is much more significant and has a far greater impact than choosing which advertisement to display on a web page or which video to suggest on a streaming platform. This situation is more akin to decision-making in a commercial, medical, or judicial environment, where the decision process is aided by a computer, but where human governance is necessary. Hence our choice of ultimately designing a DSS.

1.7 Explainable Artificial Intelligence and clustering for the TTPEMP

1.7.1 Introduction to Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) represents a fundamental shift in the paradigm of AI systems, particularly in their application to complex domains like energy systems [14]. At its core, XAI aims to make the decisions and functioning of AI models transparent and understandable to human users. This shift is driven by the increasing complexity and ubiquity of AI systems, where decisions made by black-box models can have significant and far-reaching consequences [54]. XAI is not merely a technical necessity but also an ethical imperative, ensuring that AI systems are accountable, fair, and align with human values [15]. As AI systems become more involved in critical decision-making processes in energy management, the ability to interpret and trust their outputs becomes paramount. This necessity births various XAI techniques, each striving to peel back the layers of complex AI algorithms, making them more interpretable to users and stakeholders [14].

The relevance of XAI in energy systems is further underscored by the sector's increasing reliance on AI for managing complex tasks such as demand forecasting, grid optimization, and renewable energy integration [14]. Traditional black-box models, while efficient, often lack the transparency needed for stakeholders to fully trust and understand their decision-making processes [131].

1.7.2 Explainable Artificial Intelligence for the TTPEMP's Decision Support System

The integration of XAI into DSS represents a significant advancement in energy management. This synergy is critical in making complex AI-driven decisions transparent, understandable, and actionable for stakeholders involved in energy systems. DSS, which are crucial in aiding decision-making through data analysis and model-based insights, can greatly benefit from XAI's ability to elucidate the inner workings of AI models. In the context of energy systems, where decisions have far-reaching impacts, the clarity provided by XAI is not just a value addition ; it's a necessity for informed decision-making.

XAI enhances the functionality of DSS in energy management by providing a layer of interpretability over complex AI algorithms. This transparency is essential for stakeholders to trust the recommendations made by the DSS. For example, in scenarios like energy load forecasting or optimization of energy distribution, stakeholders can make more informed decisions if they understand the rationale behind the AI's predictions or recommendations. XAI techniques, such as feature importance analysis or model-agnostic methods, can be incorporated into DSS to provide clear explanations of AI outputs. These explanations enable energy managers and decision-makers to comprehend the factors driving AI decisions, thereby fostering a higher degree of confidence and trust in the system.

Furthermore, the implementation of XAI within DSS aligns with regulatory and ethical standards, ensuring that AI-based decisions in energy management are both accountable and transparent. As energy systems increasingly rely on AI for critical operations, the demand for governance and compliance with regulatory

frameworks grows. XAI-driven DSS meet these demands by offering not only high-performance analytics but also ensuring that these analytics are understandable and justifiable. This aspect is particularly pertinent in scenarios involving stakeholder engagement, policy implementation, and strategic planning in energy management. Thus, the marriage of XAI and DSS in the energy sector paves the way for more responsible, efficient, and transparent decision-making processes, enhancing the overall efficacy and reliability of energy management systems.

1.7.3 Pretopology in Clustering Mixed Data

Pretopology offers a novel and robust approach to clustering in AI, particularly within the domain of complex energy systems. As a generalization of classical topology, pretopology extends beyond the limitations of traditional methods in handling diverse data types, a common characteristic in energy systems. This flexibility is particularly relevant when dealing with mixed data, encompassing time series, numerical, and categorical variables.[96] Pretopology excels in providing a hierarchical understanding of such data structures, crucial for dissecting and interpreting the multifaceted relationships within energy systems. The hierarchical clustering facilitated by pretopology aligns well with the layered nature of energy systems, from individual consumer behaviors to large-scale grid dynamics. This alignment enables a more intuitive and insightful grouping of data points, enhancing the interpretability and applicability of clustering results in practical energy management scenarios.

A pivotal aspect of pretopological space definition which we will use in the context of XAI is the use of Disjunctive Normal Form (DNF) [85]. DNF in pretopology explicitly outlines the logical rules underpinning the construction of clusters, thus contributing significantly to the explainability of the clustering process. This is particularly advantageous in energy systems, where understanding the 'why' behind groupings can be as critical as the groupings themselves. [91]. For instance, in load forecasting or anomaly detection in energy consumption patterns, knowing the logical rules that lead to certain groupings can provide insights into underlying causes or potential areas of intervention. Moreover, the DNF representation aligns with human cognitive processes, making the explanations generated by such models more accessible and actionable for decision-makers and stakeholders in the energy sector.

This first chapter laid the foundational groundwork for understanding the multifaceted nature of CSTSs, with a special focus on energy systems. We started by defining Complex Systems, unraveling their characteristics such as multiplicity of space and time scales, heterogeneity of components, and emergent properties. These definitions set the stage for deeper discussions on the interactions and behaviors that typify Complex Systems.

Through various models and theoretical discussions, we explored how Complex Systems manifests in the energy sector, particularly within smart energy systems and their associated stakeholders. We examined how these systems neces-

sitate a blend of technological, organizational, and human components, and how their interplay requires an approach that is both holistic and nuanced.

We presented our use case, a new actor in the energy domain : the TTPEMP. We have identified one of its role as a DSS.

The chapter highlighted the challenges inherent in designing and implementing DSS for these complex environments. It underscored the need for a deep understanding of the systems' properties to ensure that solutions are not only technically sound but also socially acceptable and practically implementable. The discussions emphasized that the long-term viability of these systems depends on their adaptability to changing conditions and sensitivity to diverse stakeholder needs.

We also analyzed various modeling approaches that are instrumental in capturing the essence of Complex Systems, providing insights into their dynamics and guiding the development of effective strategies for managing them. We presented a hybrid model for the energy system of our case study, which is both practical in a technical environment and considerate of the complex properties of Socio-Technical Systems.

Finally we presented XAI as a central concept for DSS allowing users to grasp the mechanisms leading to a recommendation, prediction or analysis. We then introduced the application of pretopology in clustering mixed data, highlighting its potential to provide a more intuitive understanding of data structures in energy systems. This approach promises to improve the interpretability of clustering outcomes, facilitating more informed decision-making in energy management practices.

Summary of Chapter 1

Complex Systems : Complex systems are characterized by features such as multiplicity of scales, heterogeneity, emergence, irreducibility, non-linearity, feedback, self - organization, distributed control, hierarchical organization, and numerosity. These systems require a holistic approach for understanding, focusing on how various subsystems interact. Complex Socio-Technical System (CSTS) are especially noted for their blend of socio-organizational and technical factors, including human activities and organizational structures, indicating the intricate relationships that define their operations.

Modeling of Building : In the realm of energy consumption, buildings are recognized as CSTS due to the significant impact of socio-technical factors. The modeling of buildings, particularly through the application of Grey Box models, exemplifies the need for a method that merges physical and statistical models. This approach facilitates a deeper analysis of energy dynamics within buildings, accounting for the complex interplay between various influencing factors.

Modeling of Smart Energy Systems : A comprehensive review of smart energy systems highlights the necessity for innovative modeling techniques. This review brings to the forefront key concepts such as Smart Grids, the Internet of Energy (IoE), Corporate Real Estate Management Systems, and Multi-

agent Systems. Within this ecosystem, the Trusted Third Party for Energy Measurement and Performances (TTPEMP) emerges as a pivotal mediator, ensuring trustworthiness in energy performance verification and recommendation by connecting consumers, and efficiency solution providers.

Monitoring of Smart Energy Systems : The importance of real-time management and decision support is underscored through the integration of IoT and analysis platforms. Systems such as Building Energy Management Systems (BEMSs), microgrids, digital twins, and Smart Grid Architecture Models (SGAM) are surveyed for their contributions to optimizing energy management. Feedback systems, in particular, are highlighted for their crucial role in analyzing the impact of various energy management strategies.

TTPEMP : Our case study is identified as a TTPEMP, recognized as a complex socio-technical system that operates at the intersection of social, organizational, and technical dimensions. We focus on its natureas a Decision Support System (DSS). Indeed the TTPEMP aims to facilitate decision-making through a comprehensive suite of IT solutions, encompassing data collection and systematic modeling. However, the development of an architecture for such an intricate system presents significant challenges, including the integration of human factors to ensure robustness and system stability.

Modeling the TTPEMP : We explores a hybrid modeling approach for the TTPEMP, integrating both analytical and systemic methodologies. The analytical approach focuses on data-driven functions and parameters, while the systemic approach views the system through its subsystems and actors, emphasizing a holistic perspective. The hybrid model combines these strategies to address complex systems' challenges, accommodating emergent phenomena and interactions, particularly within the realm of building energy efficiency. It aims to create a resilient, scalable model that incorporates data analysis, subsystem identification, and actor integration, facilitated by a feedback loop for continuous refinement and adaptation.

DSS : Designing a relevant DSS architecture is pivotal for strategic functionality of the TTPEMP, integrating and analyzing complex datasets to aid in informed decision-making. The incorporation of DSS enhances the capabilities of energy management, leveraging both systemic and data-driven approaches for a holistic decision-making process. These systems face challenges in development complexity and must remain flexible to adapt to the evolving needs of the organization.

Explainable Artificial Intelligence (XAI) : The paradigm shift towards XAI emphasizes the need for transparency in AI decisions, ensuring these decisions are accountable and aligned with human values. XAI's relevance to energy systems is particularly critical for tasks such as demand forecasting and grid optimization, where understanding the basis of AI-driven decisions is paramount.

Pretopology in Clustering Mixed Data : Pretopology introduces a novel approach for handling mixed data types within AI, offering a hierarchical understanding of data structures. Utilizing Disjunctive Normal Form (DNF) within pretopology significantly enhances the explainability of clustering processes. This method proves beneficial for energy systems by providing actionable in-

sights into underlying causes or potential interventions, thereby facilitating informed decision-making.

Chapter 2

Governance and Architecture of the Decision Support System

In Chapter 2, we delve into the intricacies of governance and architecture within the realm of Decision Support System (DSS), specifically tailored for managing Complex Socio-Technical System (CSTS) in the context of energy management.

We will analyze the theoretical underpinnings and practical methodologies essential for the effective governance of semi-automated DSS, addressing the pivotal balance between automation and human oversight in critical decision-making processes.

Through a comprehensive exploration of semi-automation, algorithm-in-the-loop, and human-in-the-loop frameworks, we attempt to sheds light on the ethical, reliable, and accurate deployment of automation within significant societal domains.

Furthermore, we will discuss the challenges and opportunities presented by distributed and big data architecture in DSSs. We will examine the strategies for handling and analyzing vast datasets to inform and improve decision-making processes. The importance of integrating various modeling techniques and adapting to the continuously evolving energy sector landscape are highlighted.

By providing insights into the development, implementation, and continual improvement of the DSS, a comprehensive understanding of the tools and strategies necessary for effective energy management in complex systems is given.

Finally we propose a DSS architecture building upon the state of the art. The overview of this DSS allows us to analyses its complex nature and to present different use cases.

2.1 Theory and Method for the Governance of Decision Support Systems

This section presents the methods and tools needed for the governance of a supervision system. The need for these methods and tools was made apparent by the systemic and analytical modeling of a CSTS of energy management.

2.1.1 Governance of Semi-Automated Decision Support Systems

2.1.1.1 State of the art

The idea of governance is closely related to the concept of monitoring but also to the concept of automation. Crucial elements of our society are being automatized in areas such as health, justice, and banking. Just as energy consumption management, these elements cannot be automatized without the need for an efficient and reliable governance so that the trust in these key institutions is maintained. In the past, decision-making was a social issue; today, it has become a socio-technical problem.

The first thing to note is that automation involving important decision making is rarely total. Most of the time it involves a contribution of an automated system and of a human. Many terms are employed to fill this gray area such as *semi-automation*, *quasi-automation*, *algorithm in the loop*, or *human in the loop* decision making. These terms will be discussed below.

In this case where the decision process is aided by a computer, but where human governance is necessary, we talk about « **algorithm-in-the-loop** » decision-making, or, in the case of a more automated process where human intervention is simply intended to prevent a significant and consequential error, we talk about « **human-in-the-loop** » decision-making. [59] These new **algorithm-in-the-loop** decision-making processes raise two questions - one normative, the other empirical - that must be resolved before machine learning is integrated into some of the most important decisions in society : (1) What criteria characterize an ethical and responsible decision when a person is informed by an algorithm ? (2) Do the ways in which people make decisions when informed by an algorithm meet these criteria ?

In their article « The Principles and Limits of Algorithm-in-the-Loop » Green and Chen [59] identify the following 3 desiderata :

- **Accuracy** : People using the algorithm should make more accurate predictions than they could without the algorithm.
- **Reliability** : Users should accurately assess their own performance and that of the algorithm and calibrate their use of the algorithm to account for its accuracy and errors.
- **Fairness** : People should interact with the algorithm impartially concerning race, gender, and other sensitive attributes.

The Trusted Third Party for Energy Measurement and Performances (TTPEMP), being involved in social issues such as assistance with energy poverty, all these questions may arise. The degree of automation of a system can also be assessed using the following 7 points, identified by Wagner [151] :

- The time available to the individual human operator relative to the task : the less time allocated to the human operator, the higher the probability of quasi-automation.
- The degree of qualification of the human operator of the system to perform the specific task : the less qualified the individual is to perform a specific task, the more likely it is to be quasi-automated.
- The degree of responsibility that will be attributed to the human operator in case of failure : the higher the amount of legal responsibility attributed to a human operator in case of failure, the more likely it is that humans

are engaged in the process only to ensure that they can assume responsibility in case of failure of the automated system (Dannenbaum, 2010 [37]; Maurino, Reason, Johnston, & Lee, 2017 [104]).
- Level of support that the individual receives to carry out the task sustainably : many involved tasks require very high degrees of concentration and often involve making very disturbing decisions in a short time. Here, higher levels of psychosocial support or other forms of support are likely to be an indicator that runs counter to quasi-automation.
- Adaptation : the more a human operator must adapt to the system, rather than the system being designed to serve the operator, the more the system is quasi-automated.
- Access to information : the human operator must have access to all relevant information in order to make the right decision.
- Agency : the human operator must have enough « authority [. . .] to modify the decision » (Article 29 Data Protection Working Party, 2017, p. 10) and must actually do so regularly. If the only function of the human operator is to regularly agree with the machine and is very rarely in disagreement with it, it is highly likely that the human operator's agency is insufficient.

If the TTPEMP remains in its role of decision aid, then it only recommends actions, one can then wonder to what extent the responsibility for the energy performance action rests entirely on the actor who used the recommendations. However, the role of the TTPEMP should be to enable actors to make informed decisions in order to recognize aberrant recommendations that could be made following an error or erroneous information, or simply to have the elements allowing them to make a choice themselves, perhaps depending on external factors unknown to the TTPEMP? If the TTPEMP wishes in the future to transform and become an actuator of energy performance, then it will have to take into account all the elements presented above.

2.1.1.2 European Regulation

Article 22 of the EU GDPR includes a « prohibition of fully automated individual decision-making, including profiling, which has a legal effect or a similarly significant effect » (Article 29 Data Protection Working Party, 2017, p. 9), as well as specific safeguards for fully automated decisions. Thus, by putting humans in the loop, companies can attempt to escape some of these limitations and safeguards. Moreover, many companies currently resort to human intervention in systems supposed to be fully automated - that is, they pretend that an advanced AI system transcribes your voicemail rather than a call center in the Philippines - without informing users[143].

This approach is summarized in Regulation (EU) No 376/2014 on the reporting, analysis, and follow-up of events in civil aviation, which calls for a « just culture » that is « *a culture in which frontline operators or other persons are not punished for actions, omissions, or decisions that they take which are proportionate to their experience and training, but in which gross negligence, willful violations, and destructive acts are not tolerated.* » This approach - also evident in several legal decisions made in the aviation sector, which focus on organizational responsibility [25] rather than individual responsibility - is increasingly common in judicial decisions [137], which focus on improving Socio-Technical Systems at scale rather than solely identifying

culprits. [151]

In this context, abiding to the recommendation described above is crucial. To do so, it is important to be able to monitor the system and to guarantee the accuracy and reliability of this CSTS. We will see in the next session how these issue relates to Development and Operations (DevOps)

2.2 Addressing Governance with Devops and Data Lineage Methodologies

2.2.1 DevOps Methodology

DevOps is a set of practices and cultural philosophies that aims to unify software development (Dev) and software operation (Ops). The primary goal of DevOps is to shorten the system development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives. This approach emphasizes the collaboration and communication of both software developers and IT professionals while automating the process of software delivery and infrastructure changes. It aims to establish an environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably.

2.2.1.1 Continuous Integration, Continuous Delivery, and Continuous Deployment

DevOps is strongly associated with Continuous Integration, delivery, and deployment [139]. We refer to the works of Shahin et al. [140] for a complete review. For a detail review, refer to Continuous Integration, Delivery and Deployment : A Systematic Review on Approaches, Tools, Challenges and Practice s[140].

Continuous integration (CI) is the practice of integrating code changes of a project frequently from multiple developers. This practice is well established in the software development industry. CI allows software companies to increase the frequency of the release cycle, the quality of software, and the productivity of their teams. This practice includes automated software building and testing. Automated tools are used to build then assert the new code's correctness before integration.

Continuous DElivery (CDE) ensures that an application is ready to go into production environment after passing automated testing and quality control. CDE automatically delivers software into a production environment using IC and deployment automation. This practice offers reduced deployment risk, lower costs and faster user feedback.

Continuous Deployment (CD) is the automatic release of approved changes from a developer to production, where they are usable by customers. It addresses the problem of overloading operations teams with manual processes that slow down application delivery. It builds on the benefits of continuous delivery by automating the next step in the pipeline.

2.2.1.2 Addressing accuracy and reliability

One of the primary reasons DevOps has gained prominence is its ability to address issues of **accuracy** and **reliability** in software development and deployment . By fostering a culture of continuous improvement and collaboration, DevOps practices help reduce human errors typically associated with manual processes. Automated testing and continuous integration enable early detection of errors, ensuring higher accuracy in the final product. Additionally, DevOps promotes the use of monitoring and logging practices throughout the software development lifecycle, enhancing the reliability of applications by allowing teams to quickly identify and respond to issues in real-time [90].

2.2.1.3 Adressing biases

Addressing the question of whether DevOps can solve issues of unwanted biases in DSS requires a nuanced understanding. DevOps itself primarily focuses on improving the efficiency and reliability of software development and operations. It streamlines processes, encourages frequent testing, and fosters a culture of continuous improvement and feedback, which can indirectly contribute to identifying and addressing biases in software systems.

However, the specific issue of unwanted biases in DSS is more closely related to the fields of data science and AI ethics rather than DevOps. Biases in DSS usually stem from biased data sources, flawed algorithms, or biases in the users or developpers. While DevOps practices can facilitate quicker iterations and responses to identified issues, including biases, it doesn't inherently provide tools or methodologies to detect or correct biases in data or algorithms.

Ensuring data quality and transparency in the dataflow through data lineage, as well as avoiding black box models through Explainable Artificial Intelligence (XAI) must be therefore be taken into account.

2.2.2 Data lineage

Data lineage refers to the lifecycle of data, including its origins, what happens to it, and where it moves over time. It creates a comprehensive view of the data flow through the entire system, from source to destination, including all the transformations it undergoes. Understanding data lineage is crucial for ensuring reliability, correctness, and fairness in automated decision-making systems.

In terms of reliability, data lineage provides a clear map of where data comes from and how it's processed. This transparency helps in identifying and correcting errors quickly, ensuring that the data used for decision-making is accurate and trustworthy. For example, if an automated decision system makes unexpected recommendations, tracing the data lineage can help pinpoint whether incorrect data input, a processing error, or a flawed transformation led to the issue.

Correctness is closely related to reliability but focuses more on the integrity of data throughout its journey. Data lineage ensures that data transformations and aggregations are performed correctly, maintaining the integrity of the data. This aspect is critical in automated decision systems, where the correctness of data directly impacts the validity of the decisions made. By having a detailed account of each step in the data's lifecycle, organizations can verify that the data meets

all the required standards and compliances before it's used in decision-making processes.

Fairness in automated decision systems is a complex issue, particularly with the growing concern over biases that can be embedded in algorithms. Data lineage can play a pivotal role in addressing fairness by providing visibility into the data sources, transformations, and decisions based on that data. With a clear understanding of the data's origin and how it's used, analysts can identify potential biases in the dataset or in the data processing stages. For instance, if a dataset predominantly includes data from a certain demographic, leading to biased decisions, data lineage can help trace back to this source issue. Moreover, by ensuring transparency in how data is processed, data lineage supports the implementation of fairness algorithms and the adjustment of processes to mitigate bias, thereby promoting equity in automated decisions.

2.3 Decision Support System architecture

Energy performance of building requires DSSs because of the complexity of the energy systems, the overwhelming amount of information, and the plurality of solutions (see 1.6). Indeed, the energy performance issues require to be able to apply different clustering, classification and prediction methods on buildings, which requires the training and inference of several models. We will address them in the management of data using the Data Lake and the Datamart, in the choice and training of Machine Learning (ML) algorithms using the Machine Learning Factory (ML-Factory), and we will present a global architecture of the DSS. The DSS for TTPEMP is presented in the Figure 2.2 and 2.1 . We will explain in the following subsections each part of the DSS.

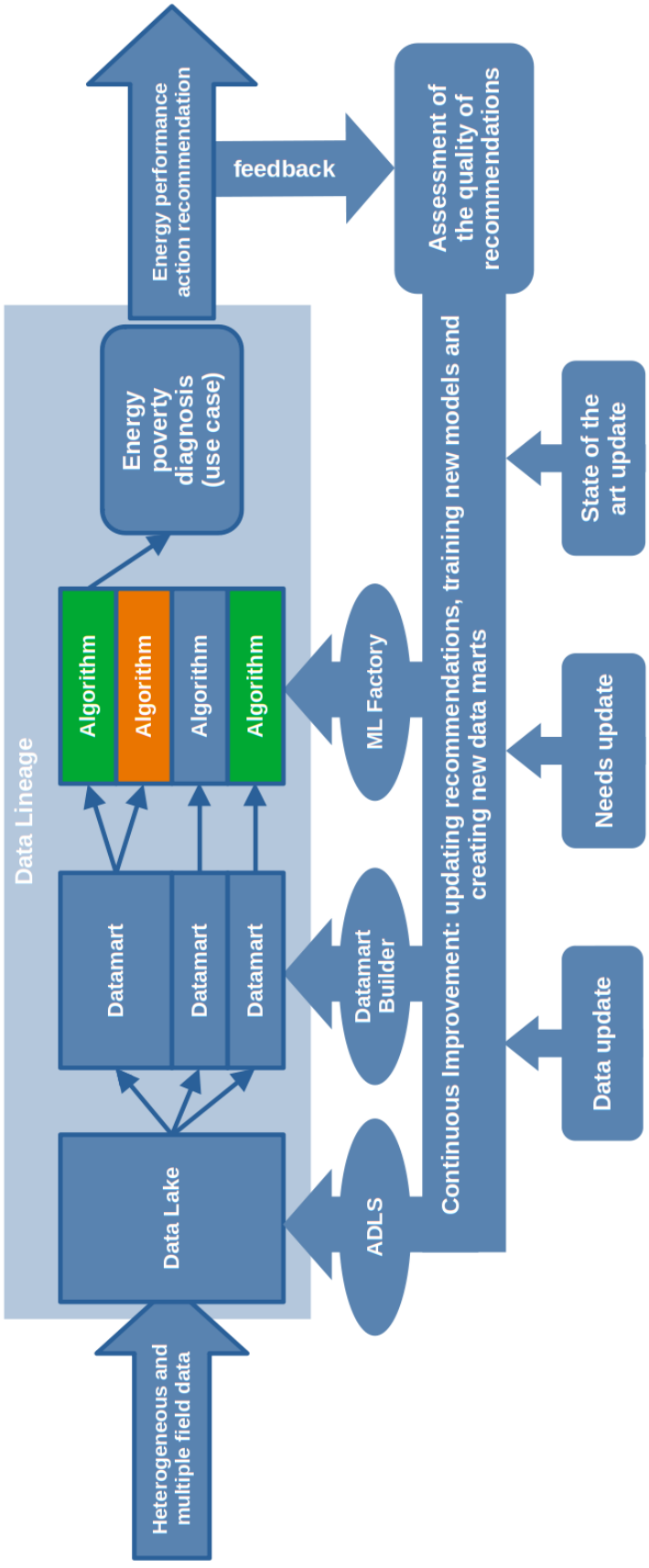


Figure 2.1 – The DSS conceptual architecture

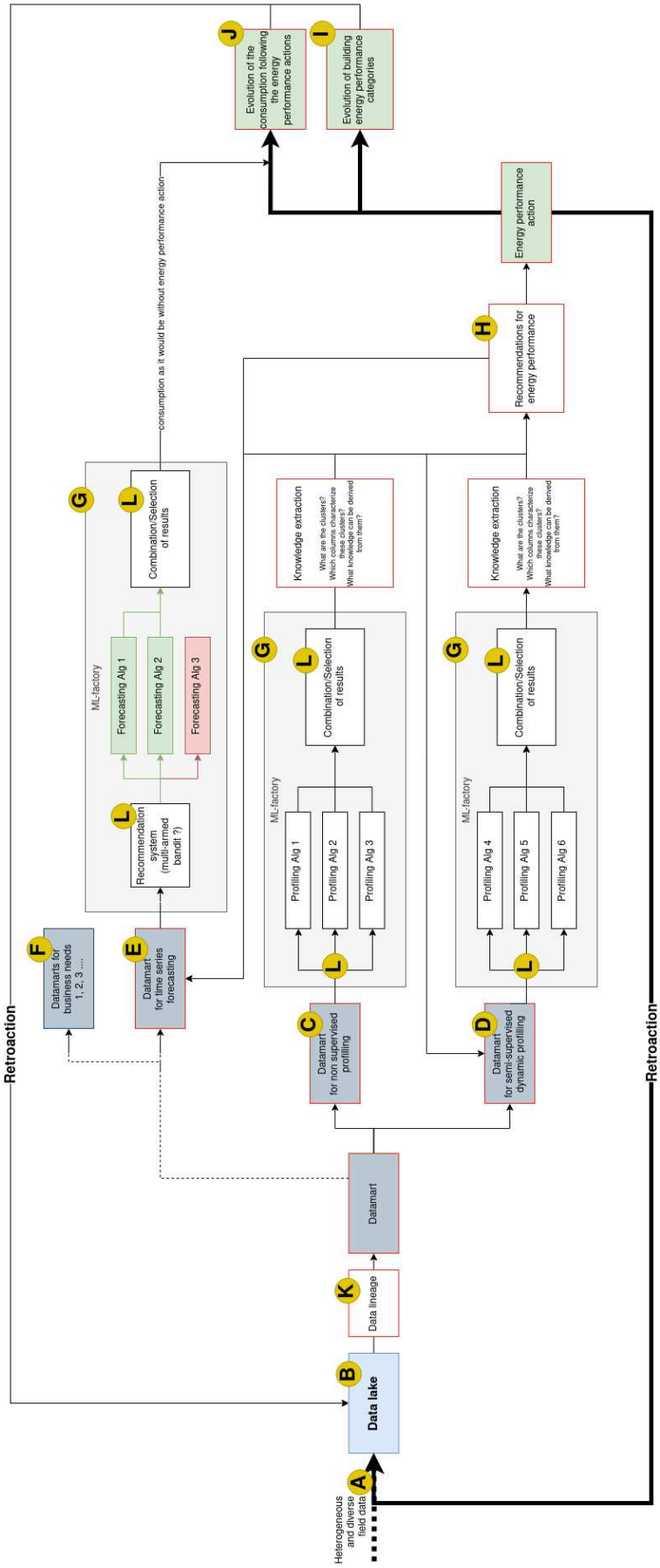


Figure 2.2 – The DSS global architecture.

2.3.1 Distributed Architecture

Distributed architecture refers to a system design paradigm that divides tasks among multiple components or services, which communicate over a network to achieve a common goal. This approach contrasts with traditional monolithic architectures, where all components of a system are tightly integrated and run as a single service. In the context of a DSS for governance of semi-automated systems, employing a distributed architecture offers several advantages that directly align with the objectives of correctness, reliability, and fairness.

Correctness : In distributed architectures, the separation of concerns allows for modular development, where each module can be designed, implemented, tested, and deployed independently. This modularity facilitates thorough verification of each component's functionality, contributing to the overall system's correctness. Additionally, updates or bug fixes can be rolled out to individual components without affecting the entire system, ensuring that the system remains in a correct state even as changes are made.

Reliability : Distributed systems are inherently designed for high availability and fault tolerance. By distributing tasks and data across multiple nodes, the system can continue to operate even if one or more nodes fail. This redundancy ensures that the DSS remains operational, providing consistent support for governance decisions. Moreover, distributed architectures can dynamically adjust to workload changes by scaling components independently, thereby maintaining performance and avoiding bottlenecks that could compromise decision-making processes.

Fairness : The decentralized nature of distributed architectures can enhance the fairness of a DSS by promoting transparency and accountability in decision-making. With data and processing logic spread across different nodes, it becomes easier to audit and trace how decisions are made, ensuring that no single component has undue influence over the outcomes. This setup supports implementing data lineage practices effectively, allowing stakeholders to understand how data is used and transformed within the DSS, thus promoting fairness and trust in the system.

Big Data : Incorporating distributed architecture is particularly advantageous in the context of big data, a domain characterized by the immense volume, velocity, and variety of data. Distributed systems are inherently designed to handle big data challenges by distributing the data storage and processing tasks across multiple nodes in the network. This parallel processing capability allows for the efficient handling of large datasets, ensuring that data can be processed quickly and accurately, which is crucial for the timely and correct functioning of a DSS. Moreover, distributed architectures facilitate scalability, enabling the system to accommodate growing data demands without significant reengineering. As the volume of data increases, additional nodes can be seamlessly integrated into the system to maintain performance levels. This scalability is essential for governance systems that may experience fluctuating data loads due to regulatory changes, system upgrades, or evolving operational requirements. Additionally, distributed systems can leverage advanced data storage solutions, such as distributed databases and data lakes, which are optimized for big data. These technologies support sophisticated data management practices, including data lineage, by efficiently organizing, indexing, and providing access to vast amounts of data across distributed

environments. Thus, a distributed architecture not only addresses the immediate needs of handling big data but also ensures the system's long-term adaptability and resilience.

Furthermore, distributed architectures support DevOps practices by enabling continuous integration and deployment (Continuous Integration/Continuous Deployment (CI/CD)) pipelines that streamline development, testing, and deployment processes. This integration ensures that the system can rapidly adapt to new requirements or changes in governance policies while maintaining high standards of quality and performance.

In summary, adopting a distributed architecture for a DSS in the governance of semi-automated systems presents a strategic approach to addressing the challenges of correctness, reliability, and fairness. It leverages modularity, redundancy, and decentralization to create a robust, adaptable, and transparent system capable of supporting complex governance tasks in a dynamic environment.

2.3.2 Big Data Processing

In this subsection, we specify the implemented solutions in order to ensure the performances during the data processing. The energy data to be processed falls into the category of Big Data.

Définition 2.1. The 5 V's of Big Data (velocity, volume, value, variety and veracity) are the five main and innate characteristics of Big Data.

A large amount of data must be processed (**Volume**). Indeed, it is necessary to process the power consumption history of several hundreds of thousands of buildings over several years with a time step often of the order of a minute. Moreover, the consumption data are updated day after day (**Velocity**). The energy data are also heterogeneous (**Variety**). Because of the multitude of sources, all buildings do not have the same descriptive data or the same consumption data (electricity, gas, water…), but even when the same values are described there is heterogeneity of formats (unit, time step, file type, indexing…). Some buildings do not even have meter collecting energy data and the only information available is accessible through PDF files of energy invoice. In addition to consumption data and physical description of the system, external data such as the meteorological consumption are also relevant. Some data describing for example the type of use of the building are entered by the customers and can present errors (**Veracity**). Of course, the data must also undergo numerous pre-processing operations in order to manage anomalies and to process outliers or missing values (**Variability**).

The architectural solutions proposed to deal with these issues are the Data Lake for the storage and integration of energy data, and the Datamarts for the processing and specific formatting of these data in order to respond to the different issues of energy actors.

2.3.2.1 Data Lake (see 2.2 : B)

Storing such massive and heterogeneous data, coming from different sources, requires an adapted storage repository.

Définition 2.2. A data lake is a centralized repository that provides massive storage of unstructured or raw data fed via multiple sources, the information has not yet been processed or prepared for analysis, there is no need to clean and process data before ingesting. [149]

However, the data stored in a data lake is not easily exploitable. And a data lake can easily turn into a *Garbage Dump* : a one way data lake in which a massive amount of unstructured data are stored but never exploited [72]. The data lake therefore needs to be used in conjunction with a more structured form of database called Datamarts.

2.3.2.2 Datamart (see 2.2 : C, D, E, F)

Datamarts (see figure 2.2 : C, D, E, F) The recommendation system relies on various methods (clustering, classification, prediction). Spatio-temporal data can be structured in different ways depending on the question being investigated, therefore the data marts C, D, E, and F in figure 2.2 will not extract the same data and will not format it in the same way [16].

In Kimball's vision of the data mart : the data warehouse is nothing more than the union of all the Datamarts [82]. Contrary to Kimball, Inmon considers that a data warehouse and a Datamart are physically separate. The data warehouse has its own physical existence and is oriented towards storage, traceability, and scalability in response to new requirements. Meanwhile, Datamarts have their own physical existence and offer a structure oriented towards the performance of data retrieval in response to user requirements. We are much closer to Inmon's conception of the Datamart. [159]

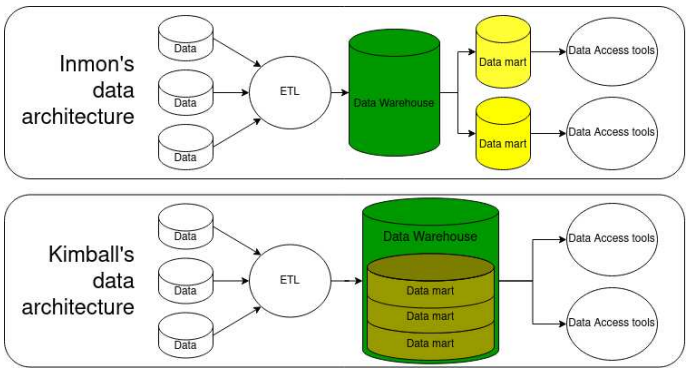


Figure 2.3 – Inmon vs Kimball architecture [159]

IBM defines a Datamart as a targeted version of a data warehouse that contains a more restricted subset of important and necessary data for a single team or a selected group of users within an organization. A Datamart is built from an existing data warehouse (or other data sources) through a complex procedure that involves multiple technologies and tools to design and construct a physical database, feed it with data, and set up complex access and management protocols. [1]

In our case, the main data source will be the data lake, but information such as national weather data could be integrated into the Datamarts without coming

1. https ://www.ibm.com/cloud/learn/data-mart

from the data lake. With its smaller, targeted design, a Datamart presents several benefits for the end user, including the following :

Cost-efficiency : There are many factors to consider when setting up a Datamart, such as scope, integrations, and the extraction, transformation, and loading process (ETL : which stands for extract, transform and load, is a data integration process that combines data from multiple data sources into a single, coherent data store that is loaded into a data warehouse or another target system). However, a Datamart generally incurs only a fraction of the cost of a data warehouse.

Simplified data access : Datamarts contain only a small subset of data, so users can quickly retrieve the data they need with less effort than they would when working with a larger data set from a data warehouse.

Faster access to information : Insights obtained from a data warehouse favor strategic decision-making at the enterprise level, impacting the entire business. A Datamart fuels business intelligence and analytics that guide decisions at the department level. Teams can exploit data targeted to their specific goals. As teams identify and extract valuable data in a shorter timeframe, the business benefits from accelerated business processes and increased productivity.

Simplified data maintenance : A data warehouse contains a multitude of business information, with scope for multiple business sectors. Datamarts focus on a single line, housing less than 100 GB, resulting in less clutter and easier maintenance.

Simpler and quicker implementation : Setting up a data warehouse involves significant implementation time, especially in a large company, as it collects data from a multitude of internal and external sources. On the other hand, a small subset of data is required when setting up a Datamart, so the implementation tends to be more efficient and involve less setup time.

2.3.3 Machine Learning Methods

Increasing energy efficiency of a system correspond to making that system consume less energy to produce the same amount of services or useful output [125]. Energy performance however is more difficult to define since energy performance indicators are multiple and complex and depend on a methodology which may vary on the national or even regional level [47]. Energy performance indicators that can be considered are not the same depending on the building type, hence the need for accurate estimates and reliable benchmarks for each type of building [22]. Therefore one of the first step in evaluating energy performances is to classify buildings based on their type. The type of a building corresponds to its main usage. Types of building can be identified by two approaches, supervised or unsupervised.

2.3.3.1 Unsupervised (Clustering/Profiling) (see 2.2 : C)

The first possibility is non supervised learning such as clustering methods.

Définition 2.3. Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each

other than to those in other groups.

In this process homogeneous groups of sites are identified based on their physical characteristics, their meteorological situation and their consumption.

Because the TTPEMP is bound to have a very high heterogeneity in the system it studies (is a hydraulic pump for public garden comparable to an office?) and the quality of the clustering made by its clients is never perfectly assured (see **Veracity** of Big Data). Because of this, being able to cluster the entirety of the clients into similar groups presents both a challenge and an opportunity to gain insight into the profiles and groups that we are studying. This might allow us to discover profiles of buildings sharing similar characteristics but that have not been classified together before.

After having identified the profiles, a knowledge extraction step is necessary in order to identify the **key indicators** that define the groups. This step is necessary to give meaning to the classification, and therefore to determine the energy performance actions which is the most relevant to each building type.

2.3.3.2 Supervised and Semi-Supervised (see 2.2 : D, E)

Contrary the non-supervised learning, the supervised learning consider data (or most of them) have already (or partially) a label/group. Through supervised learning, the objective is to classify sites into already established groups. It can be building usage, which has been identified to be a key element in evaluating energy performance of buildings [22], or it can be a site profile that has been identified using clustering methods.

2.3.3.3 Forecasting/Prediction (see 2.2 : E)

The prediction of consumption based on consumption history, building descriptive data, and building identified profile, can help evaluate energy performance evolution. Indeed, by comparing predicted energy consumption with measured energy consumption, one can detect improvement or deterioration of consumption of a building.

2.3.3.4 Model Selection in ML-Factories (see 2.2 : G, L)

Using the features extracted in the Datamarts, the system should be able to select one or several algorithms to apply to the dataset. For example in the case of forecasting, Feature-based FORecast Model Averaging (FFORMA) can outperform simple averaging methods in the forecasting of time series [114]. Similarly, several clustering methods can be combined to give a more relevant and robust clustering [158]. Finally, this process also perfectly works on deep-learning forecasting methods [147].

2.3.4 Machine Learning Factory (see 2.2 : G)

We want to extend the application of automation methods recommended by DevOps to ML algorithms. ML models require special processing because they must be trained on previously defined and labeled data sets. An ML-Factory is a model life cycle including ML methods.

Définition 2.4. We identify three main stages in an ML-Factory :
- A prototyping phase during which the developer explores the data, defines the format of the expected dataset and compares several candidate models ;
- A model training phase that can last several days and that is carried out, usually, on a larger dataset ;
- An inference phase, making the model callable, preferably with the best possible performance.

In others words, an ML-Factory, is an iterative process that covers iteratively : (1) the constitution of the training data ; (2) the identification of the ML algorithms ; (3) the training of the models and their hyper-parametrization ; (4) the recording of the trained models in an adequate library ; (5) the deployment on the chosen platform ; (6) the supervision and monitoring of the model in production ; (7) the iterative improvement of the models.

In the context of TTPEMP, it is preferable to rely on algorithms that will be trained from start to finish and to orchestrate the entire pipeline. Indeed, this control provides flexibility for each step and allows data scientists better introspection into the models they develop. An ML-Factory can include other tools such as a specialized database to allow data enrichment (*features*).

2.3.5 Recommendation, Feedback and Continuous Improvement

The goal of the ML-Factory is to determine profiles of consumers to determine which strategies can be used to improve their energy efficiency.

By applying the DevOps principle described in Section 2.2.1, the DSS must be able to identify critical points in the system. These points require human intervention to make sense of the algorithmic results and prevent aberrant responses. In addition to ensuring the quality of the recommendations, it is necessary to ensure the system's ability to adapt to changes in data and data sources, but also to integrate new methods of processing information in a fluid manner. Thus, the system will be resilient to the transformations of the studied building stock as well as to the integration of new methods of data processing. This requires integrating the DevOps methods described in Section 2.2.1 in the design, maintenance and operation of our system as follows :
- Take into account the feedback of its clients and partners : by evaluating the energy saving that have been made and improving the DSS accordingly (scoring algorithm based on the energy savings, using ML to improve the quality of the recommendation…) ;
- Thanks to the Microservices Architecture (MSA), adaptation of the information system to new technology can be relativelly seameless and easy.

Already present in Figure 2.1, the feedback loop which is so essential for governance and resilience can be found in Figure 2.2 representing the DSS global architecture.

With each element and their role clearly described let us now look at the DSS architecture as a whole. In accordance with the hybrid approach model (1.5.1.3, our system begins with the input of information from energy actors feeding us consumption and descriptive data (fig. 2.2 : A). By identifying consumption type

and learning from historical energy recommendations, the DSS must allow to identify leverage point for the client to undertake change in its consumption (whether it be behavioral or material) (fig. 2.2 : H. Following the Hybrid approach the feedback loop is central in the design of our system, allowing for constant update by integrating the feedback of the energy system on the DSS (fig. 2.2 : J). This is made possible by the methodology of DevOps, and by the tools presented above.

2.3.6 Overview of the Decision Support System

The energy data is multiple and heterogeneous. It consists of time series of consumption in electricity, but also in gas or water (A). These time series are not in the same format, do not have the same time steps or the same units. They are stored in their raw form in a Data Lake (B). In addition to this, there are descriptive data of the buildings as well as meteorological data, which are also uneven and heterogeneous. Moreover, some data will be missing or false. The data preprocessing will be carried out in different Datamarts. As the preprocessing is specific to the machine learning methods to be applied, several different Datamarts exist and will be specific to the various issues related to energy performance. In our system, we designed 3 « final » Datamarts. A Datamart for unsupervised profiling (C), a Datamart for semi-supervised profiling (D), and a Datamart for consumption prediction (E). (Other Datamarts concerning different business uses case can be used by the TTPEMP (F))

An important aspect of preprocessing for machine learning is feature extraction, these features are also specific to the methods to be applied. For profiling, the features used will be elements that allow distinguishing relevant consumption categories from the point of view of energy performance. For example, consumption during the night can give indications of superfluous energy consumption. Moreover, high consumptions at certain hours of the day or week can inform about the type of use of the building (housing, office, gym, commerce). In return, the type of use of the building can be a new feature added to the Datamart in order to make finer and more relevant diagnostics. (see figure 2.5 The building profiles that will be identified are not necessarily defined in advance, so it is a matter of unsupervised learning (C). The profiles identified by our algorithms will then be analyzed to extract specific diagnostics and recommendations (H) and help user evaluate their consumption using specific performance scales (see figure 2.4.

One of the objectives of our system is to follow the evolution of the performance of buildings and therefore to verify if they change consumption profile category. For this, it is necessary that these categories be fixed (one cannot follow the passage from one category to another if the categories themselves are movable). Placing an element in a set of predefined categories corresponds to semi-supervised learning. This is the second type of method that we will apply (D). This will allow us to detect evolutions in the consumption of buildings (change of type of activities, deterioration of the building) and to evaluate if the recommended energy performance actions have been effective (I).

Finally, another branch of the system will consist of comparing the predicted consumption of a building with its actual consumption. This will allow evaluating the energy savings resulting from the proposed energy performance actions (J). This will also identify abnormal consumptions of buildings. See figure 2.6

Figure 2.4 – Specific performance scales allowed by the clustering

In order to stay at the forefront of advances in Machine Learning, it is necessary to be able to use new methods and train new models easily, as well as to select and hybridize the most efficient models over time. This will be done using Energisme's ML-Factory and the development of complex inference graphs (G).

Finally, data lineage will be used to allow the system to adapt to the evolution of the data and context, as well as to ensure the quality and veracity of the data exploited.

2.3.7 Complex properties of the Decision Support System

After this overview of the DSS system, let us explore the complex properties of our DSS for energy management. Key features of the DSS—scalability, resiliency, feedback loops, distributed control, and heterogeneity (see 1.1)—mirror the dynamics of complex systems, enabling it to effectively navigate and adapt to the evolving energy sector. We delve into how these properties enhance the system's functionality, ensuring it remains robust and responsive to the challenges and opportunities in energy management.

Scalability (Distributed Architecture) : Our DSS leverages a distributed architecture to handle the multiple and heterogeneous nature of energy data. This design choice not only accommodates the vast volume of data from different sources but also allows the system to scale up or down efficiently as the data volume grows or the processing needs change. Scalability is essential in complex systems for managing dynamic environments without compromising performance.

Resiliency (Distributed Architecture and DevOps) : The combination of distributed architecture and DevOps practices enhances the system's resiliency. By distributing data and processing across multiple nodes, the system can maintain operations even if parts of it fail. DevOps practices further contribute to resiliency by enabling continuous integration, continuous deployment, and rapid response

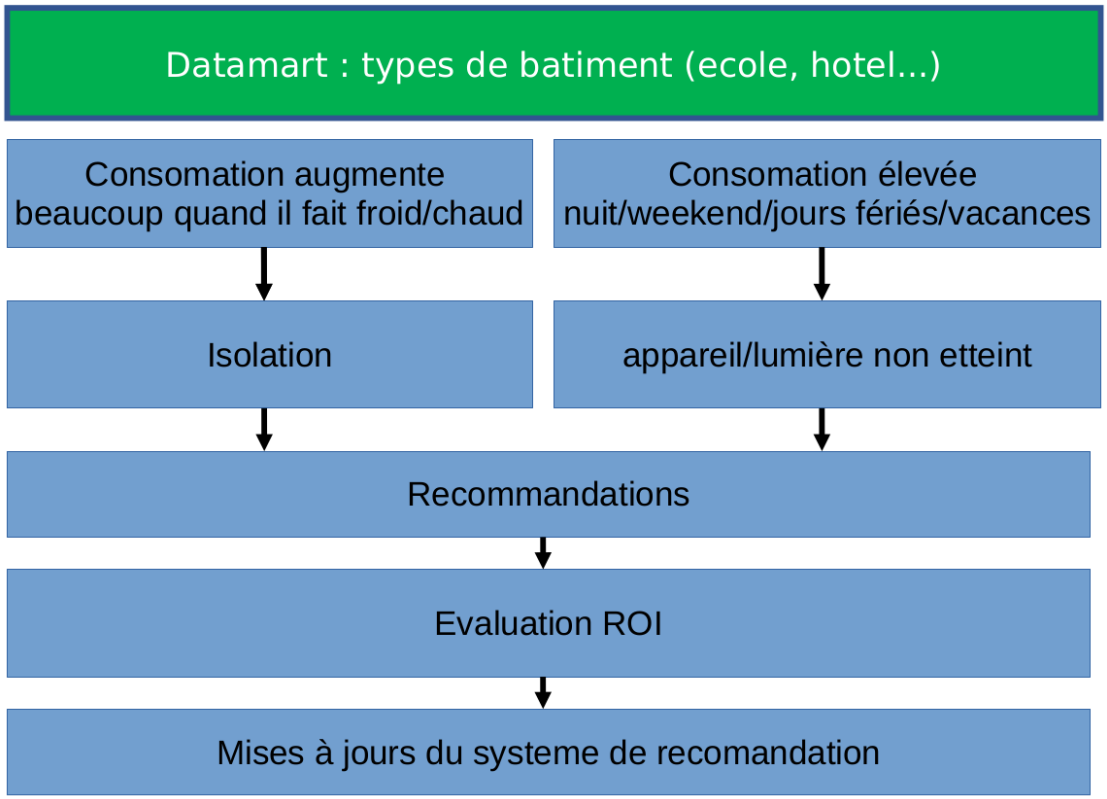


Figure 2.5 – Specific diagnosis allowed by the clustering

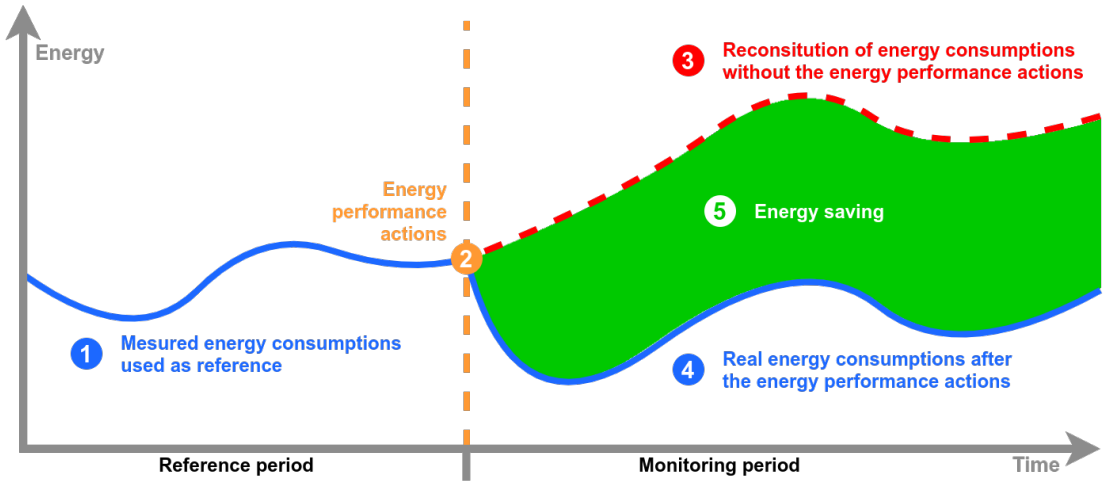


Figure 2.6 – Energy perfomance actions evaluation allowed by forecasting

to issues, ensuring the system remains robust and reliable in the face of changes and challenges.

Feedback Loops (DevOps) : DevOps facilitates a culture of continuous learning and improvement through iterative development and feedback loops. In our DSS, these feedback loops allow for the constant refinement of machine learning models and preprocessing techniques based on performance outcomes and evolving data characteristics. This adaptive approach is reflective of feedback mechanisms in complex systems, where the system evolves through interactions within itself and with its environment.

Distributed Control (Distributed Architecture) : The distributed nature of our DSS implies that control and decision-making are not centralized but rather spread across different components of the system. This distributed control mirrors complex systems where numerous agents or entities interact under local rules without a central command, leading to more flexible and adaptive behaviors.

Heterogeneity (Complex Architecture) : The inherent heterogeneity of components and actors within the DSS for energy management underscores its identity as a CSTS. This diversity spans various dimensions, including the multiplicity of data types—ranging from energy consumption metrics across different utilities to descriptive building data and variable meteorological conditions—and the spectrum of stakeholders involved, from system developers and data scientists to policy-makers and end-users. Each component and actor brings unique requirements and perspectives to the system, contributing to its complexity and necessitating a nuanced approach to its governance and operation.

The application of DevOps methodologies plays a crucial role in harmonizing these diverse elements, enabling the alignment of disparate actors towards the efficient functioning of the DSS. By fostering a culture of continuous integration, continuous deployment, and collaborative feedback, DevOps facilitates a dynamic environment where the evolving needs of various stakeholders are met with agility and precision. This collaborative approach ensures that technological developments are in step with the needs of the system's users and the objectives of energy management, thereby enhancing the system's overall coherence and effectiveness.

Self-organization and evolution : The DSS for complex energy systems exemplifies the principles of self-organization and evolution, crucial for navigating the rapidly changing technological, environmental, economic, and political landscapes that challenge entities like the TTPEMP Energisme. This adaptability is not merely a feature but a core aspect of the system's design, allowing it to dynamically adjust to new data types, emerging consumption patterns, and evolving regulatory frameworks. By leveraging data lineage, the system ensures the integrity and quality of data are preserved as the data landscape transforms.

Additionally, the integration of Energisme's ML-Factory into our DSS facilitates the continuous adoption and optimization of novel machine learning methodologies. This capability for self-organization extends the system's evolution, enabling it to autonomously refine its performance and strategies in response to external changes. Such evolutionary adaptability guarantees that the DSS not only maintains its effectiveness amidst shifts in the energy sector but also empowers the

TTPEMP Energisme to proactively tackle forthcoming challenges and seize new opportunities. This self-organizing and evolutionary property ensures the system's sustained relevance and operational excellence in a dynamic environment.

Emergence? : This inherent capacity for self-organization and evolution within the DSS means that the architecture itself is inherently mutable, subject to changes and evolutions in ways that are unpredictable. Consequently, the specifics of the current architecture may eventually become obsolete, as the system continuously adapts to new challenges and technologies in a manner that we cannot precisely foresee.

In conclusion, the DSS for energy management is a quintessential example of a complex system in action, embodying key characteristics such as scalability, resiliency, feedback loops, distributed control, heterogeneity, self-organization, and the potential for emergent behavior. These properties are not merely incidental; they are integral to the system's design, enabling it to effectively manage and navigate the complexities of energy data and its governance. The DSS's distributed architecture and DevOps practices underpin its adaptability and robustness, ensuring that it can scale and evolve in response to the rapidly changing landscape of energy management. By embracing the principles of complex systems, the DSS is poised to offer sustainable, adaptable, and innovative solutions to the challenges faced by entities like the TTPEMP Energisme, demonstrating the profound impact of applying complex systems theory to practical, real-world problems in energy management.

In this chapter, we explored the governance and architecture of DSS within the context of managing complex socio-technical systems, with a specific focus on energy management. The discussion underscored the importance of balancing automation with human oversight to ensure ethical, reliable, and accurate decision-making in critical societal domains. Through the lens of semi-automation, algorithm-in-the-loop, and human-in-the-loop frameworks, we addressed the challenges and opportunities these systems present in achieving fair and effective governance.

The adoption of DevOps methodologies and data lineage principles was highlighted as essential for maintaining the accuracy, reliability, and fairness of DSS. These approaches not only facilitate rapid adaptation to new requirements but also enhance the system's ability to manage and process big data effectively. We discussed the significance of distributed architectures in supporting scalability, reliability, and transparency in decision-making processes.

Through the introduction of a proposed DSS architecture, we provided a detailed overview of how such a system can be structured to handle the complexity of energy data and decision-making processes. This architecture incorporates Big Data processing techniques, machine learning methods, and a ML-Factory to ensure continuous improvement and adaptation of models and recommendations.

The chapter concluded by emphasizing the complex properties of the DSS, including scalability, resiliency, feedback loops, distributed control, and heterogeneity. These characteristics are indicative of a complex socio-technical system

capable of self-organization and evolution, making it well-equipped to address the dynamic challenges in energy management. By applying complex systems theory to the practical issues of energy management, the DSS demonstrates its potential to offer sustainable, adaptable, and innovative solutions.

In essence, the governance and architecture of DSS for energy management represent a confluence of technology, ethics, and policy, requiring a nuanced understanding of automation, human oversight, and data processing. The strategies and methodologies discussed in this chapter provide a roadmap for developing DSS that are not only technologically advanced but also ethically sound and socially responsible.

Summary of Chapter 2

Governance of Semi-Automated DSSs : Governance in semi-automated DSS integrates monitoring and automation, essential in high stakes sectors like health, justice, and banking, to maintain trust in automated decisions. The blend of automated systems and human intervention, known as « algorithm-in-the-loop » or « human-in-the-loop » decision-making, ensures ethical, responsible decisions are made with algorithmic assistance.

Criteria for Ethical Decision-Making : Ethical algorithm-informed decision-making requires accuracy, reliability, and fairness, with systems providing better predictions than humans alone, users accurately assessing performance and errors, and impartial interaction regardless of sensitive attributes.

Assessment of Automation Degree : The degree of automation can be evaluated based on time allocation for human operators, their qualification level, the legal responsibility in case of failure, support for sustainable task performance, system adaptation by the operator, access to relevant information, and the operator's agency to modify decisions.

European Regulation on Automated Decision-Making : EU GDPR Article 22 restricts fully automated decision-making with significant effects, advocating for human intervention to ensure decisions adhere to ethical standards and « just culture » principles, promoting a system where frontline operators aren't penalized for actions on which they had little responsibility but where gross negligence and willful violations are not tolerated.

DevOps Methodology : Emphasizes the unification of software development and operations, aiming to shorten development cycles and enhance reliability through continuous integration, delivery, and deployment. Automated processes and collaborative practices are key to its success, addressing issues of accuracy and reliability in software deployment.

Addressing Accuracy, Reliability, and Biases : DevOps promotes a culture of continuous improvement to minimize human errors and improve software quality. While it can indirectly contribute to identifying biases in DSS through

frequent testing and iteration, addressing biases directly requires specific data science and AI ethics considerations.

Data Lineage for DSSs : Critical for ensuring the reliability, correctness, and fairness of automated decision-making systems by providing a comprehensive view of data's journey and transformations. Data lineage helps identify and correct errors, maintain data integrity, and address potential biases by ensuring transparency in data processing.

Distributed Architecture : Facilitates scalability, reliability, and fairness, with modular development, high availability, fault tolerance, and transparent decision-making, suitable for managing big data and supporting DevOps practices for rapid adaptation.

Big Data Processing : Utilizes Data Lake for raw data storage and Datamarts for structured data processing, addressing the 5 V's of Big Data—velocity, volume, value, variety, and veracity—through specific, efficient storage and processing solutions.

Machine Learning Methods : Employs unsupervised clustering for building profiling, supervised and semi-supervised learning for categorization, and forecasting for performance evaluation, utilizing a ML-Factory for continuous improvement and adaptation of models.

ML-Factory Integration : Adopts DevOps principles to extend automation to ML algorithms, ensuring flexibility, iterative improvement, and adaptability in processing and analysis, supporting continuous development and deployment.

Recommendation, Feedback, and Continuous Improvement : Focuses on system resilience and adaptability to changes in data and methodologies, incorporating feedback mechanisms to refine recommendations and enhance energy efficiency strategies.

Architecture of the DSS The proposed DSS detailed architecture can address the governance and big data issues while maintaining a high level of automation and leveraging the state of the art Machine Learning methods for Decision Support in energy managment.

Complex System Properties : The DSS architecture demonstrates scalability, resiliency, feedback loops, distributed control, and heterogeneity, indicative of a complex socio-technical system capable of self-organization and evolution, prepared to address the dynamic challenges in energy management.

Chapter 3

Clustering Complex Systems

This chapter delves into the advanced algorithms for the clustering of complex systems, with a particular focus on mixed data sets. It introduces the first necessary family of methods employed by the Decision Support System (DSS) detailed in Chapter 2. Our objective is to unpack and explore a variety of clustering techniques, including partitional, hierarchical, model-based methods, and the innovative pretopology-based clustering, each meticulously tailored to navigate the intricacies of mixed data.

A pivotal aspect of our discussion centers on Dimensionality Reduction (DR) techniques. These techniques play a critical role in preprocessing mixed data, simplifying its complexity to facilitate more effective clustering.

Moreover, we introduce pretopology-based clustering as a novel mixed hierarchical clustering approach. This method stands out for its high degree of customization and parametrization, offering substantial interpretability and applicability across various domains.

The evaluation of clustering outcomes is another focal point, where we delve into metrics indispensable for assessing the quality of clusters in terms of cohesion, separation, and overall structure, enabling the validation of clustering methodologies.

We introduce Complex Clustering as the clustering of dataset comprising numerical, categorical and time series data. We present innovative solution to cluster and evaluate these data, one of which is the use of pretopology.

This chapter not only bridges the gap to understanding the sophisticated methodologies enabling effective clustering of complex datasets but also highlights their potential utility in energy systems and various other complex fields. Furthermore, it provides a solid foundation for analysing the datasets and results discussed in the following chapter.

3.1 State of the Art

Research in the domain of mixed data clustering has mainly focused on modifying existing clustering algorithms designed for either numerical or categorical datasets to perform well on mixed data. From the survey [8], we can distinguish four main types of clustering on which these works are based : *partitional clustering*, *hierarchical clustering*, *model-based clustering* and *neural-network based clustering*.

Partitional clustering works by dividing the dataset into a set of disjoint clusters and evaluating the partition according to a defined cost function. Each cluster is defined by a centroid created so that for each cluster, the distance between the data points in this cluster and its centroid is minimum compared to the other clusters' centroids. Datapoints are iteratively relocated between clusters until an optimal partition is reached, minimizing the cost function. The latter is generally the summation of the distance between each datapoint of the dataset and the cluster centroid nearest to it.

Hierarchical clustering aims to group the data into a hierarchy of clusters. It can be divided into two main types : agglomerative and divisive. In the agglomerative approach, we start by considering each datapoint as a cluster of its own. Then, the algorithm calculates the similarity of each cluster with all the other clusters by computing the similarity matrix. After this, it merges the nearest pairs of clusters into one cluster. These two steps are repeated until only one cluster is remaining, forming the hierarchy tree of clusters. The divisive approach is the opposite, starting with one cluster containing all datapoints and recursively performing splits as we move down the hierarchy. In both approaches, the observations of any number of clusters can be selected by cutting the hierarchy at the appropriate level.

Model based clustering is an approach in which we consider that a data object matches a model, which in many cases, is a statistical distribution. From the data objects we try to recover this original model which defines the clusters and the assignment of data objects to these clusters.

One of the most well-known models in model-based clustering is the *finite mixture model* (FMM). In this approach, we assume that the dataset was generated from a finite mixture of probability distributions, and aim to partition the dataset into G different clusters. An FMM with G components is a probability distribution in which the probability density function is a weighted summation of G distributions. Model-based clustering is an estimation problem that attempts to estimate the value of G and the best partitioning based on the data.

Let $X = (X_1, ..., X_M)$ be a M-dimensional random variable. X represents the different features values that a data object can have in a given dataset. If we denote $x = (x_1, ..., x_M)$ as one particular outcome of X, x is a possible data object drawn from this dataset. It is said that X follows a finite mixture distribution of G components if its probability density function denoted ψ can be written as :

$$\psi(x; \alpha) = \sum_{g=1}^{G} \pi_g h_g(x; \alpha_g) \tag{3.1}$$

where every distribution h_g is parameterized by α_g, $\{\pi_g\}_1^G$ are the mixing probabilities such that : $\sum_{g=1}^{G} \pi_g = 1$ and $\alpha = ((\pi_g, \alpha_g); g = 1, ..., G)$. The distributions can be from the same family or from different families, for example from beta and normal distributions.

Neural-network based clustering methods mostly use deep neural networks to transform input dataset into clustering-friendly representations ([12, 110, 77]). Neural networks such as Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN) or Generative Adversarial Network (GAN) can be used for this purpose. Then, the representations, called *latent features*, are extracted from one or more layers and are used as inputs of a specific clustering method. Typically, the

loss function used for the network training is a combination of a *network loss* and a *clustering loss*. Network loss constraints the network during the learning of latent features to avoid the loss of information. Clustering loss is specific to the clustering method and the clustering-friendliness of the learned representations.

In our context, we may encounter mixed datasets with various characteristics. Since some algorithms do not handle mixed data, we also introduced some DR methods in the next section.

We implemented each of the following algorithms with similar input and output. We present the following algorithms (many algorithms from the survey by Ahmad and Khan [8] need a lot of change to be adapted to any dataset) :

- **Dimensionality reduction**
  - Factorial Analysis of Mixed Data (FAMD), introduced by [46], see section 3.2.1 ;
  - Laplacian Eigenmaps, introduced by [18], see section 3.2.2 ;
  - Uniform Manifold Approximation and Projection (UMAP), introduced by [107], see section 3.2.3 ;
  - Pairwise Controlled Manifold Approximation and Projection (PaCMAP), introduced by [154], see section 3.2.4.
- **Partitional**
  - K-prototypes, introduced by Huang [70], see section 3.3.1 ;
  - Convex K-Means also known as Modha-Spangler, introduced by Modha and Spangler [113], see section 3.3.2 ;
- **Model-based**
  - KAy-means for MIxed LArge data (KAMILA), introduced by Foss et al. [48], see section 3.3.3 ;
  - Model Based Clustering for Mixed Data (ClustMD), introduced by McParland and Gormley [108], see section 3.3.4 ;
  - Mixed Dataset and Dataset with Missing Values (MixtComp), introduced by Biernacki [21], see section 3.3.5.
- **Hierarchical**
  - Philip and Ottaway, introduced by Philip and Ottaway [127], see section 3.3.6 ;
  - HDBSCAN with dimensionality reduction (DenseClus), introduced by McInnes and Healy [106], see section 3.3.7 ;
  - Pretopology, introduced by Lévy et al. [93], see section 3.3.8.

3.2 Dimensionality Reduction

To handle high-dimensional mixed data, and translate it into lower-dimensionality numerical data, we need DR techniques. We use these techniques for preprocessing data, evaluating or visualizing results.

3.2.1 Factorial Analysis of Mixed Data (FAMD)

FAMD is a factorial method used to analyze mixed data. The idea here is to apply factorial analyses on 2 separate groups of features (numerical and categorical), then to combine the results.

On a dataset including K numerical variables $k = 1, ..., K$ and Q categorical variables $q = 1, ..., Q$. With z a numerical variable, we consider $r(z, k)$ the correlation coefficient between z and k and $\eta^2(z, q)$ the squared correlation ratio between z and q. The mains steps of FAMD are :

1. Split data into 2 groups : one for numerical features, and one for categorical features
2. Perform a Multiple Correspondance Analysis (MCA) over categorical features. The objective is to maximize $\sum_k r(z, k)$.
3. Perform a Principal Component Analysis (PCA) over the numerical features. The objective is to maximize $\sum_q \eta^2(z, q)$.
4. Perform a global PCA over the results of the 2 previous factorial analyses. Then, the global objective of FAMD is to maximize :

$$\sum_k r(z, k) + \sum_q \eta^2(z, q) \tag{3.2}$$

The explained inertia represents the amount of variance in the data that is explained by the principal components obtained from the analysis. It is similar to the concept of explained variance in PCA for numerical data. With FAMD, the explained inertia is known (as it is a factorial method) and it does not require hyperparameter tuning, which can lead to instability. However, FAMD may be limited when there are too few observations (MCA becomes unstable) or when the number of numerical features is much smaller than the number of categorical features. An example of FAMD is shown in Figure 3.1a.

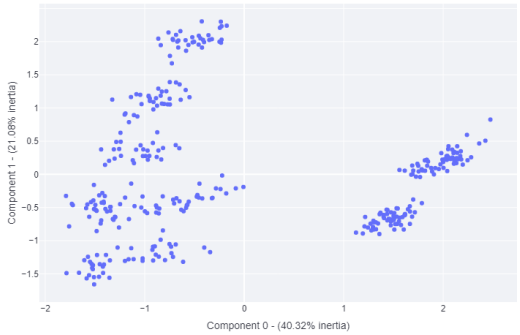
3.2.2 Laplacian Eigenmaps

Laplacian Eigenmaps is a spectral embedding technique, used for non-linear DR. Laplacian Eigenmaps main steps are :

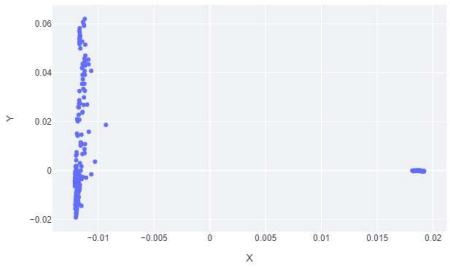
1. Compute the pairwise distance matrix of the dataset. To compute it over mixed data, we use Huang's distance :

$$d_{ij} = d_{ij}^N + \gamma d_{ij}^C \tag{3.3}$$

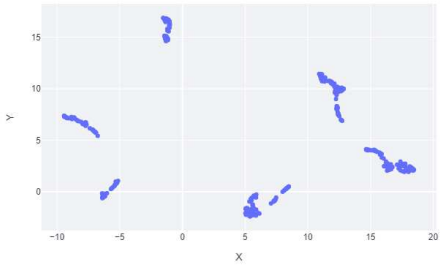
   where : d_{ij} is the distance between two data points ; d_{ij}^N is the squared Euclidean distance over numerical features ; d_{ij}^C is the Hamming distance over categorical features ; γ is proportional to the average standard deviation of numerical features. Ratio is user defined, usually its half.
2. Build an adjacency matrix from the distance matrix. This matrix W represents the edges of a weighted graph. To compute it, multiple solutions exist, but the most common is the Heat Kernel method. The adjacency W_{ij} is computed from the distance d_{ij} using $W_{ij} = exp(-\frac{d_{ij}}{t})$ where t is user-defined.
3. Compute the Laplacian matrix. Using a diagonal matrix D representing the sum of the weights for every node, $D_{ii} = \sum_j W_{ji}$. The Laplacian matrix is $L = D - W$. Then, compute the eigenvectors f for the problem : $Lf = \lambda Df$.
4. Select the eigenvectors that form our embedded low-dimensional space. As the first eigenvector corresponds to the eigenvalue 0, the m next following eigenvectors are selected to build a m dimension embedding.



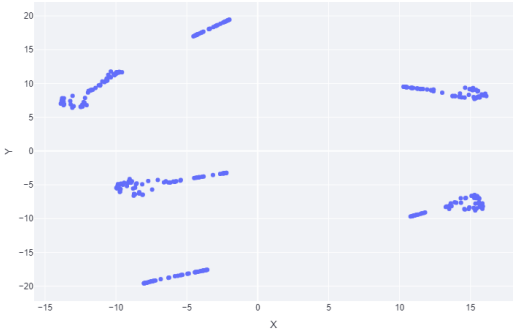
(a) FAMD in 2D with explained inertia 61.40%.



(b) Laplacian Eigenmaps 2D with $t = 1$.



(c) UMAP 2D with $k = 15$.



(d) PaCMAP in 2 dimensions on the Palmer Penguins Dataset with FAMD initialization

Figure 3.1 – Dimensionality reduction on the Palmer Penguins dataset.

This approach is similar to the one used by spectral clustering, and its results may be interpreted within a clustering framework, see [18]. However, multiple techniques can be used to compute the adjacency matrix, and these techniques require hyperparameters, such as the parameter t, which can lead to significant variations in the results. Additionally, unlike with factorial methods, the axes in the low-dimensional space obtained through spectral embedding do not have a specific meaning, leading to less interpretable results as shown in Figure 3.1b.

3.2.3 Uniform Manifold Approximation and Projection (UMAP)

UMAP is a non linear dimension reduction algorithm, that seeks to preserve the local topological structure of the data. The idea is to initialize a first embedding, then optimize it using gradient descent. The process is as follows :

1. Compute the pairwise distance matrix of the dataset. For mixed data, we use Huang's distance (Equation 3.3).
2. Compute the adjacency matrix, representing the edges of a weighted graph. To do so, we need an hyperparameter k, that is user defined (most common value is 15). The k nearest points are the « neighbors » of each node. The similarity sim_{ij} between nodes i and j is computed using Equation 3.4.

$$sim_{ij} = exp(\frac{d_{ij} - d_N}{\sigma}) \tag{3.4}$$

   where : d_{ij} is the distance between i and j ; d_N is the distance between i and its nearest neighbor ; σ is adjusted for each node, so that the sum of weights for each node is $log_2(k)$. As this similarity is not symmetrical ($sim_{ij} \neq sim_{ji}$), the final weight stored in the adjacency matrix is $W_{ij} = (sim_{ij} + sim_{ji}) - sim_{ij} * sim_{ji}$.
3. Initialize UMAP with Spectral Embedding (such as Laplacian Eigenmaps) over this graph. Any dimension reduction technique could be used for the initialization (even a random projection in a low-dimension space), but Spectral Embedding gives faster convergence.
4. Optimization. For a datapoint A, select one neighbor N and one not-neighbor F datapoints. For those two points, compute the similarity score s with A such that

$$s = \frac{1}{1 + \alpha d^{\beta}} \tag{3.5}$$

   with d the distance in the low-dimension space, $\alpha = 1.577$ and $\beta = 0.8951$. From s_{AN} and s_{AF} the similarities between A and N and between A and F, compute the cost function (3.6) :

$$cost = log(\frac{1}{s_{AN}}) - log(\frac{1}{1 - s_{AF}}) \tag{3.6}$$

   From this cost function, the point A is moved, using Stochastic Gradient Descent to find the optimal position is the low-dimensional space.

UMAP preserves coherence over the local structure of the data as illustrated in Figure 3.1c. This focus on the local structure leads to well defined groups of datapoints in the projected space. Indeed, UMAP improves the results of numerical clustering algorithms as shown by [13]. However, the number of neighbors k has an impact on the results and could cause bias in their interpretation.

3.2.4 Pairwise Controlled Manifold Approximation and Projection (PaCMAP)

PaCMAP method is quite similar to UMAP. Its idea is also to initialize a first low-dimensional embedding, then to optimize it. However, PaCMAP aims to preserve both local and global structures, whereas UMAP focuses mainly on the local structure. PaCMAP's main steps are :

1. Initializiation. With numerical-only data, Principal Component Analysis is used to build a first low-dimensional embedding. With mixed-data, we use FAMD, as it is considered as PCA's mixed-data counterpart. Note that contrary to UMAP, initialization changes the results of PaCMAP.
2. Optimization : the following steps are repeated for 450 iterations. PaCMAP relies on the concepts of Neighbors, Mid-Near Pairs and Further Pairs. For a given datapoint A, the Neighbors are the pairs formed by A and its k (hyperparameter, typically 10) closest neighbors. Then, to define Mid-Near pairs, sample 6 observations, and select the pair of A 2nd closest sampled observation. The Further Pairs are the pairs of A and every other datapoint. The number of Mid-Near and Further Pairs given by hyperparameters, relying on the number of Near-Pairs.
3. Define a weighted graph (new graph for each iteration). The weights w_{NB}, w_{MN} and w_{FP} for Neighbors, Mid-Near Pairs and Further Pairs are defined depending on the kind of pair and the iteration.
   - - First 100 iterations : $w_{NB} = 2$, w_{MN} linearly going from 1000 to 3, $w_{FP} = 1$;
     - Iteration 101 to 200 : $w_{NB} = 3$, $w_{MN} = 3$, $w_{FP} = 1$;
     - Last 250 iterations : $w_{NB} = 1$, $w_{MN} = 0$, $w_{FP} = 1$.
4. For each datapoint i, compute the loss function given by :

$$Loss = w_{NB} \times \sum_{J} \frac{\tilde{d_{ij}}}{10 + \tilde{d_{ij}}} + w_{MN} \times \sum_{K} \frac{\tilde{d_{ij}}}{1000 + \tilde{d_{ik}}} + w_{FP} \times \sum_{L} \frac{\tilde{d_{il}}}{1 + \tilde{d_{il}}} \quad (3.7)$$

   where : J the neighbors j of i; K the mid-near points k of i; L the further points l of i; $\tilde{d_{AB}} = 1 + ||y_A - y_B||^2$.
5. Move i using Stochastic Gradient Descent over the computed Loss function (Equation 3.7), to find its optimal position in the low-dimension space.

The use of Mid-Near Pairs makes PaCMAP preserve the global structure of data better than UMAP does. Yet, in some cases, preserving the global structure (and not only the local structure) might offer no benefits. An example of PaCMAP is shown in Figure 3.1d.

3.3 Algorithms

In this section, we describe each implemented algorithm with its hyperparameters and its pros and cons. We summarize the methods to provide a good understanding and overview of their process with a homogeneous vocabulary and notations.

3.3.1 Partitional clustering – K-prototypes

The most known partitional clustering algorithm for mixed dataset is K-prototypes. By combining the dissimilarity measure between two numerical features, taken from K-Means algorithm, and a matching dissimilarity measure between two categorical features, taken from K-Modes algorithm, the paper proposes a new dissimilarity measure (see Equation 3.3) between two mixed-type objects. In K-prototype, a prototype is a mix between a centroid for numerical features and a mode for categorical features.

Let γ be a user-defined hyperparameter. It is a weight for categorical attributes in a cluster, in order to balance the influence of the two types of features. Let k be a user-defined hyperparameter.

The algorithm can be decomposed in three steps :

1. Initial prototypes selection : For each of the k clusters, it selects a data object from the dataset as the initial prototype.
2. Initial allocation : From the set of the initial prototypes, it allocates each data object of the dataset to a cluster of data objects having the same closest prototype, according to the proposed dissimilarity measure. The prototypes are updated after each data object assignment.
3. Re-allocation : After the assignment of each data object to a cluster, the dissimilarity measures of each data object against each prototype is computed. If a data object is more similar to the prototype of another cluster, it is reallocated to this cluster. Then, the prototypes of the modified clusters are updated.

The step (3) is repeated until no data objects have changed of cluster after all the objects of the data dataset have been tested.

The main problem of the K-prototype is that the Hamming distance does not capture well the similarity between categorical features, which is represented as a 0 or 1 value depending on whether they are the same or different. To overcome this problem, [7] modify Huang's approach.

A major difference is in the similarity measure used for categorical features, which is not anymore binary like the Hamming distance. Given a categorical feature, its distance with another categorical feature value is computed regarding the overall distribution in the dataset of these two features and their co-occurrence with the other features, i.e the different combinations of the other features' values with these two features that are encountered in the observed datapoints.

Another major difference lies in the cost function, which uses for each numerical features a weight that represents their *significance*, i.e. their importance in the dataset and how they will influence the clustering. This parameter is not user-defined like the previous γ parameter, but is determined from the dataset using the proposed similarity measure. The numeric features are divided into intervals, which are assigned to a categorical value, to compute this parameter. This discretization do not happen for clustering, which still use the Euclidean distance. The new cost function also has a new centroid representation, represented by the mean of all numerical values in the cluster (like in Huang's algorithm) and the proportional distribution of categorical values in the cluster.

3.3.2 Partitional clustering – Convex K-Means

In Convex K-Means, given a dataset S of N data objects such that $S = (x_i; i = 1, ..., N)$, each data object x_i is represented as a tuple of M components of column vectors such that $x_i = (F_{(i,m)}; m = 1, ..., M)$ where $F_m = (F_{(i,m)}; i = 1, ..., N)$ is a column vector denoted as a *feature vector*.

Given a dataset, a feature space is defined by a set of features chosen from this dataset. This feature space contains all the values possible that the features of this set can take. The dimension of this space is equal to the number of features in the set.

Given a feature vector F_m, its components are all the values that the feature m takes along the different data objects of S. For all x_i, the components $F_{(i,m)}$ lie in the same feature space $\mathcal{F}_m$.

Given a data object $x_i = (F_{(i,1)}, F_{(i,2)}, ..., F_{(i,m)}, ..., F_{(i,M)})$, x_i lies in a feature space $\mathcal{F}$, created by the M-fold Cartesian product of the features spaces $\{\mathcal{F}_l\}_{m=1}^{M}$, such that : $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2 \times ... \times \mathcal{F}_m \times ... \times \mathcal{F}_M$. A feature vector F_m can differ from the other feature vectors by its properties, especially by the type of the feature m. Then, each feature space $\mathcal{F}_m$ has its own properties (dimensions, topologies...) and is different from the others.

Given two data objects $x_i = (F_{(i,m)}; m = 1, ..., M)$ and $x_j = (F_{(j,m)}; m = 1, ..., M)$, they propose a *distortion measure* D_m between the two corresponding feature vectors components $F_{(i,m)}$ and $F_{(j,m)}$. D_m is assigned to the feature space $\mathcal{F}_m$ where $F_{(i,m)}$ and $F_{(j,m)}$ lie. From the M distortion measures that can be obtained, they define a *weighted distortion measure* D^α between x_i and x_j as a weighted sum of the M distortion measures, such that : $D^\alpha(x_i, x_j) = \sum_{m=1}^{M} \alpha_m D_m(F_{(i,m)}, F_{(j,m)})$. The features weighting is represented by the vector $\alpha = (\alpha_m; m = 1, ..., M)$, which contains the weights relative to each D_m. They are refered to as *feature weights* and define the importance of a feature vector in the clustering.

To adapt their algorithm to the mixed data case, they consider their dataset to have two feature spaces : one consisting of numerical features and the other consisting of categorical features. They represent a data object x_i as a tuple of a numerical feature vector component $F_{(i,1)}$ and a categorical feature vector component $F_{(i,2)}$, such that : $x_i = (F_{(i,1)}, F_{(i,2)})$. The distortion measures D_1 and D_2 are respectively the Euclidean distance and the cosine distance.

Let the dataset be partitioned by the clusters $\{C_u\}_{u=1}^{U}$. Given a cluster C_u, the cluster centroid is a tuple c_u of M components, such that : $c_u = (c_{(u,m)}; m = 1, ..., M)$. Along the features spaces $\{\mathcal{F}_m\}_{m=1}^{M}$, they denote the vector component $c_{(u,m)}$ as the centroid of the cluster u lying in $\mathcal{F}_m$ with all the components of the feature vector F_m. The centroid c_u is determined by the data object that minimizes the sum of the D^α between this data object and all the other data objects contained in C_u. To do so, each component $c_{(u,m)}$ is determined in the same way as c_u, but by minimizing the sum of the D_m between the feature vector components, such that : $c_{(u,m)} = \underset{F_{(j,m)} \in \mathcal{F}_m}{\arg\min} \left(\sum_{x_i \in u} D_m(F_{(i,m)}, F_{(j,m)})\right)$.

They propose a method to automatically identify feature weights in order to reach a good discrimination between clusters along the features spaces $\{\mathcal{F}_m\}_{m=1}^{M}$. To do so, they define in the feature space $\mathcal{F}_m$ the *average within-cluster distortion* denoted Γ_m and the *average between-cluster distortion* denoted Λ_m. A given features weighting α gives : $\Gamma_m(\alpha) = \sum_{u=1}^{U} \sum_{x \in C_u} D_m(F_m, c_{(u,m)})$ where $x = (F_m; m =$

$1, ..., M)$ and $\Lambda_m(\alpha) = \sum_{i=1}^{N} D_m(F_{(i,m)}, \overline{c_m}) - \Gamma_m(\alpha)$ where $\overline{c} = (\overline{c_m}; m = 1, ..., M)$ denotes the generalized centroid of the dataset. The « best » α minimizes the M-product of the ratio between Γ_m and Λ_m and this optimal weighting scheme is found through an exhaustive grid-search. This is done by repeatedly running the algorithm with different features weightings over a fine grid on the interval $[0, 1]$.

The algorithm can be decomposed into three steps :

1. Start with an arbitrary partitioning by selecting initial centroids.
2. Find the closest centroid for each data object using the proposed distortion measure.
3. Compute the new centroids using the centroids definition mentioned above.

Steps (2) and (3) are repeated until a stopping criterion is met.

The main drawback of this algorithm is its computational cost, which is high due to the brute-force search of feature weightings. The hyperparameters of the algorithm are the number of clusters to determine k and the granularity of the exhaustive grid-search. Due to its limitations, Convex K-Means does not meet our needs and often fails to provide satisfactory results on large datasets.

3.3.3 Model-based clustering – KAMILA

KAMILA is a combination of k-means clustering with the Gaussian-multinomial mixture model.

Parametric assumptions refers to how algorithms assume the data to be « shaped ». For example, k-means clustering typically assumes that the clusters' shapes are spherical and are of similar size. A Model-based clustering assume that the clusters' shapes are defined by a given statistical distribution. Some parametric assumptions are more restrictive than others and algorithms' performances depend on how strong the parametrics assumptions are but also if the data meet them.

Like the K-means clustering algorithm, KAMILA assumes that the clusters' shapes for numerical data are spherical or elliptical, which are not strong parametric assumptions. KAMILA also uses the properties of Gaussian-multinomial mixture model [71] to equitably balance the effects of numerical and categorical data without making the user specify the weights of both.

The use of the Kernel Density Estimation (KDE) to estimate the mixture distribution of numerical data relaxes the Gaussian assumption. Indeed, assuming that numerical data follows a Gaussian distribution with its parameters (mean and covariance), a non parametric methods like KDE can estimate the probability density function without information about the distribution.

Let the dataset S consists of N observations, such that : $S = (X_i; i = 1, ..., N)$ where X_i is the i-th observation. P denotes the number of numerical features and Q the number of categorical features. Each X_i is a $(P + Q)$-dimensional vector of random variables $(V^T, W^T)^T$, such that : $X_i = (V_i^T, W_i^T)^T$ where $V = (V_i; i = 1, .., N)$ and $W = (W_i; i = 1, .., N)$. V_i is a $P \times 1$ vector of numerical random variables and W_i is a $Q \times 1$ vector of $q = 1, 2, ..., Q$ categorical random variables, such that : $W_i = (W_{i1}, ..., W_{iq}, ..., W_{iQ})^T$ where W_{iq} is a categorical random variable that can have L_q categorical levels, i.e the L_q different categorical values that W_{iq} can take, such that : $W_{iq} = \{1, ..., l, ..., L_q\}$. Then, a mixed data object x_i is modeled as vector

composed of a numerical part represented by a vector v_i and a categorical part represented by a vector w_i, such that : $x_i = (v_i, w_i)$

Each V_i follows a finite mixture of G spherical or elliptical distributions (choice made by the user) such that in this case (see Equation 3.1) : $h_g(x;\alpha_g) = f_{V,g}(v_i;(\mu_g,\Sigma_g))$ where μ_g denotes the centroid of the g-th cluster and Σ_g the scaling matrix of the g-th cluster.

Each W_i follows a finite mixture of G multinomial distributions such that in this case (see Equation 3.1) : $h_g(x;\alpha_g) = f_{W,g}(w_i;\theta_g) = \prod_{q=1}^{Q} \eta(w_{iq};\theta_{gq})$ where $\theta_g = (\theta_{qg}; q = 1,...,Q)$, θ_{qg} denotes the parameters vector of the multinomial distribution corresponding to the q-th categorical variable contained in the cluster g and η is the multinomial mass function. θ_{qg} is a $L_q \times 1$ vector such that $\theta_{qg} = (\theta_{gql}; l = 1,...,L_q)$. Each θ_{gql} is the probability that the q-th categorical variable has the categorical level l if the data object x_i is in cluster g. The multinomial mass function is written as :

$$\eta(w_q;\theta_{gq}) = \prod_{l=1}^{L_q} \theta_{gql}^{I\{w_q=l\}} \tag{3.8}$$

where $I\{\cdot\}$ denotes the indicator function.

Under the assumption that V and W are independent, the dataset S follow a finite mixture of G joint probability distributions of $(V^T, W^T)^T$ such that in this case (see Equation 3.1) : $h_g(x;\alpha_g) = f_{V,W,g}(v,w;(\mu_g,\Sigma_g,\theta_g)) = f_{V,g}(v;(\mu_g,\Sigma_g)) \times f_{W,g}(w;\theta_g)$.

We denote $\hat{\mu}_g$ the estimator of μ_g and $\hat{\theta}_{gq}$ the estimator of θ_{qg}. The algorithm starts by initializing at iteration $t = 0$ a set of centroids $\hat{\mu}_g^{(t)}$ and a set of parameters $\hat{\theta}_{gq}^{(t)}$. $\hat{\mu}_g^{(0)}$ can be initialized by random draws from an uniform distribution, but another work of [49] specifies that random draws from the numerical variables of the observations give better results. $\hat{\theta}_{gq}^{(0)}$ is initialized by a random draw from a Dirichlet distribution.

First comes the *partition step*, which assigns each observation i to a cluster g according to the quantity $H_i^{(t)}(g)$. At the t-th iteration, with the set $\hat{\mu}_g^{(t)}$ and $\hat{\theta}_{gq}^{(t)}$, the assignment of an observation i can be decomposed in 4 steps :

1. For the numerical features, the Euclidean distances $d_{ig}^{(t)}$ between v_i and each $\hat{\mu}_g^{(t)}$ are computed before extracting the minimum distance. These two substeps are performed for the N observations before obtaining the set $r^{(t)}$ of the N minimum distances.
2. $r^{(t)}$ is used to estimate f_V through an univariate KDE step. KDE is a non-parametric estimation method used to estimate a density function of a random variable. This estimation is denoted $\hat{f_V}$.
3. For the categorical features, the probability $f_{W,g}(w_i;\theta_g)$ of observing w_i in cluster g is calculated.
4. The function $H_i^{(t)}(g) = \log(f_{W,g}(w_i;\theta_g)) + \log(\hat{f}_V(d_{ig}^{(t)}))$ is calculated. The observation i is assigned to the cluster that maximizes $H_i^{(t)}(g)$.

Then comes the *estimation step*, where $\hat{\mu}_g^{(t+1)}$ and $\hat{\theta}_{gql}^{(t+1)}$ are calculated. They are computed respectively as the mean of the numerical values in cluster g over

the number of data objects in g and the mean of the number of occurrences of the categorical level l in cluster g over the number of data objects in g. The two estimators are then used as inputs for the partition step of the next iteration.

The process consisting of these two steps is repeated until a partition with stable clusters. Multiple runs of this process are performed with different initialization. At the final iteration of a given run, the sum over the N observations of the highest value of $H_i^{(final)}$ between the G clusters. The algorithm outputs the partition generated by the run that maximizes the objective function.

The hyperparameter of this algorithm is the number of runs to perform.

3.3.4 Model-based clustering – ClustMD

ClustMD uses a latent variable model (LVM). LVM's main idea is that the observed datapoints are correlated and forms particular patterns because they are influenced by hidden variables, called *latent variables*.

Let S denote a dataset of N observed data objects, such that $S = (x_i; i = 1, ..., N)$. Each observed data object is a vector that contains m mixed types variables (numerical, ordinal or categorical), such that : $x_i = (x_{im}; m = 1, ..., M)$. The proposed model assumes that a given observed data object x_i is the manifestation of an underlying latent numerical vector z_i, such that $z_i = (z_{im}; m = 1, ..., M)$. This representation enables to represent the different types of data with one unified type of variable.

The model proposes 3 ways to represent an observed datapoint regarding its type :

- Case of numerical data : A given numerical variable x_{im} is a numerical manifestation of a latent numerical variable z_{im} that follows a Gaussian distribution. Both are of the same type, then : $x_{im} = z_{im} \sim \mathcal{N}(\mu_m, \sigma_m^2)$.
- Case of ordinal data : A given ordinal variable x_{im} with L_m levels is a categorical manifestation of a latent numerical variable z_{im} following a Gaussian distribution, i.e $z_{im} \sim \mathcal{N}(\mu_m, \sigma_m^2)$. Both are of different type, so an adaptation is needed. Let γ_m denotes a $L_m + 1$ vector of thresholds that partition the real line, such that : $\gamma_m = (\gamma_{(m,l)}; l = 1, ..., L_m)$. The observed datapoint x_{im} is defined such that if $\gamma_{(j,l-1)} < z_{im} < \gamma_{(j,l)}$, then $x_{im} = l$. After this adaptation, x_{im} is numerical so : $x_{im} = z_{im} \sim \mathcal{N}(\mu_m, \sigma_m^2)$.
- Case of categorical data : A given categorical variable x_{im} with L_m levels is a categorical manifestation of the components of a numerical latent vector z_{im} of dimension $L_m - 1$. The vector z_{im} follows a Multivariate Gaussian (MVN) distribution, i.e. $z_{im} = (z_{im}^l; l = 1, ..., L_m - 1) \sim \textbf{MVN}_{L_m-1}(\underline{\mu}_m, \Sigma_m)$, where $\underline{\mu}_m$ is the mean vector and Σ_m is the covariance matrix. The observed data object x_{im} is defined such that :

$$x_{im} = \begin{cases} 1 & \text{if } \max_l\{z_{im}^l\} < 0; \\ l & \text{if } z_{im}^{l-1} = \max_l\{z_{im}^l\} \text{ and } z_{im}^{l-1} > 0 \text{ for } l = 2, ..., L_m \end{cases}$$

They represent the dataset as a matrix of N rows and M columns. Supposing that the numerical variables are in the first C columns, the ordinal and binary variables in the following O columns and the categorical data in the final $M - (C + O)$ columns. Let $P = C + O + \sum_{m=C+O+1}^{M}(L_m - 1)$, which is equal to the number of

mixed type variables M. In ClustMD, z_i follows a mixture of G multivariate Gaussian distributions of P dimensions, i.e $z_i \sim \sum_{g=1}^{G} \pi_g \text{MVN}_P(\underline{\mu}_g, \Sigma_g)$, where π_g is the marginal probability of belonging to cluster g, $\underline{\mu}_g$ the mean for cluster g and Σ_g the covariance for cluster g.

The ClustMD model is fitted, i.e obtaining the parameters of the statistical distributions in the mixture for which ClustMD describes the best the observed data, using an Expectation-Maximization (EM) algorithm. EM is an iterative method used to find the maximum likelihood estimate of a latent variable, in our case z_i. The ClustMD model derives firstly the complete data log-likelihood. Then, the Expectation step will compute the expectation of this complete data log-likelihood with respect to z_i. If categorical variables are present, a Monte Carlo approximation algorithm is used for the Expectation step. Finally, the Maximisation step will maximize the value of this expectation with regard to the model parameters.

3.3.5 Model-based clustering – MixtComp

This model-based clustering aims to cluster mixed dataset and dataset with missing values in a moderate dimensional setting. It is a statistical method for clustering mixed data, which combines the strengths of model-based clustering and Bayesian approaches. The method models mixed data as a mixture of multivariate distributions, with each component representing a cluster. It can handle different types of data, including continuous, discrete, and mixed data, as well as missing data. The method incorporates a latent variable model that captures the hidden structure of the data, enabling it to handle complex data structures. The clustering is performed through a Bayesian inference process, which estimates the number of clusters, cluster parameters and the latent variables that capture the underlying structure of the data.

Let the dataset S consists of N observations, such that : $S = (X_i; i = 1, ..., N)$ where X_i is the i-th observation. One particular outcome of X_i is the data object x_i, which has M different features, such that : $x_i = (x_{im}; m = 1, ..., M)$. A data object is decomposed in three parts : numerical, categorical and integer, such that : $x_i = (x_i^{\textbf{num}}, x_i^{\textbf{cat}}, x_i^{\textbf{int}})$. Each x_{im} is contained in one of the three parts.

Each X_i follows a finite mixture distribution of G probability distributions such that (see Equation 3.1) : $h(x_i; \alpha_g) = f(x_i^{\textbf{num}}; \alpha_g^{\textbf{num}}) \times f(x_i^{\textbf{cat}}; \alpha_g^{\textbf{cat}}) \times f(x_i^{\textbf{int}}; \alpha_g^{\textbf{int}})$ where $\alpha_g = (\alpha_{gm}; m = 1, ..., M)$. The density function f is an univariate distribution associated to the feature m if the data object x_i is in cluster g.

The probability distribution of g is chosen depending on the type of its corresponding feature m :

- Numerical type : the Gaussian model of [28] is used.
- Categorical type : the multinomial model is used in the same way as KAMILA algorithm described in section 3.3.3. In this case, let the data object x_{im} have L_m categorical levels, i.e $x_{im} \in \{1, ..., l, ..., L_m\}$. Then, $f(x_{im}; \alpha_{gm}) = \eta(x_{im}; \alpha_{gm})$ where $\alpha_{gm} = (\alpha_{gml}; l = 1, ..., L_m)$ (see Equation 3.8).
- Integer type : the Poisson distribution of parameter α_{gm} is used, such that : $f(x_{im}; \alpha_{gm}) = \frac{(\alpha_{gm})^{x_{im}} e^{-\alpha_{gm}}}{x_{im}!}$.

To fit the model, MixtComp uses a variation of EM algorithm.

3.3.6 Hierarchical clustering – Philip and Ottaway

[127] propose to use Gower's similarity measure to obtain a similarity matrix, which is then used as input for a hierarchical clustering algorithm. Gower's similarity measure separates categorical and numerical features into two subsets, creating one categorical feature space and one numerical feature space. In the categorical feature space, the similarity between two datapoints is computed by a weighted average of similarities between all categorical features, which is calculated using Hamming distance. In the numerical feature space, the similarity between two datapoints is computed by the sum of the similarities between all numeric features.

The equation for Gower's similarity measure is (by [58]) :

$$s_{ij} = \frac{\sum_{k=1}^{p} w_{ij}^{(k)} s_{ij}^{(k)}}{\sum_{k=1}^{p} w_{ij}^{(k)}} \tag{3.9}$$

where s_{ij} is the similarity between data points i and j, $s_{ij}^{(k)}$ is the similarity between data points i and j for feature k, p is the number of features. $w_{ij}^{(k)}$ is equal to 0 when $s_{ij}^{(k)}$ cannot be calculated because of missing values (or for other reasons).

If feature k is categorical, then $s_{ij}^{(k)}$ is defined as :

$$s_{ij}^{(k)} = \begin{cases} 1 & \text{if data points } i \text{ and } j \text{ have the same value for feature } k, \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

If feature k is numerical, then $s_{ij}^{(k)}$ is calculated as follows :

$$s_{ij}^{(k)} = \frac{|x_i^{(k)} - x_j^{(k)}|}{R_k} \tag{3.11}$$

where $x_i^{(k)}$ and $x_j^{(k)}$ are the values of data points i and j for feature k, and R_k is the range of values for feature k.

3.3.7 Hierarchical Density-Based clustering – DenseClus

Amazon proposes a python module named DenseClus [1]. This module performs a DR with UMAP method before using accelerated HDBSCAN algorithm from [106], an extension of HDBSCAN algorithm from [27].

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a hierarchical density-based clustering algorithm. A density-based clustering algorithm identifies contiguous regions of high density of objects in a data space, separated from other such clusters by contiguous regions of low density. The objects in the separating regions of low density are typically considered as noise/outliers (see [136]).

Given a dataset S of N objects, such that $S = (x_i; i = 1, ..., N)$, they define a *core distance* of a data object x_i with regard to the hyperparameter k, denoted

1. https://github.com/awslabs/amazon-denseclus

$d_{\textbf{core}}(x_i)$, as the distance from this data object to its k-nearest neighbor, i.e the k-th data object closer to it. Core distance is smaller for a data object in a dense region of data objects, while sparser regions give larger core distances to objects. Core distance enables to estimate the density of a region, by taking the inverse of the core distance.

They also define a data object x_i as an ϵ-*core object* for every value of the parameter ϵ that satisfies $d_{\textbf{core}}(x_i) \leq \epsilon$. This is equivalent to saying that the data object x_i has its k-nearest neighbors in the neighborhood defined by ϵ.

From the concept of the core distance, they define a new distance metric between two objects called *mutual reachability distance*. Given two objects x_i and x_j, the mutual reachability distance is computed as :

$$d_{\textbf{mreach}}(x_i, x_j) = \textbf{max}\{d_{\textbf{core}}(x_i), d_{\textbf{core}}(x_j), d(x_i, x_j)\}$$

where $d(\cdot, \cdot)$ denotes a metric distance. The mutual reachability distance captures not only the distance between the two objects in the Euclidean space but also the density of their neighborhood.

They represent their data as a weighted graph called the *Mutual Reachability Graph*. In this graph, the objects are considered to be the vertices. An edge between any two objects is considered to have a weight equal to the mutual reachability distance between the two objects. To model the cluster, all edges having weights greater than ϵ are removed and the remaining groups of connected ϵ-core objects constitutes the clusters. The remaining unconnected objects are considered as « noise ».

Clusters hierarchy is built with a divisive fashion (considering firstly all objects being contained in a single cluster) and by varying the value of ϵ. After computing the core distance with regard to k for all data objects in S, the algorithm computes the graph and extract the Minimum Spanning Tree (MST) from it using the Prim's algorithm. A MST is a subset of a graph that connects all the vertices of this graph together such that the sum of the edges weight is minimum. Then, it iteratively removes all edges from the MST in decreasing order of weights. This is done by sorting the edges of the MST in an increasing order and gradually decreasing the value ϵ so that a given edge with a weight above ϵ is removed. ϵ acts as a distance threshold, so that its variation gradually disconnect objects from their clusters. This is equivalent to gradually increasing a density threshold $\lambda = \frac{1}{\epsilon}$, so that a cluster not dense enough will be split. λ is increased until no split is performed anymore.

Splits are not performed in a classical way but occurs under particular constraints. A *minimum cluster size* parameter ω defines the minimum number of objects accepted in a cluster. When a parent cluster is split into two child clusters, if any of the two child cluster contains fewer objects than ω, the split is considered as « spurious ». The child cluster in question will be considered as « falling out of the parent cluster » at the given λ value, labelled as « noise » and removed from the cluster. Three cases can be encountered after a cluster split :

1. The two child clusters' sizes are below ω. The child clusters are removed from the parent cluster. No other splits are executed after.
2. If only one child cluster's size is higher than ω, it is considered as the continuation of the parent cluster and takes its parent cluster's label. The same

cluster size evaluation process is repeated on it while the other child cluster is removed.

3. If more than one child cluster contains more than ω data objects, the split is considered as « true ». Two child clusters are obtained and the same cluster size evaluation process is repeated on them.

We can consider that the parent cluster is « shrinking » through the splits of case (2), until case (1) or case (3) is encountered.

From the obtained dendrogram, the clusters extraction is applied according to the *stability* of the clusters, i.e their capacity to keep shrinking until a « true » split occurs as λ increases. Let S be partitioned by the clusters $\{C_u\}_{u=1}^{U}$. Given a cluster C_u and a data object x, they define $\lambda_{\textbf{min},C_u}(x)$ as the minimum λ value for which the data object x is contained in C_u. In other words, $\lambda_{\textbf{min},C_u}(x)$ is the value of λ at which this cluster became a cluster of its own (after a split or from the root of the dendrogram). They define $\lambda_{\textbf{max},C_u}(x)$ as the value of λ when the data object x falls out of cluster C_u. Then, the stability of a cluster C_u, denoted $\sigma(C_u)$, is determined as : $\sigma(C_u) = \sum_{x \in C_u} \lambda_{\textbf{max},C_u}(x) - \lambda_{\textbf{min},C_u}(x)$. The partition of U clusters that maximizes the score $\sum_{u \in U} \sigma(C_u)$ is selected under the following constraint : the partition cannot contain overlapping clusters. This is equivalent to the following condition : if a cluster is selected, its child clusters cannot be selected.

The algorithm has a quadratic complexity, which limits its applicability for large amount of data. To overcome this problem, [106] proposes an accelerated version of HDBSCAN. In this algorithm, Prim's algorithm is replaced by the Dual Tree Boruvka algorithm proposed by [103], which is designed to determine MST in a metric space. Accelerated HDBSCAN adapted this algorithm to the mutual reachability distance and presents a log-linear complexity.

3.3.8 Hierarchical clustering – pretopoMD

Pretopology allows for the extraction, organization, and structuring of data into homogeneous groups, as well as the integration of multicriteria analysis (using quantitative data, qualitative data, and other types of characteristics describing complex systems, such as time series). Pretopology-based clustering exploits the logical construction of pretopological spaces to define the construction of hierarchical structures according to the similarity between elements on specific characteristics. The theory of Pretopology is described in more details in the next section (3.4), Pretopology-based clustering and its application for clustering complex energy systems have been presented in [93], and it's application on different dataset are discussed in Chapter 4 and in Subsection 5.2.6).

A pretopological space is based on the concept of pseudoclosure : let $(U, a(.))$ be a tuple, where U is a set of elements and $a(.)$ is a pseudoclosure function on U, constitutes a pretopological space.

We define a pseudoclosure function $a : \wp(U) \to \wp(U)$ on a set U, is a function such that : $a(\emptyset) = \emptyset$; $\forall A \mid A \subseteq U : A \subseteq a(A)$, where $\wp(U)$ is the power set of U.

The mathematical formalization of a pretopological space used in the clustering algorithm presented is based on three elements :

- A set of weighted directed graphs $G = G_1(V_1, E_1), G_2(V_2, E_2), ..., G_n(V_n, E_n)$,
- A set of thresholds $\Theta = \theta_1, \theta_2, ..., \theta_n$

- A boolean function $DNF(.) : (\wp(U), U) \rightarrow True, False$, expressed as a positive Disjunctive Normal Form (DNF) in terms of n boolean functions $V_1(A,x), \ldots, V_n(A,x)$, each associated with a graph, and whose truth value depends on the set A and the item x.

We determine if an item $x \in U$ belongs to the pseudoclosure of a set A in the following way :

- $\forall V_i(A,x), V_i(A,x) = True \iff \sum_{e_{xy} \in G_i, y \in A} w(e_{xy}) \geq \theta_i$, where e_{xy} is the edge going from x to y, and $w(e)$ is the weight of the edge e.
- The item $x \in U$ will belong to the pseudoclosure of $A \iff$ the $DNF(.)$ evaluates to True

This formalization was introduced in [85].

Exploiting the built preotopological space, the construction of a hierarchical clustering is applied following the following algorithm :

1. Determine a family of elementary subsets called seeds.
2. Construct the closures of the seeds by iterative application of the pseudo-closure function.
3. Construct the adjacency matrix representing the relations between all the identified subsets (even the intermediate ones).
4. Establish the quasi-hierarchy by applying the associated algorithm on the adjacency matrix.

This pretopological-based clustering approach is being implemented in a Python library and can be applied simultaneously to various data types, making it a versatile and powerful clustering method.

3.3.9 In short

Table 3.1 shows the characteristics of the different algorithms such as its type or the use of tandem analysis. An algorithm's ability to produce outliers, or to handle missing values might differentiate it from others. Also, algorithms needing a hyperparameter k for the number of clusters to find must use the Elbow Method to find k, which could extend the computation time artificially.

Algorithm	Type	Needs K	Tandem	Missing Values	Outliers
K-Prototypes	Partitional	Yes	-	No	No
Modha-Spangler	Partitional	Yes	-	Yes	No
Phillip & Ottaway	Hierarchical	Yes	-	No	No
Kamila	Model-Based	Yes	-	No	No
ClustMD	Model-Based	Yes	-	No	Yes
MixtComp	Model-Based	Yes	-	Yes	Yes
DenseClus	Hierarchical	No	UMAP	No	Yes
Kmeans-FAMD	Partitional	Yes	FAMD	No	No
Pretopology	Hierarchical	No	FAMD UMAP PaCMAP	No	Yes

Table 3.1 – Characteristics of the different algorithms.

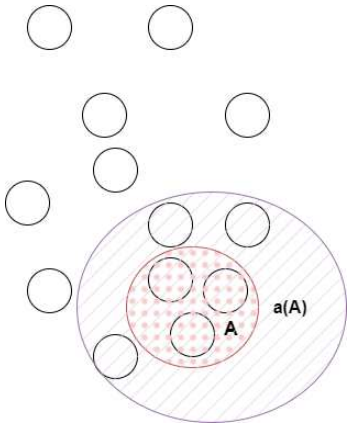


Figure 3.2 – Example of a pseudoclosure function.

3.4 Pretopology

3.4.1 Theoretical Framework of Pretopology

In this section, we provide an overview of the fundamental concepts and definitions in pretopology. We will start with a description of pretopological space and pseudoclosure.

Définition 3.1. A pseudoclosure function $a : \wp(U) \to \wp(U)$ on a set U of elements is a function that satisfies :

- $a(\emptyset) = \emptyset$
- $\forall A \mid A \subseteq U : A \subseteq a(A)$

where $\wp(U)$ represents the power set of U.

The pretopological space for a dataset is constructed based on the features of the dataset, taking into account the different types (numerical, categorical, etc.). Through the pseudoclosure function, it establishes relationships between sets of elements and their subsets.

Définition 3.2. A pretopological space is a tuple $(U, a(.))$, where U is a set of elements and $a(.)$ is a pseudoclosure function on U.

The previous definition determines the most general pretopological space. By asking the function to fulfill some additional conditions we get more specific pretopological spaces :

Définition 3.3. If $\forall\ A, B \mid A \subseteq U,\ B \subseteq U : A \subseteq B \implies a(A) \subseteq a(B)$, then we get a pretopological space of type V. Otherwise we call it a $non\text{-}V$ space.

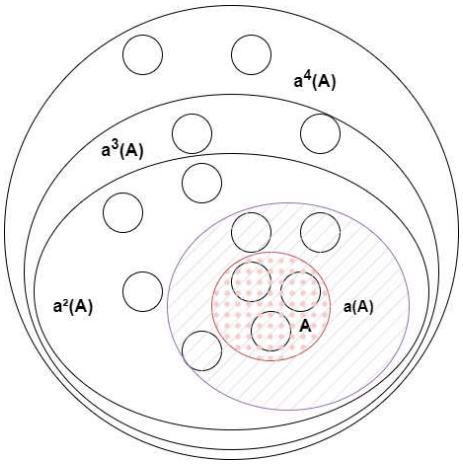


Figure 3.3 – Closure of set A, $a^4(A) = F(A)$.

Définition 3.4. In a pretopological space, the closure of a set A denoted as $F(A)$ is determined by iteratively applying the pseudoclosure operator to the set and its subsequent images until no further expansion occurs (see Figure 3.3).

Définition 3.5. Given a pretopological space $(U, a(.))$, any subset A of U is said to be a closed subset of U if and only if $A = a(A)$

Définition 3.6. In a pretopological space, the closure of a subset A of U is the smallest closure that contains A, denoted as $F(A)$.

Now we introduce our framework for formalizing a pretopological space, which is based on, and adapts the work of Julio Laborde [85]. This framework is illustrated in figure 3.4 . In this framework, a pretopological space is characterized by a tuple $(G, \Theta, DNF(.))$, where :

- $G = G_1(V_1, E_1), G_2(V_2, E_2), ..., G_n(V_n, E_n)$ represents a collection of n weighted directed graphs.
- $\Theta = \theta_1, \theta_2, ..., \theta_n$ is a set of n thresholds, each associated with a specific graph.
- $DNF(.) : (\wp(U), U) \rightarrow True, False$ is a boolean function defined as a positive DNF involving the n boolean functions $V_1(A, x), ..., V_n(A, x)$, each associated with a graph. The truth value depends on the set A and the element x.

To determine if an element $x \in U$ belongs to the pseudoclosure of a set A, we determine the values for V function and the DNF :

- For each $V_i(A, x)$, $V_i(A, x) = True$ if and only if $\sum_{e_{xy} \in G_i, y \in A} w(e_{xy}) \geq \theta_i$, where e_{xy} denotes the edge from x to y, and $w(e)$ represents the weight of the edge e.
- The element $x \in U$ belongs to the pseudoclosure of A if and only if the $DNF(.)$ evaluates to True.

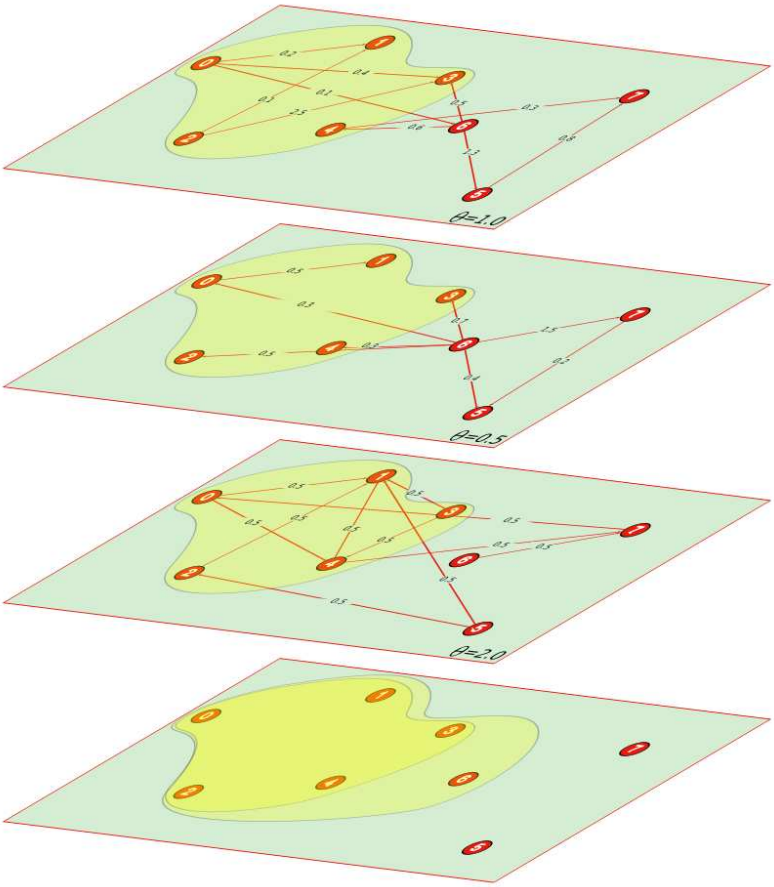


Figure 3.4 – Illustration of the framework formalizing a V-type pretopological space [85]

In essence, this process checks if the sum of the edge weights connecting the element x to the elements within A is greater than the threshold associated with the graph in each graph. If this condition is met, the boolean variable corresponding to that graph takes the value True; otherwise, it takes the value False. If $DNF(.)$ evaluates to True given the values of the boolean functions $V_i(A, x)$, then the element belongs to the pseudoclosure.

A pretopological space is created from a dataset in the following way. Each graph in G corresponds to a characteristic of the dataset. In each graph, a vertex represents an element of the system under study, and an edge represents the similarity between two elements with respect to the corresponding characteristic.

The thresholds, similarity functions, and the DNF are hyperparameters determined based on the nature of the characteristic and its importance in the clustering process. Default values and functions for these hyperparameters are discussed in subsection 3.4.3.

3.4.2 PretopoMD Algorithm

This section outlines the algorithms developed in a Python library for constructing closures and building hierarchical clustering of mixed data. The algorithm, provided as pseudocode in Algorithm 1, is organized into four stages :

- Identify a family of elementary subsets, referred to as seeds.
- Construct closures of seeds through iterative application of the pseudoclosure function.
- Create the adjacency matrix representing relationships between all recognized subsets, including intermediate ones.
- Determine the quasi-hierarchy by applying the corresponding algorithm to the adjacency matrix.

Algorithm 1 QuasistructuralAnalysis : Algorithm for building a quasi-hierarchy from pretopological space.

Require: $((U, a(.)), d, seed\_Func(.), th_{qh})$
Ensure: $Sets_{qh}, Adj_{qh}$
$seed\_List \leftarrow Set\_Seeds((U, a), d, seed\_Func)$
$Sets_{ipc} \leftarrow Iterative\_Pseudoclosure((U, a), seed\_List)$
$Atr \leftarrow Attraction\_Matrix(Sets_{ipc})$
$Sets_{qh}, Adj_{qh} \leftarrow QuasiHierarchy(Sets_{ipc}, Atr, th_{qh})$

Several methods can be employed to identify seeds. As a result, the algorithm is influenced by the following two hyperparameters :

- The $seed\_Func(.)$ function, which determines a set of nearby elements for a given element, constituting a seed.
- The degree d specifies the size of the seeds.

An additional hyperparameter is required by the *Extract_Quasihierarchy* algorithm to establish the quasi-hierarchy : th_{qh}, representing the threshold above which two sets are considered related in the hierarchy.

We will now discuss each stage of the algorithm in detail.

Computation of a Family of Elementary Sets or Seeds

The goal here is to determine elementary subsets of size d, referred to as seeds, using the $seed\_Func(.)$ function, which is responsible for finding the required d neighbors. This is accomplished by iterating over all points in the set U, associated with the pretopological space p. The pseudocode for the resulting algorithm (named $Elem\_Quasiclosures$) is provided in Algorithm 2.

Algorithm 2 Set_Seeds : Construction of the seeds of size d by applying the function $seed\_Func(.)$ on all the elements of the set U.

Require: $((U, a(.)), d, seed\_Func(.))$
Ensure: $seed\_List$
  $seed\_List \leftarrow list()$
  for all $x \in U$ **do**
    $seed \leftarrow seedFunc(x, d)$
    $seed\_List.append(seed)$
  end for

Creation of Subsets through Iterative Pseudoclosure Applications

The *Set_Seeds* function constructs the subsets that will be organized by the quasi-hierarchy algorithm, utilizing the seed list $seedList$ previously computed by *Elem_Quasiclosures*. For each seed in $seed\_List$, the membership function is iteratively applied until the pseudoclosure no longer results in larger sets.

The subsets are stored in a list of sets called QF_{tmp}, which indexes the subsets according to the number of elements they contain. The subsets of size s are stored in the s-th position of QF_{tmp}. Because the pseudoclosure function $a(.)$ only returns a set that is larger or equal in size, applying the pseudoclosure function to the sets in ascending order of size ensures that all elements are processed once and only once.

The list $Sets_{ipc}$, constructed from the lists in QF_{tmp}, is then returned. The corresponding pseudocode is provided in Algorithm 3.

Creation of the Attraction Matrix

The iterative application of a pseudoclosure to two seeds can create distinct sets that have non-empty intersections. Traditional hierarchies of sets only deal with sets that either have no intersection or are contained within one another (i.e., subsets and supersets). Hence, another type of relationship must be defined, called a quasi-hierarchy.

First, in algorithm 4, an attraction matrix is built, representing the “attraction” that sets have for each other. We use the term attraction to represent a non-symmetrical relationship between two intersecting sets, based on the size of each set and the size of their intersection. It is based on the following principles :

- Two subsets should only be attracted to each other if their intersection is non-empty (i.e., $A \cap B \neq \emptyset$),
- The larger the cardinality of the intersection $A \cap B$ relative to that of A, the stronger the attraction between A and B,

Algorithm 3 Iterative_Pseudoclusure : Calculation of subsets by iterative application of the pseudo-closure function.

Require: $((U, a(.)), seed\_List)$
Ensure: $Sets_{ipc}$
  QF_{tmp} a list of $Size(U)$ of empty sets
  for all $seed \in seed\_List$ **do**
    $QF_{tmp}[Size(seed)].append(seed)$
  end for
  for all $i \in range(1, Size(U) + 1)$ **do**
    for all $s \in QF_{tmp}[i]$ **do**
      $a_s \leftarrow a(s)$
      if a_s **not in** lists of QF_{tmp} **then**
        $QF_{tmp}[Size(a_s)].append(a_s)$
      end if
    end for
  end for
  $Sets_{ipc} \leftarrow list()$
  for all $i \in range(Size(QF_{tmp}))$ **do**
    $Sets_{ipc}.extend(QF_{tmp}[i])$
  end for

- The larger the cardinality of the subset B relative to that of A, the less necessary it is for $A \cap B$ to be large for the relation between A and B to be strong. In other words, a very large set will attract smaller sets even if their intersection is not very large.

Algorithm 4 Attraction_Matrix : Construction of the attraction matrix for the quasihierarchy.

Require: $(Sets_{ipc})$
Ensure: Atr
  $Atr \leftarrow Squared\_Matrix\_Zeros(size(Sets_{ipc}))$
  for all $A, B \in Sets_{ipc}$ **do**
    $A\_has\_B \leftarrow Size(A \cap B)/Size(B)$
    $B\_has\_A \leftarrow Size(A \cap B)/Size(A)$
    $A\_bigger\_B \leftarrow Size(A)/Size(B)$
    $B\_bigger\_A \leftarrow Size(B)/Size(A)$
    $Atr[B\_index, A\_index] = B\_bigger\_A * B\_has\_A$
    $Atr[A\_index, B\_index] = A\_bigger\_B * A\_has\_B$
  end for

Creation of the Quasi-Hierarchy

The quasi-hierarchy is defined by a list of sets and an adjacency matrix. The adjacency matrix is derived from the attraction matrix by determining whether the attraction values in the attraction matrix surpass the threshold th_{qh}. The quasi-hierarchy is established by applying the following rules to the values of Atr :

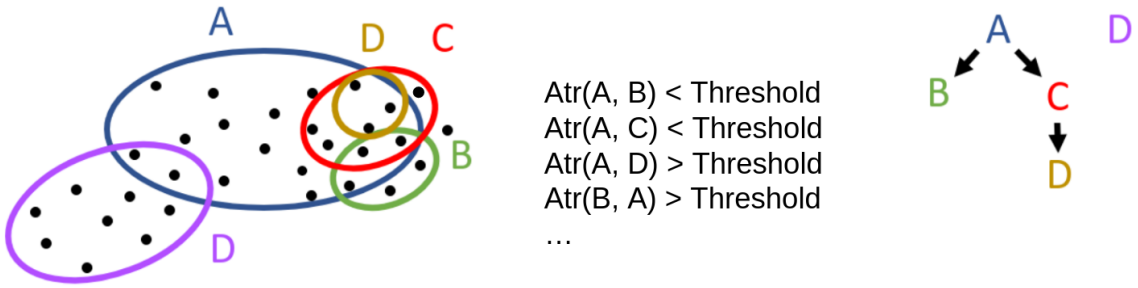


Figure 3.5 – The quasi-hierarchy allows to define relationship between sets that are intersecting

- A link between two subsets is established in the quasi-hierarchy if their attraction exceeds the threshold th_{qh}.
- Two subsets with strong mutual attraction (i.e., surpassing the threshold th_{qh}) are considered equivalent, and only one of them is retained. If the sets are of equal size, one of them is selected at random; otherwise, the smaller set is removed.
- The updated list of sets, along with their adjacency matrix, determines the quasi-hierarchy.

Algorithm 5 Quasi_Hierarchy : Ensures QuasiHierarchy

Require: $Sets_{ipc}$, Atr, th_{qh}
Ensure: $Sets_{qh}$, Adj_{qh}
  $Adj_{qh} \leftarrow Squared\_Matrix\_Zeros(size(Sets_{ipc}))$
  $Adj_{qh}[Atr > threshold] \leftarrow 1$
  for all $i, j \in Range(size(Sets_{ipc}))$ **do**
    if $Adj_{qh}[i,j] = 1$ **&** $Adj_{qh}[j,i] = 1$ **then**
      if $size\_of\_set(i) >= size\_of\_set(j)$ **then**
        remove set j from Adj_{qh} and $Sets_{ipc}$
      else
        remove set i from Adj_{qh} and $Sets_{ipc}$
      end if
    end if
  end for

3.4.3 Hyperparameters

The definition of the pretopological space has a significant influence on the formation of clusters. For instance, all n numeric features can be considered together, with their Euclidean distances calculated in one graph of G. Similarly, the Hamming distances for all categorical values of the dataset can be calculated in another graph of G. Through this generic parsing of data, we obtain a simple pretopological space. The DNF could be a logical « AND » or « OR » combination of the Euclidean and Hamming distances. However, features can be considered individually, each with its own graph, similarity measure, and threshold. This approach makes the DNF more extensive and specific.

Thresholds are automatically calculated to adapt to the number of points, the number of close neighbors each point has, and the dispersion in the dataset, or they can be set manually. Parameters in the threshold calculation function can be adjusted to obtain either high thresholds, yielding small clusters with low inner dispersion and a high number of outliers, or lower thresholds, yielding larger clusters with fewer outliers.

The threshold th_{qh} used in the construction of the quasi-hierarchy is usually fixed (to 0.1).

The DNF function defines the logical rules determining the formation of clusters. Using a logical AND (i.e., G_i AND G_j) creates a more constrained clustering, wherein clusters exhibit similar values for characteristics i and j. On the other hand, a logical OR (i.e., G_i OR G_j) results in less constrained clusters, wherein clusters show similar values for either characteristic i or j.

3.5 Metrics for clustering evaluation

To establish a benchmark, we need metrics. Some of those metrics are used to assess the cluster tendency of a dataset, while others are used to evaluate the result of a cluster analysis [122].

An important proportion of the datasets we use to compare the different algorithms have no feature considered as « true clusters », or this feature might not be relevant. Therefore, we do not focus on external indices that compare a clustering with « true clusters ». We mainly use internal indices, that evaluate the quality of a partition.

One of the characteristics of this study is the use of mixed data. As we do not use numerical-only data, we cannot use traditional clustering evaluation indices without preprocessing, as they often require a Euclidean space to compute. To use them, we use dimension reduction techniques to translate our data into a Euclidean space, then compute evaluation indices in this space.

3.5.1 Cluster tendency – Hopkins Statistic

To evaluate the results of a dimension reduction, or simply to discuss the cluster tendency of a dataset, we use the Hopkins Statistic from [65]. It behaves like a statistical hypothesis test, where the null hypothesis is that the datapoints are uniformly distributed. To compute it on a set X of n points in d dimensions :

- Generate $\tilde{X}$, a random sample of $m \ll n$ datapoints from X. [88] suggests sampling 5% of X.
- Generate Y, a set of m randomly and uniformly distributed datapoints.
- Define u_i the minimum distance of $y_i \in Y$ to its nearest neighbor in X.
- Define w_i the minimum distance of $\tilde{x_i} \in \tilde{X}$ to its nearest neighbor in X.

Then compute H the Hopkins Statisitc defined by :

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d} \tag{3.12}$$

H is bounded between 0 and 1. A value close to 1 indicates that the data has a high clustering tendency, its data points are typically much closer to other data

points than to randomly generated ones. A value close to 0 indicates uniformly spaced data, and values around 0.5 indicate random data. The Hopkins Statistic usually is a useful measure. However datasets with only one very dense cluster might obtain a high score, although running a cluster analysis over them would be pointless.

3.5.2 Cluster tendency – Improved Visual Assessment of Cluster Tendency

In partitional clustering, the question of cluster tendency, i.e the number of clusters necessary to obtain a good partitioning, can have a high influence of the final performance of an algorithm. Usually, it is manifested by an hyperparameter k inputted by the user before running the algorithm (e.g k-means, k-prototypes, ...). To address this question, [62] propose the Improved Visual Assessment of Cluster Tendency (iVAT) algorithm.

Given a dataset, a dissimilarity matrix can be computed. It is a square and symmetric matrix where each element represent the dissimilarity between two data objects of the dataset. Each element is scaled to the range $[0, 1]$, the value 0 describes the highest dissimilarity between two objects and the value 1 the lowest. From this matrix, a visual interpretation can be extracted which is an image of greyscale pixels, where each pixel represent the dissimilarity between two objects. Each pixel's colors depends on the value of the corresponding dissimilarity, such that the darker a pixel is, the lower the dissimilarity value is. The image is characterized by a black diagonal of pixels, because each data object is exactly similar with itself.

iVAT will reorder this matrix in order to have a visualisation of the cluster tendency. Reordering is done in a way to have one or more dark blocks along the diagonal of the image. A potential cluster is represented by a dark block, which is a submatrix with low dissimilarities values. Objects that are members of a dark block are relatively similar to each other. Cluster tendency is determined by the number of black blocks along the image diagonal.

iVAT can be a good alternative to Elbow Method, which can have decreased performannce in case of outliers in the dataset. However, iVAT is a visual method and the extraction of the number of cluster must be done by the user. Different viewers can have different interpretation of cluster tendency, especially in the case of unclear boundaries between the different dark blocks. To address this problem, [153] propose a similar algorithm named aVAT that uses some image processing techniques to determine automatically the number of cluster. Unfortunately, the source code is anavailable and the algorithm is not well documented.

In our benchmark, we use iVAT to determine the relevance of the computed Hopkins statistics on a dataset. Indeed, only knowing the clustering friendliness of a dataset through Hopkins Statistic can lead to a bad evaluation of the dataset. For example, a dataset with a good Hopkins statistic can present a cluster tendency of only one cluster through iVAT. In this case, clustering would be useless despite of the different interpreation we could have with Hopkins statistic.

3.5.3 Cluster analysis – Calinski-Harabasz

A standard index to evaluate the definition of clusters is the Calinski-Harabasz index from [26], also known as the Variance-Ratio Criterion. From a set of data points, and the result of a cluster analysis, we compute s as described in Equation 3.14. For a dataset E, with n_E individuals, divided into k clusters, the Calinski-Harabasz index is is the ratio of the sum of between-clusters dispersion and of within-cluster dispersion for all clusters (respectively B_k and W_k, defined in Equation 3.13), where dispersion is defined as the sum of distances squared.

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T \ , \ B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \tag{3.13}$$

with C_q the set of n_q points in a cluster q of center c_q, and c_E the center of E. The index s is calculated by :

$$CH = \frac{tr(B_k)}{tr(W_k)} \times \frac{n_E - k}{k - 1} \tag{3.14}$$

This index returns a positive real number, where a higher Calinski-Harabasz score relates to a model with better-defined clusters.

3.5.4 Cluster analysis – Silhouette

The Silhouette Coefficient, from [135], also evaluates the definition of clusters. It is only computed using pairwise distances. Therefore, it is not only possible to use it along with dimension reduction techniques, but also with Huang's Distance (Equation 3.3). A score is computed for each data point as described in Equation 3.15, using a the mean distance of a point with the other points of its cluster, and b the mean distance with the points of the nearest cluster.

$$silhouette = \frac{b - a}{max(a, b)} \tag{3.15}$$

The Silhouette Coefficient of a set of points is the mean of the Silhouette Coefficient for each sample. It is bound between -1 for incorrect clustering, and +1 for highly dense clustering. A score of zero indicates that clusters are overlapping.

3.5.5 Cluster analysis – Davies-Bouldin

To evaluate clusters separation, we use the Davies-Bouldin index from Davies and Bouldin in (1979). For each pair of clusters i and j, a similarity R_{ij} is computed (Equation 3.16). Then, the Davies-Bouldin index DB is the mean of the highest similarities for each cluster (Equation 3.17).

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{3.16}$$

where R_{ij} is the similarity between clusters i and j; s_i and s_j are the average distances of points of clusters i and j to their centroids; d_{ij} is the distance between the centroids of clusters i and j.

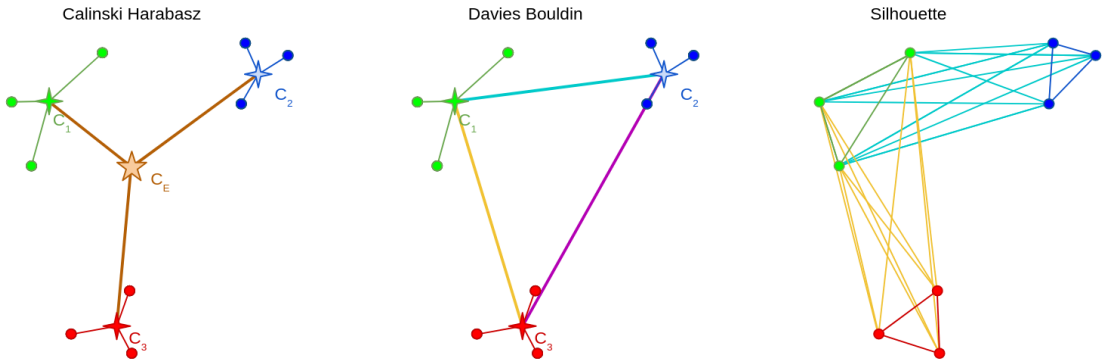


Figure 3.6 – Illustration of the different distances used in the calculation of the different cluster quality indices

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j} R_{ij} \tag{3.17}$$

A low Davies-Bouldin index indicates well-separated clusters, where zero is the lowest possible score.

3.5.6 In short

Index	Advantages	Drawbacks
CH	- Higher for well-defined clusters - Widely used in the litterature - Fast to compute	- Higher for convex clusters than other concepts of clusters (i.e. Density Based) - Needs a Euclidean Space
Silhouette	- Higher for well-defined clusters - Does not require a Euclidean Space - Bound between -1 and +1	- Higher for convex clusters - Results are often less eloquent than other indices
DB	- Low when clusters are well separated - Simple computation	- Higher for convex clusters - Needs a Euclidean Space

Table 3.2 – Advantages and Drawbacks of the different interval validation indices

Table 3.2 summarizes the advantages and drawbacks of the different internal validation indices. Those characteristics are given in terms of computation complexity, interpretability, and mathematical limitations.

3.6 Clustering of complex data (including time series)

The popularity of mixed data clustering algorithms has increased due to the prevalence of real-world datasets containing both numeric and categorical features. Various methods have been proposed for clustering mixed data, though a unified research framework is still lacking in this field [9]. Time series features

have also been an active area of research, with various methods proposed to address the challenges associated with handling time series.

However, there is little to no scientific literature addressing the clustering of elements defined by categorical, numerical, and time series features, despite the presence of such data in various fields studying complex systems. For example, a customer, a patient in a hospital, or a machine in any IoT context will not be described solely by fixed features, nor will they be described by time series alone. Fixed features are essential to cluster complex systems; for humans, it can be the date of birth, place of birth, or blood type. In a shorter timeframe, it can be address, socio-economic status, sex, height, or hobbies. For machines, buildings, or industrial plants, fixed features can be construction date, model name, specifications of any sort, and so on.

Many features describing humans, organizations, machines, or buildings are time series, such as weight, blood glucose, income, expenses, energy input or output, or temperature. In many contexts, these features are essential to administer a diagnosis, whether it is for medical diagnosis, energy performance recommendations, or predictive maintenance. Clustering involving fixed and fluctuating features is necessary to identify homogeneous groups of complex elements.

Another challenge tackled is the explainability, exploitability, and parametrization of heterogeneous and complex system clustering. Since unsupervised methods identify clusters in data without predefined labels, no clustering is inherently considered as the 'true' clustering. Ideally, the number of clusters, where they separate, and how depends on the specific needs and vision of each user and their context.

Hierarchical clustering is useful for handling this complexity, as it allows the user to identify coherent structures within each cluster, providing scalability and interpretability to the clustering. Allowing the user to adjust several clustering method parameters and easily understand their role in constructing clusters is also a way to enable more parametrization and interpretability of the clusters.

In the existing scientific literature, there is minimal focus on clustering data composed of numerical features, categorical features, and time series. However, there is substantial literature on mixed data clustering and time series clustering. We present relevant concepts and state-of-the-art methods for clustering and cluster evaluation of mixed data and time series.

The Curse of Dimensionality is a phenomenon that arises in high-dimensional spaces, particularly in clustering and machine learning tasks, where the increase in dimensions leads to exponentially larger search spaces, making it difficult for algorithms to operate efficiently [20, 150]. Furthermore, distance metrics that work well in lower-dimensional spaces may not be as effective in higher-dimensional spaces, leading to poor performance in clustering tasks [145]. This problem is particularly relevant in the context of complex data containing time series, as time series often have high dimensionality due to the numerous time points involved [150]. A solution to break this curse is often dimensionality reduction.

DR can be used to break the Curse of Dimensionality, and it is often employed as a preliminary step for clustering high-dimensional data. It can also be used to reduce mixed data. FAMD [46] is a DR technique specifically designed for such tasks. Additionally, although not initially adapted for mixed data reduction, Uniform Manifold Approximation and Projection (UMAP) [107] or Pairwise Control-

led Manifold Approximation Projection (PaCMAP) [154] can be adapted. UMAP is adapted by using the Huang Distance, that is suited for mixed data, and PaCMAP can be initialized with FAMD. Then, these techniques are able to convert a high-dimensional mixed dataset into a low-dimensional numerical dataset. Subsequently, state-of-the-art numerical clustering algorithms, such as K-means, can be applied to the transformed dataset, and cluster visualization on the low-dimensional data can be performed. DR is also a prevalent preprocessing approach for time series clustering, aiming to decrease the complexity and computational cost associated with high-dimensional data.

3.6.1 State of the Art on Time Series Clustering

Time series clustering has been an area of active research for several years due to its widespread applicability in fields such as finance, healthcare, and IoT [5] . The primary goal of time series clustering is to group similar time series, considering their temporal dynamics and patterns. This section reviews the state of the art in time series clustering, focusing on the major methods and techniques developed to address the unique challenges associated with time series data.

Distance-based Clustering

is one of the most common approaches for clustering time series. This approach computes pairwise distances between time series, using a distance metric to measure similarity. The most widely used distance metrics for time series clustering are Euclidean distance, Dynamic Time Warping (DTW) [116], and **Longest Common Subsequence (LCSS)** [41]. DTW is particularly popular because it allows for non-linear alignment between time series, providing a more flexible similarity measure compared to the Euclidean distance.

Feature-based Clustering

It involves extracting **time series features (TSF)** and using these features to represent the time series in a lower-dimensional space. This approach can reduce the dimensionality of the data and the computational complexity associated with clustering. Common features extracted from time series include statistical features (e.g., mean, standard deviation), frequency domain features (e.g., Fourier transform, wavelet transform), and shape-based features [51].

Multivariate time series feature extraction

It involves deriving additional features or new time series from the analysis of links between two or more time series. Specific features can be extracted depending on the case study; for instance, in building classification and clustering, features are often calculated based on outside temperature and energy consumption. In general, the extracted features involve the evaluation of the correlation between different time series [138]. The use of autoregressive modeling to form augmented-feature vectors [80] is also an option.

After feature extraction, traditional clustering algorithms, such as k-means or hierarchical clustering, can be applied to the reduced feature space.

Model-based Clustering

Those methods assume that each time series is generated by an underlying model, and the goal is to group time series based on the similarity of their generative models. Some popular model-based clustering techniques include clustering based on **Hidden Markov Models (HMM)** [119], **autoregressive models** [101], and **Gaussian Process models** [105]. These methods often require fitting models to each time series and comparing the models to compute pairwise similarities, which can be computationally expensive.

3.6.2 Cluster evaluation for complex data

Evaluating the quality of clusters is more challenging than evaluating classifications due to the absence of ground truth for comparison. Instead, the focus is on the quality of a partition, based on metrics such as dispersion and distances within and between clusters [122].

Calinski-Harabasz (CH)

The CH index [26] is a well-established metric for evaluating the definition of clusters. The index, also known as the Variance-Ratio Criterion, is computed as the ratio of the sum of between-cluster dispersion and within-cluster dispersion for all clusters, where dispersion is the sum of squared distances. A clustering with a high CH score indicates a model with well-defined clusters. This method provides a robust approach to assess the quality and explainability of mixed data clustering. A higher CH score is indicative of a model that exhibits more distinct and well-defined clusters.

Davies-Bouldin (DB)

The DB index [38] is used to assess the separation of clusters. The similarity between a pair of clusters is the ratio of the sum of the average distance in each of the two clusters and the distance between the centroids of the two clusters. The DB index is then computed as the average of the maximum similarities for each individual cluster. Lower values of the DB index signify better-separated clusters, with the minimum possible score being zero.

Silhouette Coefficient (SC)

The SC indicates how well-defined the clusters are. A score is calculated for each data point as shown in Equation 3.15. The SC for a group of points is determined by averaging the SC of each individual sample. The coefficient ranges between -1 for improper clustering and +1 for highly compact clustering. A score of zero implies that clusters are overlapping.

One issue with mixed data clustering is that these metrics are defined for numerical spaces. Therefore, the application of the DR techniques described earlier is necessary for any cluster evaluation. Similarly, for complex data, it must be reduced before the clusters are evaluated. In order to calculate the CH, DB, and SC

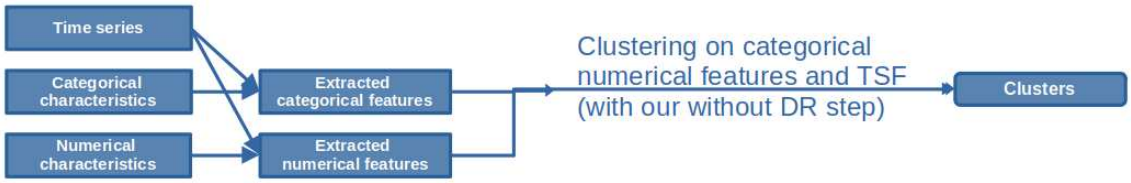


Figure 3.7 – Clustering on Mixed Features and Time Series based on distance or model.

scores, datasets are transformed into Euclidean spaces using FAMD, which ensures that the output space has the same number of dimensions as the original space. FAMD is chosen for its known inertia, deterministic nature, and minimal reliance on hyper-parameters. Additionally, since the SC score is the only index in the study that accepts a pairwise distance matrix as input, it is computed using the Gower matrix to prevent any bias towards FAMD or provide extra insights when FAMD has low inertia. We will call it the Gower Silhouette Coefficient (GSC).

3.6.3 Pretopology-based clustering for complex data

This subsection introduces the essential concepts and definitions in pretopology, such as pretopological space and pseudoclosure, before describing the primary algorithm for pretopological hierarchical clustering.

The primary insight obtained from this pretopological framework and its associated algorithm is that pretopology enables the abstraction of the complex nature of the elements being studied by focusing on the relationships between them based on their characteristics. Each characteristic has its own weighted graph, which allows the calculation of a distance for each characteristic. For example, a Manhattan distance can be computed for a pair of longitude and latitude coordinates, while a corresponding volume difference can be calculated for a 3D space describing an object's dimensions. Similarly, the distance between two highly time-dependent series can be measured using Euclidean space, while DTW can be employed to compare time series where the overall profile is more relevant. Once this set of graphs is defined, the DNF establishes the logical rules by which pseudoclosure, and consequently hierarchical clustering, are generated.

3.6.4 Methods for clustering complex data

We introduce various approaches that appear relevant for clustering complex data. These approaches are combinations of the different components presented in the state of the art. The figure illustrates a case study in which time series correspond to energy consumptions and weather values.

Method1 : Clustering on Mixed Features and Time Series using each time step as dimension

(Figure 3.7)

This approach involves using each time step of the time series as a numerical feature and applying mixed clustering methods such as K-prototype.

Advantages : Simple to implement and uses state-of-the-art mixed clustering methods.

Disadvantages : In this case, the « weight » of each measure of the time series is the same as other features, and simple numerical or categorical features will be overshadowed due to the sheer volume of time series values, leading to inadequate consideration in the resulting clusters.

Explainable Artificial Intelligence (XAI) : Medium.

Method2 : Clustering on Mixed Features and Time Series using specific distances

(figure 3.7)

In this approach, specific distances are calculated for particular features or groups of features. These distances are subsequently aggregated in the clustering process, using either a weighted sum or logical rules.

Advantages : All available information is fully exploited to create the most relevant clusters possible.

Disadvantages : This approach requires a deep understanding of the dataset and specification of the appropriate AHC or PretopoMD.

XAI : High to very high.

Method3 : Clustering on numerical data only via DR

(figure 3.8)

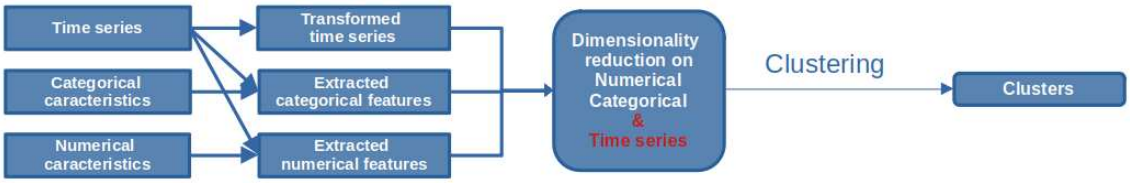


Figure 3.8 – Clustering on numerical data only via dr.

In this method, we use DR to create a low-dimensional numerical representation of all features (numerical, categorical, and time series). To apply DR on time series, we consider a time series as a point in a high-dimensional space, where each time step is a dimension.

Advantages : State-of-the-art numerical clustering methods can be applied.

Disadvantages : DR of long time series can be challenging due to the « Curse of Dimensionality ».

XAI : Very low.

Method4 : Clustering on mixed features and pre-clustered Time Series labels as categorical features

(figure 3.9)

In this method, we apply one or several time series clustering methods on each time series. We obtain a label corresponding to which cluster the time series belongs to. This label is then considered as a categorical feature. Mixed data clustering methods are then used on the enriched dataset.

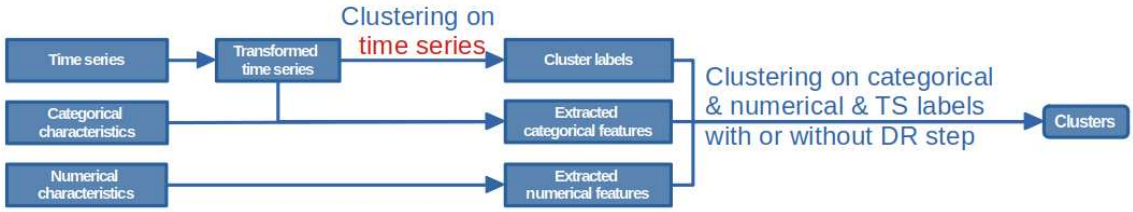


Figure 3.9 – Clustering on mixed features and pre-clustered Time Series labels as categorical features.

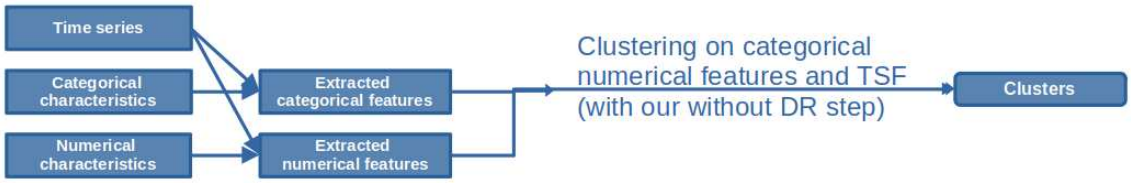


Figure 3.10 – Clustering on Mixed Features and Time Series Features

Advantages : The similarity between time series is considered through the time series clustering methods. State-of-the-art mixed dataset clustering methods can be applied.

Disadvantages : The choice of methods (including metrics and hyperparameters) can affect the quality of time series clusters labels.

XAI : Very high.

Method5 : Clustering on mixed features and pre-clustered time series labels as categorical features using DR

(figure 3.9)

In this method, the same preliminary steps as in method 4 are executed to obtain time series labels. Next, a DR method is applied to create a numerical dataset on which numerical clustering is performed.

Advantages : State-of-the-art numerical clustering methods can be applied.

Disadvantages : DR can create a loss of information and explainability.

XAI : Low.

Method6 : Clustering on Mixed Features and Time Series Features

(figure 3.10)

In this method, as in methods 4 and 5, we do not apply clustering on time series in their raw form. Instead, we extract TSFs to capture their essence in numerical and categorical representations. By doing so, we enable the application of mixed clustering methods on the extracted data.

Advantages : Mixed clustering methods can be applied. The features extracted are specific to the context and therefore can allow the clustering to be relevant from a field point of view

Disadvantages : The amount of information lost during preprocessing is important, though varying depending on the nature of the time series and the quality of the feature extraction process.

XAI : High.

Method7 : Clustering on Mixed Features and Time Series Features using DR

(figure 3.10)
We add a DR step to the sixth method.
Advantages : State-of-the-art numerical clustering methods can be applied.
Disadvantages : Information loss both during feature extraction and DR
XAI : Low.

3.6.5 Cluster Quality Indicators

Using the evaluation metrics presented in Section 3.5, we can asses the outcomes of various clustering algorithms.

To calculate the CH, DB, and SC scores, we transform datasets into Euclidean spaces using FAMD, ensuring that the output space has the same number of dimensions as the original space. We choose FAMD as the DR method because it is not too dependent on hyperparameters, because the inertia of the model is known (as it is a factorial method), and for its deterministic nature.

Moreover, since the SC score is the only index that can accept a pairwise distance matrix as input, we also compute it using the Gower matrix. This may prevent any bias towards FAMD and provide additional insights in cases where FAMD achieves low inertia.

It should be noted that in certain situations, an algorithm might produce a single cluster or only outliers. In such cases, we present the worst possible score or an infeasible value.

In this chapter, we explored the landscape of clustering mixed data, particularly focusing on what we call complex datasets that incorporate numerical, categorical, and time series features. We began by examining the current state of mixed data clustering research, identifying key methodologies such as partitional, hierarchical, model-based, and neural-network-based clustering. Each of these approaches offers unique approchaes for handling mixed data.

We delved into dimensionality reduction techniques, such as Factorial Analysis of Mixed Data (FAMD), Laplacian Eigenmaps, Uniform Manifold Approximation and Projection (UMAP), and Pairwise Controlled Manifold Approximation and Projection (PaCMAP), which are crucial for simplifying mixed datasets. These methods enhance algorithm compatibility and computational efficiency and allow powerfull visualisation, thereby facilitating more effective and insightful clustering.

Throughout our exploration, we reviewed specific clustering algorithms, including K-prototypes, Convex K-Means, KAMILA, ClustMD, MixtComp, DenseClus, and Pretopology-based clustering. Each algorithm was discussed in detail, with an emphasis on its suitability for mixed datasets and the unique advantages it offers for clustering such data.

Furthermore, we introduced the key concepts of pretopology and a foundational framework for organizing and analyzing data within a pretopological space.

The PretopoMD algorithms were proposed, illustrating the process of constructing closures and building hierarchical clustering from pretopological spaces. This approach emphasized the role of hyperparameters and function definitions in shaping the clustering process and the structure of the pretopological space.

To evaluate the quality of clusters, we presented metrics such as the Hopkins Statistic, Improved Visual Assessment of Cluster Tendency (iVAT), Calinski-Harabasz (CH), Silhouette Coefficient (SC), and Davies-Bouldin (DB) Indexes. These metrics allow us to assess the cohesion and separation of clusters, providing valuable insights into the effectiveness of different clustering methods.

In our discussion on clustering complex data, including time series, we emphasized the challenges and outlined several methods for clustering data with mixed features and time series. Each method was evaluated in terms of its advantages, disadvantages, and levels of explainability and interpretability (XAI), highlighting the importance of selecting the right approach for each specific dataset.

Through the detailed exploration of these methodologies, we have seen that the proper selection and tuning of these algorithms are crucial for achieving meaningful and actionable clusters. The challenges in clustering mixed data types have underscored the importance of understanding both the theoretical underpinnings and practical implications of each method. Our methodology has also highlighted the necessity of considering both the local and global structures of data, the balance between numerical and categorical data, and the need for algorithms to adapt to various data characteristics.

The insight given by this chapter will allow us to understand the implementation and results presented in the next chapter.

Summary of Chapter 3

Mixed Data Clustering Research : Explored various key approaches such as partitional, hierarchical, model-based, and neural-network-based clustering, identifying the diversity and scope of current methodologies.

Dimensionality Reduction Techniques : Highlighted the critical role of techniques like FAMD, Laplacian Eigenmaps, UMAP, and PaCMAP in simplifying mixed datasets, facilitating subsequent clustering processes.

Specific Clustering Algorithms : Reviewed algorithms specifically tailored for mixed datasets, including K-prototypes, Convex K-Means, KAMILA, ClustMD, MixtComp, DenseClus, and Pretopology-based clustering, offering insights into their unique advantages and applications.

Importance of Dimensionality Reduction and Algorithm Selection : Emphasized the necessity of effectively managing the diversity inherent in mixed datasets through careful selection of dimensionality reduction methods and clustering algorithms.

Pretopology Theory : Introduced fundamental concepts such as pretopological space and pseudoclosure function, establishing a comprehensive theoretical framework for organizing and analyzing complex datasets.

PretopoMD Algorithm : Detailed a comprehensive approach for constructing closures and building hierarchical clustering from pretopological spaces, underscoring the critical influence of hyperparameters in shaping the clustering outcome. The pseudocode for the algorithms is detailed.

Clustering Evaluation Metrics : Discussed key metrics including the Hopkins Statistic, iVAT, Calinski-Harabasz Index, Silhouette Coefficient, and Davies-Bouldin Index, crucial for assessing cluster quality in terms of cohesion, separation, and overall structure.

Clustering Methods for Complex Data : A classification of various approaches is made based on the possibilities offered by features creation and manipulation in complex datasets and their interaction with dimensional reduction and with numeric, mixed, or complex clustering methods, balancing the need to preserve data integrity, clustering explainability and with achieving meaningful clustering results, and adapting evaluation metrics to complex data scenarios.

Chapter 4

Datasets and results

This chapter delves into the comparative analysis of various clustering algorithms applied to a range of datasets, from publicly available to custom-generated, to private ones, each presenting unique challenges due to their mixed and complex nature. Through the lens of benchmark datasets such as Palmer Penguins, Heart Disease, and Sponge datasets, alongside a sophisticated custom dataset generator, we explore the performance, limitations, and practical implications of state-of-the-art clustering techniques. This investigation not only highlights the capabilities and shortcomings of each algorithm but also provides insights into their suitability for specific types of data challenges.

We will commence with a detailed examination of the datasets employed in our study, underscoring the rationale behind their selection and the specific challenges they pose. Following this, we will present the results of applying various clustering algorithms presented in chapter 3 to these datasets. This section will not only showcase the algorithms' performance but also discuss the computational costs, technical limitations, and the impact of dataset characteristics on the effectiveness of each clustering approach.

This analysis extends to complex data clustering, a crucial aspect where the interplay between numerical, categorical, and time-series data within a single dataset requires sophisticated strategies for effective clustering. These strategies are explored using a complex dataset generation tool. Finally the clustering of a private energy dataset is analyzed.

Through comprehensive analysis and evaluation, we seek to offer a roadmap for selecting appropriate clustering techniques based on specific dataset characteristics and clustering objectives.

4.1 Datasets

To compare the results of different clustering algorithms, we need to establish a benchmark. Therefore, we use multiple datasets used in the literature (Palmer Penguins [1], Heart Failure [2], Sponge [3]). Additionally, we have implemented a dataset generator to compare the algorithms across as many configurations as possible.

1. https://www.kaggle.com/datasets/parulpandey/palmer-archipelago-antarctica-penguin-data
2. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction
3. https://archive.ics.uci.edu/ml/datasets/sponge

4.1.1 Public datasets

Palmer Penguins

The first results we present (others are available on the Github) are done on the Palmer Penguins dataset. This dataset is built upon physical measurements of 344 penguins in the Palmer Archipelago, in Antarctica [55]. It contains 4 numerical and 4 categorical features. We use it as a base case, as it is widely used in the literature, and its shape is pretty common. Also, it has high clustering tendency over the different dimension reductions (Figure 4.1), especially over UMAP and PaCMAP.

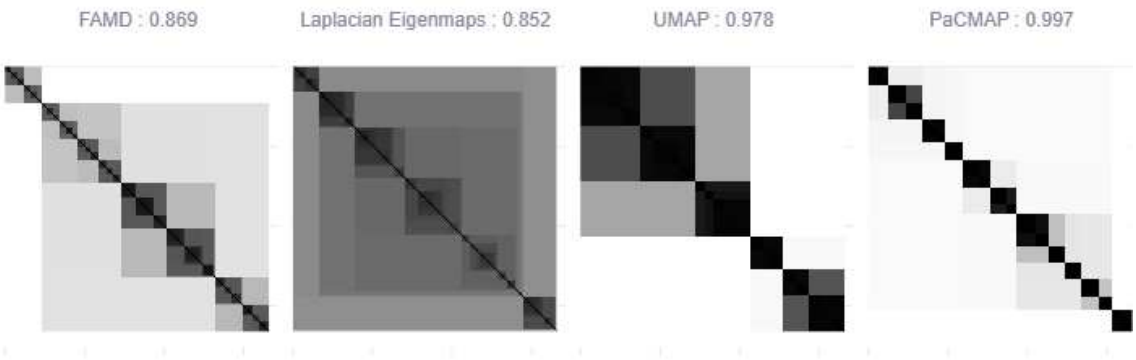


Figure 4.1 – Hopkins Statistic and iVAT for every dimension reduction over the Palmer Penguins dataset.

Heart Disease

The Heart Disease dataset belongs to the field of medicine. It combines 5 datasets over 13 features (5 numerical, 4 categorical, 4 ordinal). It contains 918 observations. Mixing in equal numbers each kind of features makes this dataset complex and the choose of metric or Dimensionality Reduction (DR) may completely change the clustering results.

Sponge

The Sponge dataset also belongs to the field of marine biology. Its aims is to describe and classify marine sponges. It has a pretty uncommon shape, as it only contains 75 individuals, with 42 categorical and 3 numerical features. Having both few individuals and a lot of categorical features makes this dataset harder to process, therefore interesting in the context of benchmarking.

4.1.2 Dataset generator

4.1.2.1 Mixed dataset

To evaluate the different algorithms over every desired configurations, we use a dataset generator. The most common way to generate datasets to benchmark and evaluate clustering algorithms is to generate isotropic gaussian blobs. This

method is natively present in the widely used scikit-learn for Python by [126], MixSim for R by [109] and Linfa for Rust[4].

First, we generated cluster centers, with an average pairwise distance of 1. Then we generate samples from a gaussian mixture model with the density described by :

$$p(x) = \frac{1}{k}\sum_{i=1}^{k}\mathcal{N}(\mu_i, \Sigma_i) \tag{4.1}$$

where :
- k is the number of clusters
- μ_i are the cluster centers
- Σ_i refers to the cluster covariances. Here, it is a diagonal matrix of the clusters variance.

Inspired by [35], we split features upon quantiles to transform them into categorical features. Thus, we get a mixed dataset. With this method, the different parameters we can tune to obtain different configurations are :

- The number of samples to generate (the number of individuals);
- The number of clusters k;
- The number of numerical features;
- The number of categorical features;
- The number of unique values taken by categorical variables;
- The standard deviation of clusters.

4.1.2.2 Complex dataset generator (with time series)

Since health datasets can be technically long to explain and to display, we present a generated dataset with categorical, numerical features and time series.

To evaluate the clustering methods, we generated a dataset consisting of elements characterized by four features : their position in a 2D space (numerical), their size (numerical), their shape (categorical with four possible values), and a time series consisting of a hundred data points. The dataset comprises 50 elements. The motivation for selecting such a dataset was to enable visualization without the need for DR, allowing for a direct understanding of the cluster construction (see Figures 4.2 and 4.23). This approach demonstrates how the logical rules defining the PretopoMD algorithm can enable customized clustering that addresses specific field requirements.

4.1.3 Private Datasets

4.1.3.1 Energy dataset

A proprietary dataset was constructed using data from Energisme, encompassing a diverse range of building characteristics. This dataset is comprehensive, encompassing categorical, numerical, and time series data to describe each building thoroughly.

4. https://rust-ml.github.io/linfa/

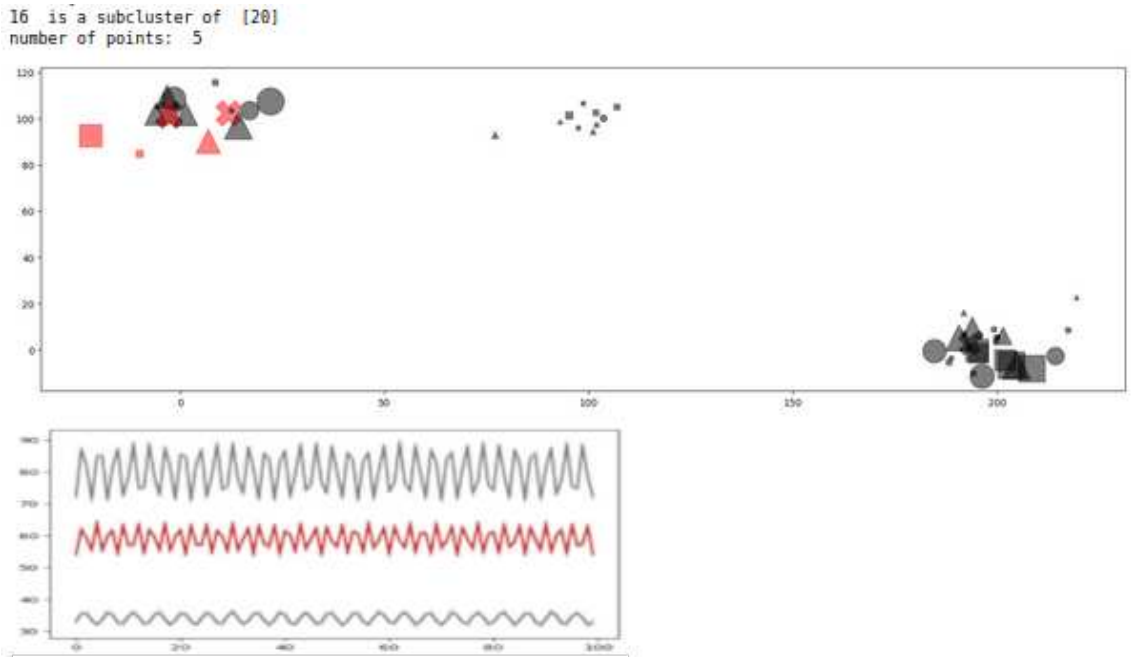


Figure 4.2 – In this example, the hierarchical clustering has been made using the Disjunctive Normal Form (DNF) condition *Position AND TS*. Thus, the subcluster elements are spatially close and have similar time series.

For categorical data, buildings are classified based on a detailed typology system, including Section, Division, and specific activities. An example of this classification would be : Section : Trade ; Division : Retail Trade, Excluding Automobiles and Motorcycles ; Activity : Retail Sale of Clothing in Specialized Stores.

The numerical data encompasses geographical coordinates (latitude, longitude, altitude), the building's surface area, and the year of construction.

Time series data provides a dynamic view of each building, capturing electricity and/or fuel consumption, along with meteorological conditions such as temperature, humidity, and sunlight.

A critical aspect of preparing this dataset involves cleaning and preprocessing the time series data. This process includes standardizing the time steps across the series and filling in minor gaps to maintain data continuity.

Additionally, we derive new data from the existing ones. For instance, consumption per square meter is calculated to provide a more nuanced understanding of energy usage. Thermal sensitivity is determined through a linear regression between the change in consumption and the variation in Heating degree days

Several transformations are applied to the consumption time series data to aid in analysis. The first transformation involves smoothing the consumption over a two-year period to facilitate straightforward comparison of consumption patterns. Another transformation creates an 'average week' of consumption, calculated by averaging the values for each 10-minute interval over a week. Therefore 1008 average values are calculated. This approach effectively captures the building's typical energy usage pattern without losing crucial temporal information.

The use of Dynamic Time Warping (DTW) was initially considered for calculating the average week in our analysis. However, a significant limitation of DTW in this context is its potential to obscure the precise timing of consumption events. In

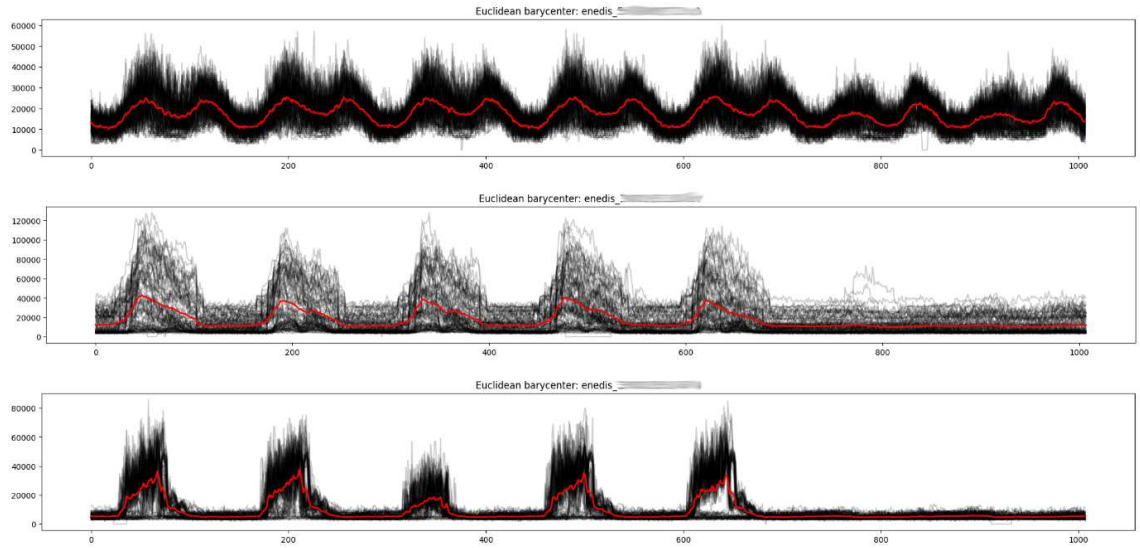


Figure 4.3 – Example of week euclidan barycenters, or « average weeks »

our case study, two similar consumption patterns occurring at different times are not equivalent, as they provide crucial insights into the building's usage patterns. Therefore, to preserve this vital temporal information, DTW was ultimately not employed.

Similarly, similarity metrics based on DTW were deemed unsuitable for the clustering process due to their potential to misrepresent the timing of consumption patterns.

To supplement our analysis, we extracted various numerical features from the time series data. These included the average, standard deviation, percentiles, and the dates of maximum and minimum consumption. These calculations were performed not only on the entire time series but also on the derived average week time series, providing a comprehensive view of consumption patterns over different time scales.

4.2 Clustering Results

For the results, we will provide the analysis over the Penguins dataset and the dataset generator. More results are available on the Github.

4.2.1 Computation cost and technical limitations

Clustering algorithms are generally computation-heavy. Their respective computation times and memory usage should not be neglected, as they could cause technical limitations. The following execution statistics are obtained upon testing on a configuration with an AMD Ryzen 7 5800H CPU, on a 3.20GHz frequency with 512KB of L1 cache and 32GB of DDR4 RAM.

To benchmark the memory usage and computation time of the different algorithms, we measure those indices over several different generated datasets. The aim is to determine the impact of the dataset characteristics (number of individuals, number of numerical and categorical features) on its computation cost.

To do so, we start from a « base configuration » (Figure 4.4) with 500 individuals, 5 numerical and 5 categorical features. Then, we evaluate the impact of those 3 characteristics on the memory usage and computation time. Our measurements only include the clustering algorithm (not the data generation phase). We then measure the memory usage of this algorithm every 10 times/second, and keep the maximum.

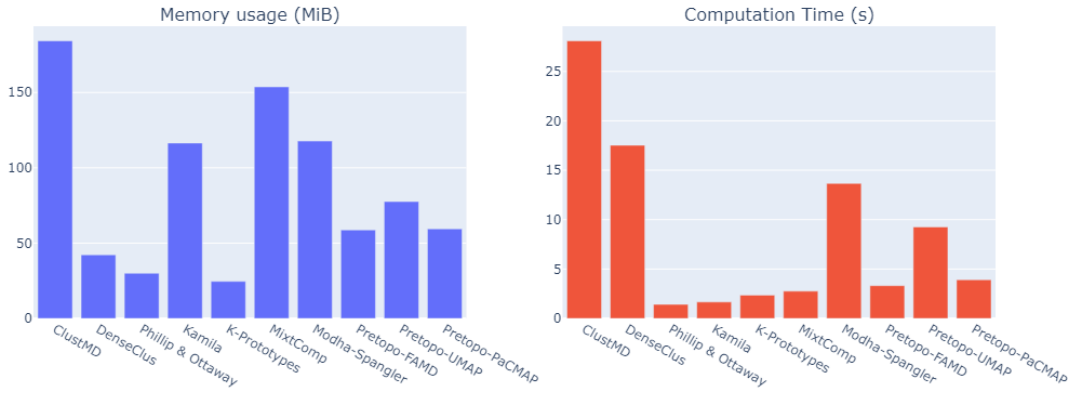


Figure 4.4 – Time and Memory usage of the different algorithms, on a base case with 500 individuals, 5 numerical and 5 categorical features.

4.2.1.1 Number of Individuals

First, we evaluate the impact of the number of individuals on the computation time and memory usage. We include configurations with 50, 100, 250, 500, 1000, 1750, 2500 and 5000 individuals. From Figure 4.5, we note that we have significative differences between the algorithms. The different variations of the pretopological algorithm have a steep curve, meaning that their memory usage increases faster than the other algorithms. On the other hand, we may note that most of the algorithms have similar memory usages with 5000 individuals.

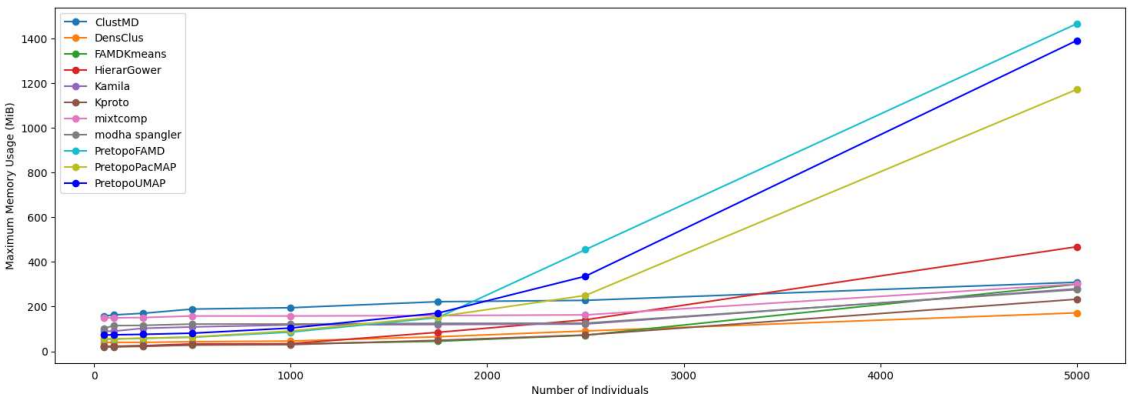


Figure 4.5 – Maximum memory usage depending on the number of individuals

Concerning the computation time (Figure 4.6), the pretopological algorithms' curves are closer to being linear, even if steeper than most algorithms. Yet the

UMAP version takes 6 times more time with 5000 than with 2500 individuals. We may also note that ClustMD obtains very high computation time that may cause technical limitations, even if it seems linearly related to the number of individuals.

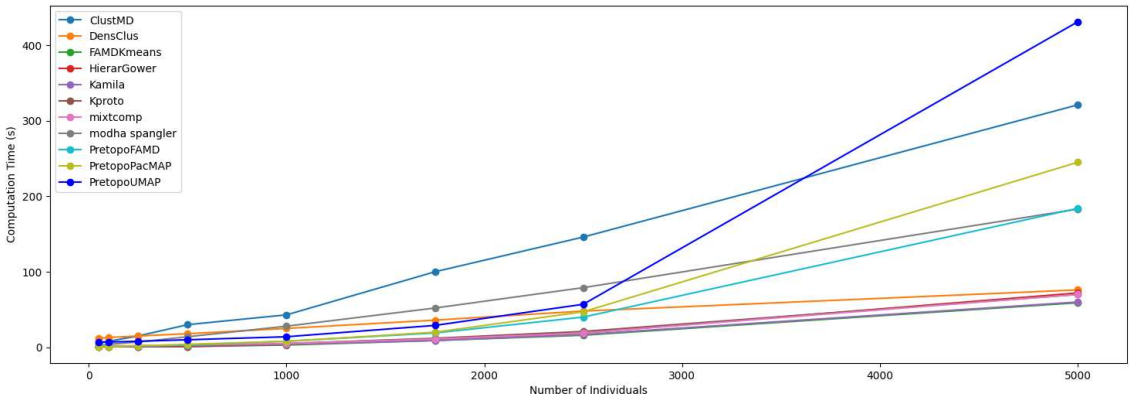


Figure 4.6 – Computation time depending on the number of individuals

4.2.1.2 Number of dimensions

Then, we must evaluate how the number of dimensions impact the computation time and memory usage of the algorithms. As some algorithms treat numerical and categorical features in a totally different fashion, we evaluate their respective impacts separately. We measure the computation time and memory usage on generated datasets with 2, 5, 10, 20, 50 and 100 numerical/categorical features (depending on the characteristic we evaluate).

Number of Numerical Features

The number of numerical features seemingly has less impact on memory usage than the number of individuals (Figure 4.7). Most algorithms barely use more memory with 100 numerical dimensions than with 2, so their results in terms of memory stay close to the base case. However, ClustMD's results are close to quadratic, and could cause limitations.

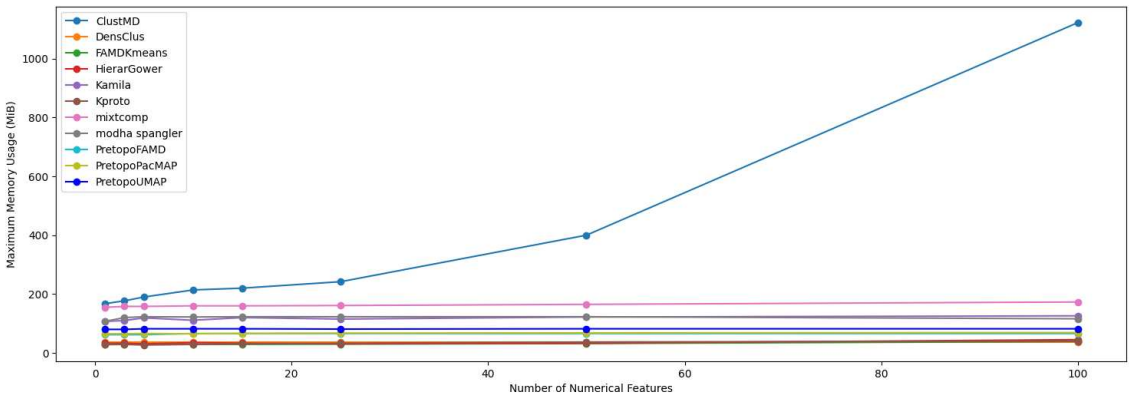


Figure 4.7 – Maximum memory usage depending on the number of numerical features.

Observing the execution time leads to similar results than the memory usage (Figure 4.8). The number of numerical seems to have close to no impact there, except on ClustMD. We might also note that Modha-Spangler's execution time also increases slightly.

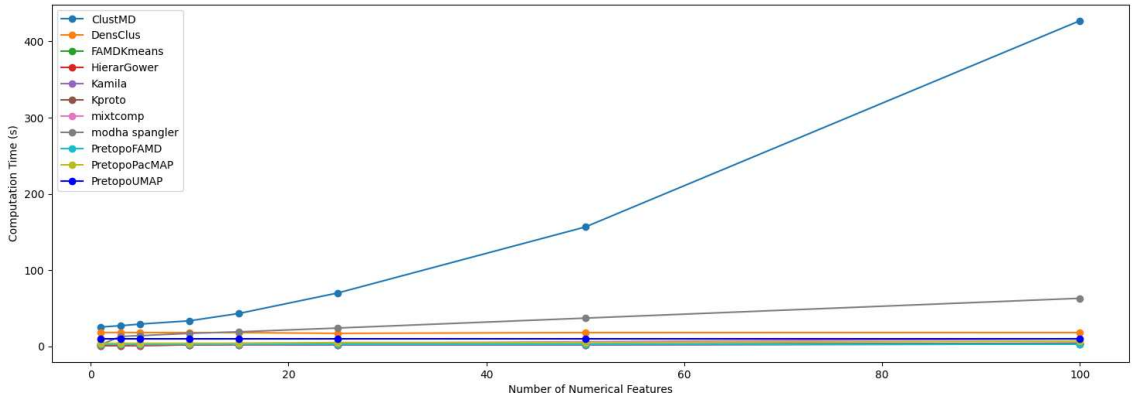


Figure 4.8 – Computation time depending on the number of numerical features.

Number of Categorical Features

Measuring the memory usage of the algorithms over datasets over the number of categorical features leads to results very close to numerical features' ones, for every algorithm (Figure 4.9).

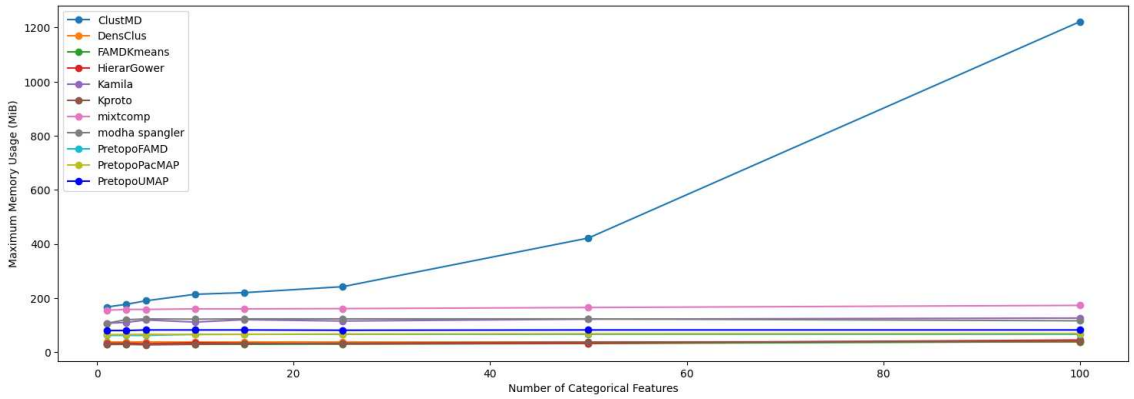


Figure 4.9 – Maximum memory usage depending on the number of categorical features.

In terms of computation time (Figure 4.10), the main difference in the impact of the number of categorical and numerical features can be observed in MixtComp. Its computation time is close to linearly related to the number of categorical features, while it was increasing slower upon the number of numerical features.

4.2.1.3 Discussion

Determining the number of clusters

A significant number of algorithms require K, the number of clusters, as a parameter. When K is unknown, the Elbow Method is commonly employed to de-

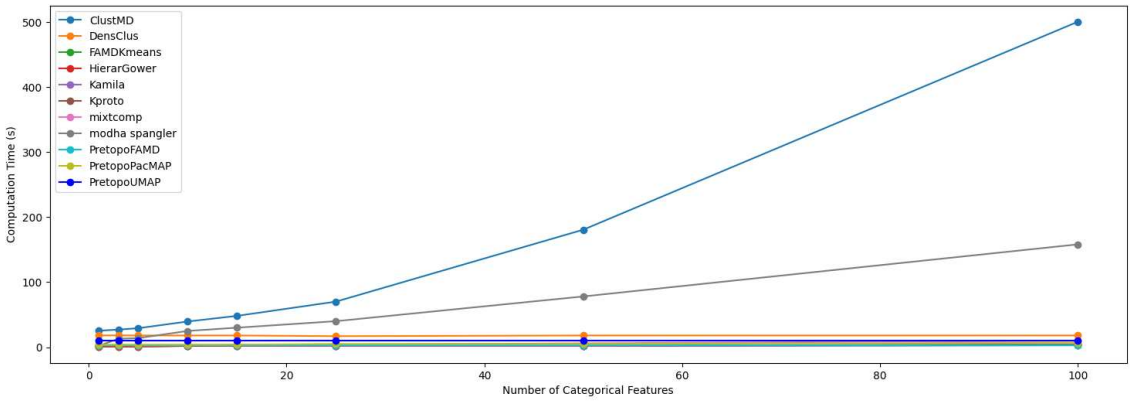


Figure 4.10 – Computation time depending on the number of categorical features

termine it. Typically, this involves plotting the explained variation as a function of the number of clusters and selecting the 'elbow' of the curve as the optimal number of clusters. The underlying intuition is that increasing the number of clusters naturally improves the fit, explaining more variation due to the increased number of parameters, but that beyond a certain point, it leads to overfitting. The 'elbow' in the plot reflects the point between optimization and overfitting.

Nevertheless, in mixed data clustering, more clusters rarely equate to better performance indicators, diverging from this traditional approach. Therefore, for mixed data, the Elbow Method combines the Gower distance with the Calinski-Harabasz metric for each value of K in a K-Means algorithm. With the Calinski-Harabasz index, we look for the maximum value, which signifies the 'elbow', contrasting with the traditional method. This adaptation is demonstrated in figure 4.11.

However, employing the Elbow Method is computationally demanding, sometimes exceeding the time required for the actual clustering process. Figure 4.12 illustrates the memory usage over time during various phases of the Philip and Ottaway algorithm, applied to a generated dataset comprising 1000 individuals and 50 features of each type. While the Elbow Method does not consume more memory than the actual clustering process, it requires more than twice the time. Yet, every algorithm needing K as a parameter would take at least this time to process.

4.2.2 Mixed Clustering Results

With the discussed materials and measures, we are able to evaluate the results of the different clustering algorithms.

In order to compute the Calinski-Harabasz, Davies-Bouldin and Silhouette scores, we translate datasets into Euclidean spaces using Factorial Analysis of Mixed Data (FAMD), with the output space having the same number of dimensions as the initial space. Here, FAMD is chosen over other techniques for the following reasons :

- It is a factorial method, the inertia of the model is known
- It is deterministic
- It does not rely heavily on hyper-parameters

Also, as the Silhouette score is the only index of the study that can take a pairwise distance matrix as an input, we compute it with the Gower matrix. It might avoid a bias towards FAMD, or just add to the analysis in cases FAMD obtains low inertia.

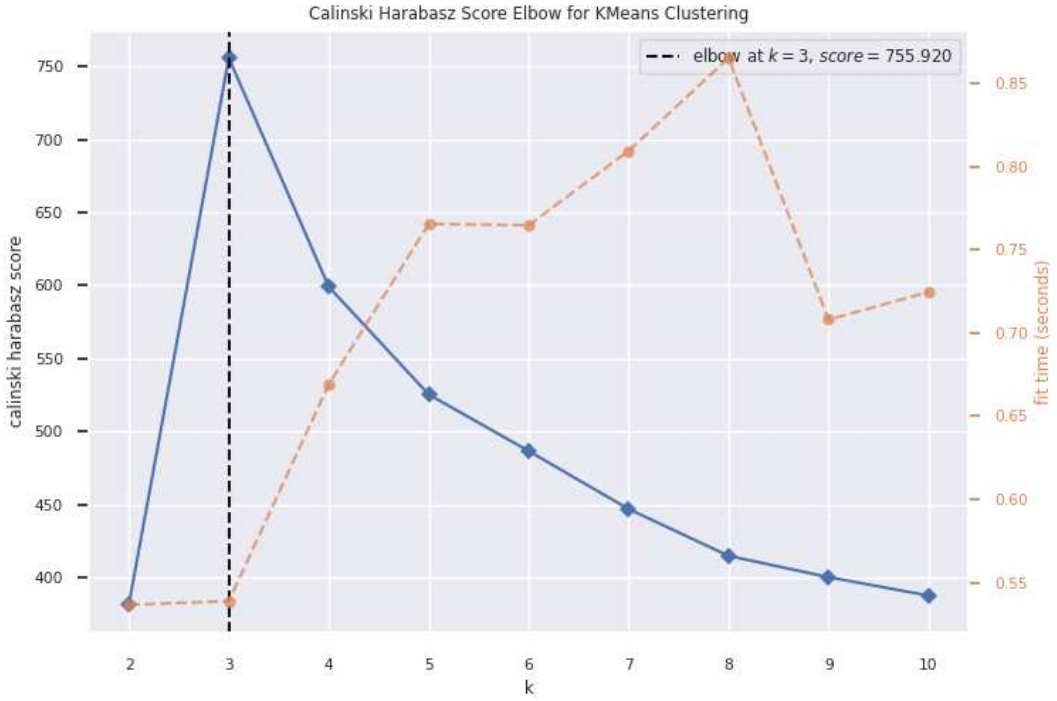


Figure 4.11 – Determining the number of clusters using k-means and the Gower distance with Calinski-Harabasz metric on the base generated dataset

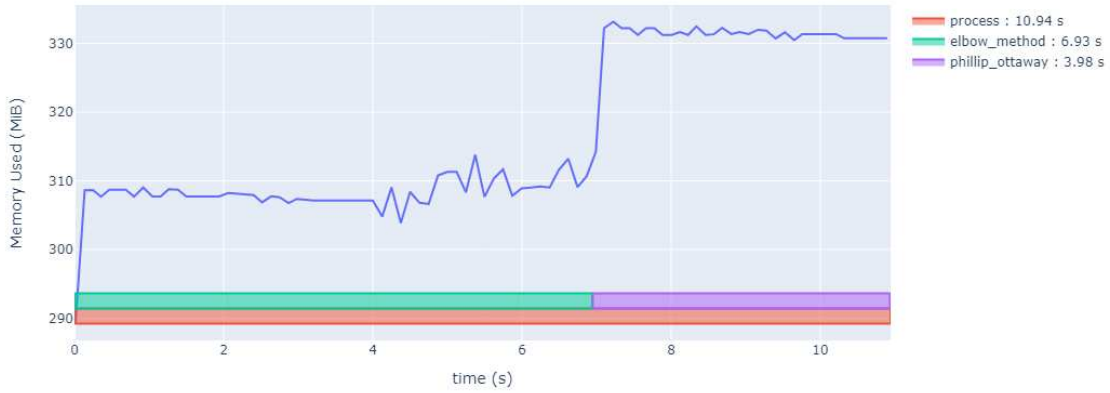


Figure 4.12 – Impact of the Elbow Method on the computation cost.

Also, note that in some cases an algorithm may return only one cluster, or only outliers. In those cases, we display « - » in the results table. The following results come from the Penguins dataset, more results are available on the Github.

Palmer Penguins

DenseClus and algorithms utilizing the Elbow Method segregate the Palmer Penguins dataset into two clusters. Being constrained by the Elbow Method to identify the same quantity of clusters—two in this instance—these algorithms se-

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	**169.74**	0.33	0.44	1.24
DenseClus	**169.74**	0.33	0.44	1.24
Phillip & Ottaway	**169.74**	0.33	0.44	1.24
Kamila	**169.74**	0.33	0.44	1.24
K-Prototypes	**169.74**	0.33	0.44	1.24
MixtComp	**169.74**	0.33	0.44	1.24
Modha-Spangler	**169.74**	0.33	0.44	1.24
Kmeans-FAMD	**169.74**	0.33	0.44	1.24
Pretopo-FAMD	158.70	**0.65**	**0.65**	**0.75**
Pretopo-UMAP	**169.74**	0.33	0.44	1.24
Pretopo-PaCMAP	52.93	0.29	0.24	1.12
Pretopo-Louvain	70.05	0.20	0.29	2.38
Pretopo-Laplacian	2.05	-0.41	-0.52	2.36
PretopoMD	105.17	0.24	0.26	1.72

Table 4.1 – Results of the selected Algorithms on the Palmer Penguins dataset.

lect an identical partitioning of the dataset, resulting in equal outcomes (refer to Table 4.1 for details). The Pretopo MD algorithm too identifies two clusters, and with more balanced quantities. However, these clusters yield lower scores when evaluated with the chosen metrics. This is not mirrored in the three versions of the pretopological algorithm using dimenssionality reduction. The UMAP variant delineates three distinct clusters, while the PaCMAP version identifies eleven, accompanied by 112 outliers. The FAMD version of the pretopological algorithm, on the other hand, segments the data into twenty-six clusters. It is noteworthy that the algorithm producing the highest number of clusters achieves the most optimal indices, suggesting that its partition carries extensive information about the dataset. The FAMD inertia for this dataset reaches a noteworthy 98.2%, indicating a high level of data variance representation.

Sponge

The Sponge dataset has many categorical features, yet has a relatively small sample size. FAMD applied to this dataset has a lower inertia than the Penguin dataset (86.13%). It also results to a low Hopkins statistic (0.63), indicative of a weak clustering tendency. Visualization of internal clustering structure (iVAT) further reinforces this observation, providing no clear evidence of inherent cluster structure. Conversely, PaCMAP yields a substantially higher Hopkins statistic (0.88) and a more distinct iVAT, suggesting a comparatively easier task for subsequent clustering algorithms, though not necessarily better clusters in the end. The Calinski-Harabasz (CH) scores are roughly an order of magnitude lower than those observed with the Penguin dataset, signaling the existence of less well-defined clusters. In fact, all cluster evaluation metrics for the Sponge dataset are poorer than those for the Penguin dataset, consistent with the reduced clustering tendency. Pretopological FAMD clustering demonstrates a notably low Davies-Bouldin (DB) score, yet underperforms on other indices. It identified 62 outliers and created 6 clusters,

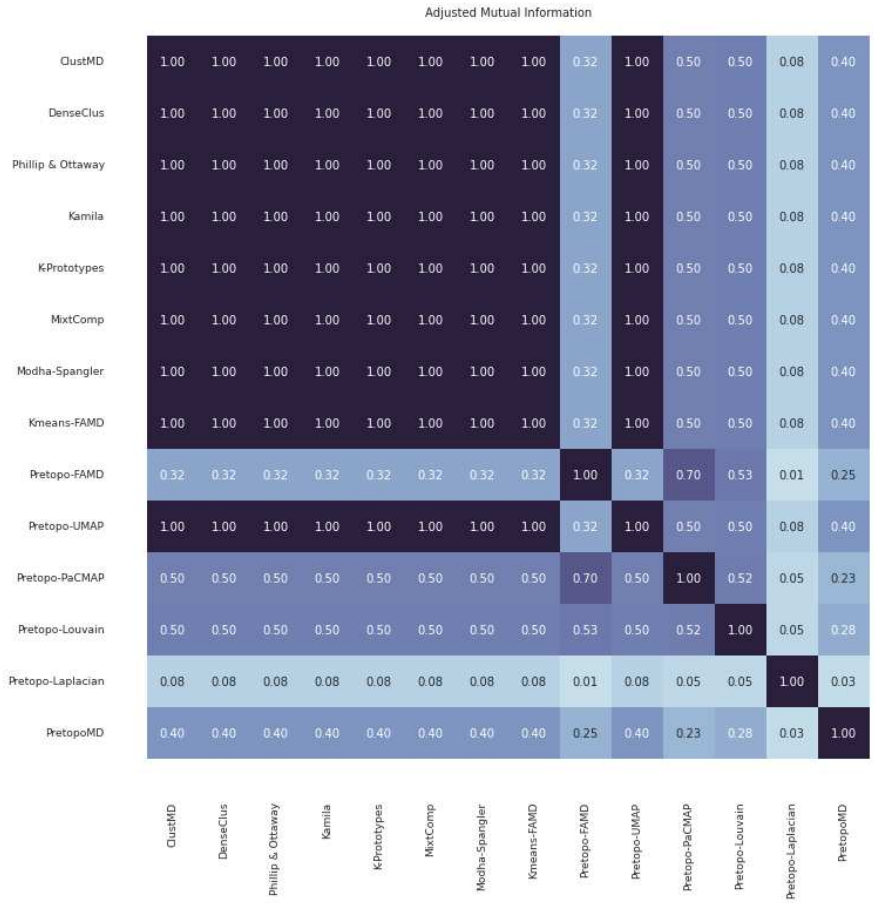


Figure 4.13 – Adjusted Mutual Information (AMI) of the selected Algorithms on the Palmer Penguins dataset.

each comprising 2 to 3 elements. In contrast, PretopoMD managed to identify a singular, prominent cluster of 74 elements and a single outlier. Considering the weak clustering tendency of the dataset, this result appears pertinent, earning it the highest FAMD Silhouette score and the lowest DB score.

Heart Disease

On this large dataset, the Elbow Method suggests 2 clusters. The algorithms utilizing the Elbow Method consequently identify 2 clusters, decompozing the dataset in two with one cluster representing the two third of the dataset. In this dataset, the best-performing algorithms are Phillip & Ottway, Kmeans on FAMD reduced dataset, PretopoMD on Louvain reduced dataset, and PretopoMD, with PretopoMD obtaining the top score on two indicators with two clusters. Pretopo-Louvain identifies four distinct clusters and no outliers. The inertia from FAMD is notably high at 99.9%, indicating that the dimension reduction process successfully captured all the variance present in the original dataset.

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	0.000	-1.000	-1.000	-1.000
DenseClus	0.000	-1.000	-1.000	-1.000
Phillip & Ottaway	9.706	0.168	**0.418**	2.226
Kamila	10.220	0.121	0.224	2.454
K-Prototypes	10.239	0.108	0.311	2.726
MixtComp	0.000	-1.000	-1.000	-1.000
Modha-Spangler	**10.322**	0.112	0.234	2.367
Kmeans-FAMD	10.228	0.141	0.328	2.471
Pretopo-FAMD	1.520	-0.071	-0.168	1.500
Pretopo-UMAP	6.884	0.074	0.142	2.828
Pretopo-PaCMAP	0.000	-1.000	-1.000	-1.000
Pretopo-Louvain	5.107	0.041	-0.037	2.680
Pretopo-Laplacian	5.805	0.068	0.109	2.342
PretopoMD	6.371	**0.484**	0.013	**0.384**

Table 4.2 – Results of the selected Algorithms on the Sponge dataset.

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	32.416	0.090	0.117	2.855
DenseClus	0.000	-1.000	-1.000	-1.000
Phillip & Ottaway	**72.814**	0.214	**0.416**	1.755
Kamila	51.922	0.161	0.209	2.298
K-Prototypes	37.488	0.115	0.109	2.708
MixtComp	61.022	0.212	0.414	1.947
Modha-Spangler	50.625	0.159	0.196	2.326
Kmeans-FAMD	**72.814**	0.214	**0.416**	1.755
Pretopo-FAMD	12.021	-0.029	-0.049	1.910
Pretopo-UMAP	0.000	-1.000	-1.000	-1.000
Pretopo-PaCMAP	40.654	0.116	0.211	1.904
Pretopo-Louvain	22.982	0.054	0.003	2.524
Pretopo-Laplacian	8.916	**0.404**	0.216	2.120
PretopoMD	67.883	0.248	**0.416**	**1.311**

Table 4.3 – Results of the selected Algorithms on the Heart Disease dataset.

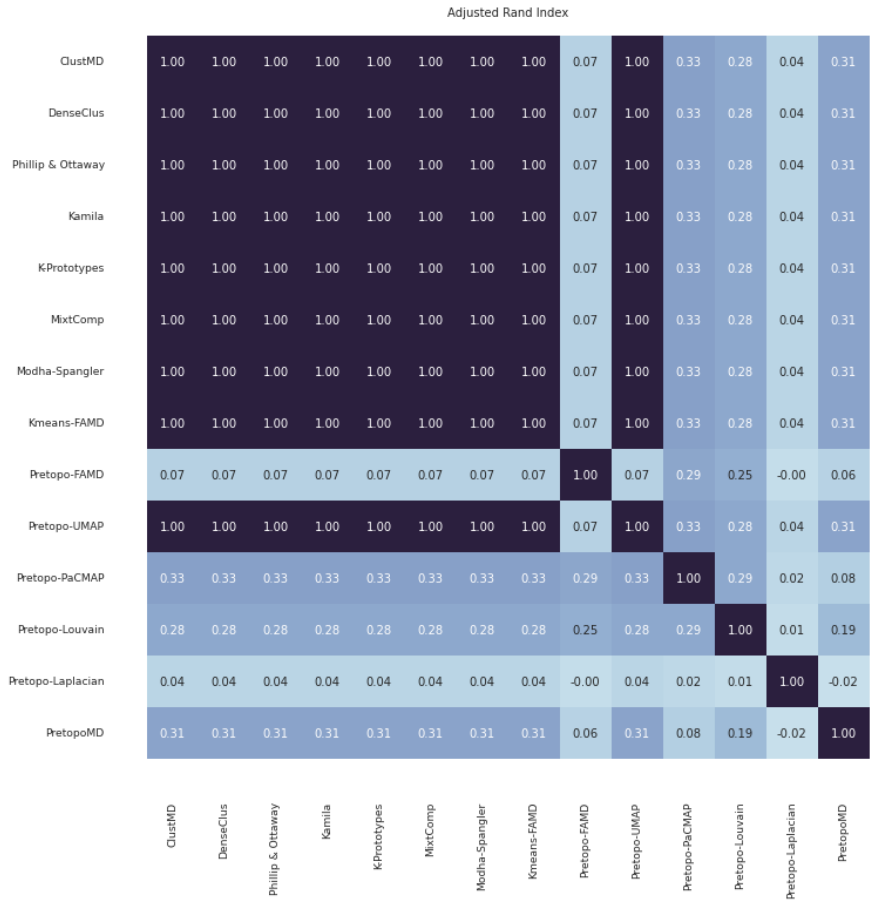


Figure 4.14 – Adjusted Rand Index (ARI) of the selected Algorithms on the Palmer Penguins dataset.

Base Generated Case (500 individuals, 5 features num/cat, 3 clusters, 3 cat uniques, 0.1 std)

The algorithms were also tested on generated datasets. In the base configuration, the Elbow Method determines k=3 as the optimal number of clusters. Since this aligns with the intended number of clusters, algorithms that utilize the Elbow Method have an advantage. Consequently, these algorithms produce very similar partitions of the dataset, with results closely aligned across all four indices. The two algorithms that employ UMAP, DenseClus and PretopoMD on UMAP reduced dataset, yield similar results even without the Elbow Method, while Pretopo-FAMD reports 444 outliers. PretopoMD on PaCMAP reduced dataset identifies six clusters and 124 outliers. Meanwhile, PretopoMD detects three sizable clusters and 60 outliers. As observed with other datasets, adjusting the hyperparameters might either improve or deteriorate the results obtained with PretopoMD.

Generated dataset with 10 clusters

We generated a dataset using the same parameters as the « base case, » but with a distinction : it now contains 10 clusters. If a dataset has more clusters, while retaining the same number of individuals and deviation, it might pose a challenge

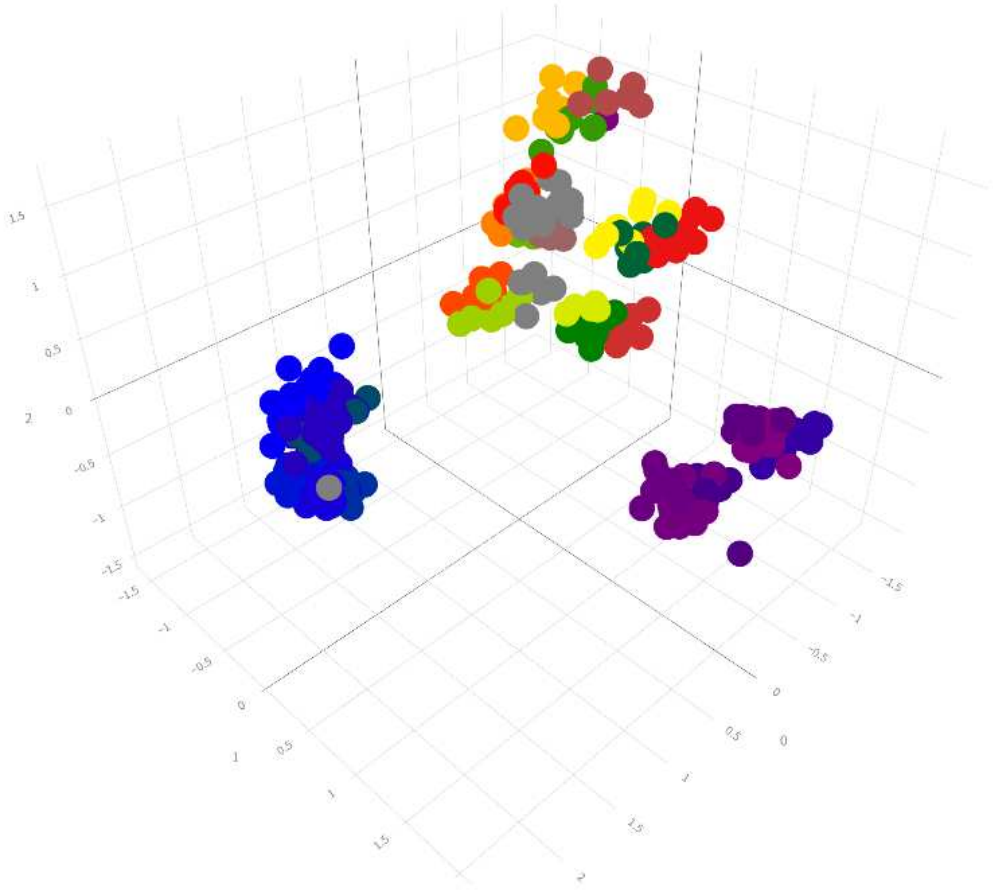


Figure 4.15 – 26 clusters identified by PretopoMD on FAMD reduced dataset in the FAMD reduced space representing the penguins dataset

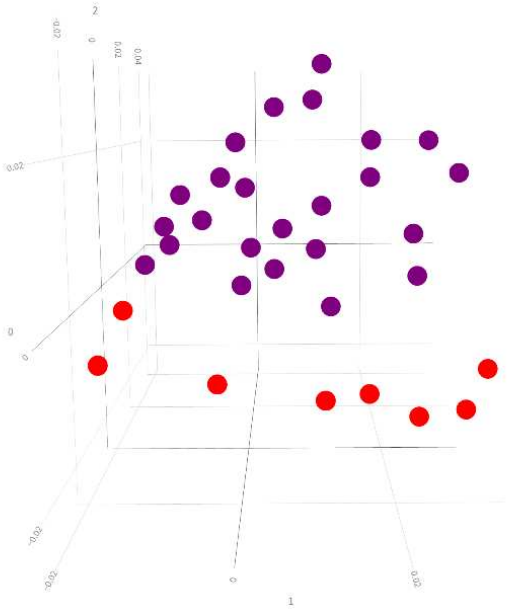


Figure 4.16 – 2 clusters identified by both KAMILA and K-Prototype in the Laplacian Eigenmap reduced space of the sponge dataset

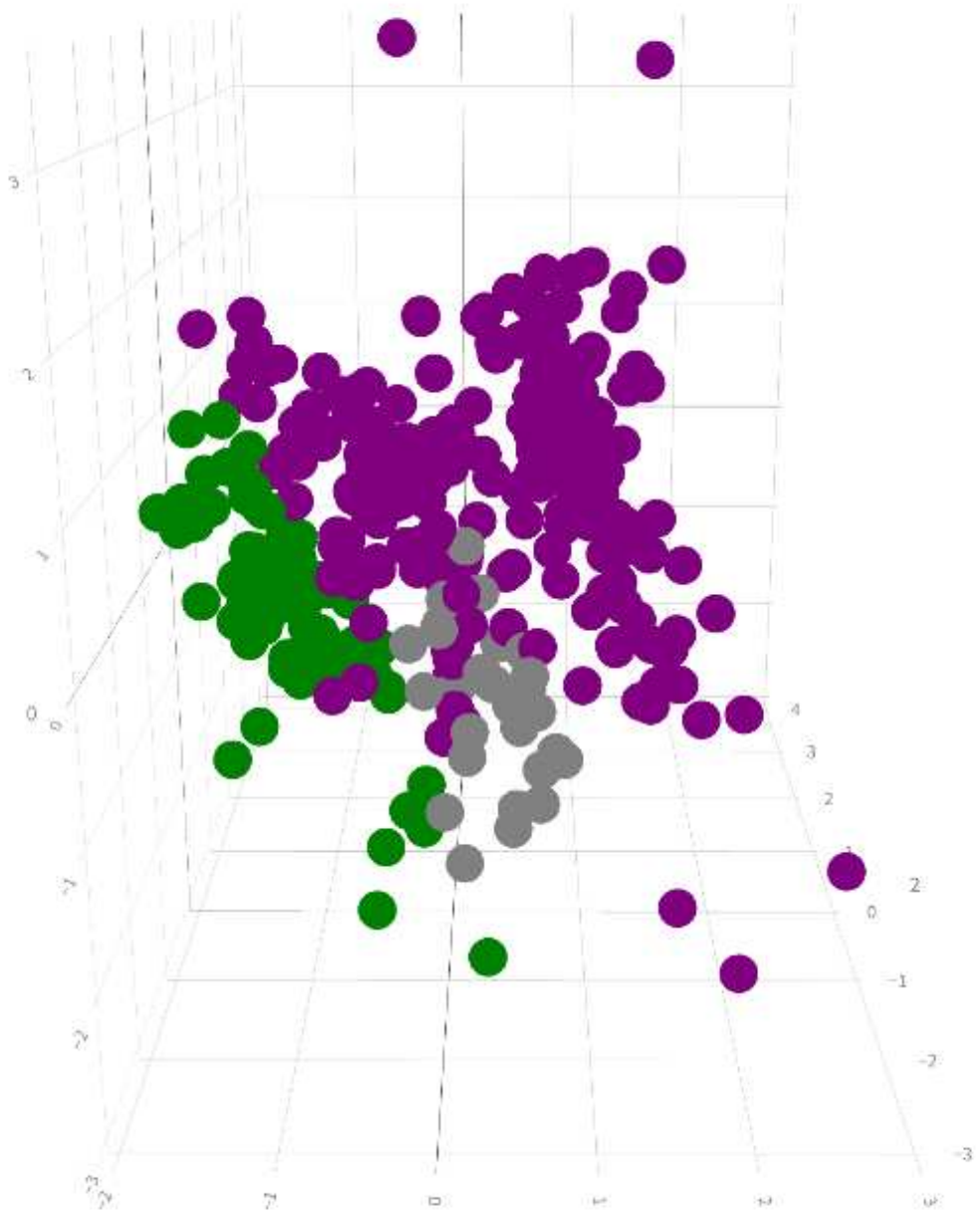


Figure 4.17 – 2 clusters and outliers (in grey) identified by Pretopo-MD in the FAMD reduced space of the Heart Dataset

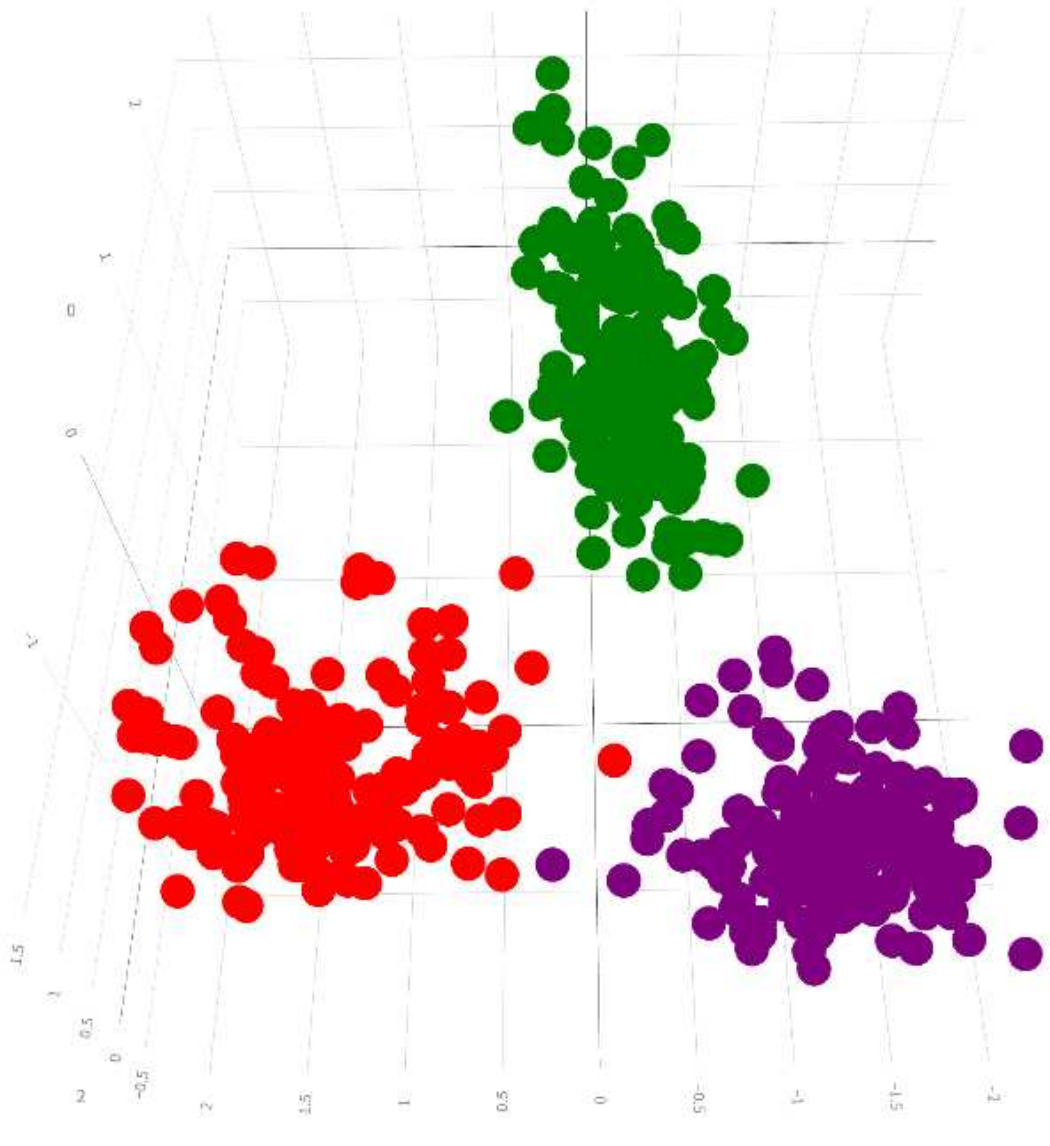


Figure 4.18 – 3 clusters identified by almost all methods in the reduced FAMD space of the Base Generated Case

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	**296.330**	**0.395**	0.521	**1.071**
DenseClus	295.224	**0.395**	0.519	1.074
Phillip & Ottaway	290.557	0.389	0.515	1.084
Kamila	**296.330**	**0.395**	0.521	**1.071**
K-Prototypes	**296.330**	**0.395**	0.521	**1.071**
MixtComp	294.322	**0.395**	0.519	**1.071**
Modha-Spangler	**296.330**	**0.395**	0.521	**1.071**
Kmeans-FAMD	296.246	**0.395**	**0.523**	**1.071**
Pretopo-FAMD	24.146	0.053	0.061	2.968
Pretopo-UMAP	293.835	0.392	0.517	1.078
Pretopo-PaCMAP	129.698	0.195	0.243	1.709
Pretopo-Louvain	77.688	0.028	0.008	3.394
Pretopo-Laplacian	1.177	-0.358	-0.453	2.231
PretopoMD	127.339	0.230	0.308	1.508

Table 4.4 – Results of the selected Algorithms on the Base Generated Case

for clustering because the individual clusters are less dense. The Elbow Method identifies k=2 as the optimal number of clusters, which significantly deviates from the intended 10.

DenseClus and PretopoMD both identify approximately 300 outliers from the 500 data points, even though the generated datasets are not designed to contain noise. Lastly, Pretopo-UMAP divides the dataset into 10 distinct clusters and achieves the highest score across all indicators. This indicates that Pretopo-UMAP is very effective in such case. K-means On the UMAP reduced space didn't identify all the clusters and fused some of them, giving it a much lower score.

Generated dataset with 15 categorial features and 15 categorical unique values

Then, we analyze how the different algorithms perform in a high dimension context. To do so, we generate a dataset with 15 categorical features with 15 different values of each size. There, the Elbow Method finds $k = 2$ clusters. ClustMD, PretopoMD, Pretopo-UMAP and MixtComp don't converge on such a dataset, and only produce noise. Pretopo-FAMD finds 498 outliers out of the 500 individuals. Pretopo-UMAP produces 1 cluster of 332 individuals and 168 outliers (that might be merged into another cluster). DenseClus and Pretopo-PaCMAP both find 3 balanced clusters, and the latter finds more balanced clusters and therefore obtains a slightly better score on the 4 indices. KAMILA, K-prototype and Philip and Ottaway have merged two clusters into one and therefore find One cluster of approximately 333 elements and one cluster of around 137 elements.

Generated Dataset with 1000 individuals

Gave fairly equilibrated results with most methods identifying the 3 clusters well apart from Pretopo-FAMD, Pretopo-PaCMAP, Pretopo-Louvain, Pretopo-Louvain,

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	0.000	-1.000	-1.000	-1.000
DenseClus	91.514	0.184	0.180	1.866
Phillip & Ottaway	105.311	0.297	0.348	1.389
Kamila	111.239	0.258	0.257	1.464
K-Prototypes	96.461	0.230	0.199	1.487
MixtComp	95.644	0.223	0.200	1.514
Modha-Spangler	112.780	0.296	0.258	1.314
Kmeans-FAMD	131.886	0.341	0.373	1.209
Pretopo-FAMD	30.539	0.048	0.028	1.552
Pretopo-UMAP	**134.738**	**0.372**	**0.375**	**1.152**
Pretopo-PaCMAP	48.594	0.160	0.141	1.561
Pretopo-Louvain	88.408	0.294	0.292	2.166
Pretopo-Laplacian	2.599	-0.358	-0.450	1.578
PretopoMD	88.679	0.203	0.278	1.658

Table 4.5 – Results of the selected Algorithms on a generated dataset with 10 clusters.

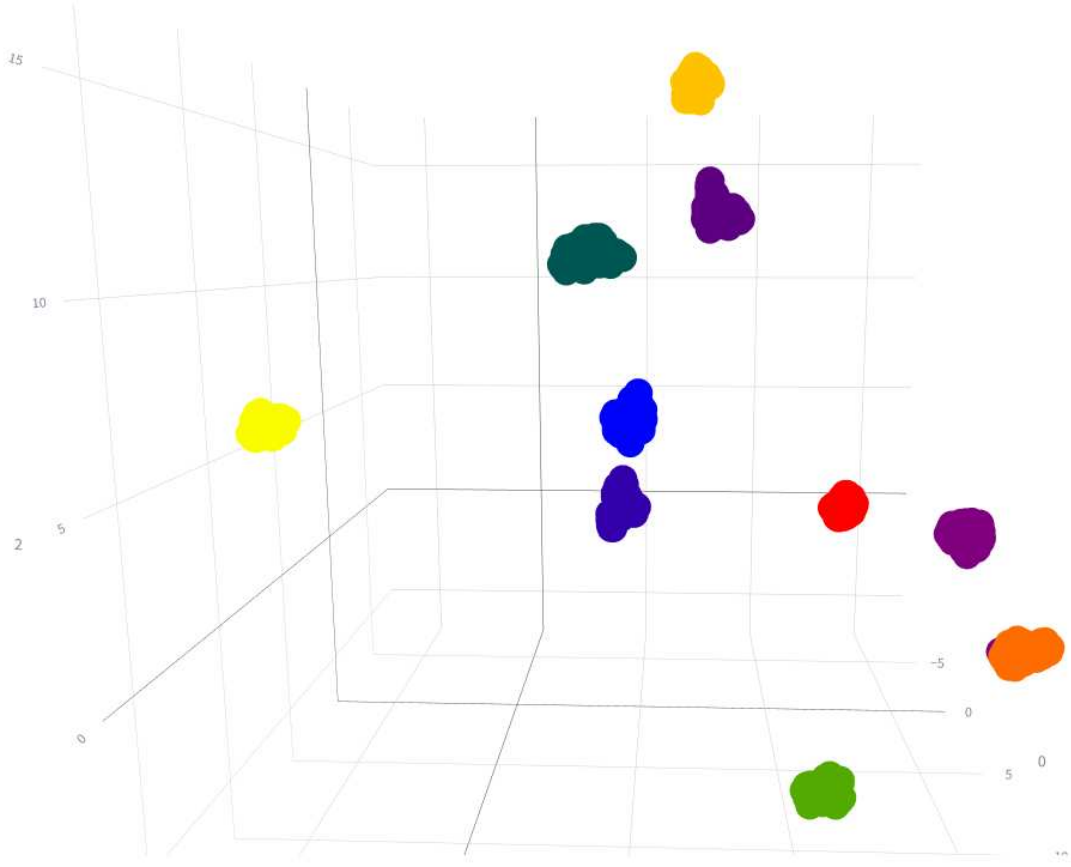


Figure 4.19 – The ten generated clusters identified by Pretopo-UMAP in the UMAP reduced space

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	0.000	-1.000	-1.000	-1.000
DenseClus	118.504	0.216	0.102	1.708
Phillip & Ottaway	119.712	0.197	0.087	1.839
Kamila	121.389	0.199	0.088	1.825
K-Prototypes	118.497	0.194	0.085	1.869
MixtComp	0.000	-1.000	-1.000	-1.000
Modha-Spangler	121.389	0.199	0.088	1.825
Kmeans-FAMD	121.389	0.199	0.088	1.825
Pretopo-FAMD	0.946	-0.092	-0.015	2.080
Pretopo-UMAP	90.615	0.153	0.073	2.096
Pretopo-PaCMAP	**123.388**	**0.226**	**0.107**	**1.664**
Pretopo-Louvain	37.727	0.029	0.001	3.847
Pretopo-Laplacian	0.436	-0.095	-0.019	3.501
PretopoMD	0.000	-1.000	-1.000	-1.000

Table 4.6 – Results of the selected Algorithms on a generated dataset with 15 categorial features and 15 categorical unique values

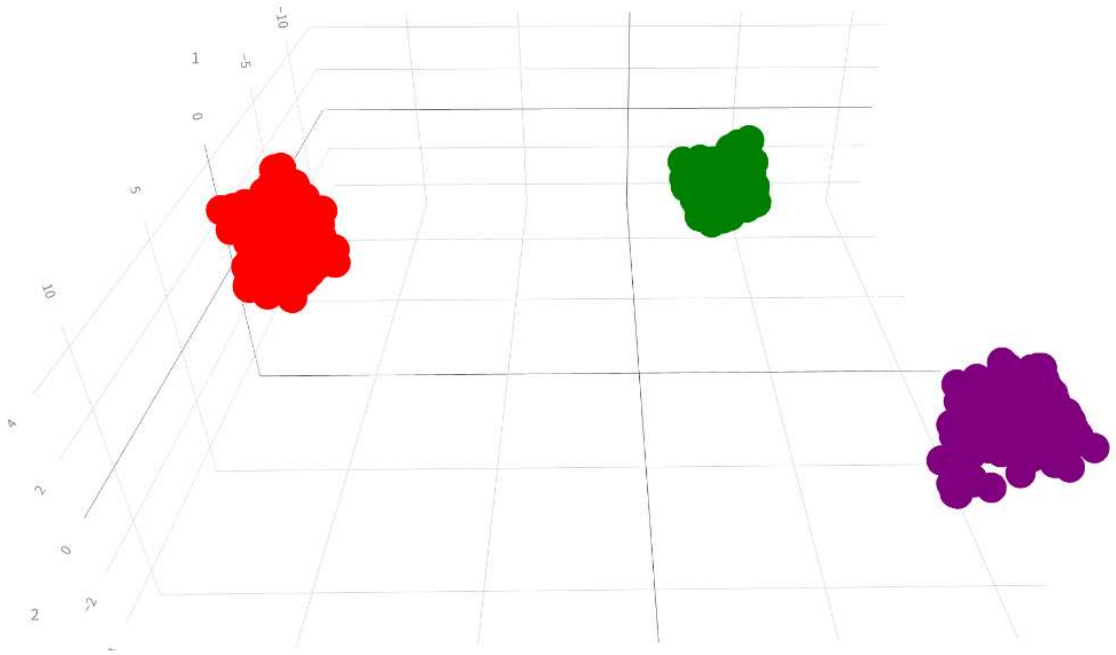


Figure 4.20 – Three clusters identified by Pretopo-PaCMAP in the PaCMAP reduced generated dataset with 15 categorical values

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	183.449	0.150	0.131	2.118
DenseClus	132.056	0.121	0.143	2.711
Phillip & Ottaway	3.512	0.088	0.169	2.157
Kamila	183.719	0.150	0.131	2.127
K-Prototypes	180.026	0.147	0.127	2.155
MixtComp	120.243	0.106	0.111	2.369
Modha-Spangler	168.726	0.142	0.144	2.218
Kmeans-FAMD	184.044	0.150	0.131	2.120
Pretopo-FAMD	1.316	0.035	0.043	2.104
Pretopo-UMAP	188.827	0.179	0.175	1.915
Pretopo-PaCMAP	**200.106**	**0.189**	**0.188**	**1.861**
Pretopo-Louvain	39.915	0.017	-0.007	4.050
Pretopo-Laplacian	1.199	0.059	0.093	2.340
PretopoMD	19.286	-0.017	-0.040	2.868

Table 4.7 – Results of the selected algorithms on a generated dataset with 1000 individuals, 10 dimensions of each type, and a deviation of 0.15

and PretopoMD

Generated Dataset with 1000 individuals, 10 dimensions of each type and a deviation of 0.15

For the moment, we only studied the results of our algorithms on configurations with a clusters deviation of 0.10. Therefore, we generated a dataset to analyze how the different algorithms perform on sparser clusters. This dataset contains 1000 individuals, 10 dimensions of each type, and has a deviation of 0.15 (while the base case has 0.10). On this dataset, the Elbow Method determines $k = 2$ as the optimal number of clusters, while the generated dataset is supposed to contain 3. The algorithms that obtain the more optimal scores are the algorithms that use UMAP and PaCMAP. As those reduction move the neighbors closer to each other, it is not surprising to see them perform well on datasets with a higher clusters deviation. Among those 3, Pretopo-PaCMAP is the only one that produces no outlier, therefore it obtains the highest Calinski-Harabasz and Silhouette scores.

4.2.2.1 Results analysis

In conclusion of this subsection, after conducting a comprehensive analysis of the results obtained from executing the algorithm on the generated dataset and other datasets, we can draw several significant observations.

First and foremost, it is clear that both ClustMD and MixtComp often struggle to achieve convergence, highlighting a notable limitation in these approaches. Similarly, while DensClus and PretopoMD generally shows promise, they are not immune to convergence issues either.

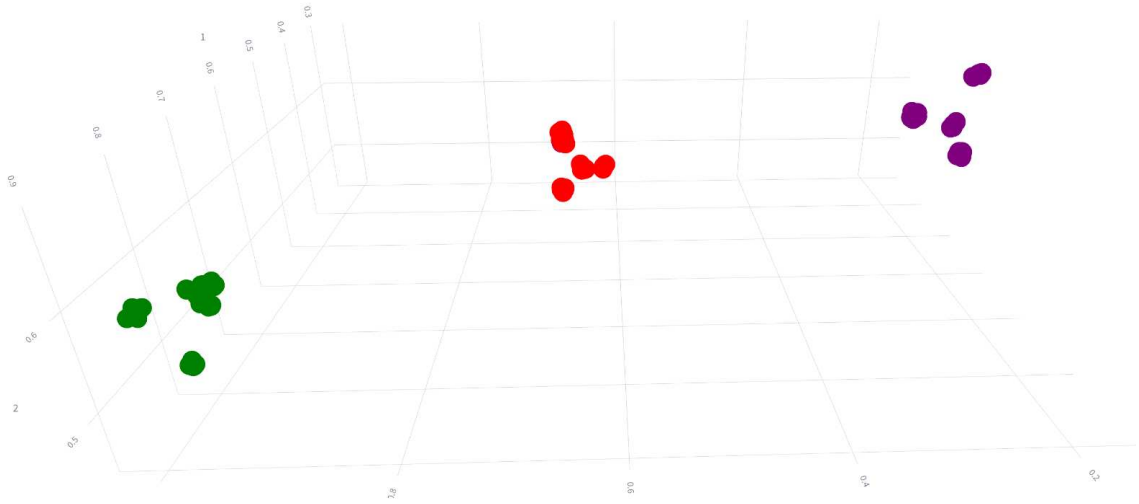


Figure 4.21 – 3 clusters identified by Pretopo-PaCMAP in the Louvain reduced generated dataset with 1000 individuals

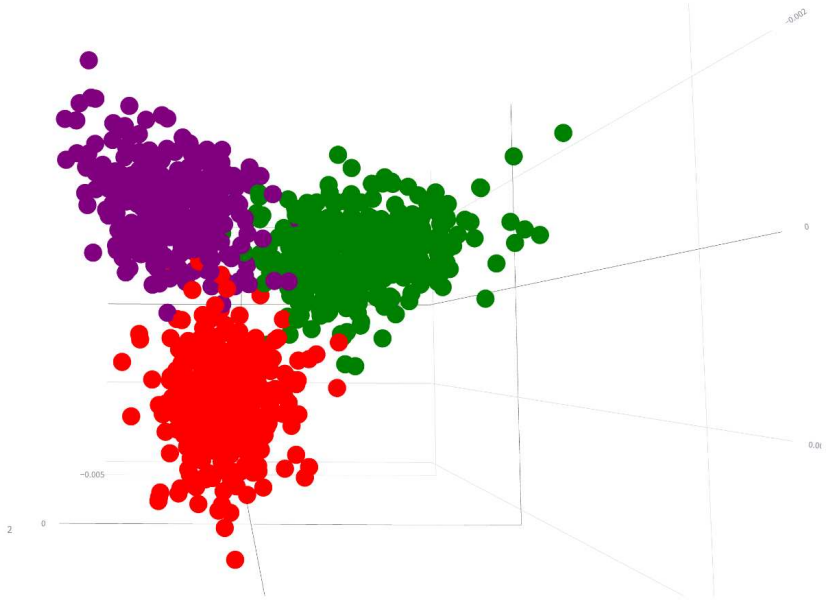


Figure 4.22 – 3 clusters identified by Pretopo-PaCMAP in the Laplacian Eigenmap generated dataset with 1000 individuals

The Pretopo-UMAP algorithm sometimes group the dataset into a single, large cluster, accompanied by outliers. This clustering pattern does not receive sufficient penalization according to our current calculation method for the indicators (please refer to 5.2.3 for more details). This aspect deserves attention for future improvements in metric design.

In cases of ambiguous data patterns, Pretopo-FAMD tends to create a dominant central cluster surrounded by several smaller clusters. This can complicate the interpretability of the resulting clusters. It's important to note that Pretopo-FAMD did not perform well on the presented dataset, except for the penguin dataset, where it achieved the highest scores on 3 out of 4 indicators after identifying 28 clusters.

Algorithms that rely on the Elbow Method face a significant limitation. Regardless of the algorithm's sophistication, if it is set to identify an incorrect number of clusters, it cannot achieve optimal partitioning. This makes it challenging to analyze each algorithm individually in detail, especially since they often produce very similar partitions.

In contrast, Pretopo-PaCMAP consistently delivers superior results across various configurations. Its independence from the constraints of the Elbow Method, combined with its ability to avoid the convergence issues observed in other algorithms, positions it as a robust and reliable approach for data partitioning.

PretopoMD, which is independent of both DR and the Elbow Method, tends to identify a small number of sizable clusters along with a few outliers. Across the datasets on which it was evaluated, it sometime outperformed other methods on at least one of the employed indicators but didn't shine on the generated datasets and often couldn't identify any clusters.

It is worth noting that across each of the seven presented datasets, the various applications of the pretopological algorithms consistently ranked among the top two to four performers. This underscores their potential as a valuable addition to the toolkit of clustering techniques when dealing with mixed datasets.

After analyzing the performance of the algorithms on the quality indexes we have selected, the following recommendation could be made.

On a datasets that contain a high number of clusters, the algorithm Pretopo-UMAP seems to be the most relevant on all indexes. On a dataset with a high number of categorical features, the algorithm that seems to be the most relevant is Pretopo-PaCMAP, followed closely by KAMILA, Modha-Spangler, Philip and Ottaway, DenseClus and K-prototypes. On a dataset with a high number of elements, Pretopo-PaCMAP seem to be the most relevant, followed by ClustMD and KAMILA. On a dataset with a high deviation, Pretopo-PaCMAP seem to be the most relevant, followed by ClustMD and KAMILA.

On average with all scores being normalized between 0 and 1 with 1 being the best score of the table, the best algorithms on average for CH and for Silhouette Coefficient (SC) are Kmeans-FAMD, KAMILA and Modha-Spangler because they have the most consistently good score even when they are not always the best scoring algorithms. For Gower Silhouette Coefficient (GSC) the best algorithms on average are Philip and Ottaway, Kmeans-FAMD and Modha-Spangler. On the DB score, the best scoring on average were attained by DenseClus, Pretopo-UMAP and Pretopo-PaCMAP (see : table 4.8).

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	0.575	0.367	0.298	0.084
DenseClus	0.596	0.403	0.365	**0.000**
Phillip & Ottaway	0.798	0.939	**1.000**	0.155
Kamila	**0.938**	**0.975**	0.856	0.350
K-Prototypes	0.906	0.960	0.859	0.416
MixtComp	0.558	0.336	0.341	0.065
Modha-Spangler	**0.933**	**0.970**	**0.882**	0.379
Kmeans-FAMD	**1.000**	**1.000**	**0.938**	0.337
Pretopo-FAMD	0.128	0.538	0.449	0.429
Pretopo-UMAP	0.779	0.798	0.700	**0.020**
Pretopo-PaCMAP	0.532	0.671	0.524	**0.042**
Pretopo-Louvain	0.302	0.530	0.409	1.000
Pretopo-Laplacian	0.000	0.000	0.000	0.643
PretopoMD	0.395	0.316	0.210	0.257

Table 4.8 – Average normalized scores of the algorithms on all the datasets

4.2.3 Complex Clustering Results

4.2.3.1 Clustering complex generated data

Here we will present the results of PretopoMD. We have created four prenetworks for this instance : one for the position features, one for the size feature, one for the shapes of the elements, and one for the time series. For each prenetwork, a distance matrix is calculated. We use Euclidean distance for numerical features, Hamming distance for categorical features, and DTW for time series.

Different DNFs were used, and the DNF that scored highest on CH, SC, and GSC was the one using only TS, indicating that clustering based solely on the time series was more effective than using more complex clustering methods. The only score that did not favor this DNF was DB, which preferred *Position AND Size AND TS OR Shape*. This DNF provided better cluster separation because it had more *AND* rules, which made it more likely to divide the dataset into many clusters with similar position, size, and time series characteristics.

However, other DNF combinations could be chosen depending on the user's needs, as the relevance of the clustering varies according to the application. For illustrative purposes, a clustering using the simple *Position AND Time Series* rule is shown in Figures 4.2 and 4.23, identifying 8 clusters. Each cluster consists of elements that are close in space and have similar time series patterns. This observation also applies to the subclusters within the larger clusters. The result of a more complex DNF, corresponding to more specific needs, is also presented in Figure 4.25.

We observe that Pretopo-FAMD achieves the best results in terms of CH, DB, and SC Scores. This can be attributed to the fact that clustering in conjunction with DR is highly effective on time series data as it mitigates the Curse of Dimensionality. It is also worth noting that the CH, DB, and SC Scores are all calculated on the dataset after applying FAMD, thereby favoring clustering methods that utilize

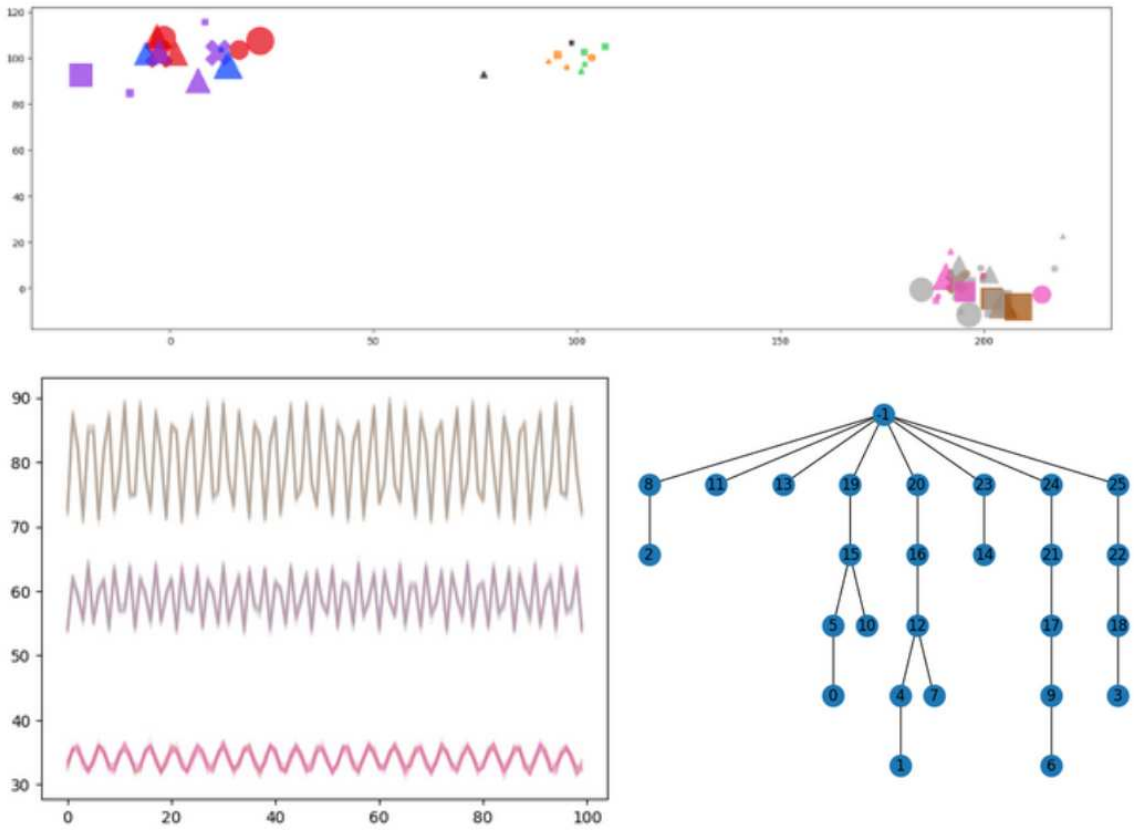


Figure 4.23 – Visualisation of mixed hierarchical clustering using time series.

FAMD in their preprocessing. Had we evaluated the clusters using CH, DB, and SC by reducing the dataset with another DR method, we would have obtained different results. Additionally, we can note that Pretopo-FAMD does not have a good score on GSC despite being the best on the other metrics.

If we normalize and add up our scores, the best algorithms in descending order are : Pretopo-FAMD, Kmeans-FAMD, Pretopo-PaCMAP, Kmeans-FAMD with TSF instead of the whole time series, and Pretopo-PaCMAP with TSF instead of the whole time series. Just below these are AHC_Gow_DTW with three clusters, Philip and Ottaway, KAMILA, K-prototypes, and PretopoMD using only time series values. These methods have identified clusters that correspond exclusively to the time series.

Interestingly, these methods that identified only the time clusters (AHC with three clusters, Philip and Ottaway, KAMILA, K-Prototype, and PretopoMD using the DNF Time Series) achieved the highest GSC Score, all at equal values. It means that the other features are not only deemed irrelevant by these clustering methods but also the GSC Score. For example, AHC with more than three clusters employ other features for clustering but are considered worse than AHC_3.

There are several interpretations of this phenomenon. First, K-prototypes, KAMILA, and Philip and Ottaway treat each time step as a feature, making non-time series features less significant in the resulting clusters due to their comparatively low numbers.

As for AHC, it was specifically designed to address this issue by incorporating a distance specific to time series in addition to the Gower distance for other fea-

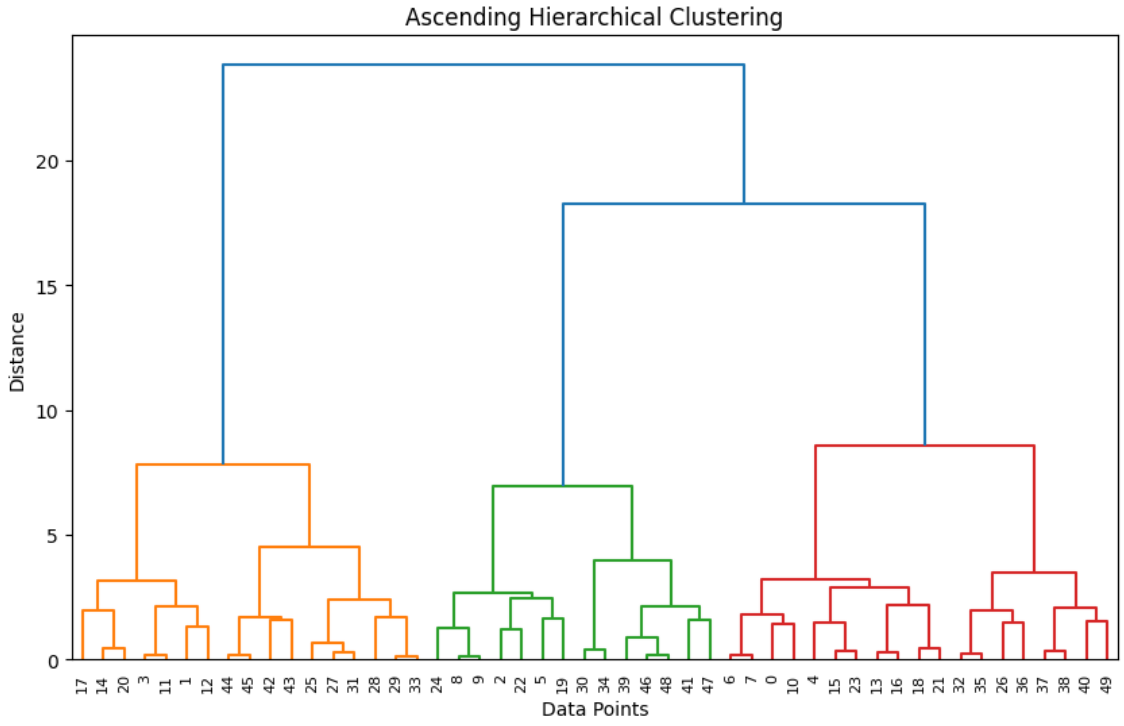


Figure 4.24 – The DTW distance between the time series slightly outweighs the Gower distance between the mixed features in this dataset. All evaluation metrics rewards the separation in 3 clusters

tures. By examining the dendrogram in Figure 4.24, we can observe that the three clusters are well-separated because the distance between time series is more pronounced than the distance between other features. However, this is more attributable to the test dataset, in which the time series are extremely similar, rather than the Hierarchical Clustering methods itself. In this instance, when clustering into three clusters, it made sense to cluster based on time series similarity. When more clusters were demanded from AHC, it provided a finer separation of the dataset, taking into account other features. However, no indicators rewarded such behavior. This is the case with and without normalized distance in AHC. What was not attempted here was assigning weights to different distances based on specific needs or characteristics. In a case study, one could decide to give more weight to a certain set of parameters for them to have a more significant influence on the resulting hierarchical clustering.

Another point concerning the evaluation metrics is that none of the extracted features used for some of the clustering were added to the dataset. Adding the dataset with pre-identified time series clusters or extracted features might have changed the way the clusters are evaluated.

Exploring further cluster evaluation metrics and aggregating them might be a solution for hyperparameterization. The objective might not simply be to have the highest average score but to find a clustering that has scores relatively high for all metrics.

However, one must accept that the quality of clustering is highly dependent on the objectives of the user, especially in the case of complex data. Depending on the case study, the relevance of one aspect of the data can vary significantly.

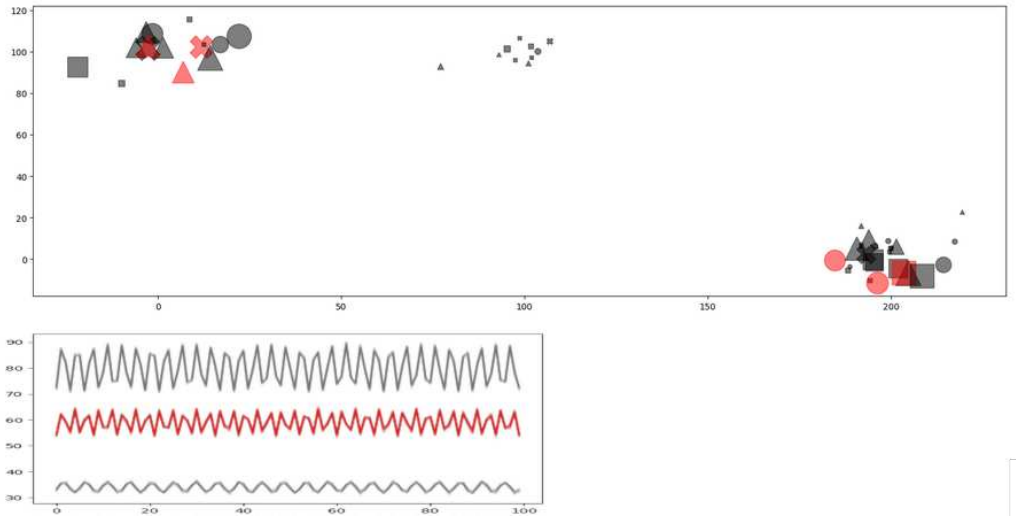


Figure 4.25 – A subcluster of the hierarchy build with the DNF *(Position AND Shape AND TS) OR (Size AND TS)* prioritizes *TS*, then *Size* then equally *Position* and *Shape*

Visually analyzing the data in its raw decomposed form, such as in time series, or visualizing it through different DR techniques can allow users to view it from various perspectives (quite literally) and realize how one clustering might seem more appropriate when viewed through FAMD and another more relevant when viewed through UMAP. Ultimately, it is the meaning behind the features and the coherence of the final clusters that give relevance to a clustering method.

Therefore, hierarchical clustering techniques such as PretopoMD, which function extremely well with DR, might actually be more relevant without DR when complex rules and distances must be used to identify clusters according to specific requirements. AHC might also be used in this manner simply through the use of weights. Both have the advantages of allowing the user to zoom in on a cluster to identify subgroups, which is often relevant in complex data contexts. For example, the AHC dendrogram allowed us to view how the relatively high distance between the time series cluster influenced the separation of the complex dataset and how weighting the different distance might have changed this separation (see figure 4.24).

Regarding pretopology, the example in figure 4.2, as well as more complex DNFs allow for some very interesting hierarchie. For example, a hierarchical clustering built with the DNF *(Position AND Shape AND TS) OR (Size AND TS)* (see figure 4.25) will return the same clusters as the DNF *TS*, but will return a hierarchy with subclusters of elements that are necessarily close in terms of time series but are also as close as possible in terms of position, size, or shape, with size being the first criterion of aggregation. That is, the smaller clusters are necessarily close in time series and are mostly close in size. Then they expand by integrating other elements according to the other criteria. Adjusting the DNF in this manner enables the construction of hierarchies tailored to meet the complex requirements specific to various case studies. Furthermore, besides the DNF, the diverse parameters of PretopoMD facilitate extensive customization of the dispersion, size, and number of outliers within the clusters.

Method	CH	DB	SC	GSC
Method 1				
Phillip & Ottaway	8.01	2.69	0.12	0.93
Kamila	8.01	2.69	0.12	0.93
K-Prototypes	8.01	2.69	0.12	0.93
Pretopo_Euclid_Hamm	4.27	2.16	-0.04	-0.26
Method 2 AHC				
AHC_Gow_DTW_6	3.61	4.64	-0.03	0.38
AHC_Gow_DTW_5	4.38	4.55	0.01	0.52
AHC_Gow_DTW_4	5.79	3.07	0.06	0.73
AHC_Gow_DTW_3	8.01	2.69	0.12	0.93
Method 2 PretopoMD				
Pos_&_Size_or_Shape_&_TS	4.10	4.90	0.06	0.54
Pos_or_Size_&_Shape_or_TS	4.10	4.90	0.06	0.54
Pos_&_Size_&_TS_or_Shape	7.48	1.01	0.12	0.19
Pos_&_Size_or_Shape_&_TS	1.20	3.22	-0.27	-0.49
Pos_&_TS	2.58	3.75	-0.14	0.28
TS	8.01	2.69	0.12	0.93
Method 3				
DenseClus	0.00	-1.00	-1.00	-1.00
Kmeans-FAMD	26.84	1.03	0.48	-0.07
Pretopo-FAMD	28.06	0.81	0.51	-0.07
Pretopo-Laplacian	0.40	3.62	-0.12	-0.28
Pretopo-UMAP	7.49	2.76	0.11	0.87
Pretopo-PaCMAP	25.84	1.03	0.47	-0.08
Pretopo-Louvain	2.97	3.30	-0.16	0.21
Method 4				
DenseClus	0.00	-1.00	-1.00	-1.00
Phillip & Ottaway	1.40	5.77	0.01	0.00
K-Prototypes	1.08	6.57	0.00	0.01
Pretopo_Eucl_Hamm	1.80	2.49	-0.25	-0.51
Method 5				
Kmeans-FAMD	1.08	6.57	0.00	0.01
Pretopo-FAMD	3.86	2.07	-0.06	-0.15
Pretopo-Laplacian	1.05	5.81	-0.18	-0.34
Pretopo-UMAP	0.95	7.38	-0.09	-0.12
Pretopo-PaCMAP	11.04	1.79	0.23	-0.04
Pretopo-Louvain	1.08	5.32	-0.05	-0.15
Method 6				
Phillip & Ottaway	8.43	2.51	0.11	0.57
K-Prototypes	7.67	2.51	0.08	0.38
Pretopo_Eucl_Hamm	5.99	2.10	0.10	0.11
Method7				
DenseClus	0.00	-1.00	-1.00	-1.00
Kmeans-FAMD	16.73	1.37	0.32	0.08
Pretopo-Laplacian	2.21	3.66	0.02	0.01
Pretopo-UMAP	6.34	2.65	0.09	0.49
Pretopo-PaCMAP	15.79	1.40	0.32	0.06
Pretopo-Louvain	6.45	2.21	0.06	0.42
Pretopo-FAMD	2.48	2.36	-0.03	-0.13

Table 4.9 – Cluster evaluation scores of the 7 different complex clustering methods on the generated dataset.

	Calinski-Harabasz	Silhouette FAMD	Silhouette Gower	Davies-Bouldin
ClustMD	0.000	-1.000	-1.000	-1.000
DenseClus	0.000	-1.000	-1.000	-1.000
Phillip & Ottaway	14.443	0.157	**0.437**	2.145
Kamila	**15.519**	0.167	0.421	2.070
K-Prototypes	**15.519**	0.167	0.421	2.070
MixtComp	0.000	-1.000	-1.000	-1.000
Modha-Spangler	15.411	0.165	0.404	2.079
Pretopo-FAMD	1.502	-0.063	-0.173	1.641
Pretopo-UMAP	6.933	0.063	0.138	2.734
Pretopo-PaCMAP	0.000	-1.000	-1.000	-1.000
PretopoMD	6.371	**0.484**	0.013	**0.384**

Table 4.10 – Results of the selected algorithms on a private dataset of energie consumption data and building characteristics

4.2.3.2 Clustering complex energy data

The methods described here is the methods number 4 (see figure 3.9) consisting in applying one or several clustering on the time series in order to extract labels before applying mixed clustering on the entire enriched dataset.

We applied three clustering methods on the smoothed time series as well as on the average week :

- Self Organising Map (SOM)
- Kmeans
- Kmeans with DR

Both SOM and Kmeans necessitate the number of cluster. On figure 4.28, 4.29 and 4.31, you can see the result of the different clustering, with the line in red representing the average values for the cluster.

SOM, K-means, and K-means with DR produced different clusters. Subsequently, we used the results from these three clustering methods as labels, creating three additional categorical variables.

Both KAMILA and K-prototypes identified the same clustering pattern, consisting of three clusters of varying sizes, whereas PretopoMD identified two clusters, with one encompassing most of the dataset. This clustering is supported by SC and DB scores, even though it does not seem highly relevant. However, because PretopoMD is hierarchical, we identified more balanced clusters within the hierarchy. Similarly, Philip and Ottaway found one dominant cluster containing almost the entire dataset and two very small clusters. This demonstrates that the indicators do not always select the most balanced clustering. Upon examining the reduced dataset (see figure 4.26), this clustering seems to somewhat align with the general 'shape' of the dataset.

This chapter has delved into the intricate domain of datasets and the performance analysis of diverse clustering algorithms, highlighting the complex challenges and methodologies applicable to mixed and complex data. The exploration began with a focus on

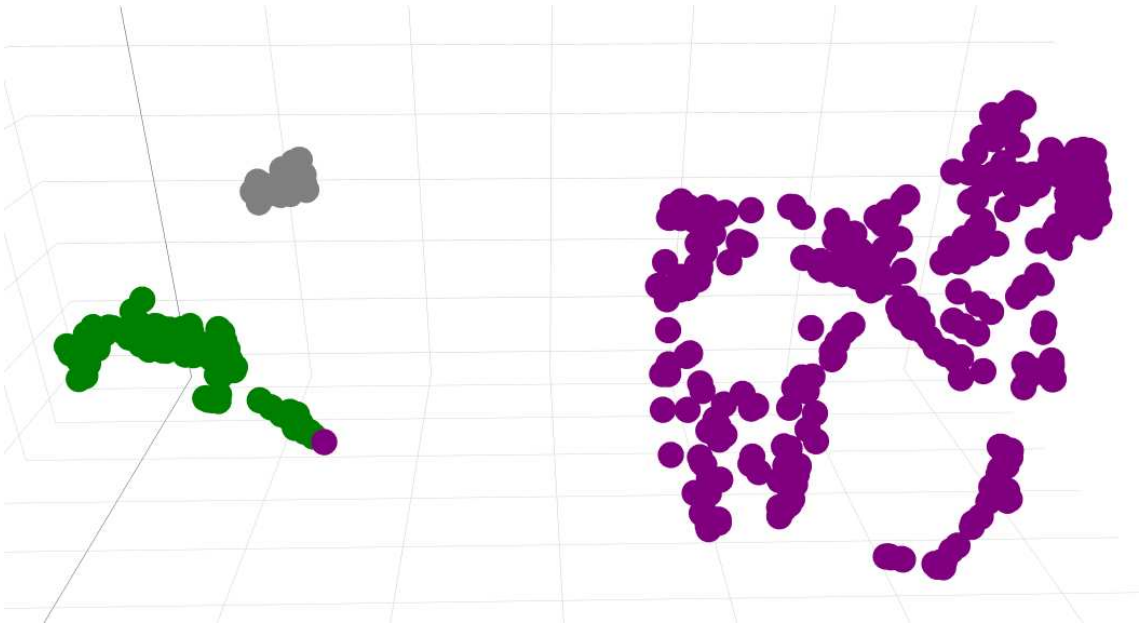


Figure 4.26 – 3 clusters identified by Pretopo-UMAP in the UMAP reduced energisme dataset

several public datasets—namely, the Palmer Penguins, Heart Disease, and Sponge datasets—each chosen for its unique characteristics and relevance to mixed data clustering. These datasets served as a foundation for comparing the efficacy of various clustering algorithms, showcasing the algorithms' capabilities and limitations in handling datasets with varying complexity.

A significant part of the analysis was dedicated to a custom dataset generator, designed to facilitate the comparison of algorithms under a wide range of configurations. This tool allowed for the generation of isotropic Gaussian blobs, which are crucial for benchmarking clustering algorithms' performance across numerous scenarios, including different numbers of clusters, samples, and feature types. The generator's ability to transform numerical features into categorical ones further enriched the dataset's complexity, offering a closer simulation of real-world mixed datasets.

The chapter also introduced a complex dataset generator, incorporating numerical, categorical, and time series data. This advanced generator aimed to replicate the multifaceted nature of real-world data, highlighting the challenges inherent in clustering such diverse datasets. Through this generator, the chapter provided an insightful examination of various clustering algorithms, including Pretopo-PaCMAP and PretopoMD, emphasizing their performance in specific data configurations.

The analysis underscored several key findings, notably the critical impact of dataset characteristics on the algorithms' computational costs and technical limitations. Factors such as the number of individuals, the diversity of feature types, and the clusters' dispersion significantly influenced the memory usage and computation time, revealing stark differences in the efficiency of different algorithms. This section shed light on the necessity of choosing the right algorithm based on specific data characteristics and the intended clustering objectives.

Moreover, the exploration of clustering results on mixed and complex datasets offered valuable insights into the algorithms' performance. It highlighted the effectiveness of Pretopo-PaCMAP in handling datasets with a larger number of clusters and high-dimensional data. The chapter also discussed the challenges faced by algorithms like ClustMD and MixtComp, particularly in achieving convergence on complex datasets.

Summary of Chapter 4

Benchmark Establishment for Clustering Algorithms : Utilized public datasets and a custom dataset generator for algorithm comparison. Public datasets included Palmer Penguins, Heart Disease, and Sponge datase.

Palmer Penguins Dataset Analysis : Demonstrated high clustering tendency, especially with UMAP and PaCMAP dimension reductions.

Utilized for benchmarking due to its commonality in literature and balanced mix of numerical and categorical features.

Heart Disease Dataset : A medical dataset combining various feature types over 918 observations, illustrating the complexity of clustering mixed datasets.

Sponge Dataset : Focused on marine biology, characterized by a small sample size and a high number of categorical features, highlighting the challenge in clustering datasets with few individuals and numerous categorical features.

Custom mixed Dataset Generator : Enabled testing of algorithms under diverse configurations, emphasizing the generation of isotropic Gaussian blobs and the transformation of numerical features into categorical ones to simulate mixed datasets.

Custom Complex Dataset Generation : Included numerical, categorical, and time series data, underlining the intricacy of clustering in real-world scenarios.

Evaluation of Clustering Algorithms Mixed and Complex Datasets : Discussed computation cost and technical limitations, noting significant differences in memory usage and computation time across algorithms.

Highlighted the impact of dataset characteristics on algorithm performance, including the number of individuals, the number of clusters, the number of dimensions or the dispersion.

Provided detailed analysis of clustering performance on various datasets, showing the effectiveness of certain algorithms like Pretopo-PaCMAP in specific configurations.

Examined the role of dimensionality reduction in clustering and the challenges in clustering high-dimensional and complex datasets.

Comprehensive Analysis of Clustering Techniques :

Illustrated the necessity of customizing clustering approaches to fit specific data characteristics and objectives.

Emphasized the importance of understanding algorithm limitations and selecting appropriate techniques based on dataset intricacies.

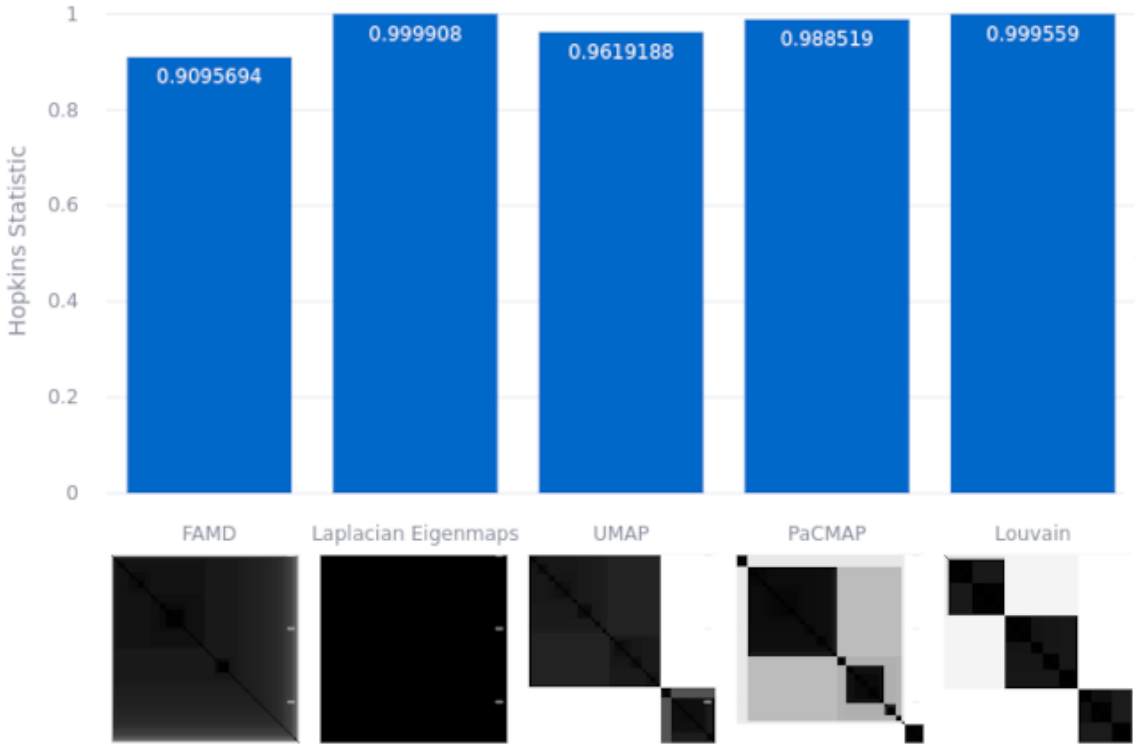


Figure 4.27 – iVAT of the energie data

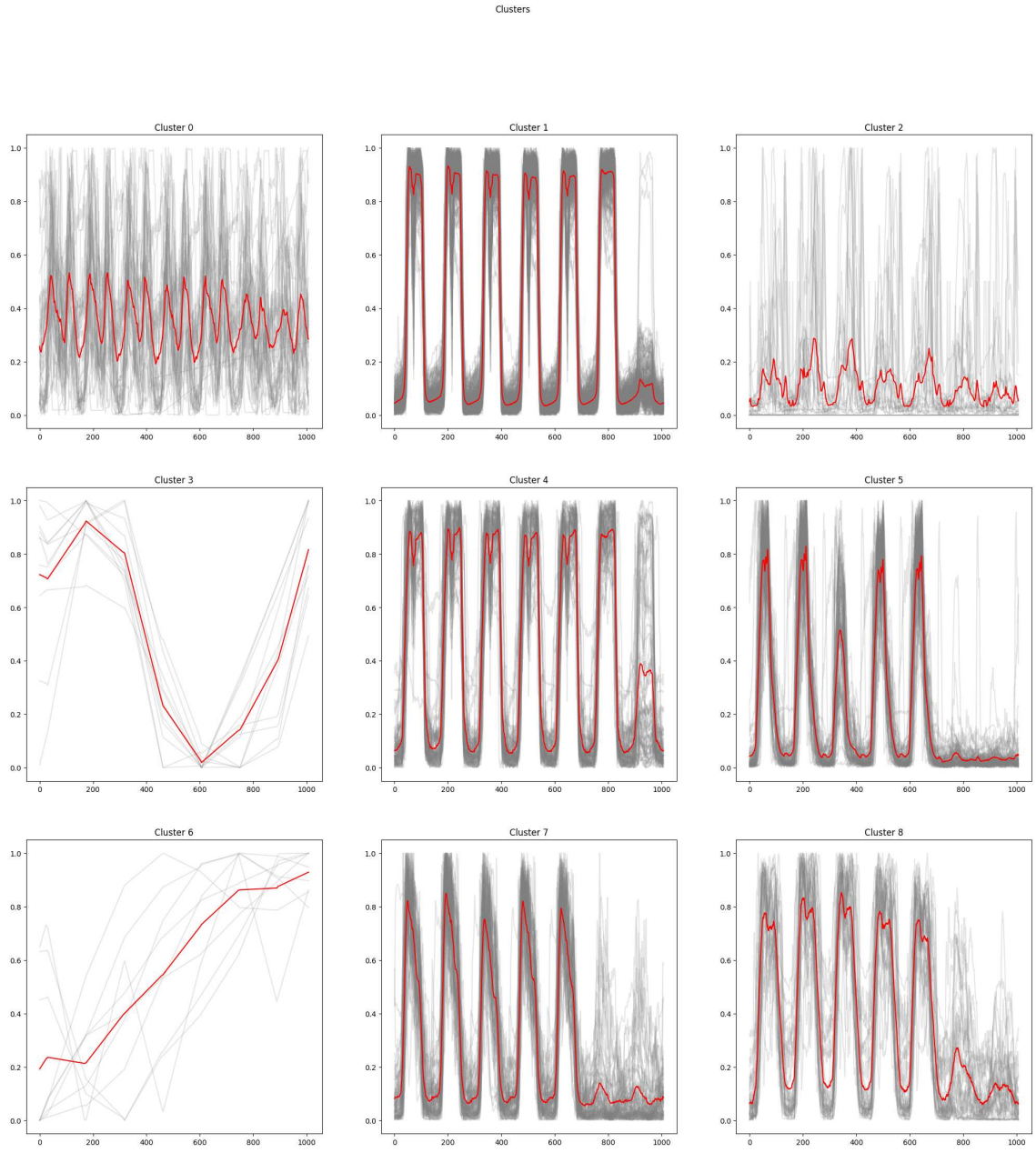


Figure 4.28 – Kmeans with 9 clusters

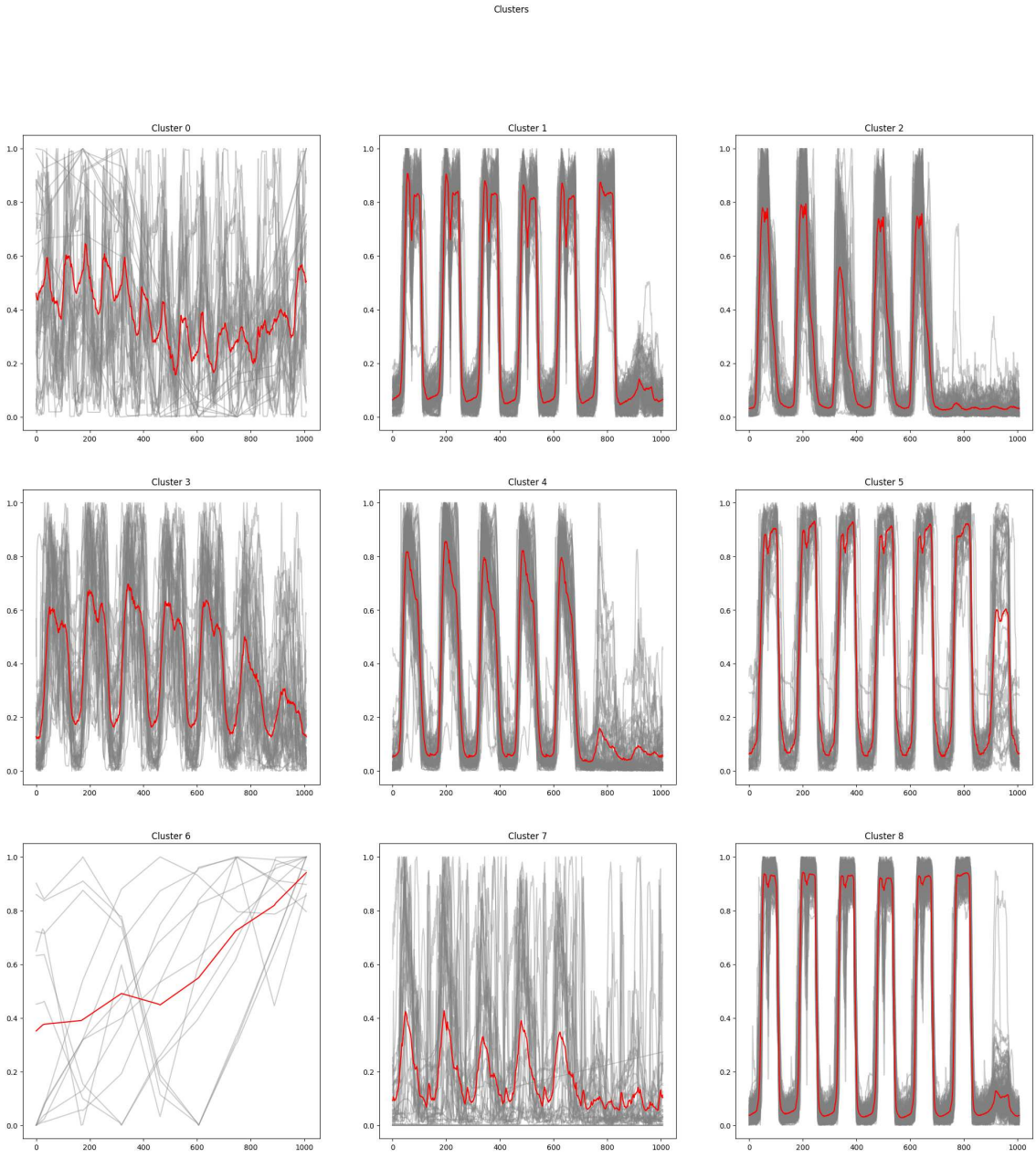


Figure 4.29 – Kmeans after reduction with 9 clusters

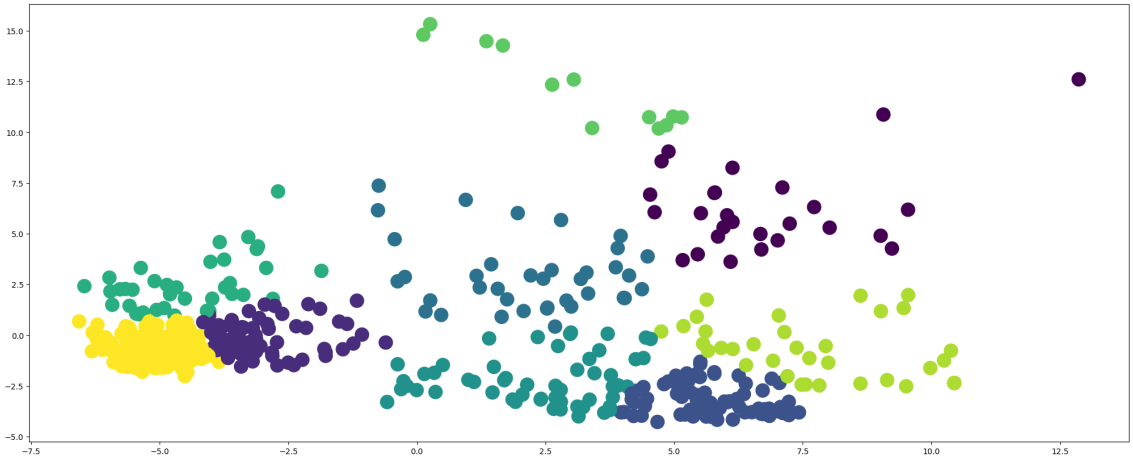


Figure 4.30 – Kmeans after reduction with 9 clusters, viewed from the reduced dataset point of view

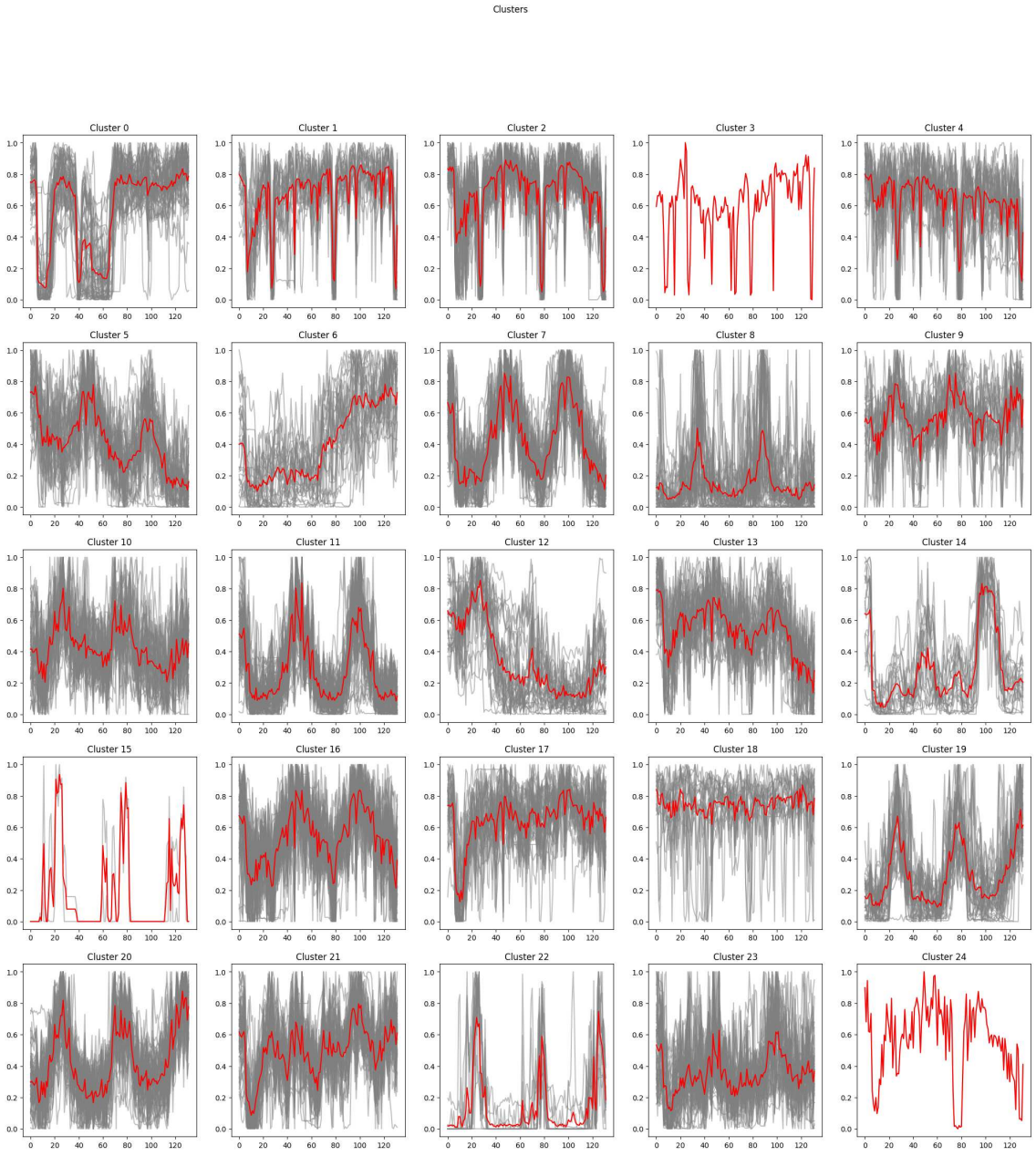


Figure 4.31 – SOM applied to the whole smoothed time series over two years

Chapter 5

Discussion

This final chapter serves as a reflective discussion on the journey undertaken in this thesis, focusing on Advanced Clustering and AI-Driven Decision Support System (DSS) for Smart Energy Management. We'll revisit the key themes explored, from the conceptual framework of complex socio-technical systems to the intricate challenges of developing ai-driven DSS and applying clustering methods to mixed data types. Each section aims to critically assess our approaches, acknowledge the limitations encountered, and suggest directions for future work. This chapter is an opportunity to consider the broader implications of our findings, the gaps in current methodologies, and the potential for further research in the field. Through this discussion, we hope to encapsulate the insights gained and contribute to the ongoing dialogue in energy management and data analysis.

5.1 Limits of the recommender system

5.1.1 Limits of the Decision Support System Architecture's Automation

The DSS has several limitations on its entire chain of information. Each paragraph is focused on a limit.

Datalake : The selection of the relevant data in the datalake must be performed with extreme seriousness and irrelevant data sources must be dealt with promptly. Otherwise the datalake will become a « garbage dump ». This process must be conducted continuously as the data sources are always evolving. This process is not automatable as it requires understanding of the clients and company needs.

Datamarts : Similarly, the preprocessing of the data must be adapted to the evolution of data being sent as well as to the evolution of the clients needs. Monitoring changes and adapting the entire data chain is the role of data lineage.

The **automated selection of algorithms** based on certain performance criteria is accomplished through tools such as multi-armed bandit. However, identifying the relevant criteria for the selection of algorithms requires understanding of the clients' needs. They may want explainable models (White Box models) or simply want efficient models (more likely Black Box models).

Knowledge extraction of building clustering is bound to be to be one of the most challenging aspect for the DSS. Indeed, **clustering** being a non-supervised method, there is no certainty of obtaining an interpretable result and the interpretation requires a profound understand of building consumptions types. Even with algorithms displaying the importance of each feature in the clustering process, human expertise is still required.

Setting up the **recommendations** for energy performance based on building profile can only be performed either by Machine Learning (ML), or by human input and with the knowledge of the existing and experimented recommendations. To calibrate the recommendation system through ML, one would need a database of building energy performance actions and their effects on building consumption. Using this, one would correlate building type with successful building performance actions. The other option is to analyse the clusters through human expertise and to calibrate the recommendations of the system based on that human expertise.

Retroaction based on the effect of the energy performance recommendations will be difficult because of the time factor in most energy performance actions, the time between energy performance action and analysable result might be long and therefore the system will learn slowly.

5.1.2 Limits regarding the data

In this thesis, we have emphasized the critical roles of Microservices Architecture (MSA), Development and Operations (DevOps) methodologies, feedback cycles, and quality assessments in the domain of data-driven decision-making. These aspects are crucial across various sectors, including energy systems, medicine, marketing, finance, law, and biology, where poor data quality—marked by incompleteness, inconsistency, or noise—can significantly skew clustering results and the effectiveness of subsequent recommendations. The challenge of clustering becomes even more pronounced when dealing with diverse data types, such as numerical, categorical, and time series data.

Furthermore, we highlighted the necessity for robust data pre-processing, sophisticated cleansing techniques, the development of resilient algorithms, adherence to DevOps best practices, and the implementation of a Distributed Architecture.

However, it's important to recognize that even the most rigorous practices and tools cannot overcome the fundamental limitations posed by the quality of the dataset itself. Creating a comprehensive and high-quality dataset is frequently obstructed by privacy concerns and the proprietary nature of data, particularly in sensitive sectors like medicine and finance. Within the context of an industrial thesis with a Trusted Third Party for Energy Measurement and Performances (TTPEMP), assembling a large dataset of clean, detailed, and comprehensive data for analysis has proven to be a surprisingly hard challenge. As of this writing, we are in the process of compiling a dataset that includes tens of thousands of buildings, encompassing more than two years of historical data at a detailed time step, and furnished with sufficient descriptive data for effective clustering.

This thesis highlights the fact that while methodologies and architectures can establish a foundation for efficient data management and analysis, the quality and breadth of the underlying dataset are crucial in realizing the full potential of any data-driven system, particularly in the context of complex clustering and recommendation scenarios.

5.1.3 Complex System Analysis

Complex system analysis plays a crucial role in understanding and optimizing energy building clustering for energy efficiency recommendation systems. However, it's essential to acknowledge its inherent limitations. Firstly, complex systems are by nature intricate and dynamic, comprising numerous interconnected components that can exhibit emergent behaviors. This complexity often leads to challenges in accurately modeling and predicting system dynamics, especially when considering the diverse interactions within energy building clusters.

Secondly, the availability and quality of data significantly impact the effectiveness of complex system analysis. In the context of energy efficiency recommendation systems, data collection and integration from various sources such as building sensors, weather databases, and occupant behavior records is often incomplete or unreliable. This can introduce uncertainties and biases into the analysis, potentially limiting the accuracy of recommendations generated by the system.

Moreover, complex system analysis often relies on simplifying assumptions and models to make the problem tractable. While these simplifications aid in understanding system behavior, they can oversimplify reality and overlook critical nuances present in the actual system. In the context of energy building clustering, factors such as building heterogeneity, temporal variability, and external influences may not be fully captured by simplified models, leading to suboptimal recommendations.

Furthermore, the scalability of complex system analysis poses a challenge when dealing with large-scale energy building clusters. As the number of interconnected buildings increases, the computational complexity of analyzing the system grows exponentially. This can strain computational resources and limit the feasibility of performing comprehensive analyses, especially in real-time or near-real-time scenarios where timely recommendations are essential.

5.1.4 Machine Learning Factory implementation

Implementing an Machine Learning Factory (ML-Factory) within a DevOps methodology presents unique challenges and considerations, particularly in the context of energy building clustering for energy efficiency recommendation systems. Firstly, integrating ML pipelines into the DevOps workflow requires careful orchestration to ensure seamless collaboration between data scientists, software developers, and operations teams. This involves establishing automated processes for model training, testing, deployment, and monitoring that align with the principles of Continuous Integration/Continuous Deployment (CI/CD).

Secondly, managing the lifecycle of ML models within a DevOps framework entails addressing version control, reproducibility, and scalability challenges. Unlike traditional software artifacts, ML models are sensitive to changes in data distributions, feature engineering techniques, and hyperparameters, necessitating robust versioning and tracking mechanisms. Additionally, deploying ML models at scale requires efficient resource allocation, monitoring, and scaling strategies to accommodate varying workloads and ensure reliable performance in production environments. (see [23]

Furthermore, ensuring the reliability and interpretability of ML-driven recommendations in energy building clustering presents additional complexities. The opaque nature of some ML algorithms may hinder stakeholders' understanding of model predictions and recommendations, posing challenges for validation, trust, and regulatory compliance. Therefore, incorporating explainability and interpretability techniques into the ML pipeline is essential for enhancing transparency, accountability, and user acceptance of the recommendation system.

Moreover, integrating feedback loops and continuous learning capabilities into the ML-Factory enables adaptive optimization of energy efficiency recommendations over time. This involves collecting real-time feedback from building sensors, user interactions, and environmental factors to refine and update ML models iteratively. However, implementing effective feedback mechanisms requires robust data infrastructure, anomaly detection algorithms, and model retraining pipelines to detect drift, mitigate biases, and adapt to evolving system dynamics.

To enhance the efficiency and effectiveness of the ML-Factory, integrating AutoML

(Automated Machine Learning) capabilities can offer several compelling advantages. AutoML streamlines and automates various stages of the machine learning pipeline, from data preprocessing to model selection and hyperparameter tuning, thereby accelerating model development and deployment processes. By leveraging AutoML within the ML-Factory framework, organizations can significantly reduce the time and resources required for building and maintaining machine learning models, allowing data scientists and engineers to focus on higher-level tasks and innovation.

AutoML also democratizes machine learning by enabling individuals with varying levels of expertise to develop sophisticated models without extensive knowledge of machine learning algorithms or programming languages. This democratization expands the pool of individuals capable of contributing to the ML-Factory, fostering collaboration and innovation across diverse teams within the organization.

Furthermore, AutoML enhances model performance and generalization by systematically exploring a wide range of algorithms, preprocessing techniques, and hyperparameter configurations. This exhaustive search helps identify optimal model architectures and configurations tailored to specific datasets and problem domains, ultimately leading to more accurate and robust machine learning models.

Additionally, AutoML facilitates model reproducibility and transparency by automatically documenting the entire model development process, including data preprocessing steps, model architectures, hyperparameters, and evaluation metrics. This transparency enhances trust and accountability in the ML-Factory's outputs, enabling stakeholders to understand and validate model decisions effectively.

Incorporating AutoML into the ML-Factory not only accelerates model development but also ensures scalability and adaptability to evolving data and business requirements. By automating repetitive tasks and leveraging computational resources efficiently, AutoML enables the ML-Factory to handle large-scale datasets and complex modeling tasks effectively, positioning organizations to derive actionable insights and maintain a competitive edge in dynamic market environments.

In conclusion, integrating AutoML capabilities into the ML-Factory represents a strategic investment to enhance productivity, democratize machine learning, improve model performance, ensure transparency and reproducibility, and foster scalability and adaptability. By harnessing the power of AutoML, organizations can unlock the full potential of their data assets and accelerate innovation in delivering AI-driven solutions to address complex business challenges.

5.1.5 Complex data clustering

The current state-of-the-art clustering methods face several limitations when dealing with mixed data types, such as combining numerical, categorical, and time series data within energy building clustering for energy efficiency recommendation systems. One challenge lies in appropriately representing and handling diverse data types within the clustering algorithm. Traditional methods may struggle to effectively capture the inherent relationships and dependencies present in mixed data, leading to suboptimal clustering results.

Another limitation arises from the differing scales and distributions of various data types. Numerical data, for example, may have different ranges and variances compared to categorical or time series data. Clustering algorithms designed for homogeneous data may struggle to accommodate such differences, potentially leading to biased cluster assignments or inaccuracies in representing the underlying structure of the data.

Furthermore, mixed data clustering often requires specialized techniques for similarity or distance computation between heterogeneous data types. While some methods

exist for handling specific combinations of data types, such as numerical and categorical data, integrating time series data adds another layer of complexity. Time series data inherently contains temporal dependencies and patterns that may not be adequately captured by traditional distance metrics, requiring tailored approaches for effective clustering.

Additionally, the scalability of clustering methods when dealing with mixed data and time series can be a significant concern. As the dimensionality and volume of data increase, computational resources and processing time may become prohibitive, particularly in real-time or large-scale applications. This scalability issue poses challenges for implementing clustering algorithms within energy efficiency recommendation systems for complex building clusters, where timely and resource-efficient analysis is crucial.

Moreover, the interpretability and explainability of clustering results in the context of mixed data and time series can be limited. While clustering algorithms can identify patterns and group similar data points, understanding the underlying reasons for cluster assignments may be challenging, especially when dealing with heterogeneous data types. This lack of interpretability can hinder the adoption of clustering-based approaches in practical applications where actionable insights are essential for decision-making.

In conclusion, while clustering methods have shown promise in analyzing mixed data types, including time series data, several limitations must be addressed to enhance their effectiveness in energy building clustering for energy efficiency recommendation systems. Overcoming these limitations requires advancements in algorithmic development, scalability, interpretability, and integration with domain-specific knowledge to enable more robust and actionable insights for optimizing energy efficiency in complex building environments.

5.1.6 Discussion on Design Formalism

In contemplating future work on the design and development of the DSS detailed in Chapter 2, the integration of Unified Modeling Language (UML) offers a compelling pathway forward. UML's role as a standardized design pattern, endorsed both in academia and industry, provides a robust foundation for articulating the system's architecture explicitly. Its capacity to adapt and describe emerging technologies, including blockchain-based IT systems as noted by Górski [60], renders it particularly suitable for our purposes. Given UML's prevalence and its regular updates, employing UML for both theoretical validation and practical applicability ensures the DSS's relevance and usability within the developer community.

Future work will focus on employing UML for a thorough modeling of the DSS's components, as outlined in Section 2. This includes leveraging UML to delineate dynamic behaviors, data structures, and component interactions, which are crucial for the DSS's operation within energy systems management. Specifically, the creation of UML class diagrams could provide deep insights into the system's data model, while sequence diagrams could shed light on operational workflows. Deployment diagrams would also be vital, offering a view of the DSS's infrastructure and elucidating how components are distributed across physical and virtual resources.

The synergy between UML and DevOps methodologies, particularly the Model-to-Code Transformation, presents a vital area for future exploration, as discussed by Górski [57]. This approach aligns with the continuous practices essential in today's agile development environments, enabling rapid prototyping, continuous integration, and deployment. Integrating UML into the development workflow could significantly streamline the transition from design to implementation, enhancing the DSS's development cycle and facilitating its iterative refinement.

Engaging the developer community through UML modeling of the DSS offers a struc-

tured framework that fosters better understanding and collaboration. This strategy not only supports the technical realization of the DSS but also ensures the design's robustness, scalability, and alignment with user needs.

In summary, future endeavors should aim to leverage UML's comprehensive modeling capabilities to encapsulate the intricate dynamics of the DSS, emphasizing its integration with DevOps practices for a more collaborative and responsive development process. These efforts are anticipated to significantly advance the design, implementation, and usability of the DSS in managing complex energy systems, as envisaged in this thesis.

5.2 Improvements concerning Complex Data Clustering

5.2.1 Elbow Method

In many instances, clusterings relying on the elbow method have shown unsatisfactory results. The core of the issue might be in the application of Gower's distance matrix across the entire dataset without dimensionality reduction or data preprocessing. This might lead to an « inadequate » number of clusters. Gower's distance is especially advantageous for mixed data types, as it adeptly manages both categorical and numerical data, creating a comprehensive distance matrix. Nevertheless, this method can sometimes veil the inherent clustering patterns within the data, especially in high-dimensional spaces where the curse of dimensionality might dilute meaningful distances between observations.

One possible solution to enhance the relevance of the elbow method would involve « correcting » the approach by calculating the elbow on a reduced dataset rather than using the Gower distance matrix on the complete dataset. This strategy aims to simplify the data structure and highlight more pronounced clustering tendencies, potentially leading to more accurate and meaningful determination of the optimal cluster count. Adopting a dimensionality reduction step could mitigate the effects of irrelevant features and reduce noise, allowing the elbow method, when applied to the resulting simplified dataset, to more effectively discern the point at which adding more clusters yields diminishing returns. Consequently, this « correction » could lead to significantly better clustering results, offering a more precise understanding of the data's inherent groupings and facilitating more informed decision-making based on the identified clusters.

5.2.2 Feature selection

We have not explicitly addressed the feature selection step in every machine learning pipeline. Feature selection, a critical phase in every machine learning pipeline, involves identifying and selecting a subset of relevant features from the dataset to simplify the model, improve its performance, or reduce overfitting. This process is particularly pivotal when dealing with mixed datasets, as highlighted by Li et al. [97], due to their susceptibility to the Curse of Dimensionality. Feature selection effectively addresses this challenge by reducing the dataset's dimensionality, thereby enhancing the efficiency and accuracy of subsequent analyses.

In mixed datasets, where both categorical and continuous variables coexist, feature selection must navigate the complexity introduced by this diversity. While categorical features can be directly compared with other types of features, analyzing multicollinearity without resorting to Dimensionality Reduction (DR) techniques proves to be challenging. Converting categorical into numerical features facilitates this analysis, allowing for the

application of statistical tests like chi-square, t-tests, or mutual information to assess the importance and information content of features.

However, in distance-based clustering or classification methods, the distinction between continuous and discrete values diminishes, overshadowed by the challenges of feature scaling and normalization. Such challenges are exacerbated in datasets characterized by sparsity or high levels of noise. While exploring features can unveil varying insights into the dataset, these insights may not be directly comparable across categorical and numerical features due to their inherent differences.

Furthermore, employing metrics such as mutual information or entropy necessitates discretizing numerical features, potentially oversimplifying their complexity. This simplification, while reducing computational demands, may also diminish the nuanced information these features carry. Additionally, the statistical assessment of feature significance often requires distinct approaches for categorical and numerical features, emphasizing the need for meticulous feature selection in mixed data contexts to preserve the integrity and interpretability of the analysis.

For a deeper understanding of feature selection techniques and their critical role in managing mixed datasets, consulting resources like the SPMF database by Philippe Fournier-Viger or the comprehensive review by Li et al. [97] can provide valuable insights. Moreover, embracing feature selection not only as a preprocessing step but also as a strategy for enhancing model interpretability and performance is essential in the realm of machine learning, especially when navigating the complexities of mixed datasets.

5.2.3 Distance metric selection and Clustering Comparison

Many of the metrics commonly applied in the context of mixed data comparison are originally designed for classification tasks rather than clustering, as the Adjusted Rand Index (ARI) for example. Presently, the prevailing best practice involves utilizing metrics tailored for quantitative data and adapting them for mixed data.

The primary limitation of these methods lies in the considerable loss of information due to DR. Moreover, a substantial portion of these metrics primarily assesses the compactness of clusters and their overall design. However, it is important to note that, similar to the behavior of the DBSCAN algorithm, a clustering algorithm for mixed data may not necessarily identify spherical clusters. Consequently, the challenge of identifying suitable metrics for assessing mixed data clusters persists.

Some measures in this domain rely on information-theory concepts such as entropy or mutual information but often demand a significant amount of memory for computation. In the realm of mixed data clustering, there is a pressing need to adapt these methods to gauge how the quality of clustering might degrade if the clusters were to undergo changes. Introducing a sense of mathematical logic that governs the relationships between elements could enhance our understanding of both metrics and algorithms in this context.

Another avenue to address the challenges of mixed data clustering involves examining entanglement, which refers to the similarity between two different clusterings. Given the absence of ground truth and the limitations of internal measures, introducing a measure to assess the similarities or differences between clusterings can provide valuable insights into their results. For instance, a strategy could entail selecting the clustering solution that exhibits the highest average ARI in regard to the other clustering solutions.

Furthermore, comparing two clusterings, independent of the methods/preprocessing used, that yield substantially different numbers of clusters poses a challenge. Even if one clustering appears to exhibit worse indicators, it can be challenging to immediately conclude its inferiority, particularly when the differing numbers of clusters offer distinct

interpretations of the dataset.

5.2.4 Data Mining

Data mining, as a practice, entails the automated exploration of vast datasets to uncover underlying trends and patterns that extend beyond conventional analysis. Data mining often finds application in Exploratory Data Analysis, facilitating a deeper understanding of the inherent relationships among data objects. For an exhaustive review of data mining methods, metrics, and algorithms, we recommend consulting the SPMF database curated by Philippe Fournier-Viger [1]. Additionally, Mirkin [111] provides an extensive introduction to clustering methods specifically tailored for data mining, with a particular emphasis on mixed data types.

One promising avenue within the domain of data mining is the potential creation of logical graphs or graph structures based on metrics (such as confidence or lift) to leverage their insights. Analysis of such graphs can encompass techniques like community clustering (akin to the Louvain algorithm, as discussed in [44]) or multi-level clustering approaches, as proposed by [42].

In practice, data mining often serves as a preprocessing step that enables the modeling of relationships between data objects, thereby providing a novel perspective for applying clustering methods.

5.2.5 Time series

Another method for handling time series comparisons is the Temporal Distortion Index (TDI) proposed by [52]. TDI is a dimensionless metric ranging between 0 and 1, where 0 signifies zero temporal distortion and 1 represents maximum temporal distortion. The bounded nature of this measure enhances its interpretability compared to Dynamic Time Warping (DTW).

Additionally, we introduce the RdR score [2] as a novel approach. It involves comparing the time series curves to a ground truth. In this context, consider a k-Means clustering approach where the means represent the ground truth. It becomes feasible to compute RdR scores for each time series in relation to each ground truth, and assign them to the cluster that yields the best score. Subsequently, new ground truths are computed as the means of the time series within each cluster. This process iterates until the clusters stabilize.

Furthermore, a persistent challenge lies in dealing with complex data. Just as metrics and algorithms are often ill-suited for mixed data, analogous challenges emerge when grappling with datasets that incorporate time series information.

5.2.6 Clustering methods and limitation

As our datasets comprise mixed data and necessitates a structure between clusters to establish a clear relationship while ensuring their interpretability and explicability, hierarchical clustering emerges as the most suitable approach. Notably, pretopological clustering methods have demonstrated superior performance in several datasets. Subsequent research endeavors will be directed toward further refining and enhancing this methodology.

Interpreting the results of mixed data clustering poses a notable challenge. This complexity arises from the amalgamation of diverse data types within the clustering process,

1. https://www.philippe-fournier-viger.com/spmf/
2. https://github.com/CoteDave/blog/tree/master/RdR%20score

resulting in clusters that may not readily lend themselves to intuitive interpretation. Additionally, some clustering algorithms may lack transparency in elucidating the mechanics behind cluster formation, rendering it difficult to discern the underlying data patterns. Interpretability is a crucial factor, particularly in applications where clustering outcomes inform decision-making processes or guide subsequent analyses.

Furthermore, the realm of eXplainable Artificial Intelligence (Explainable Artificial Intelligence (XAI)) assumes paramount importance for any novel algorithm. Hierarchical clustering, such as the pretopological clustering employed here, offers a valuable advantage in this regard. The dendrogram generated by hierarchical clustering can be harnessed to gain deeper insights into the inherent relationships between each cluster within the hierarchy, thus enhancing the algorithm's transparency and interpretability.

Conversely, the concept of robustness in clustering pertains to an algorithm's ability to consistently produce reliable results despite the presence of noise, outliers, and various sources of data variability. Mixed data clustering introduces unique challenges in maintaining robustness, given the diverse nature of data types that may be influenced by distinct sources of variability.

To foster a comprehensive understanding of clustering results, we recommend employing multiple clustering algorithms for evaluation. Each algorithm possesses its own strengths and weaknesses, and a multifaceted analysis approach can enrich discussions by providing a more holistic perspective on the outcomes. Detecting variations between results can be particularly informative, as it aids in identifying biases, such as systematic errors or distortions, which may lead to inaccurate or misleading interpretations of the clustering results.

5.2.7 Deep Learning

Neural Network, and more specificaly, deep learning was presented as part of the state of the art on mixed clustering.

Deep learning, with its robust feature extraction capabilities and flexibility in handling various data types, is a powerful approach for tackling the intricacies of mixed and complex datasets [17]. Deep learning models, renowned for their ability to learn hierarchical representations, offer a unique advantage in processing mixed data. Through layers of abstraction, these models can uncover latent structures and relationships within the data, facilitating more nuanced clustering and predictive analytics. Techniques like autoencoders for dimensionality reduction and recurrent neural networks (RNNs) for time-series analysis exemplify the potential of deep learning in extracting meaningful insights from complex datasets.

However, the application of deep learning to mixed and complex data goes beyond mere data representation. It encompasses the development of novel architectures and training methodologies that can inherently accommodate the heterogeneity of data types, enhancing the models' ability to learn from and interpret such data effectively.

Despite the apparent potential of deep learning in this domain, its exploration within the scope of this thesis was constrained by several factors. Primarily, the intensive computational resources required for training deep learning models posed a significant challenge. Deep learning's reliance on large datasets for training, coupled with the need for high-performance computing infrastructure (e.g., GPUs), rendered it a less feasible option given the time and resource limitations encountered during the research period.

Additionally, the development and fine-tuning of deep learning models for mixed and complex data require specialized knowledge and expertise. The intricacies involved in designing models that can seamlessly integrate and analyze diverse data types necessitate a thorough understanding of both the data and the underlying deep learning technologies.

Looking ahead, the integration of deep learning into our research framework for clustering mixed and complex data is a central component of our planned future work. Recognizing its untapped potential, we aim to allocate resources towards developing deep learning models tailored to our specific dataset characteristics.

5.2.8 Dataviz

An effective approach to address the explainability issues associated with clustering, particularly in the case of complex data, is to have a comprehensive understanding of both the dataset and the various steps involved in a clustering method, while adhering to logical rules and parameter settings. Developing improved visualization tools that can help in understanding the value of clusters in high-dimensional contexts is also an important area of research.

In the realm of mixed data clustering, we often encounter datasets with multiple types of data interrelated in intricate ways. Representing these complex relationships in a meaningful manner can pose a significant challenge, particularly when dealing with non-linear or high-dimensional relationships among data types.

To address these formidable challenges, it would seem interesting to employ a combination of visualization techniques, including heatmaps, scatterplots, and network graphs. These tools would enable us to effectively depict the various data types and their intricate relationships. Another approach involves DR to transform the data into a more manageable Euclidean space. However, it's crucial to recognize that such approaches may present only one facet of the problem or potentially distort the true nature of the dataset. As previously discussed regarding clustering method limitations, maintaining explainability and robustness is essential for results usability the results fully. Proper interpretation is vital to ensure that the clustering outcomes carry meaningful insights and can guide informed decision-making processes.

One notable bottleneck encountered during our study is the absence of dedicated methods for visualizing mixed data. This deficiency becomes evident when examining resources like data-to-viz [3], a platform that catalogs data visualization methods across R, Python, and d3.js [4] (used as the foundation for Plotly in Python). Notably, there are currently no established techniques capable of effectively handling both quantitative and qualitative features concurrently. Consequently, the most comprehensible approach for presenting results from mixed data, as demonstrated in our paper, often involves DR followed by the application of conventional visualization methods.

This domain remains an uncharted challenge, and the potential solution may lie in dynamic graph representations, such as those demonstrated in d3.js. However, addressing the challenge of displaying dynamic graph-based results in a traditional paper format remains an ongoing area of exploration.

5.2.9 Application to our case study

We are currently undertaking the project of assembling a comprehensive database that will include detailed information on over 10,000 buildings. Each entry will feature more than two years of energy consumption history, alongside descriptive data such as building typology, geographic coordinates (latitude and longitude), altitude, and internal surface area. This endeavor forms the cornerstone of our thesis, enabling us to explore the intricate patterns of energy usage across a diverse array of buildings.

3. https://www.data-to-viz.com/
4. https://d3js.org/

Upon completion of the database, a meticulous statistical analysis will be carried out to discern the most suitable clustering technique for our dataset. This analysis will draw upon insights from the preliminary discussions presented in Chapter 4 of this thesis. Preliminary evaluations suggest that PretopoMD with PaCMAP, K-means with FAMD, and AHC and PretopoMD may emerge as promising candidates. The precise hyperparameters, including the construction of the pretopological space for PretopoMD and the distance function for AHC, will be fine-tuned in collaboration with energy management experts from Energisme.

The analysis of the clusters, conducted with the valuable input of Energisme's experts in the field of energy management, aims to fulfill two critical objectives. Initially, it will facilitate the generation of initial hypotheses regarding potential energy inefficiencies and recommendations tailored to distinct clusters. This process is essential for translating complex data patterns into practical, actionable insights, enhancing the efficacy of energy efficiency measures and deepening our understanding of energy consumption behaviors across different building types through a collaborative effort between data-driven methodologies and domain-specific knowledge.

In concluding this chapter, we've explored the complexities of developing AI-driven DSS and applying advanced clustering techniques within the context of smart energy management. This discussion has emphasized the nuanced challenges faced in modeling complex socio-technical systems, the critical importance of data quality, and the intricacies of algorithm selection and system design. Through a reflective lens, we have identified key limitations and areas for future improvement, grounding our exploration in the realities of current methodologies and the practicalities of implementation.

The process of compiling a comprehensive dataset for analysis, refining the architecture of decision support systems, and exploring effective clustering methods has highlighted the multifaceted nature of this research area. These tasks, while challenging, outline a clear direction for advancing the field of smart energy management through data-driven approaches.

Future work will inevitably need to address the limitations identified throughout this thesis, particularly focusing on enhancing data quality, improving system scalability and flexibility, automating hyperparameter optimization and developing more interpretable clustering techniques. The importance of interdisciplinary collaboration has also been underscored, as the convergence of expertise from various domains will be crucial in overcoming the current challenges and pushing the boundaries of what can be achieved in smart energy management.

As this thesis concludes, it's evident that the journey of innovation and discovery in AI-driven energy management systems is ongoing. The insights and reflections shared in this chapter serve as a foundation for further research, with the goal of not only addressing the challenges identified but also exploring new opportunities for advancement. The path forward is marked by a commitment to continuous improvement and the pursuit of more efficient, reliable, and user-centric energy management solutions.

Summary of Chapter 5

Conceptual Framework and Challenges : Exploration of complex socio-technical systems, development challenges for AI-driven DSS, and the application of clustering methods to mixed data types.

Recommender System Limitations : Challenges with DSS architecture automation across various stages, from data lake management to algorithm selection. The non-automatable nature of certain processes due to the need for deep understanding and continuous adaptation.

Data Limitations : The critical role of high-quality, comprehensive datasets and the challenges in creating such datasets due to privacy concerns and data availability.

Complex System Analysis : The inherent limitations in modeling dynamic, interconnected components of energy building clusters and the impact of data quality on system effectiveness.

ML-Factory Implementation : The unique challenges and considerations for integrating ML pipelines within a DevOps methodology, focusing on scalability, reliability, and interpretability. The potential of AutoML to enhance efficiency, democratization, and scalability of machine learning in energy management.

Design Formalism : Future work considerations, including the use of UML for detailed system modeling and its integration with DevOps for improved DSS design and implementation.

Improvements in Data Clustering : Addressing challenges in the elbow method, feature selection, distance metric selection, and the need for improved data visualization methods. The ongoing challenge of clustering mixed data and the potential directions for enhancing interpretability and robustness of clustering results.

Addition of Deep Learning methods to the Benchmark : Future work include the implementation of Deep Learning methods, as they are a powerful tool for clustering mixed data and time series.

Application to Case Study : The endeavor to compile a comprehensive database for energy management and the collaboration with experts to fine-tune clustering techniques and develop energy efficiency recommendations.

Conclusion

This thesis addresses the problem of clustering of complex and heterogeneous energy systems within a Decision Support System (DSS).

For this purpose, we delved into the theory of complex systems and their modeling, recognizing buildings as Complex Systems, specifically as Complex Socio-Technical Systems. We examined the state of the art of the different agents involved in energy performance within the energy sector, identifying our case study as the Trusted Third Party for Energy Measurement and Performances (TTPEMP). Given our constraints, we opted to concentrate on the need for a DSS to provide energy recommendations. We identified the necessity for explainability in AI-aided decision-making (XAI) in high stakes scenarios. Acknowledging the complexity, numerosity, and heterogeneity of buildings managed by the TTPEMP, we argued that clustering serves as a pivotal first step in developing a DSS, enabling tailored recommendations and diagnostics for homogeneous subgroups of buildings.

During this thesis we argued that our problematic can be addressed by proposing a DSS distributed architecture comprising a Data Lake, Datamarts, an ML-Factory, and an algorithm library featuring machine learning methods, including clustering. The governance of such complex semi-automated system requires methodologies such as DevOps and data lineage to address the identified needs for Accuracy, Reliability and Fairness. States of the art clustering methods had to be adapted to the specificities of our case study. Indeed, the datasets we manipulate are composed of numerical, categorical and time series data. We coined the term Complex Clustering to address the clustering of this combination of data types. Such methods include Dimensionality Reduction technics, adaptation of mixed or numerical state-of-the-art technics and the use of Pretopological clustering with custom proximity definition. Statistical evaluation of the clusters using methods such as Calinsky-Harabasz, Silhouette, or Davis-Bouldins scores were adapted to this context by using Dimensionality Reduction.

Our contributions include proposing a comprehensive DSS architecture for the TTPEMP and developing solutions that leverage state-of-the-art clustering techniques for Complex Datasets. We identified innovative ways to evaluate these clustering. We analyzed the computational performances of algorithms and the quality of clusters across datasets varying in size, number of clusters, distribution, and number of categorical and numerical parameters. We analyzed the specific strengths and shortcomings of the methods, identifying how Pretopology and Dimensionality Reduction show promising results, especially in large dataset, with high standard deviation in clusters. This effort was complemented by the creation of a dedicated library, and of interactive tools for visualization and evaluation of clustering methods. All of these works have been the subject of internal reports as well as international publications [91, 92, 93, 94, 95, 96].

Yet we have acknowledged that the proposed clustering methods and the DSS exploiting them are limited by several factors. Each step of the dataflow in the DSS architecture presents limitations, ranging from the quality and completeness of the data to the difficulty in evaluating the effectiveness of the recommendation, as well as the difficulty of

converging on a clustering that is both statistically sound and that is judged business-relevant by experts. The technical issues regarding the maintenance and update of the ML-Factory were also discussed.

Our research is actively continuing as this thesis is written. We are currently working on automating the adjustment of the pretopological space parameters to better match the data's unique characteristics, automatic hyperparametrization efforts are to be made for several other methods. Clustering using deep learning methods is also planned in order to have a more comprehensive benchmark. The effort to compile a comprehensive dataset comprising over 10,000 buildings is ongoing. Moreover, the evaluation of our clustering results by industry experts has just begun, with their feedback still emerging. Our ongoing research efforts are also focused on further enhancing the Datamarts, Machine Learning and Decision Support components of our infrastructure.

Résumé en Français

Cette thèse traite du regroupement de systèmes énergétiques complexes et hétérogènes au sein d'un système d'aide à la décision (SAD).

Dans le chapitre 1, nous abordons la théorie des systèmes complexes et leur modélisation, en reconnaissant les bâtiments comme des systèmes complexes, plus précisément comme des systèmes sociotechniques complexes. Nous examinons l'état de l'art des différents agents impliqués dans la performance énergétique du secteur de l'énergie, en identifiant notre cas d'étude comme étant celui du Tiers de Confiance pour la Mesure et la Performance Énergétique (TCMPE). Compte tenu de nos contraintes, nous choisissons de nous concentrer sur le besoin d'un SAD pour fournir des recommandations énergétiques. Nous comparons ce système aux systèmes de supervision et de recommandation, en soulignant leurs différences et leurs complémentarités et introduisons la nécessité d'une explicabilité dans la prise de décision assistée par l'IA (XAI). Reconnaissant la complexité, la numérosité et l'hétérogénéité des bâtiments gérés par le TCMPE, nous soutenons que le regroupement en clusters est une étape essentielle pour développer un SAD, permettant des recommandations et diagnostics adaptés à des sous-groupes homogènes de bâtiments.

Dans le chapitre 2, nous explorons l'état de l'art des SAD, en mettant l'accent sur la nécessité de la gouvernance dans les systèmes semi-automatisés pour la prise de décision à enjeux élevés. Nous examinons les réglementations européennes, en mettant en évidence le besoin de précision, de fiabilité et d'équité dans notre système de décision, et identifions des méthodologies pour répondre à ces besoins, telles que la méthodologie DevOps et la traçabilité des données (Data Lineage). Nous proposons une architecture de SAD qui répond à ces exigences et aux défis posés par le big data, avec une architecture distribuée comprenant un data lake pour la gestion des données hétérogènes, des datamarts pour la sélection et le traitement des données spécifiques, et une ML-Factory alimentant une bibliothèque de modèles. Différents types de méthodes sont sélectionnés pour différents besoins en fonction des spécificités des données et de la problématique à traiter.

Le chapitre 3 se concentre sur le regroupement en clusters en tant que méthode principale d'apprentissage automatique dans notre architecture, essentiel pour identifier des groupes homogènes de bâtiments. Compte tenu de la combinaison des données numériques, catégorielles et des séries temporelles décrivant les bâtiments, nous inventons le terme « regroupement complexe » pour aborder cette combinaison de types de données. Après avoir passé en revue l'état de l'art, nous identifions le besoin de techniques de réduction de dimensionnalité et les méthodes de regroupement mixtes les plus pertinentes. Nous introduisons également la prétopologie comme une approche innovante pour le regroupement de données mixtes et complexes. Nous soutenons qu'elle permet une plus grande explicabilité et interactivité dans le regroupement en permettant le regroupement hiérarchique et la mise en œuvre de règles logiques et de notions de proximité personnalisées. Les défis de l'évaluation du regroupement sont abordés, et des adaptations du regroupement numérique au regroupement mixte et complexe sont proposées, en te-

nant compte de l'explicabilité des méthodes.

Dans le chapitre 4, portant sur les jeux de données et les résultats, nous présentons des jeux de données publics, privés et générés utilisés pour l'expérimentation et discutons des résultats du regroupement en clusters. Nous analysons les performances computationnelles des algorithmes et la qualité des clusters obtenus sur différents ensembles de données variant en taille, nombre de clusters, distribution, et nombre de paramètres catégoriels et numériques. La Prétopologie et la Réduction de Dimensionnalité montrent des résultats prometteurs par rapport aux méthodes de regroupement de données mixtes de l'état de l'art.

Enfin, dans le dernier chapitre, nous examinons les limitations de notre système, y compris celles liées à l'automatisation du SAD à chaque étape du flux de données. Nous mettons l'accent sur le rôle crucial de la qualité des données et les défis liés à la prédiction du comportement des systèmes complexes dans le temps. L'objectivité de nos méthodes d'évaluation du regroupement est examinée à la lumière de l'absence de données de référence et de la dépendance à la réduction de dimensionnalité pour adapter les métriques de l'état de l'art aux données complexes. Nous abordons les problèmes potentiels liés à la méthode du coude choisie. Une discussion sur les perspectives de travaux futurs s'ensuit, incluant l'automatisation du réglage des hyperparamètres, la poursuite du développement du SAD, et la nécessité d'innover dans la visualisation des données complexes.

Bibliographie

[1] Energy Performance of Buildings Directive. URL https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficient-buildings/energy-performance-buildings-directive_en.

[2] Regulation (EU) 2021/1119 of the European Parliament and of the Council of 30 June 2021 establishing the framework for achieving climate neutrality and amending Regulations (EC) No 401/2009 and (EU) 2018/1999 ('European Climate Law'), June 2021. URL http://data.europa.eu/eli/reg/2021/1119/oj/eng. Legislative Body : CONSIL, EP.

[3] Nur Najihah Abu Bakar, Mohammad Yusri Hassan, Hayati Abdullah, Hasimah Abdul Rahman, Md Pauzi Abdullah, Faridah Hussin, and Masilah Bandi. Energy efficiency index as an indicator for measuring building energy performance : A review. *Renewable and Sustainable Energy Reviews*, 44 :1–11, April 2015. ISSN 13640321. doi : 10.1016/j.rser.2014.12.018. URL https://linkinghub.elsevier.com/retrieve/pii/S1364032114010703.

[4] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6) :734–749, 2005. Publisher : IEEE.

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

[6] Murat Ahat, Soufian Ben Amor, Marc Bui, Alain Bui, Guillaume Guérard, and Coralie Petermann. Smart Grid and Optimization. *Am. J. Oper. Res.*, 03(01) :196–206, 2013. ISSN 2160-8830, 2160-8849. doi : 10.4236/ajor.2013.31A019. URL http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ajor.2013.31A019.

[7] Amir Ahmad and Lipika Dey. A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63(2) :503–527, 2007.

[8] Amir Ahmad and Shehroz S Khan. Survey of state-of-the-art mixed data clustering algorithms. *Ieee Access*, 7 :31883–31902, 2019.

[9] Amir Ahmad and Shehroz S Khan. Survey of state-of-the-art mixed data clustering algorithms. *Ieee Access*, 7 :31883–31902, 2019.

[10] Tanveer Ahmad, Huanxin Chen, Yabin Guo, and Jiangyu Wang. A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand : A review. *Energy Build.s*, 165 :301–320, 2018. ISSN 03787788. doi : 10.1016/j.enbuild.2018.01.017. URL https://linkinghub.elsevier.com/retrieve/pii/S0378778817329225.

[11] Derboul Ahmed, Hadj Baraka Ibrahim, and Chafik Khalid. Contribution of industrial information systems to industrial performance : Case of industrial supervision. In *Innovations in Smart Cities and Applications : Proceedings of the 2nd Mediterranean Symposium on Smart City Applications 2*, pages 884–901. Springer, 2018.

[12] Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, Maximilian Strobel, and Daniel Cremers. Clustering with deep learning : Taxonomy and new methods. *arXiv preprint arXiv :1801.07648*, 2018.

[13] Mebarka Allaoui, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. Considerably improving clustering algorithms using umap dimensionality reduction technique : a comparative study. In *International Conference on Image and Signal Processing*, pages 317–325. Springer, 2020.

[14] Roba Alsaigh, Rashid Mehmood, and Iyad Katib. Ai explainability and governance in smart energy systems : A review. *Frontiers in Energy Research*, 11 :1071291, 2023.

[15] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai) : Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58 : 82–115, 2020.

[16] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-temporal data mining : A survey of problems and methods. *ACM Comput. Surv. (CSUR)*, 51(4) :1–41, 2018. Publisher : ACM New York, NY, USA.

[17] K Balaji, K Lavanya, and A Geetha Mary. Clustering of mixed datasets using deep learning algorithm. *Chemometrics and Intelligent Laboratory Systems*, 204 :104123, 2020.

[18] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6) :1373–1396, 2003.

[19] Lorenzo Belussi, Benedetta Barozzi, Alice Bellazzi, Ludovico Danza, Anna Devitofrancesco, Carlo Fanciulli, Matteo Ghellere, Giulia Guazzi, Italo Meroni, Francesco Salamone, et al. A review of performance of zero energy buildings and energy efficiency solutions. *J. Build. Eng.*, 25 :100772, 2019.

[20] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *Database Theory—ICDT'99 : 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7*, pages 217–235. Springer, 1999.

[21] Christophe Biernacki. Bigstat for big data : Big data clustering through the bigstat saas platform. In *Journée scientifique Big Data & Data science*, 2016.

[22] Sofia-Natalia Boemi and Charalampos Tziogas. Indicators for Buildings' Energy Performance. In Sofia-Natalia Boemi, Olatz Irulegi, and Mattheos Santamouris, editors, *Energy Performance of Buildings*, pages 79–93. Springer International Publishing, Cham, 2016. ISBN 978-3-319-20830-5 978-3-319-20831-2. doi : 10.1007/978-3-319-20831-2_5. URL http://link.springer.com/10.1007/978-3-319-20831-2_5.

[23] Jérémie Bosom. *Conception de microservices intelligents pour la supervision de systèmes sociotechniques : application aux systèmes énergétiques*. These de doctorat, Université Paris sciences et lettres, Paris, France, 2020. URL https://www.theses.fr/2020UPSLP051.

[24] Jérémie Bosom, Anna Scius-Bertrand, Haï Tran, and Marc Bui. Multi-agent Architecture of a MIBES for Smart Energy Management. In Michal Hodoň, Gerald Eichler, Christian Erfurth, and Günter Fahrnberger, editors, *Innovations for Community Services*, volume 863, pages 18–32. Springer International Publishing, Cham, 2018. ISBN 978-3-319-93408-2. doi : 10.1007/978-3-319-93408-2_2. URL http://link.springer.com/10.1007/978-3-319-93408-2_2.

[25] Gert Brüggemeier. Organisationshaftung : Deliktsrechtliche Aspekte Innerorganisatorischer Funktionsdifferenzierung. *Archiv für die civilistische Praxis*, 191(H. 1/2) : 33–68, 1991. Publisher : JSTOR.

[26] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1) :1–27, 1974.

[27] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining : 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17*, pages 160–172. Springer, 2013.

[28] Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern recognition*, 28(5) :781–793, 1995.

[29] Smart Grid Coordination CEN-CENELEC-ETSI et al. Cen-cenelec-etsi smart grid coordination group smart grid information security. *http ://ec. europa. eu/energy/gas_electricity/smartgrida/doc/xpert_groupl_security. pdf*, 2012.

[30] Jean-Marc CHARTRES. Supervision : outil de mesure de la production. *Techniques de l'ingénieur. Informatique industrielle*, 2(R7630) :R7630–1, 1997.

[31] Marshall Clemens. Visualizing complex systems sci. (css) : Complex adaptive system model.

[32] Louise K Comfort. Self-organization in complex systems. *Journal of Public Administration Research and Theory : J-PART*, 4(3) :393–410, 1994.

[33] European Commission. In focus : Energy efficiency in buildings - European Commission, 2020. URL https://commission.europa.eu/news/focus-energy-efficiency-buildings-2020-02-17_en.

[34] European Commission. New rules to boost energy performance of buildings, 2023. URL https://ec.europa.eu/commission/presscorner/detail/en/IP_23_6423.

[35] Efthymios Costa, Ioanna Papatsouma, and Angelos Markos. Benchmarking distance-based partitioning methods for mixed-type data. *Advances in Data Analysis and Classification*, pages 1–24, 2022.

[36] Gabriella Crotti, Domenico Giordano, Davide Signorino, Antonio Delle Femine, Daniele Gallo, Carmine Landi, Mario Luiso, A Biancucci, and L Donadio. Monitoring energy and power quality on board train. In *2019 IEEE 10th International Workshop on Applied Measurements for Power Systems (AMPS)*, pages 1–6. IEEE, 2019.

[37] Tom Dannenbaum. Translating the standard of effective control into a system of effective accountability : how liability should be apportioned for violations of human rights by member state troop contingents serving as United Nations peacekeepers. *Harv. Int'l LJ*, 51 :113, 2010. Publisher : HeinOnline.

[38] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2) :224–227, 1979.

[39] Jos de Haan. How emergence arises. *Ecological complexity*, 3(4) :293–301, 2006.

[40] Sidney Dekker. *The Field Guide to Understanding Human Error*. CRC Press, Aldershot, England ; Burlington, VT, 2 edition edition, 2006. ISBN 978-0-7546-4826-0.

[41] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data : experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2) :1542–1552, 2008.

[42] Sonia Djebali, Quentin Gabot, and Guillame Guérard. Tourists profiling by interest analysis. In *Advanced Data Mining and Applications : 17th International Conference, ADMA 2021, Sydney, NSW, Australia, February 2–4, 2022, Proceedings, Part II*, pages 42–53. Springer, 2022.

[43] Aicha Dridi, Hassine Moungla, Hossam Afifi, Jordi Badosa, Florence Ossart, and Ahmed E Kamal. Machine learning application to priority scheduling in smart microgrids. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1695–1700. IEEE, 2020.

[44] Scott Emmons, Stephen Kobourov, Mike Gallant, and Katy Börner. Analysis of network clustering algorithms and cluster quality metrics at scale. *PloS one*, 11(7) : e0159161, 2016.

[45] Energisme. Energisme, 2019. URL https://energisme.com/.

[46] B Escofier. Traitement simultané de variables qualitatives et quantitatives en analyse factorielle. *Cahiers de l'Analyse des Données*, 4(2) :137–146, 1979.

[47] Union European. Directive 2010/31/EU of the European Parliament and of the Council of 19 May 2010 on the energy performance of buildings, 2010.

[48] Alex Foss, Marianthi Markatou, Bonnie Ray, and Aliza Heching. A semiparametric method for clustering mixed data. *Machine Learning*, 105(3) :419–458, 2016.

[49] Alexander H Foss and Marianthi Markatou. kamila : clustering mixed-type data in r and hadoop. *Journal of Statistical Software*, 83 :1–44, 2018.

[50] Armin Fuchs. *Nonlinear dynamics in complex systems*. Springer, 2014.

[51] Ben D Fulcher, Max A Little, and Nick S Jones. Highly comparative time-series analysis : the empirical structure of time series and their methods. *Journal of the Royal Society Interface*, 10(83) :20130048, 2013.

[52] Martín Gastón, Laura Frías, Carlos Fernández-Peruchena, and Fermín Mallor. The temporal distortion index (tdi). a new procedure to analyze solar radiation forecasts. In *AIP Conference Proceedings*, volume 1850, page 140009. AIP Publishing LLC, 2017.

[53] Isaías González, Antonio José Calderón, and José María Portalo. Innovative multilayered architecture for heterogeneous automation and monitoring systems : Application case of a photovoltaic smart microgrid. *Sustain.*, 13(4) :2234, 2021.

[54] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3) :50–57, 2017.

[55] Kristen B Gorman, Tony D Williams, and William R Fraser. Ecological sexual dimorphism and environmental variability within a community of antarctic penguins (genus pygoscelis). *PloS one*, 9(3) :e90081, 2014.

[56] George Anthony Gorry and Michael S Scott Morton. A framework for management information systems. 1971.

[57] Tomasz Górski. Towards continuous deployment for blockchain. *Appl. Sci.s*, 11(24) : 11745, 2021.

[58] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[59] Ben Green and Yiling Chen. The Principles and Limits of Algorithm-in-the-Loop Decision Making. *Proceedings of the ACM on Human-Comput. Interaction*, 3(CSCW) :1–24, 2019. ISSN 2573-0142. doi : 10.1145/3359152. URL https://dl.acm.org/doi/10.1145/3359152.

[60] Tomasz Górski. The 1+5 Architectural Views Model in Designing Blockchain and IT System Integration Solutions. *Symmetry*, 13(11) :2000, October 2021. ISSN 2073-8994. doi : 10.3390/sym13112000. URL https://www.mdpi.com/2073-8994/13/11/2000.

[61] Mahammad A Hannan, Mohammad Faisal, Pin Jern Ker, Looe Hui Mun, Khadija Parvin, Teuku Meurah Indra Mahlia, and Frede Blaabjerg. A review of internet of energy based building energy management systems : Issues and recommendations. *IEEE Access*, 6 :38997–39014, 2018.

[62] Timothy C Havens and James C Bezdek. An efficient formulation of the improved visual assessment of cluster tendency (ivat) algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 24(5) :813–822, 2011.

[63] Barry Haynes, Nick Nunnington, and Timothy Eccles. *Corporate Real Estate Asset Management : Strategy and Implementation*. Taylor Francis, 2017. ISBN 978-1-317-42713-1. URL https://books.google.fr/books?id=xCQlDwAAQBAJ&lpg=PP1&dq=Google-Books-ID%3A%20xCQlDwAAQBAJ&hl=fr&pg=PP1#v=onepage&q&f=false. Google-Books-ID : xCQlDwAAQBAJ.

[64] Lawrence J. Hettinger, Alex Kirlik, Yang Miang Goh, and Peter Buckle. Modelling and simulation of complex sociotechnical systems : envisioning and analysing work environments. *Ergon.*, 58(4) :600–614, 2015. ISSN 0014-0139, 1366-5847. doi : 10.1080/00140139.2015.1008586. URL http://www.tandfonline.com/doi/full/10.1080/00140139.2015.1008586.

[65] Brian Hopkins and John Gordon Skellam. A new method for determining the type of distribution of plant individuals. *Annals of Botany*, 18(2) :213–227, 1954.

[66] Naser Hossein Motlagh, Mahsa Mohammadrezaei, Julian Hunt, and Behnam Zakeri. Internet of Things (IoT) and the Energy Sector. *Energ.*, 13(2) :494, 2020. ISSN 1996-1073. doi : 10.3390/en13020494. URL https://www.mdpi.com/1996-1073/13/2/494.

[67] Nasser Hosseinzadeh, Ahmed Al Maashri, Naser Tarhuni, Abdelsalam Elhaffar, and Amer Al-Hinai. A real-time monitoring platform for distributed energy resources in a microgrid—pilot study in oman. *Electron.*, 10(15) :1803, 2021.

[68] Chung-Chian Hsu and Wei-Hao Huang. Integrated dimensionality reduction technique for mixed-type data involving categorical values. *Applied Soft Computing*, 43 : 199–209, 2016.

[69] Alex Q. Huang, Mariesa L. Crow, Gerald Thomas Heydt, Jim P. Zheng, and Steiner J. Dale. The Future Renewable Electric Energy Delivery and Management (FREEDM) System : The Energy Internet. *Proc. the IEEE*, 99(1) :133–148, 2011. ISSN 1558-2256. doi : 10.1109/JPROC.2010.2081330. URL http://ieeexplore.ieee.org/document/5634051/. Conference Name : Proc. the IEEE.

[70] Zhexue Huang. Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)*, pages 21–34. Citeseer, 1997.

[71] Lynette Hunt and Murray Jorgensen. Clustering mixed data. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 1(4) :352–361, 2011.

[72] Bill Inmon. *Data Lake Architecture : Designing the Data Lake and Avoiding the Garbage Dump*. Technics Publications, 1er édition edition, 2016.

[73] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems : An Introduction*. Cambridge University Press, September 2010. ISBN 978-1-139-49259-1. Google-Books-ID : eygTJBd_U2cC.

[74] Neil F Johnson. *Simply complexity : a clear guide to complexity theory*. Oneworld, Oxford, 2009. ISBN 978-1-85168-630-8. OCLC : ocn647072479.

[75] Ersan Kabalci and Yasin Kabalci. *From Smart Grid to Internet of Energy*. Academic Press, 2019. ISBN 978-0-12-819711-0. Google-Books-ID : ZzamDwAAQBAJ.

[76] Yasin Kabalci, Ersan Kabalci, Sanjeevikumar Padmanaban, Jens Bo Holm-Nielsen, and Frede Blaabjerg. Internet of Things Applications as Energy Internet in Smart Grids and Smart Environments. *Electron.*, 8(9) :972, 2019. ISSN 2079-9292. doi : 10.3390/electronics8090972. URL https://www.mdpi.com/2079-9292/8/9/972.

[77] Md Rezaul Karim, Oya Beyan, Achille Zappa, Ivan G Costa, Dietrich Rebholz-Schuhmann, Michael Cochez, and Stefan Decker. Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, 22(1) :393–415, 2021.

[78] Miroslava Kavgic, Anna Mavrogianni, Dejan Mumovic, Alex Summerfield, Zarko Stevanovic, and Maja Djurovic-Petrovic. A review of bottom-up building stock models for energy consumption in the residential sector. *Building and environment*, 45(7) : 1683–1697, 2010.

[79] Peter GW Keen. Decision support systems : a research perspective. In *Decision support systems : Issues and challenges : Proceedings of an international task force meeting*, pages 23–44, 1980.

[80] Adil Mehmood Khan, Young-Koo Lee, Sungyoung Y Lee, and Tae-Seong Kim. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE transactions on information technology in biomedicine*, 14(5) :1166–1172, 2010.

[81] Muhammad Waseem Khan and Jie Wang. Multi-agents based optimal energy scheduling technique for electric vehicles aggregator in microgrids. *Int. J. Electr. Power Energy Syst.*, 134 :107346, 2022.

[82] Ralph Kimball and Margy Ross. *The data warehouse toolkit : the complete guide to dimensional modeling*. John Wiley & Sons, 2011.

[83] Laura Klein, Jun-young Kwak, Geoffrey Kavulya, Farrokh Jazizadeh, Burcin Becerik-Gerber, Pradeep Varakantham, and Milind Tambe. Coordinating occupant behavior for building energy and comfort management using multi-agent systems. *Autom. Constr.*, 22 :525–536, 2012. ISSN 09265805. doi : 10.1016/j.autcon.2011.11.012. URL https://linkinghub.elsevier.com/retrieve/pii/S0926580511002196.

[84] Lukas Kranzl, Andreas Müller, Agne Toleikyte, Marcus Hummel, Jan Steinbach, Judit Kockat, Clemens Rohde, ISI Fraunhofer, Carine Sebi, Kimon Keramidas, and others. Policy pathways for reducing energy demand and carbon emissions of the EU building stock until 2030. *Energy Econ. Group, Vienna Univ. Technol.*, 2014.

[85] Julio Laborde. *Pretopology, a mathematical tool for structuring complex systems : methods, algorithms and applications*. PhD thesis, Paris Sciences et Lettres (ComUE), 2019.

[86] James Ladyman, James Lambert, and Karoline Wiesner. What is a complex system ? *Eur. J. Philos. Sci.*, 3(1) :33–67, 2013.

[87] James Ladyman, James Lambert, and Karoline Wiesner. What is a complex system ? *Eur. J. Philos. Sci.*, 3(1) :33–67, 2013. ISSN 1879-4912, 1879-4920. doi : 10.1007/s13194-012-0056-8. URL http://link.springer.com/10.1007/s13194-012-0056-8.

[88] Richard G Lawson and Peter C Jurs. New index for clustering tendency and its application to chemical problems. *Journal of chemical information and computer sciences*, 30(1) :36–41, 1990.

[89] Jean-Louis Le Moigne. *La modélisation des systèmes complexes*. Dunod, Paris, 1999. ISBN 978-2-10-004382-8. Réedition.

[90] Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. A Survey of DevOps Concepts and Challenges. *ACM Comput. Surv. (CSUR)*, 52(6) :1–35, 2020. ISSN 0360-0300, 1557-7341. doi : 10.1145/3359981. URL http://dl.acm.org/doi/10.1145/3359981.

[91] Loup-Noé Lévy, Jérémie Bosom, Guillaume Guerard, Soufian Ben Amor, Marc Bui, and Hai Tran. Application of pretopological hierarchical clustering for buildings portfolio. In *SMARTGREENS*, pages 228–235, 2021.

[92] Loup-Noé Lévy, Jérémie Bosom, Guillaume Guerard, Soufian Ben Amor, Marc Bui, and Hai Tran. Devops model appproach for monitoring smart energy systems. *Energies*, 15(15) :5516, 2022.

[93] Loup-Noé Lévy, Jérémie Bosom, Guillaume Guerard, Soufian Ben Amor, Marc Bui, and Hai Tran. Hierarchical clustering of complex energy systems using pretopology. In *Smart Cities, Green Technologies, and Intelligent Transport Systems : 10th International Conference, SMARTGREENS 2021, and 7th International Conference, VEHITS 2021, Virtual Event, April 28–30, 2021, Revised Selected Papers*, pages 87–106. Springer, 2022.

[94] Loup-Noé Lévy, Jérémie Bosom, Guillaume Guerard, Soufian Ben Amor, and Hai Tran. Modeling and recommendation system for improving the energy performance of buildings. In *Distributed Computing and Artificial Intelligence, Volume 2 : Special Sessions 18th International Conference 18*, pages 206–209. Springer, 2022.

[95] Loup-Noé Lévy, Guillaume Guérard, and Soufian Ben Amor. Recommendation system infrastructure for the energy efficiency of buildings. In *ECML PKDD 2022*, 2022.

[96] Loup-Noé Lévy, Guillaume Guérard, Soufian Ben Amor, and Djebali Sonia. Clustering mixed data comprising time series. In *Proceedings of the 12th International Symposium on Information and Communication Technology*, pages 110–117, 2023.

[97] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection : A data perspective. *ACM computing surveys (CSUR)*, 50(6) :1–45, 2017.

[98] Xiwang Li and Jin Wen. Review of building energy modeling for control and operation. *Renew. Sustain. Energy Rev.*, 37 :517–537, 2014.

[99] Xiwang Li and Jin Wen. Review of building energy modeling for control and operation. *Renewable and Sustainable Energy Reviews*, 37 :517–537, September 2014. ISSN 1364-0321. doi : $10.1016/\mathrm{j.rser.2014.05.056}$. URL https://www.sciencedirect.com/science/article/pii/S1364032114003815.

[100] Alexandre Lucas, Dimitrios Geneiatakis, Yannis Soupionis, Igor Nai-Fovino, and Evangelos Kotsakis. Blockchain technology applied to energy demand response service tracking and data sharing. *Energ.*, 14(7) :1881, 2021.

[101] Elizabeth Ann Maharaj. Cluster of time series. *Journal of Classification*, 17(2), 2000.

[102] Hariprasath Manoharan, Yuvaraja Teekaraman, Irina Kirpichnikova, Ramya Kuppusamy, Srete Nikolovski, and Hamid Reza Baghaee. Smart Grid Monitoring by Wireless Sensors Using Binary Logistic Regression. *Energ.*, 13(15) :3974, August 2020. ISSN 1996-1073. doi : $10.3390/\mathrm{en13153974}$. URL https://www.mdpi.com/1996-1073/13/15/3974.

[103] William B March, Parikshit Ram, and Alexander G Gray. Fast euclidean minimum spanning tree : algorithm, analysis, and applications. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 603–612, 2010.

[104] Daniel E. Maurino, James Reason, Neil Johnston, and Rob B. Lee. *Beyond aviation human factors : Safety in high technology systems*. Routledge, 2017.

[105] Ian C McDowell, Dinesh Manandhar, Christopher M Vockley, Amy K Schmid, Timothy E Reddy, and Barbara E Engelhardt. Clustering gene expression time series data using an infinite gaussian process mixture model. *PLoS computational biology*, 14 (1) :e1005896, 2018.

[106] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE, 2017.

[107] Leland McInnes, John Healy, and James Melville. Umap : Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv :1802.03426*, 2018.

[108] Damien McParland and Isobel Claire Gormley. Model based clustering for mixed data : clustmd. *Advances in Data Analysis and Classification*, 10(2) :155–169, 2016.

[109] Volodymyr Melnykov, Wei-Chen Chen, and Ranjan Maitra. Mixsim : An r package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51 :1–25, 2012.

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

[111] Boris Mirkin. *Clustering : a data recovery approach*. CRC Press, 2012.

[112] Tom M Mitchell. Machine learning, 1997.

[113] Dharmendra S Modha and W Scott Spangler. Feature weighting in k-means clustering. *Machine learning*, 52(3) :217–237, 2003.

[114] Pablo Montero-Manso, George Athanasopoulos, Rob J Hyndman, and Thiyanga S Talagala. Fforma : Feature-based forecast model averaging. *Int. J. Forecast.*, 36(1) : 86–92, 2020.

[115] M Morillo and JM Casado. Behavior of collective variables in complex nonlinear stochastic models of finite size. *arXiv preprint arXiv :1703.05110*, 2017.

[116] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[117] Nikolaos Nanas, Manolis Vavalis, and Elias Houstis. Personalised news and scientific literature aggregation. *Information Processing and Management*, 3(46) :268–283, 2010. ISSN 0306-4573. doi : 10.1016/j.ipm.2009.07.005. URL https://www.infona.pl//resource/bwmeta1.element.elsevier-5d74f550-fcca-3c46-9dfe-96fb9b4b9219.

[118] M. E. J. Newman. Complex Systems : A Survey. *Am. J. Phys.*, 79(8) :800–810, 2011. ISSN 0002-9505, 1943-2909. doi : 10.1119/1.3590372. URL http://arxiv.org/abs/1112.1440. arXiv : 1112.1440.

[119] Tim Oates, Laura Firoiu, and Paul R Cohen. Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 workshop on neural, symbolic and reinforcement learning methods for sequence learning*, volume 17, page 21. Citeseer, 1999.

[120] Julio M Ottino. Complex systems. *American Institute of Chemical Engineers. AIChE Journal*, 49(2) :292, 2003.

[121] B Ozarisoy and H Altan. Developing an evidence-based energy-policy framework to assess robust energy-performance evaluation and certification schemes in the south-eastern mediterranean countries. *Energy Sustain. Dev.*, 64 :65–102, 2021.

[122] Julio-Omar Palacio-Niño and Fernando Berzal. Evaluation metrics for unsupervised learning algorithms. *arXiv preprint arXiv :1905.05667*, 2019.

[123] Deepak Kumar Panda and Saptarshi Das. Smart grid architecture model for control, optimization and data analytics of future power networks with more renewable energy. *J. Clean.. Prod.*, 301 :126877, 2021.

[124] Tulasi K Paradarami. A hybrid recommender system using artificial neural networks. *Expert. Syst. With Appl.*, page 14, 2017.

[125] Murray G Patterson. What is energy efficiency? *Energy Policy*, 24(5) :377–390, May 1996. ISSN 03014215. doi : 10.1016/0301-4215(96)00017-1. URL https://linkinghub.elsevier.com/retrieve/pii/0301421596000171.

[126] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.

[127] G Philip and BS Ottaway. Mixed data cluster analysis : an illustration using cypriot hooked-tang weapons. *Archaeometry*, 25(2) :119–133, 1983.

[128] Joern Ploennigs, Bei Chen, Paulito Palmes, and Raymond Lloyd. e2-Diagnoser : A System for Monitoring, Forecasting and Diagnosing Energy Usage. In *2014 IEEE International Conference on Data Mining Workshop*, pages 1231–1234, Shenzhen, China, 2014. IEEE. ISBN 978-1-4799-4274-9 978-1-4799-4275-6. doi : 10.1109/ICDMW.2014.56. URL http://ieeexplore.ieee.org/document/7022741/.

[129] Sergio Potenciano Menci, Julien Le Baut, Javier Matanza Domingo, Gregorio López López, Rafael Cossent Arín, and Manuel Pio Silva. A novel methodology for the scalability analysis of ict systems for smart grids based on sgam : the integrid project approach. *Energ.*, 13(15) :3818, 2020.

[130] Daniel J Power. *Decision support systems : concepts and resources for managers*. Quorum Books, 2002.

[131] Alun Preece, Dan Harborne, Dave Braines, Richard Tomsett, and Supriyo Chakraborty. Stakeholders in explainable ai. *arXiv preprint arXiv :1810.00184*, 2018.

[132] James Reason. *Managing the Risks of Organizational Accidents*. Ashgate, Aldershot, Hants, England ; Brookfield, Vt., USA, 1 edition, 1997. ISBN 978-1-84014-105-4.

[133] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[134] Jeremy Rifkin. *The third industrial revolution : how lateral power is transforming energy, the economy, and the world*. Macmillan, 2011.

[135] Peter J Rousseeuw. Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20 :53–65, 1987.

[136] Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

[137] Hanna Schebesta. Risk regulation through liability allocation : Transnational product liability and the role of certification. *Air and Space Law*, 42(2), 2017.

[138] Skyler Seto, Wenyu Zhang, and Yichen Zhou. Multivariate time series classification using dynamic time warping template selection for human activity recognition. In *2015 IEEE symposium series on computational intelligence*, pages 1399–1406. IEEE, 2015.

[139] Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. The Intersection of Continuous Deployment and Architecting Process : Practitioners' Perspectives. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '16, pages 1–10, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 978-1-4503-4427-2. doi : 10.1145/2961111.2962587. URL https://doi.org/10.1145/2961111.2962587. tex.ids : shahin_intersection_.

[140] Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. Continuous integration, delivery and deployment : a systematic review on approaches, tools, challenges and practices. *IEEE Access*, 5 :3909–3943, 2017.

[141] Dragoslav D Siljak. *Decentralized control of complex systems*. Courier Corporation, 2011.

[142] Herbert A Simon. The architecture of complexity. *Proceedings of the American philosophical society*, 106(6) :467–482, 1962.

[143] Olivia Solon. The Rise of "Pseudo-AI" : How Tech Firms Quietly Use Humans to Do Bots' Work. *The Guardian*, 6, 2018.

[144] Ralph H Sprague Jr. A framework for the development of decision support systems. *MIS quarterly*, pages 1–26, 1980.

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

[146] Khaizaran Abdulhussein Al Sumarmad, Nasri Sulaiman, Noor Izzri Abdul Wahab, and Hashim Hizam. Microgrid energy management system based on fuzzy logic and monitoring platform for data analysis. *Energ.*, 15(11) :4125, 2022.

[147] Ihab Taleb, Guillaume Guerard, Frédéric Fauberteau, and Nga Nguyen. A flexible deep learning method for energy forecasting. *Energ.*, 15(11) :3926, 2022.

[148] Ihab Taleb, Guillaume Guerard, Frédéric Fauberteau, and Nga Nguyen. A flexible deep learning method for energy forecasting. *Energ.*, 15(11), 2022. ISSN 1996-1073. doi : 10.3390/en15113926. URL https://www.mdpi.com/1996-1073/15/11/3926.

[149] Unknown. What is a Data Mart | IBM, 2024. URL https://www.ibm.com/topics/data-mart.

[150] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *Computational Intelligence and Bioinspired Systems : 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005. Proceedings 8*, pages 758–770. Springer, 2005.

[151] Ben Wagner. Liable, but Not in Control? Ensuring Meaningful Human Agency in Automated Decision-Making Systems : Human Agency in Decision-Making Systems. *Polic. Internet*, 11(1) :104–122, 2019. ISSN 19442866. doi : 10.1002/poi3.198. URL https://onlinelibrary.wiley.com/doi/10.1002/poi3.198.

[152] K. Wang, X. Hu, H. Li, P. Li, D. Zeng, and S. Guo. A Survey on Energy Internet Communications for Sustain. *IEEE Trans. on Sustain. Comput.*, 2(3) :231–254, 2017. ISSN 2377-3782. doi : 10.1109/TSUSC.2017.2707122. URL https://ieeexplore.ieee.org/document/7932928.

[153] Liang Wang, Uyen TV Nguyen, James C Bezdek, Christopher A Leckie, and Kotagiri Ramamohanarao. ivat and avat : enhanced visual analysis for cluster tendency assessment. In *Advances in Knowledge Discovery and Data Mining : 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part I 14*, pages 16–27. Springer, 2010.

[154] Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work : An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *J. Mach. Learn. Res.*, 22(201) : 1–73, 2021.

[155] Matthijs J. Warrens and Hanneke van der Hoef. Understanding the Adjusted Rand Index and Other Partition Comparison Indices Based on Counting Object Pairs. *Journal of Classification*, July 2022. ISSN 0176-4268, 1432-1343. doi : 10.1007/s00357-022-09413-z. URL https://link.springer.com/10.1007/s00357-022-09413-z.

[156] Meng-Lun Wu, Chia-Hui Chang, and Rui-Zhe Liu. Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. *Expert systems with applications*, 41(6) :2754–2761, 2014. Publisher : Elsevier.

[157] Da Yan, William O'Brien, Tianzhen Hong, Xiaohang Feng, H. Burak Gunay, Farhang Tahmasebi, and Ardeshir Mahdavi. Occupant behavior modeling for building performance simulation : Current state and future challenges. *Energy Build.s*, 107 : 264–278, November 2015. ISSN 0378-7788. doi : 10.1016/j.enbuild.2015.08.032. URL http://www.sciencedirect.com/science/article/pii/S0378778815302164.

[158] Weixin Yao, Yan Wei, and Chun Yu. Robust mixture regression using the t-distribution. *Comput. Stat. Data Anal.*, 71 :116–127, 2014.

[159] Lamia Yessad and Aissa Labiod. Comparative study of data warehouses modeling approaches : Inmon, Kimball and Data Vault. In *2016 International Conference on System Reliability and Sci. (ICSRS)*, pages 95–99, Paris, France, 2016. IEEE. ISBN 978-1-5090-3277-8 978-1-5090-3278-5. doi : 10.1109/ICSRS.2016.7815845. URL http://ieeexplore.ieee.org/document/7815845/.

[160] Hiroshi Yoshino, Tianzhen Hong, and Natasa Nord. IEA EBC annex 53 : Total energy use in buildings—Analysis and evaluation methods. *Energy Build.s*, 152 : 124–136, 2017. ISSN 0378-7788. doi : 10.1016/j.enbuild.2017.07.038. URL http://www.sciencedirect.com/science/article/pii/S0378778817318716.

[161] Shi You, Lin Jin, Junjie Hu, Yi Zong, and Henrik W. Bindner. The Danish Perspective of Energy Internet : From Service-oriented Flexibility Trading to Integrated Design, Planning and Operation of Multiple Cross-sectoral Energy Systems. *Zhongguo*

Dianji Gongcheng Xuebao, 35(14) :3470–3481, 2015. ISSN 0258-8013. doi : 10.13334/j.0258-8013.pcsee.2015.14.001. URL https://orbit.dtu.dk/en/publications/the-danish-perspective-of-energy-internet-from-service-oriented-f.

[162] Kaile Zhou, Shanlin Yang, and Zhen Shao. Energy Internet : The business perspective. *Appl. Energy*, 178 :212–222, 2016. ISSN 0306-2619. doi : 10.1016/j.apenergy.2016.06.052. URL http://www.sciencedirect.com/science/article/pii/S0306261916308273.

Acronymes

AMI **A**djusted **M**utual **I**nformation . xiv, 98, *Glossary :* adjusted mutual information

API **A**pplication **P**rogramming **I**nterface . *Glossary :* Application Programming Interface

ARI **A**djusted **R**and **I**ndex . xiv, 100, 129, *Glossary :* Adjusted Rand Index

CH **C**alinski-**H**arabasz. 79, 83, 84, 97, 109–111, *Glossary :* Explainable Artificial Intelligence

CI/CD **C**ontinuous **I**ntegration/**C**ontinuous **D**eployment . 36, 125, *Glossary :* Continuous Integration/Continuous Deployment

CS **C**omplex **S**ystem . 3–6, 8, 17, *Glossary :* Complex System

CSTS **C**omplex **S**ocio-**T**echnical **S**ystem . 2, 3, 6, 7, 16–18, 23, 24, 27, 30, 44, *Glossary :* Complex Socio-Technical System

DB **D**avies-**B**ouldin. 79, 83, 84, 97, 98, 109–111, 115, *Glossary :* Davies-Bouldin

DevOps **Dev**elopment and **Ops**erations . 19, 30, 31, 36, 39–42, 44, 45, 47, 124, 125, *Glossary :* DevOps

DNF **D**isjunctive **N**ormal **F**orm . xiii, xiv, 23, 25, 65, 67, 69, 72, 73, 80, 90, 110, 111, 113, *Glossary :* Disjunctive Normal Form

DR **D**imensionality **R**eduction . 49, 51, 52, 62, 77–79, 81–83, 88, 89, 109–111, 113, 115, 128, 129, 132, 157, *Glossary :* Dimensionality Reduction

DSS **D**ecision **S**upport **S**ystem . xiii, 2, 3, 13, 19–25, 27, 31–36, 40–42, 44–47, 49, 123, 127, 128, 133–135, *Glossary :* Decision Support System

DTW **D**ynamic **T**ime **W**arping . xiv, 78, 80, 90, 91, 110, 112, 130, *Glossary :* Dynamic Time Warping

FAMD **F**actorial **A**nalysis of **M**ixed **D**ata . xiv, 51–53, 55, 77, 78, 80, 83, 84, 95, 97, 98, 101–103, 110, 111, 113, *Glossary :* Factorial Analysis of Mixed Data

GSC **G**ower **S**ilhouette **C**oefficient. 80, 109–111, *Glossary :* Silhouette Coefficient

IoE **I**nternet **o**f **E**nergy . 8, 9, 11–13, 16, 24, *Glossary :* Internet of Energy

IoT **I**nternet **o**f **T**hings. *Glossary :* Internet of Things

IT **I**nformation **T**echnology. 13, 19

KDE **K**ernel **D**ensity **E**stimation. 58, 59

Kmeans-FAMD Kmeans on FAMD reduced dataset. 98, 109, 111

MIBES ***M**ulti-**I**nstitution **B**uilding **E**nergy **S**ystem* . *Glossary :* Multi-Institution Building Energy System

ML **M**achine **L**earning . 1, 7, 17, 32, 39, 40, 124, *Glossary :* Machine Learning

ML-Factory **M**achine **L**earning **Factory** (see chapter: 2.3.4). 32, 39, 40, 42, 44, 47, 125, 126, 134–136

MS **M**icro**S**ervices . *Glossary :* Microservices Architecture

MSA **M**icro**S**ervices **A**rchitecture . 19, 40, 124, *Glossary :* Microservices Architecture

ORC **O R C** . *Glossary :* ORC Language

Pretopo-FAMD PretopoMD on FAMD reduced dataset. xiv, 100, 101, 104, 109–111

Pretopo-Louvain PretopoMD on Louvain reduced dataset. 98, 104

Pretopo-PaCMAP PretopoMD on PaCMAP reduced dataset. xiv, 100, 104, 106–109, 111, 116, 117

Pretopo-UMAP PretopoMD on UMAP reduced dataset. xiv, 100, 104, 105, 109, 116

SC **S**ilhouette **C**oefficient. 79, 80, 83, 84, 109–111, 115, *Glossary :* Silhouette Coefficient

SG **S**mart **G**rid . 8, 11, 12, *Glossary :* Smart Grid

TTPEMP **T**rusted **T**hird **P**arty for **E**nergy **M**easurement and **P**erformances . xiii, 2, 3, 9, 11, 13, 15, 16, 18–21, 24, 25, 28, 29, 32, 39–41, 44, 45, 124, 135, 161, *Glossary :* Trusted Third Party for Energy Measurement and Performances

UML **U**nified **M**odeling **L**anguage . 127, 128, *Glossary :* Unified Modeling Language

XAI e**X**plainable **A**rtificial **I**intelligence . 3, 22–25, 31, 81–84, 131, 135, *Glossary :* Explainable Artificial Intelligence

Glossary

Adjusted Mutual Information Use ARI when the ground truth clustering has large equal sized clusters. Use AMI when the ground truth clustering is unbalanced and there exist small clusters [5] [155]. 153

Adjusted Rand Index The Adjusted Rand Index (ARI) is a measure used to evaluate the similarity between two data clusterings. It is an adjusted version of the Rand Index, modified to account for the chance grouping of elements. ARI is particularly useful in the context of clustering validation, as it provides a way to compare the agreement between two different clusterings of a dataset, independent of the number of clusters. ARI values range from -1 to 1, where 1 indicates perfect agreement between two clusterings, 0 indicates random labeling, and negative values suggest greater than random dissimilarity. [6] [155]. 153

Application Programming Interface An Application Programming Interface (API) is a set of rules, protocols, and tools for building software and applications. It specifies how software components should interact and provides a way for different software applications to communicate with each other. APIs are used to enable the integration of systems, allowing them to exchange data and functionalities easily and securely. They play a crucial role in modern software development, facilitating the creation of complex systems and services by providing modular components that can be reused and interconnected.. 153

Big Data Big Data refers to extremely large datasets that are beyond the capability of traditional data processing tools to capture, store, manage, and analyze effectively. Characterized by the three Vs: Volume (immense amount of data), Velocity (high speed of data in and out), and Variety (range of data types and sources), Big Data requires advanced techniques and technologies for proper handling and analysis. This concept is significant in various fields like business, science, and technology, where it is used to uncover hidden patterns, correlations, and insights through sophisticated analytics. The rise of Big Data has been facilitated by the proliferation of data-generating devices, the Internet of Things (IoT), and the advancement of storage and computational resources.. 1, 36, 39

Black Box Black box model is also known as purely data driven model. Statistical models are directly applied to capture the correlation between building energy consumption and operation data. This type of models needs on-site measurements over a certain period of time to train the models to be able to predict the building operation under different conditions. These black box models are also widely applied in existing studies to determine building control strategies to reduce energy consumption and energy cost. Black box models are easy to build and computationally efficient, however, such models often require long training

5. https://stats.stackexchange.com/questions/260487/adjusted-rand-index-vs-adjusted-mutual-inform
6. https://stats.stackexchange.com/questions/260487/adjusted-rand-index-vs-adjusted-mutual-inform

period and are bounded to building operating conditions that they are trained for which sometimes can cause huge forecasting error when training data does not cover all the forecasting range. [99]. 7, 8, 123, 159, 160

Chaos Theory Chaos theory is a branch of mathematics focusing on the behavior of dynamical systems that are highly sensitive to initial conditions, a phenomenon popularly referred to as the butterfly effect. This theory suggests that small changes in the initial conditions of a system can lead to vastly different outcomes, making long-term prediction of their behavior impossible in general. This is often due to these systems possessing non-linear dynamics.. 6, 10

ClustMD Model Based Clustering for Mixed Data (ClustMD) introduced by [108]; it uses a latent variable model (LVM). LVM's main idea is that the observed datapoints are correlated and form particular patterns because they are influenced by latent variables. The clustMD model is fitted using an Expectation-Maximization (EM) algorithm. EM is an iterative method used to find the maximum likelihood estimate of a latent variable. If categorical variables are present, a Monte Carlo approximation algorithm is used for the Expectation step. See 3.3.4. 51, 60, 61, 83, 84, 93, 94, 104, 107, 109, 116

Complex Clustering In this thesis, Complex Clustering refers to the process of grouping data that includes a mix of different types, notably numerical, categorical, and time series data, among others. This approach is distinguished by its capacity to handle the intricacies and nuances of mixed data types within a single dataset, employing specialized algorithms and methodologies to discern patterns and relationships. The utilization of innovative techniques for Complex Clustering like pretopology are discussed in this thesis. 49, 135

Complex Socio-Technical System Complex Socio-Technical Systems (CSTS) refer to the integrative study of complex systems where social and technological elements are deeply intertwined. These systems are characterized by multiple interacting components, both human and machine, whose collective behavior exhibits properties not evident from the individual parts. Key features include emergent behavior, non-linearity, and selfk-organization. CSTS are prevalent in contexts like urban infrastructure, organizational networks, and energy systems, where human decisions and technological processes coexist and influence each other. The study of CSTS focuses on understanding these interactions to improve system design and management.. 6, 135, 153

Complex System There is no concise definition of a complex system on which all scientists agree. And arriving at a definition of complexity through necessary and sufficient conditions seems difficult if not impossible. In their article Ladyman et al. [87] identify that complex systems are associated to the following concepts, read their article for more details.

- Nonlinearity
- Feedback
- Spontaneous order
- Robustness and lack of central control
- Emergence
- Hierarchical organisation
- Numerosity

. 6, 23, 24, 135, 153

Continuous Integration/Continuous Deployment Continuous Integration (CI) and Continuous Deployment (CD) refer to the practices in software engineering where

developers regularly merge their code changes into a central repository, followed by automated builds and tests. The primary goal of CI is to provide rapid feedback so that if a defect is introduced, it can be identified and corrected as soon as possible. Continuous Deployment extends this to automatically deploy all code changes to a testing or production environment after the build stage. These practices are part of the DevOps approach and aim to increase software delivery speed, improve software quality, and enhance the responsiveness to changes.. 153

Convex K-Means Convex K-means also known as Modha–Splanger introduced by [113] is partitional Clustering method; the cluster centroids are forced to lie within the convex hull of the data points assigned to that cluster. To compute the distortion measures, they use the Euclidean distance between the numerical features vectors and the cosine distance between the categorical features ones. The method minimizes the average within-cluster distortion and the average between-cluster distortion. The hyperparameters of the algorithm are the number of clusters to determine k and the granularity of the exhaustive grid search. See 3.3.2. 51, 57, 58, 83, 84, 161

Curse of Dimensionality The curse of dimensionality is a phenomenon that arises in high-dimensional spaces, particularly in clustering and machine learning tasks, where the increase in dimensions leads to exponentially larger search spaces, making it difficult for algorithms to operate efficiently [20, 150]. Furthermore, distance metrics that work well in lower-dimensional spaces may not be as effective in higher-dimensional spaces, leading to poor performance in clustering tasks [145]. This problem is particularly relevant in the context of complex data containing time series, as time series often have high dimensionality due to the numerous time points involved [150]. A solution to break this curse is often DR.. 77, 81, 110, 128

Data Lake A Data Lake is a centralized repository that allows for the storage of structured, semi-structured, and unstructured data at any scale. It is designed to store vast amounts of data in its native, raw format. Data Lakes are used for storing big data and are an essential component of big data analytics frameworks. They enable the storing of data in various formats, including files, images, audio, and video, and facilitate flexible, large-scale data analysis and machine learning. A Data Lake provides a high level of scalability and can support various analytics and machine learning tools, allowing for comprehensive data processing and insights extraction.. 32, 135

Data Lineage Data Lineage refers to the lifecycle of data as it moves through various stages in the data ecosystem, including its origins, what happens to it, and where it moves over time. It encompasses the data's journey from its initial source to its final destination, including all the processes it undergoes, such as transformation, storage, and aggregation. Data lineage is crucial for data governance, quality, and compliance, offering visibility into the data's accuracy, reliability, and usage throughout an organization. It enables organizations to trace errors back to their source, understand the impact of changes in data, and ensure that data used for decision-making is trustworthy.. 135

Datamart A Datamart is a subset of an organization's data store or warehouse, often focused on a single subject area or business unit. Unlike a data warehouse, which stores data from multiple sources and covers an entire enterprise, a datamart is usually smaller in scope and size and is tailored to meet the specific needs of a particular group of users. Datamarts provide a more focused view of data and are optimized for data access and reporting. They are useful for departmental data

analysis and can improve data retrieval efficiency by reducing the volume of data to be processed.. 32, 36–39, 41, 47, 123, 135, 136

Decision Support System A Decision Support System (DSS) is a computer-based information system that supports business or organizational decision-making activities. DSSs serve the management, operations, and planning levels of an organization and help to make decisions, which may be rapidly changing and not easily specified in advance. They encompass a variety of data, including documents, raw data, and business models to assist in problem-solving and decision-making. DSSs can be either fully computerized, human-powered, or a combination of both.. 20, 135, 153

DenseClus DenseClus, developed by Amazon, integrates dimensionality reduction via UMAP with an accelerated version of the HDBSCAN algorithm, facilitating hierarchical density-based clustering. It is tailored for identifying high-density regions in datasets, thereby distinguishing between clusters and noise/outliers. It represent the data as a weighted graph called the *Mutual Reachability Graph*. In this graph, we consider the objects to be the vertices and an edge between any two objects to have a weight equal to the mutual reachability distance of the two objects. To model the cluster, all edges having weights greater than ϵ are removed and the remaining groups of connected ϵ-core objects constitute the clusters. The remaining unconnected objects are considered as "noise". This method is heavily dependent on hyperparameters but the we propose embedded ways to optimize them. The algorithm constructs clusters hierarchically, adjusting the density threshold to manage cluster connectivity. See 3.3.7. 51, 62, 83, 84, 96, 100, 104, 109

DevOps DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) with the goal of shortening the systems development life cycle and providing continuous delivery with high software quality. DevOps emphasizes collaboration, automation, and integration between developers and IT professionals, fundamentally changing how development, deployment, and operations teams work together. This approach enhances the speed and quality of software development and deployment, improves responsiveness to customer needs, and fosters a culture of continuous improvement.. 135, 153

Dimensionality Reduction Dimensionality Reduction (DR) is a process used in data analysis and processing to reduce the number of variables under consideration. It is achieved by obtaining a set of principal variables. Techniques like Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are commonly used for this purpose. DR is especially valuable in dealing with high-dimensional data, as it helps to simplify the dataset, reduce noise, and make the data more manageable for analysis. It is widely applied in fields such as machine learning, data mining, and pattern recognition. It can also be used to deal with mixed data [68] as in this thesis.. 135, 153

Disjunctive Normal Form Disjunctive Normal Form (DNF) is a standardization or normalization of a logical formula in propositional logic. It is a canonical form where the formula is expressed as an OR of ANDs. Each AND is referred to as a conjunct, consisting of literals (variables or their negations). DNF makes it easy to evaluate the truth value of a formula by breaking it down into simpler parts. It is commonly used in boolean algebra, computer science, and digital circuit design, as it provides a structured and simplified way of representing complex logical expressions.. 153

Dynamic Time Warping Dynamic Time Warping (DTW) is an algorithm used for measuring similarity between two temporal sequences which may vary in speed. It is commonly used in time series analysis, especially in speech recognition, to align

sequences of data points by stretching or compressing them in time. DTW calculates an optimal match between two given sequences with certain restrictions and rules. The algorithm allows for more flexibility than traditional methods like Euclidean distance, making it more effective for data where the timing varies, but the pattern or sequence is still meaningful.. 153

Elbow Method The Elbow Method is a heuristic used in cluster analysis to determine the optimal number of clusters. In this method, the sum of squared distances of samples to their nearest cluster center is plotted against the number of clusters. As the number of clusters increases, this sum decreases until it reaches an "elbow point," where the rate of decrease sharply changes. This point is considered to be an indicator of the appropriate number of clusters. The Elbow Method is widely used due to its simplicity and intuitive interpretation, though it is not always definitive and depends on the dataset's characteristics.. xiv, 65, 74, 94–96, 98, 100, 104, 107, 109

Energy Efficiency In general, energy efficiency refers to using less energy to produce the same amount of services or useful output. [125]. See energy efficiency index. 1, 7, 13, 18, 38, 40

Energy Efficiency Index Energy Efficiency Index (EEI), or sometimes known as Building Energy Index (BEI), is the most commonly used index as a Key Performance Indicator (KPI) to track and compare performance of energy consumption in buildings. The concept of this index is widely spread because it is beneficial to have a universal index for energy efficiency practices in buildings. Generally, EEI can be viewed as the ratio of the energy input to the factor related to the energy using component, as given in the following equation: EEI = Energy input / Factor related to the energy using component. The above definition for EEI is dependent on the parameters used as the energy input and the factor related to the energy using component. In general, the EEI for a building is tied to the size of the building as the energy used is considered to be based on the building floor area [64,65]. Some researchers define EEI as the ratio between the performance in terms of energy consumption or carbon dioxide emissions of an actual building to that of a reference building [18]. Regardless of the definition, the saving targets are always based on the lowest EEI for the building. [3]. 159

Explainable Artificial Intelligence Explainable Artificial Intelligence (XAI) refers to methods and techniques in the field of artificial intelligence (AI) that allow the results of the solution to be understood by humans. It contrasts with the Black Box nature of many AI models, where the decision-making process is not transparent or interpretable. XAI is crucial for validating and trusting AI in various applications, especially those involving critical decisions. It involves the integration of AI transparency, interpretability, and accountability into AI systems.. 154

Factorial Analysis of Mixed Data Factorial Analysis of Mixed Data (FAMD) is a statistical method introduced by [46]. It is designed for analyzing datasets that consist of both numerical and categorical variables. FAMD extends the principles of principal component analysis (PCA) to mixed data, allowing for the efficient reduction of dimensionality and the visualization of complex datasets in a lower-dimensional space. It is particularly useful in exploratory data analysis, enabling the identification of patterns and relationships within mixed datasets. For more details, see Section 3.2.1.. 153

Grey Box Grey Box Modeling is an approach that combines elements of both White Box (transparent) and Black Box (opaque) models. It integrates empirical data (as

in Black Box models) with physical, theoretical, or logical reasoning (as in White Box models) to provide a balanced understanding of the system being modeled. This approach is particularly useful in scenarios where complete information about a system is not available or where a full White Box model would be too complex. Grey box models are commonly employed in various fields, including system identification, control engineering, and machine learning, offering a pragmatic balance between simplicity and accuracy. A grey box modeling of a building can be found in [99].. 7, 8

Heating degree days Heating degree days (HDDs) are a measure of how cold the temperature was on a given day or during a period of days. For example, a day with a mean temperature of 40°F has 25 HDDs. Two such cold days in a row have 50 HDDs for the two-day period... 90

Internet of Energy The Internet of Energy [75] (IoE) refers to an advanced networking infrastructure that facilitates the integration and management of distributed energy resources. IoE encompasses the convergence of energy and information technology, enabling enhanced control, efficiency, and reliability in energy distribution and consumption. It supports the dynamic balancing of energy supply and demand, advanced energy analytics, and the integration of renewable energy sources. For further understanding.. 153

KAMILA KAy-means for MIxed LArge data (KAMILA) introduced by [48]; it is a combination of k-means clustering with a Gaussian-multinomial mixture model [71] to equitably balance the effects of numerical and categorical data without making the user specify the weights of both. The algorithm outputs the partition generated by multiple runs that maximizes the objective function. The hyperparameter of this algorithm is the number of runs to perform. see Section 3.3.3.. xiv, 51, 58, 61, 83, 84, 101, 104, 109, 111, 115

K-prototypes The K-prototypes algorithm is a partitional clustering method designed for mixed datasets, integrating the dissimilarity measure for numerical features from the K-Means algorithm and a matching dissimilarity measure for categorical features from the K-Modes algorithm. It creates prototypes as a combination of centroids for numerical features and modes for categorical features. The algorithm operates through initial prototype selection, initial allocation of data objects based on the closest prototype according to a new dissimilarity measure, and reallocation of data objects to enhance cluster quality. It addresses the limitations of the Hamming distance in capturing similarity between categorical features by employing a more nuanced similarity measure that considers the overall distribution and co-occurrence of feature values. Additionally, it introduces a cost function that assigns weights to numerical features based on their significance, determined by the proposed similarity measure. For more details, see Section 3.3.1.. 56, 83, 84, 109, 111, 115

Laplacian Eigenmaps Laplacian Eigenmaps introduced by [18]; it is a spectral embedding technique used for non-linear dimensionality reduction. This method has one hyperparameter. see 3.2.2. 51–54, 83, 84

Machine Learning Machine Learning is a branch of artificial intelligence (AI) focused on building systems that can learn from and make decisions based on data. It involves the development of algorithms that can analyze and interpret complex data, learn from it, and then apply the knowledge to make informed decisions or

predictions. Machine learning enables computers to improve their performance on a task with increasing experience or data over time, without being explicitly programmed for the specific task. This field intersects with statistics, computer science, and information theory and has widespread applications in areas like natural language processing, image recognition, and predictive analytics. For a detailed understanding, see [112], which provides an in-depth exploration of the fundamental concepts and methodologies of machine learning.. 154

Microservices Architecture Microservices Architecture (MSA) is an architectural style that structures an application as a collection of loosely coupled services, which implement business capabilities. Each microservice is a small, independent process and is deployed separately. MSA enables continuous deployment of large, complex applications and allows organizations to evolve their technology stack. This approach contrasts with traditional monolithic architecture, where different components of an application are tightly coupled and interdependent. Microservices are typically developed and deployed using containerization technologies, facilitating independence and scalability.. 154

MixtComp Mixed Dataset and Dataset with Missing Values (MixtComp) introduced by [21]. It is a statistical method for clustering mixed data, which combines the strengths of model-based clustering and Bayesian approaches. It can handle different types of data, including continuous, discrete, and mixed data, as well as missing data. The method models mixed data as a mixture of multivariate distributions, with each component representing a cluster. The clustering is performed through a Bayesian inference process, which estimates the number of clusters, cluster parameters, and latent variables. See 3.3.5. 51, 61, 83, 84, 94, 104, 107, 116

ModhaSpangler see Convex K-Means. 51, 94, 109

Multi-Institution Building Energy System The *Multi-Institution Building Energy System* (MIBES), introduced by Bosom et al. [24] is a hierarchical graph model used to model the TTPEMP [24, 23]. 153

ORC Language . 154

PaCMAP Pairwise Controlled Manifold Approximation and Projection (PaCMAP) is a dimensionality reduction technique that balances the preservation of local and global structures within high-dimensional data, making it suitable for visualization and exploratory analysis. It controls the attraction and repulsion between data points in the reduced space to accurately represent both nearby and distant relationships from the original high-dimensional space. This approach addresses common issues like crowding and loss of global structure, making PaCMAP applicable across various fields for data visualization, exploratory data analysis, and as a preprocessing step for further machine learning tasks.. xiv, 51, 53, 55, 78, 83, 84, 88, 97, 106, 107, 117, 133

Philip and Ottaway Philip and Ottaway propose to use Gower's similarity measure to obtain a similarity matrix, which is then used as input for a hierarchical clustering algorithm. Gower's similarity measure separates categorical and numerical features into two subsets, creating one categorical feature space and one numerical feature space. In the categorical feature space, the similarity between two datapoints is computed by a weighted average of similarities between all categorical features, which is calculated using Hamming distance. In the numerical feature space, the similarity between two datapoints is computed by the sum of the similarities between all numeric features.. 51, 95, 104, 109, 111, 115

PretopoMD PretopoMD leverages pretopology for clustering, organizing data into homogeneous groups while accommodating multicriteria analysis across diverse data types, including quantitative, qualitative, and time-series data. Pretopology utilizes the concept of pseudoclosure in a set to define hierarchical structures based on element similarities. A pretopological space is defined by a set of elements and a pseudoclosure function, defining how elements relate and group together. The clustering process involves creating weighted directed graphs, setting thresholds, and applying a boolean function in Disjunctive Normal Form (DNF) to determine element inclusion in group closures. This methodology facilitates the construction of a hierarchical clustering by identifying elementary subsets (seeds), expanding these through iterative pseudoclosure application, and mapping relationships via an adjacency matrix to establish a quasi-hierarchy. Its implementation in a Python library underscores its potential for broad applicability in data analysis and system categorization.. 81, 84, 89, 98, 100, 104, 107, 109–111, 113, 115, 116, 133

Smart Grid The Smart Grid (SG) is an intelligent, robust, and flexible energy grid which includes communication between each element of the grid. It integrates various technologies like advanced metering infrastructure, renewable energy sources, and energy-efficient resources. Smart Grids play a crucial role in modernizing the energy system, enhancing energy efficiency, and ensuring sustainable energy management. Refer to [6] for a detailed exploration of Smart Grid concepts and applications.. 154

Trusted Third Party for Energy Measurement and Performances This term refers to an independent and impartial entity responsible for measuring and assessing energy performances. Trusted Third Parties in the context of energy efficiency play a crucial role in verifying and ensuring the accuracy and reliability of energy data, providing an essential service in the validation of energy-saving measures and performance contracts. . 135, 154

UMAP Uniform Manifold Approximation and Projection (UMAP) is a dimensionality reduction technique that maps high-dimensional data to a lower-dimensional space, aiming to preserve the local and global structure of the original data. It operates under the assumption that the data lies on a uniform manifold and uses a fuzzy simplicial set approach to model the high-dimensional geometric structure. By optimizing a cost function that aligns this structure with a lower-dimensional representation, UMAP seeks to maintain the topological relationships among data points. This makes UMAP particularly useful for exploratory data analysis, visualization, and improving the performance of machine learning models by reducing the complexity of data. UMAP's flexibility, speed, and preservation of both local and global data structures distinguish it from other dimensionality reduction techniques, offering a powerful tool for uncovering insights in complex datasets.. xiv, 51, 53–55, 62, 77, 78, 83, 84, 88, 93, 97, 100, 104, 105, 107, 113, 116, 117

Unified Modeling Language The Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. UML provides a unified way to visualize the design of a system. It is used for specifying, constructing, visualizing, and documenting the artifacts of software systems. UML combines a variety of modeling techniques, including structural, behavioral, and interaction modeling. It is widely used in object-oriented analysis and design and facilitates the process of understanding and designing software systems, particularly in complex software projects.. 154

White Box White box or forward modeling approach uses detailed physics based equations to model building components, sub-systems and systems to predict whole buildings and their sub-systems beha- viors, such as their energy consumption and indoor comfort. Due to the detailed dynamic equations in white box models, they have the potential to capture the building dynamics well, but they are time consuming to develop and solve.[99] Even though these elaborate simulation tools are effective and accurate, they require detailed information and parameters of buildings, energy systems and outside weather conditions. These parameters, however, are always difficult to obtain, and even sometimes are not available. What's even more challenging in creating these white box models is that they normally require expert work, and the calculation is extremely time-consuming, which is the major barrier for white box building models to be used in on-line model based control and operation. [99]. 7, 8, 123, 159, 160