Intelligent VPP Cluster's Segmentation v1.0

AUTHORS:

N. Bezas, N. Foutakis, P. Tzallas, I. Moschos, D. Ioannidis [CERTH]

H2020-LC-SC3-2018-2019-2020 / H2020-LC-SC3-2020-EC-ES-SCC

EUROPEAN COMMISSION

Innovation and Networks Executive Agency

Grant agreement no. 957810

PROJECT CONTRACTUAL DETAILS

Project title	IntegrAted SolutioNs for the DecarbOnization and Smartification of Islands
Project acronym	IANOS
Grant agreement no.	957810
Project start date	01-10-2020
Project end date	30-09-2024
Duration	48 months
Project Coordinator	João Gonçalo Maciel (EDP) - JoaoGoncalo.Maciel@edp.com

DOCUMENT DETAILS

Deliverable no.	D4.5
Dissemination level	Public
Work package	WP4 IANOS Multi-Layer VPP Operational Framework
Task	T4.3 Intelligent VPP Cluster's Segmentation
Due date	31st March 2022
Actual submission date	31st March 2022
Lead beneficiary	CERTH

VERSION HISTORY

v	**Date**	**Beneficiary**	**Changes**
0.1	08/09/2021	CERTH	TOC
0.2	13/10/2021	CERTH	Literature Review and methodology
0.3	11/01/2021	CERTH	Methodology implementation
0.4	17/02/2022	CERTH	Consolidated version for review
0.5	17/03/2022	ETRA ID	Peer revision
0.6	18/03/2022	TNO	Peer revision

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

1.0	29/03/2022	CERTH	Final Version

Disclaimer

Table of contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

List of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

List of Figures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

Abbreviations

Abbreviations	Full Description
API	Application Programming Interface
DBI	Davies Bouldin Indicator
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNOs	Distribution Network Operators
DR	Demand Response
DSO	Distribution System Operator
ESB	Enterpise Service Bus
FC	Feature Construction
FE	Feature Extraction
FS	Feature Selection
HDL	Hourly Daily Load
iVPP	inteligent Virtual Power Plant
ITI	Information Technologies Institute
LOF	Local Outlier Factor
PAA	Piecewise Aggregate Approximation
PCA	Principal Component Analysis
RF	Random Forest
SAX	Symbolic Aggregate Approximation
SOM	Self Organizing Maps
SSE	Sum of Squared Error
SVM	Support Vector Machine
XGB	Extreme Gradient Boosting

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

1 Introduction

1.1 Scope of the deliverable

This document is a technical deliverable that reports the aggregation and classification intelligence module that will be developed within IANOS iVPP platform. This is the first version of the deliverable related to Task 4.3 “Intelligent VPP Cluster’s Segmentation”. In this task, multiple techniques will be implemented in order to form different cluster groups from the available portfolio of assets based on certain strategies and objectives provided by the portfolio manager (i.e. Aggregator/Prosumers Retailer). The module will be integrated into the iVPP platform and it will be utilized as a preprocessing tool to extract important insights regarding the assets within IANOS and to assist in the development of useful tools for Demand Response (DR) applications. These modules will be developed and deployed into an architectural layer to address non-functional needs such as stability and scalability. This deliverable refers to the first part of the task where the preparatory work takes place, while in the final part of the task another deliverable (D4.6) will address the actual development of the solutions.

1.2 Relation to other deliverables

This deliverable is based on the use cases and requirements described in D2.1 “Report on Islands Requirements Engineering and Use Cases Definitions”. There is also an initial description of the functionality of the component detailed in this deliverable, as well as its interconnections and dependencies, in T2.5 “System Architecture”, which provides a description of the system’s architecture.

All of the tasks from WP4 “IANOS Multi-Layer VPP Operational Framework” are closely related to the current deliverable. In T4.1 “Cyber-Secure data monitoring and VPP Governance”, the Enterprise Service Bus (ESB) is described, which is responsible for providing the necessary inputs to the Aggregation and intelligent Segmentation module. Furthermore, the component will provide the Forecasting Engine related to T4.2 “iVPP Forecasting Engine” with inputs and more specifically the residential load forecasting sub-module.

1.3 Structure of the deliverable

The deliverable is structured as follows:

Section 2 presents the literature review on the topic of clustering techniques, as well as, the clustering algorithms and evaluation metrics that are used.

Section 3 includes a description of the output format as well as an overview of the IANOS Aggregation and intelligent Segmentation module within the iVPP.

Section 4 presents in detail the pre-processing and feature engineering techniques applied prior to the deployment of the clustering techniques.

Section 5 discusses clustering in terms of providing valuable insights to the relevant Manager/Aggregator/Retailer operator in relation to demand side management, as well as doing a full research of the application of clustering for load forecasting.

The conclusions and next steps are discussed **in Section 6.**

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement Nº 957810

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

2 Literature review

Typical load profiles are important for evaluating schedulable load capacity, developing price-based or incentive-based DR programs and contributing to the operation planning of the smart grid. In addition, with the rapid development of smart grid technologies, residential and commercial loads have great potential for participating in demand response (DR) programs [1], [2], [3]. Thus, the need for the energy suppliers/aggregators and distribution network operators (DNOs) to better understand how to extract and exploit important insights of the load profiles retrieved from the energy consumers/prosumers makes the deployment of clustering and classification techniques important [4], [5], [6].

Many studies have focused on the application of cluster analysis, which is the process of classifying unlabeled data into groups of similar entities. Clustering residential and commercial loads based on attributes derived from smart meters contribute to the identification of different energy behavioral patterns. By using the correct attributes, the energy retailers/aggregators and the DNOs can use clustering to identify the most suitable groups of energy assets for demand reduction solutions and hence, to reduce the overall network demand and maintain grid stability. An indicative example of such use is the identification of customers with high but regular demand in the evening period that could be ideal candidates for peak demand reduction. Another case would be the mining of socio-demographic characteristics through the load profiles, such as the type of household (residential, commercial, etc). However, besides those two, clustering can be used in cases of large volumes of data to reduce complexity and consequently the computation cost (use only the representative of the clusters as inputs instead of the entire portfolio of energy assets).

Prior to the deployment, aggregation and pre-processing techniques are of high importance for efficient clustering. Both aggregation and pre-processing techniques are utilized to extract the necessary attributes from load time series and deliver them as input features in the appropriate format. As energy clustering involves mainly the features extracted from the load consumption time series, time series aggregation is considered one of the most adequate techniques [7]. Temporal aggregation allows for significantly reduced operational cost by aggregating the raw data into user-defined time windows. One of the most obvious options of temporal aggregation is to estimate the average values of different periods, providing them as input features to the clustering algorithms in order to identify different energy behaviors [8]. Another important aggregation technique that could be implemented for dimensional reduction is using the symbolic aggregate approximation (SAX) algorithm on the load profiles [9]. The aim of SAX is to represent the load profiles into a symbolic string by transforming the load data into a piecewise aggregate approximation (PAA) and afterwards a string is used to symbolize the PAA resulting in the reduction of the storage space.

In terms of pre-processing techniques, data normalization is performed to ensure that the distance measures assign equal weight to each value. The most widely used normalization techniques in clustering applications are: statistical normalization and scaling normalization. Data cleaning is another important process involved in pre-processing techniques. Data quality is crucial for effective clustering, since missing values, outliers and discontinuities within load consumption data is a frequent phenomenon that is facilitated by applying interpolation and outlier detection techniques.

Choosing correct attributes is one of the most important aspects, as it was mentioned above, clustering and feature selection is responsible for this. The selected features are subject to the objective defined by the respective operator. Therefore, in most cases, the selected input

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

features are based on intuition and domain knowledge. The number of the features should be defined optimally in order to ensure that the data distribution is dense, to reduce the computational costs and to ensure that the results can be easily interpreted. After the feature selection, clustering techniques are employed to evaluate and test the hypothesis defined by the objective function. Clustering algorithms fall into the following categories: centroid-based clustering, density-based clustering, distribution-based clustering and hierarchical clustering [10], [11]. Most of the studies on clustering analysis focus on the K-means [12], [13], [14] algorithm, which belongs to the centroid-based clustering algorithms, due to its robustness and high efficiency performance. In [15] K-means is performed in order to identify and classify into the same groups regional industries having similar energy consumption. [16] conducts a comparison assessment between different clustering methods regarding electricity consumption pattern recognition. It was found that the K-means algorithm is more useful for segmenting customers in contrast to the other clustering algorithms based on clustering validity indicators. [17] utilizes the raw data of electricity consumption of residential buildings to extract a representative load profile for every home for each season, applying K-means. Another reference [18] examines and compares three different clustering techniques based on their pattern of electricity use across the day: K-medoid, K-means and Self Organizing Maps (SOM), where SOM proved to be the most suitable option. More generally, though, the choice of the clustering algorithms depends on the application side as well. Density-based clustering is used to identify noise data indicating them as outliers. Thus, the algorithms that fall to this category are considered for outlier detection. In [19] Density-Based Spatial Clustering of Application with Noise (DBSCAN) is used to evaluate the consistency in the individual daily load profiles of the residential customers as it plays an important role for an efficient load forecasting. Therefore, customers having inconsistent behavior in terms of load consumption will present poor load forecasted results.

2.1 Clustering algorithms description

A wide variety of techniques and methods can be found in the literature, where an assessment of the advantages and strengths of each clustering algorithm is performed and documented in detail. The clustering algorithms that are mostly used and implemented in smart grids are the following:

k-means: k-means [20] is one of the simplest and most popular clustering unsupervised machine learning algorithms. The word “means” refers to the averaging of data and it belongs to the centroid-based clustering algorithms. The aim of this category of algorithms is to split and allocate the data into different groups or clusters. In general, the centroid-based methods detect and identify a number of partitions that satisfy a given constraint or function, where in the case of k-means is the minimization of the sum of squared error (SSE) of the set of points that have been assigned to each group respectively. SSE is the sum of squared differences between each observation and the mean value of the group, while indicating the variation within a cluster as well. SSE’s formula is described in equation (1).

$$SSE = \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad (1)$$

The first step of k-means includes the random selection of K centroids, where K corresponds to the number of clusters that are initially established. The centroids refer to the data points representing the center of a cluster. Once the centroids are assigned, the distance of each element of the data set is calculated from all the centroids and are allocated to the closest

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement Nº 957810

centroid. Once all elements have been associated to a centroid, a preliminary partition of the data set is done. The reference metric for the estimation of the distance between the data points and the centroids is the sum of the squared Euclidean distances. After the initial allocation, the next step involves the re-calculation of the centroids and the data points and the formulation of new clusters. This process is performed in an iterative manner until the centroid positions reach convergence and no variation is calculated resulting in clusters with minimum SSE [21]. Thus, the final form of the SSE is described in equation (2)

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{k} ||d_i^{(j)} - c_j||^2 \quad (2) \ ,$$

where:

- $d_i^{(j)}$ is the data point of the i-th element of j-th cluster
- c_j is the j-th centroid, and
- $||d_i^{(j)} - c_j||^2$ is the squared distance between the data point and the respective centroid

Essentially, k-means is widely used, as it is simple to implement, fast in the execution, the computation burden is less than other techniques and it is able to detect and identify well-defined groups of clusters.

Density based spatial clustering with noise (DBSCAN): DBSCAN [22] belongs to the density-based data clustering algorithms. The goal of DBSCAN is to "reveal" high-density regions of data points identifying them as clusters. The algorithm groups together points that are close to each other (points with many neighbors); while considering as outliers points that appear alone in low-density regions. Thus, a data point belongs to a cluster, if it is "close" to many points from that cluster. However, since adopting the definition "close" is a relative concept, the data points are classified as "core" points, "border" points and outliers. A point is considered a "core" point if the number of its neighboring points within a radius (eps) that is defined by the user is greater than a predefined number of points (minPoints). If the neighboring points are less than the minimum points that are chosen at the start, then they are considered "border" points. Finally, if no points exist the algorithm marks the data point as an outlier. In contrast with K-means, in DBSCAN there is no need to predefine the number of the clusters, but two other parameters have to be declared prior to the implementation as it was mentioned, namely the radius of the neighborhood (epsilon) and the minimum number of the data points (minPoints) .

- **eps:** the distance that specifies when two points are "neighbors" and can be considered a part of a cluster. If the distance between these two points is less than the epsilon value, these points are marked as neighbors.
- **minPoints:** the minimum number of points to form a dense region (cluster).

Unlike other clustering algorithms, DBSCAN is able to discover clusters of arbitrary shapes and noise. Moreover, it is applicable to spatial datasets and additionally there is no need to predefine the number of clusters. Due to the way it performs the identification of the clusters using the density, DBSCAN can be considered an ideal option for outlier detection applications.

Spectral Clustering: Spectral clustering is a clustering methodology that is based on graph connectivity theory. Given a dataset in the n-dimensional space and a matrix A that describes the similarity between the data points based on a distance metric, spectral clustering attempts

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

to partition the dataset into k clusters. The clustering is performed on the eigenvectors that correspond to the first k eigenvalues of the Laplacian matrix and is usually done with the K-Means algorithm [23].

The algorithm for Spectral Clustering is described below:

1. Define and compute the similarity matrix A of the dataset
2. Define and compute the Laplacian matrix L of the dataset
3. Compute the first k eigenvalues of the Laplacian (excluding the zero eigenvalue).
4. Create the matrix $[x_1, x_2, x_k]$ (n being the number of datapoints in the dataset).
5. Considering each row of the matrix as a point, perform K-Means clustering on the points.
6. According to the label that was assigned to row i of the matrix, assign the same label to the original datapoint of the dataset.

Spectral Clustering does not cluster the data directly, but rather performs a dimensionality reduction first. Different definitions for the similarity matrix A as well as the Laplacian L have been given.

In [23] the similarity matrix is defined as affinity matrix, as follows, given a set of points $s = \{s_1, \dots, s_n\}\ in\ R^l$:

$$A_{ij} = \begin{cases} e^{\left(-\frac{\left(s_i - s_j\right)^2}{2*\sigma^2}\right.} & ,\ if\ i! = j \\ 0 & ,\ if\ i = j \end{cases} \quad (3)$$

While, the graph Laplacian is defined as:

$$L = D^{-\frac{1}{2}} * A * D^{-\frac{1}{2}}\ D_{ii}\ (4),$$

where D_{ii} is the sum of A' s i-th row

In [24] the Laplacian matrix is defined as:

$$L = D - A\ (5)$$

Hierarchical Clustering: Hierarchical Clustering attempts to create a hierarchy of clusters [25]. Two major approaches exist: The bottom-up approach also referred to as agglomerative approach and the top-down approach also referred to as divisive. In agglomerative clustering, each data point is initially considered as its own cluster. Based on a linkage criterion and a metric used the clusters are merged, until only one cluster remains that contains all the elements of the dataset [26]. In divisive clustering the opposite direction is followed: all the elements of the dataset are initially in one cluster and then a hierarchical tree is formed until each data point belongs in one cluster [26]. By “cutting” through the tree at a certain height, the formed clusters are returned as output. The agglomerative clustering algorithm forms the clusters by following a certain criterion. Some of these criteria are listed below [27]:

- ‘ward’ minimizes the variance of the clusters being merged.
- ‘average’ uses the average of the distances of each observation of the two sets.
- ‘complete’ or ‘maximum’ linkage uses the maximum distances between all observations of the two sets.
- ‘single’ uses the minimum of the distances between all observations of the two sets.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement Nº 957810

Different distance measures can be utilized, such as Euclidean distance, Manhattan distance, and so on.

2.2 Evaluation of clustering performance based on clustering validity indexes

The evaluation analysis is used when assessing the performances of one or more clustering techniques and it is a measure of the consistency of the established clusters. The use of cluster validation indices helps to interpret the optimal number of clusters for a given algorithm and it is used to perform comparisons between different clustering techniques. Two different types of similarity metrics are involved in the evaluation process. The first similarity metric assesses the cohesion of data points within clusters (intra-cluster), and the second measures the distance between objects belonging to different clusters (inter-cluster). Furthermore, it can be used as a reference metric in order to understand and determine the proper number of clusters. Among the large variety of methods and indexes available in literature, the most relevant ones for energy and power applications are the following:

Elbow Method: The elbow method is one of the most common evaluation methods and it is adopted for general-purpose clusterization. This method is usually utilized in the k-means algorithm in order to define the optimal number of clusters. It is a graphic based method as the optimal number of the clusters corresponds to the point where the “elbow” appears. The plot refers to the value of the cost function produced by different values of clusters, where the cost function can be either the within-cluster sum of squared errors (inertia) or the average of the squared distances from the data points of the respective centers (distortion). The inertia and the distortion are estimated for each value of k in the given range. An example of the selection of the optimal number of clusters based on distortion is given in Figure 2.1:

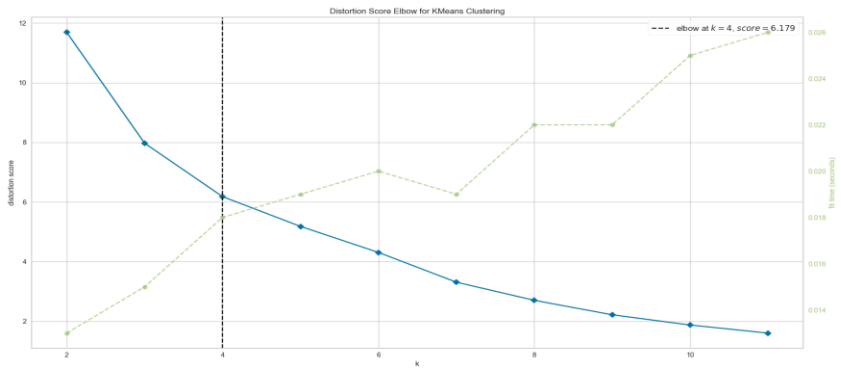

Figure 2.1: Elbow method for determining the optimal number of clusters

The dashed black line corresponds to the “elbow” and indicates the optimal number of clusters. In addition, the graph shows the training time of the clustering model as a dashed green line and as it becomes evident the time increases as the number of clusters increases.

Silhouette index: Silhouette analysis [28] is a metric indicating how well the clusters are separated. The silhouette plot displays a measure of how close each point in one cluster is to the points of a neighboring cluster (separation) or how similar a point is to its own cluster (cohesion). Thus, silhouette index provides a way to assess the clustering parameters in a visual way. The value of the metric ranges between [-1, 1]. A silhouette value of one data point i is calculated in equation (6):

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement Nº 957810

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i)-b(i)\}} \quad (6),$$

where $a(i) = \frac{1}{|C_i|-1} \sum_{jeCi} d(i,j)$ is the mean distance of data point i and all other data points within the same cluster, |Ci| is the number of points that belong to the cluster and $d(i,j)$ is the distance between the data point i and j of cluster |Ci|. Essentially, $a(i)$ is an indicator of the cohesion of the data point i to its own cluster, while $b(i)$ is a measure of the mean dissimilarity and is given by the following equation: $b(i) = min \frac{1}{|C_i|} \sum_{jeC_i} d(i,j)$ Id the smallest mean distance of the data point i to all points to its "closest" neighbor cluster, therefore the min operator is used. If the Silhouette score is close to 1 the data points have been separated efficiently, while those with a lower value can be considered less well grouped. An example of a visual representation of the silhouette metric for a given number of clusters is illustrated by the following Figure 2.2 and Figure 2.3

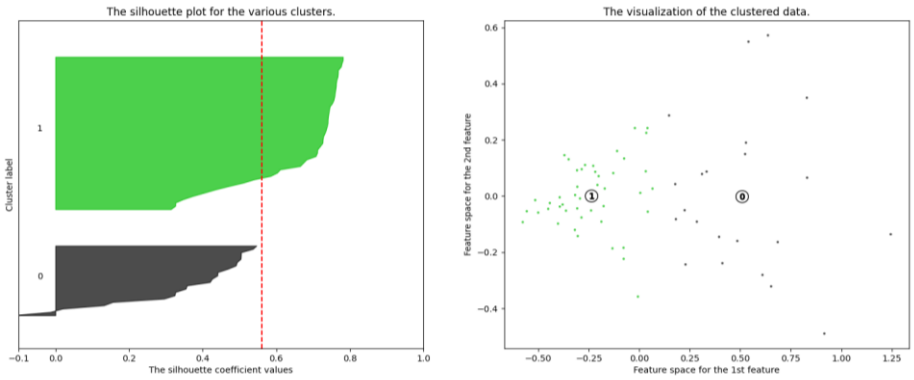

Figure 2.2: Silhouette analysis for k-means clustering on sample data with 2 number of clusters

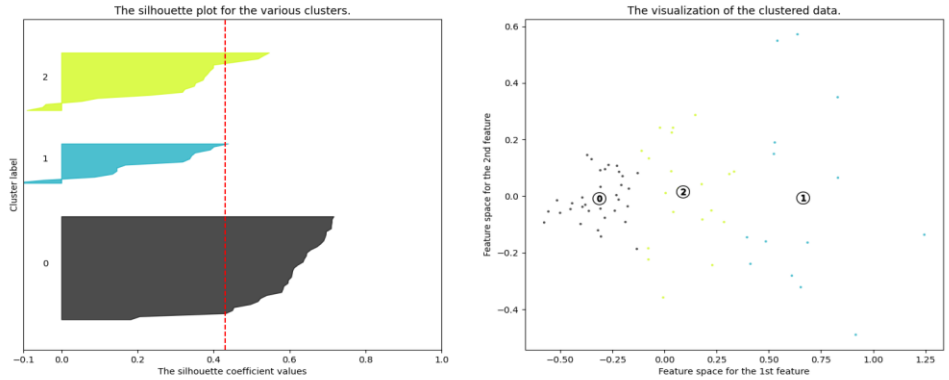

Figure 2.3: Silhouette analysis for K-means clustering on sample data with 3 number of clusters

In this example, the silhouette analysis is used to choose the optimal number of centroids. In the context of the example, the silhouette analysis is conducted only for two and three centroids respectively. The dashed red line in the left figure depicts the score of the silhouette analysis, while in the right figure a scatter plot of the established clusters is provided. The silhouette analysis is performing better in the case of two clusters, while choosing three clusters the dashed red line is shifted towards to zero value.

Davies-Bouldin Indicator (DBI): Davies-Bouldin index [29] is an average similarity metric of each cluster with its most similar cluster. The similarity is defined as the ratio between the distances of data points to its own cluster center to the 'neighbor's within-distances respectively. Lower values of DBI index imply a more efficient performance of the clustering,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

with zero being the minimum score that DBI receives. The DBI index is described in equation (7):

$$DB = \frac{1}{n}\sum_{i=1}^{n} max\, D_{ij}\ (7),$$

Where N is the number of clusters and $D_{ij} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{ij}}$, where $\bar{d}_i$ is the average distance between the data points belonging to cluster i and its centroid and similar for $\bar{d}_j$. d_{ij} refers to the Euclidean distance between the centroids of the two clusters. According to the equation if the clusters have a large distortion then the ratio is increased, meaning that the clustering separation is not successful. Additionally, if the distance between neighboring centers is large the ratio is decreased. Thus, it is evident that the DBI index is taking into consideration both the internal distortion of the clusters as well as the distance between the neighboring clusters. An example of the evaluation of the optimal number of centroids is illustrated below. The lowest value is occurred when the number of clusters is set to two and is chosen as the optimal number of clusters.

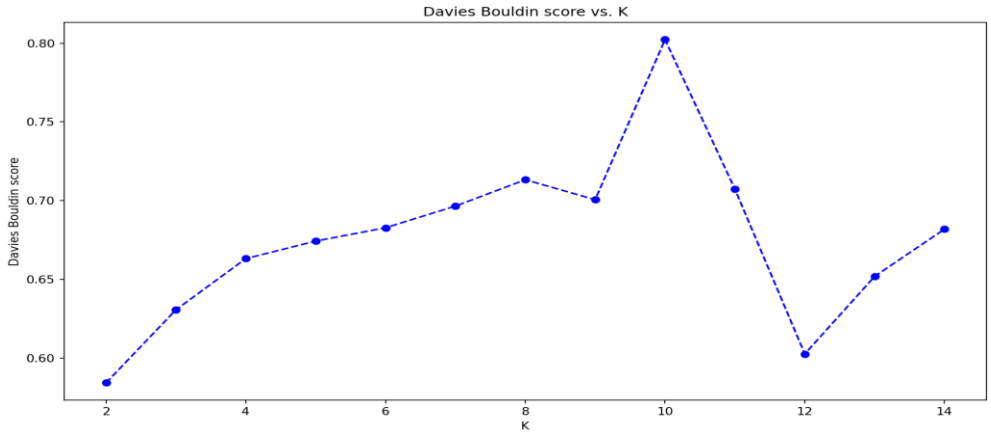

Figure 2.4: Davies Bouldin score for determining the optimal number of clusters

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

3 IANOS Aggregation and intelligent Segmentation module

3.1 IANOS Aggregation and intelligent Segmentation logical and development view

The logical and development views of the Aggregation and intelligent Segmentation engine deployed within the iVPP platform are depicted in the following figures.

Logical View

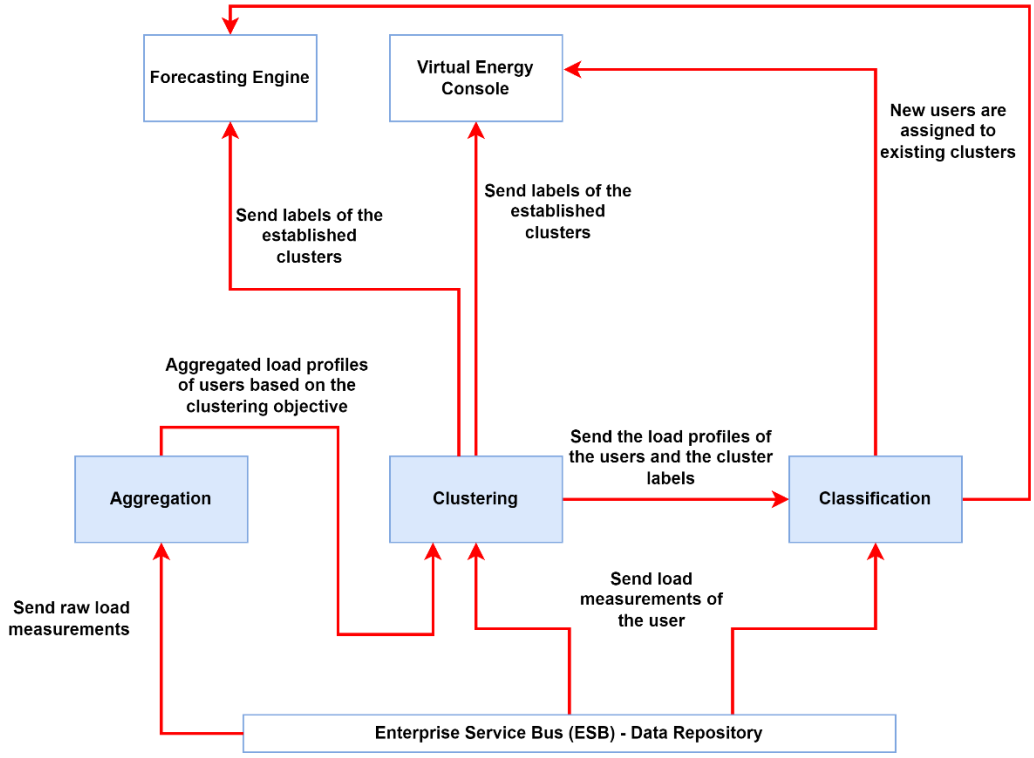

Figure 3.1: Interconnections among T4.3 components (blue boxes) and other iVPPs components (white boxes) as represented in IANOS Architecture

Development View

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

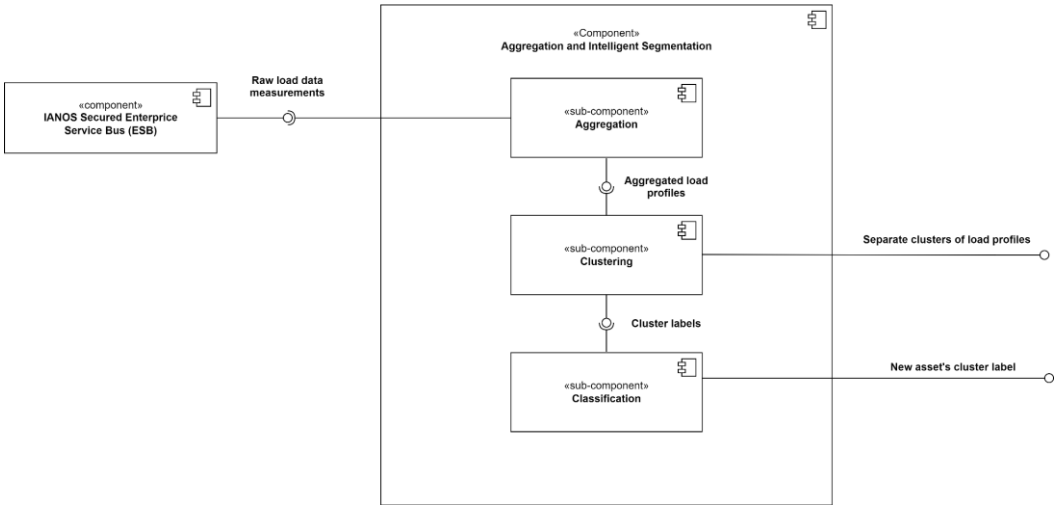

Figure 3.2: Development view of the Aggregation and intelligent Segmentation module

The Aggregation and intelligent Segmentation component can be considered one of the core components of IANOS and serves as a pre-processing component that is able to extract and provide meaningful insights of the residential and commercial load measurements to other components integrated within IANOS iVPP framework. It will receive the load profiles of residential and commercial loads via the Enterprise Service Bus (ESB). The Aggregation and intelligent Segmentation engine consists of **three different components**, each of which performs a different functionality. More precisely, the **Aggregation module** implements pre-processing techniques on the raw measurements, including data normalization and data cleaning, in order to formulate the appropriate inputs for the Clustering module depending on the respective objective defined by the iVPP framework. The **Clustering module** is the main component of the engine, aiming to provide several well-defined and separate clusters of load profiles based on different features. The results are evaluated according to specific indexes and are interpreted intuitively using visual representations of the data distribution of the clusters. Finally, the **Classification module** employs algorithms that are trained with the clustering results in order to assign new load profiles to the existing clusters.

3.2 Data model

The input and output of the IANOS Aggregation and intelligent Segmentation component follows a specified data model, and it will be provided in the form of a .json file including information on the number of generated clusters, the user's id, and the cluster label in which it was assigned. Table 3.1 describes the type and the meaning of each input variable, whereas

Table 3.2 describes the output variables. The input and output of the segmentation component are visually represented in Figure 3.3 and Figure 3.4 respectively.

Table 3.1: Description of input variables

Friendly name	Measurement name	Measurement unit	Description
Power consumption	active_power	W	Measurement name of the respective historical load measurements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

Table 3.2: Description of output variables

Friendly name	Measurement name	Measurement unit	Description
Cluster label	label	int	The label of the cluster in which the user has been assigned


```
{
  "asset_id": "bdksncjdsijcd",
  "measurements": [
    {
      "measurement_name": "activePower",
      "measurement_unit": "kW",
      "measurement_type": "float",
      "properties": [
        {
          "value": 50.53363,
          "timestamp": "2019-11-08T12:45:40.035Z",
          "quality": 1
        },
        {
          "value": 60.53363,
          "timestamp": "2019-11-08T12:50:40.035Z",
          "quality": 1
        }
      ]
    },
    {
      "measurement_name": "reactivePower",
      "measurement_unit": "kVar",
      "measurement_type": "float",
      "properties": [
        {
          "value": 15.53363,
          "timestamp": "2019-11-08T12:45:40.035Z",
          "quality": 1
        },
        {
          "value": 35.53363,
          "timestamp": "2019-11-08T12:50:40.035Z",
          "quality": 1
        }
      ]
    }
  ]
}
```


Figure 3.3: Example of the input of the segmentation component

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

```
{
  "asset_id": "bdksncjdsijcd",
  "measurements": [
    {
      "measurement_name": "label",
      "measurement_unit": "",
      "measurement_type": "int",
      "properties": [
        {
          "value": 0,
          "timestamp": ""
        }
      ]
    }
  ]
}
```


Figure 3.4: Example of the output of the segmentation component

3.3 Software pre requirements

The Aggregation and intelligent Classification component in order to be completely functional, it shall be useful to exploit the following technologies:

Python (https://www.python.org/): It is recommended to be the reference language as it is ideal for fast and easy prototyping and additionally contains suitable packages for data analysis and machine learning algorithms. Below are described some of the most important and useful packages in terms of data analysis and clustering:

- ***Numpy (http://www.numpy.org/):*** A fast and versatile package for convenient array operations, offering a comprehensive set of mathematical functions.

- ***Pandas (https://pandas.pydata.org/):*** One of the most famous and widely used "Data Analysis Library". Pandas is easy to use for handling datasets in several formats: csv, h5, parquet, etc. Pandas is ideal for time-series analysis, as it offers a variety of methods for reading, writing and pre-processing the raw time-series by extracting important information, and therefore, fits perfectly into the concept of clustering.

- ***Scikit-learn (https://scikit-learn.org):*** It offers various pre-built machine learning algorithms, including clustering and classification algorithms as well.

- ***Tsfresh (https://tsfresh.readthedocs.io/en/latest/index.html):*** It is a python package used to calculate a large number of time series characteristics, features, automatically. The explaining power and significance of the extracted attributes, whether it concerns regression or classification tasks, are evaluated by the methods provided by the package.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

4 Data-preprocessing

Clustering is a data mining process that is centered around the discovery of data groups without any prior knowledge of the target variables. Therefore, it is based on the quality and quantity of the data used for discovering the clusters of data, as well as the features that are selected based on the need of the application at hand. A high-level flow diagram presenting the entire processing pipeline is illustrated in the following figure:

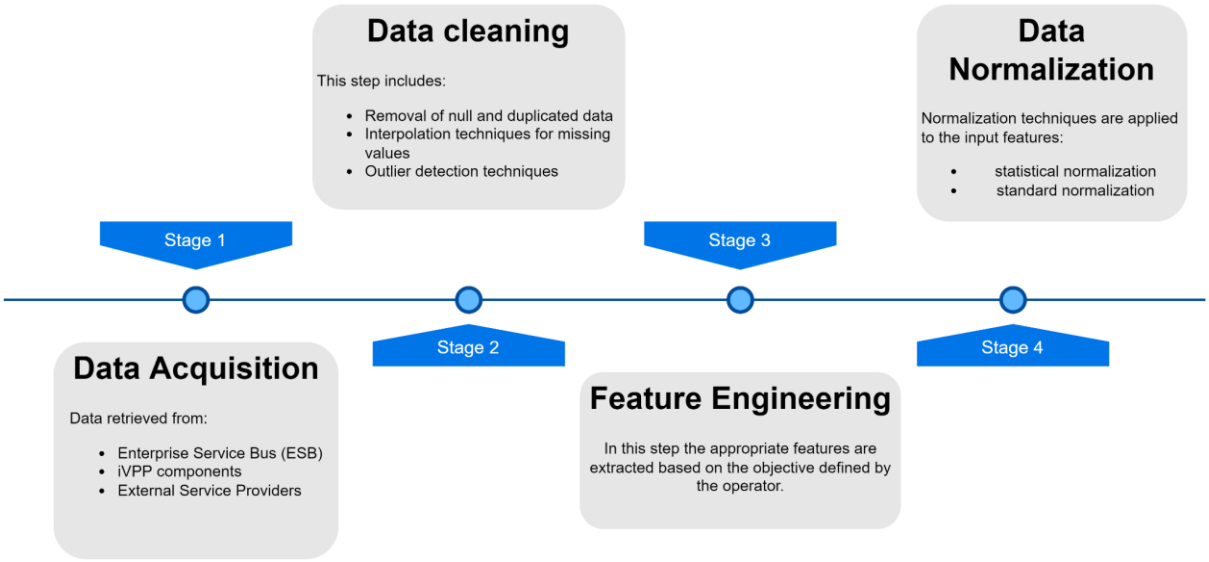

Figure 4.1: Data-preprocessing pipeline

The processing pipeline in IANOS Aggregation and intelligent Segmentation module consists of 4 steps. The first step includes the data collection through the iVPP enterprise service bus (ESB), which is the middleware for data transactions and data storage considering various energy services and assets. After that, the data cleaning module looks for any missing or erroneous data before the feature engineering module constructs the clustering model's input features. Finally, normalization techniques are implemented on top of the input features utilizing the data normalization module.

4.1 Data Acquisition

All the information required for the aggregation and clustering analysis of IANOS VPP platform are retrieved through the enterprise service bus (ESB). In general, the ESB will be the module where all the components integrated within the iVPP platform will communicate in order to collect data of all the energy assets that are distributed in different geographical locations. The ESB module will comprise and expose these methods to send and exchange the contextual data from the field components to the iVPP platform. Additional on-line connections with external end-points-databases and Operators platform will be established through Application Programming Interfaces (API's) in order to retrieve external information such as weather and price data. In addition, it is important to highlight the fact that all the cybersecurity issues regarding the transactions for the energy services through the ESB have been addressed in detail within Task 4.1 "Cyber-Secure Data Monitoring and VPP Governance".

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

4.2 Data Cleaning

The quality of data is essential for effective clustering, as most of the time, either missing values or outliers are existent within the raw measurements due to inconsistent operation of the power systems and the erroneous measurements recorded by the smart meters. In order for the data cleaning process to work, it must first be able to detect and identify the inaccurate measurements and manage them accordingly by implementing relative data cleaning techniques. Three are the most common categories of inaccurate measurements that are present with the raw measurements and are examined below:

1. Missing values: Lack of data is one of the most common problems in large real-world databases and data repositories. Some of the reasons may be transmission errors during the recording of the data, faulty smart meters and data that can be considered irrelevant and are not stored in the database. Several techniques are documented in the literature that could potentially be used [30]. Attributes that contain missing or null values can be either dropped from the dataset or filled with values that are created according to data cleaning techniques. As a rule of thumb is to split the raw load measurements into daily segments and exclude the days with less than 80% of the maximum samples due to lack of data. Afterwards, imputation techniques are applied to the remaining days in order to fill in the missing values. Fill-in approaches, such as replacing a missing attribute with a measure of the most attribute's central tendency (e.g. mean or median) or assigning the "closest" existing value either backwards or forward, are the most popular and straightforward methods. Another method of time-series data cleaning is to execute a statistical analysis on the data and replace the missing attribute with the mean of the values recorded for the same time slot of the previous days. Also, different interpolation methods could be used as well (e.g. linear or quadratic interpolation).

2. Duplicate values: Interruption when retrieving data from smart meters can result in duplicate data. Another factor that contributes to the creation of duplicate data is the end of daylight saving. Therefore, duplicate data have to be detected and treated as invalid data.

3. Outlier detection: Outliers are values that are not consistent with the rest of the dataset and may not exhibit logical cohesion within the context of the given problem. The problem with outliers is the need to identify the "true" outliers and the "false" outliers, with "true" being the outliers that are created due to: noise, system malfunction, user errors or tampering with the data, and "false" outliers being the ones that are caused by an irregular or extreme behavior and need to be identified and considered during the knowledge discovery process. Z-score and IQR methods are used to identify and remove the outliers within load consumption measurements. The z-score method transforms the data onto a distribution whose mean is defined as 0 and the standard deviation is 1. The aim of z-score is to scale the data on the same level and afterwards any data point that is far from zero is identified as outlier. The threshold of z-score usually varies from -3 up to 3, therefore any data points that are located outside the pre-defined boundaries are excluded as outliers. Interquartile range is another robust method for labeling outliers and is widely used. The detection of outliers is conducted based on the box plot, which uses quartiles to represent the shape of data. The quartiles are points that divide the data into four equal sized segments. The box plot indicates the 1st and 3rd quartiles, which are equal to the 25th and 75th percentiles,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

while the line inside the box denotes the median value. The range between the first and the third quartiles is called the interquartile range (IQR). The values that fell outside of either 1.5 times the IQR below the first quartile (minimum) or 1.5 times above the third quartile (maximum) are considered as outliers. Finally, the “whiskers” extend to the last data point that is not “outside” of the mentioned ranges. The different parts of the box plot are explained by the following figure. More specifically, the pink shaded area indicates the interquartile range, the yellow line corresponds to the median of the data points, and the “whiskers” are defining the boundaries, while the “green” data points are considered as outliers, as they are located outside from the pre-defined boundaries.

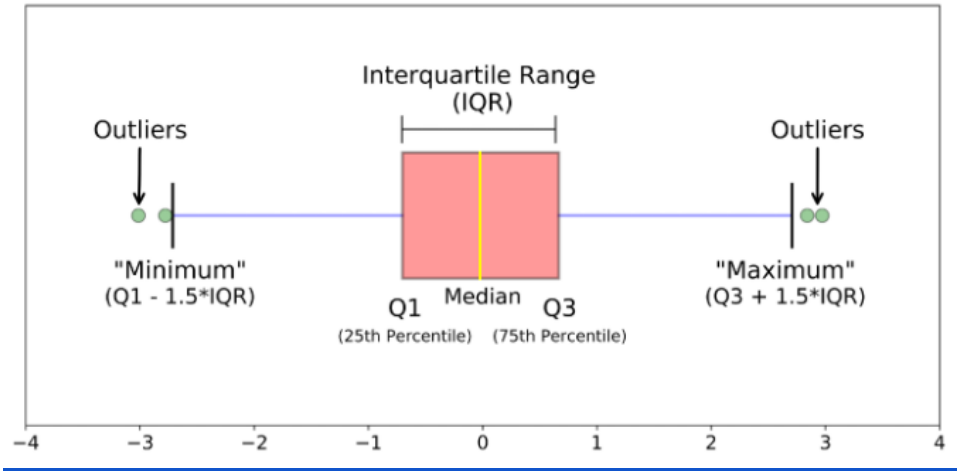

Figure 4.2: Different parts of a boxplot

4.3 Data Normalization

As the clustering requires the comparison of individual energy profiles, normalization of the data is necessary. The normalization aims at scaling the entire dataset to a uniform scale such that the energy profiles are clustered based on their “structural” similarities. Thus, a normalization procedure may be a mandatory step prior to implementation of clustering techniques. Various types of normalization techniques are available in the literature, such as the min-max and standard normalization [31], [32]. The min-max normalization is accomplished by dividing each energy value by the maximum energy value in the full dataset. Hence, the min-max normalization process transforms the consumption data of the energy values to the range of [0, 1] of the respective unit, as shown in equation (8):

$$x_i' = \frac{x_i - xmin}{xmax - xmin} \quad (8),$$

where x_i and x_i'indicate the actual and the normalized energy value. Standard normalization (z-score) is utilized in the literature and seeks to bring the entire dataset into a more confined domain, similar to min-max normalization. Standard deviation is a measure of how dispersed the data is in relation to the mean, thus the data points are transformed considering the global average and the standard deviation of the entire dataset as equation (9) indicates:

$$x_i' = \frac{x_i - \bar{x}}{\sigma_x} \quad (9),$$

where $x_i$and x_i' represent the input data value and its normalized data value respectively; $\bar{x}$ and σ_x are the global average and global standard deviation of the entire dataset used to train

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement Nº 957810

the clustering model. It is advisable to apply normalization on a daily basis rather than over long periods to smooth out and reduce the impact of anomalies with critical peaks or incorrect data injections.

4.4 Feature Engineering

Feature engineering is the process of using data science knowledge and techniques to construct the necessary input feature sets that enable the machine learning algorithms to achieve the best performance. The feature engineering includes three steps:

- feature construction (FC)
- feature extraction (FE)
- feature selection (FS)

The **feature construction** is based on the raw data and depends on the user's expertise and domain knowledge to generate new interpretable features that are relevant to the clustering objective. The feature construction procedure entails analyzing the data and extracting useful information. In the context of energy clustering, feature construction needs background information on the power industry. Aim of the feature selection process is to select the subset of features from the original one set according to specific criteria. Additionally, dimensionality reduction techniques are used in feature selection to efficiently lower the original feature set dimension, resulting in low computational complexity and, as a result, reduced execution time, enhancing clustering effectiveness. The **feature extraction** process, like feature selection, minimizes the dimension of the original feature set. Feature extraction preserves as much of the original data as feasible, while obtaining a more abstract and compact feature representation, nevertheless, unlike **feature selection**, feature reduction is accomplished by performing mathematical operations on the input features and transforming them onto a new feature representation set. The Principal Component Analysis (PCA) is one of the most extensively utilized feature extraction methods.

4.4.1 Feature construction and selection in clustering for power systems

In order to produce informative groups from clustering, it is crucial to construct the appropriate features regarding the energy applications. Clustering algorithms are used in the majority of energy applications to recognize and detect groups of customers who are eligible for energy savings through demand response services. This is accomplished by identifying consumers who have higher responsibility to the peak system, as this may lead to more effective energy reduction recommendations. As a result, the goal is to uncover clusters describing various types of peak demand, as well as variabilities and seasonalities existing within the load consumption measurements. Another application of clustering is load profile characterization based on distinctive characteristics in order to assign load labels for commercial purposes and understand the proportion of each consumer type, such as residential consumers, commercial consumers, service consumers, industries, and so on. The mean, standard deviation, and skewness of a load profile are all significant metrics for every dataset since they relate with average consumption, variation, and asymmetry, respectively. Furthermore, daily energy profile data is used to analyze the variability of a customer's electricity consumption patterns over time by determining whether there is a stable or varying electricity use pattern.

Table 4.1: Clustering input features used for power systems.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

Clustering objective	Feature construction inputs	References
Identification of customers for participation in demand response services	• Responsibility factor in different time periods • Variance of the daily energy consumption	[33]
Detect the most appropriate set of customers for demand response schemes	• Relative average power in different time periods • Mean relative standard deviation • Seasonal score for different seasons of the year • Different score for weekdays and weekends	[4]
Load profile characterization	• Mean • Standard deviation • Skewness	[34]
Examine the variability of customer's electricity use patterns	• Daily raw data of load consumption measurements	[35], [12], [36]

4.4.2 Feature Extraction in clustering for power systems

Feature extraction methods minimize the scale of input data by transforming data in a high-dimensional domain into a space with fewer dimensions. Except for dimensional reduction, feature extraction techniques are utilized for visual representation of the clustering performance in many cases. Reducing the number of a data set naturally comes at expense of accuracy, but the computational cost is reduced. One of the techniques frequently utilized to lower the size of the dimensions is Principal component analysis (PCA), which aims to preserve as much of the covariance of the data characteristics as possible while using the fewest features as possible. Essentially, PCA targets to represent all n data vectors as linear combinations of a small number of eigenvectors in order to minimize the mean-squared reconstruction error. A diagram flow of the procedure of PCA is illustrated in Figure 4.3:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

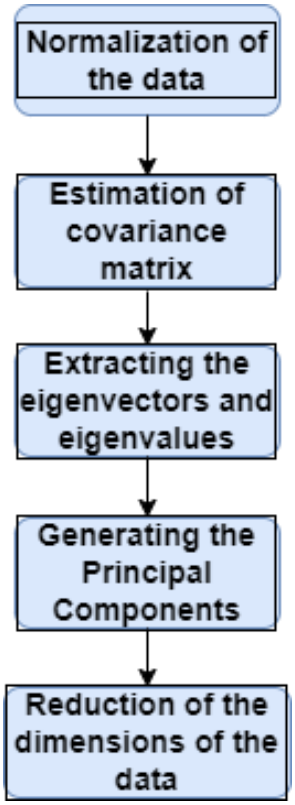

Figure 4.3: Flow diagram of PCA application

1. Data normalization is the initial step before implementing PCA to ensure that all variables and values are within a comparable range.
2. The covariance matrix is computed in the second phase. The correlation between different variables must be identified since they include biased and redundant information, affecting the model's overall performance.
3. The eigenvectors and eigenvalues are extracted from the covariance matrix and formulate the principal components of the data set.
4. The eigenvector with the highest eigenvalue is the most significant and thus forms the first principal component.
5. The last step is the implementation of PCA on the dataset.

4.5 Data Aggregation

One of the most important skills for extracting useful information from time-dependent data is the ability to perform aggregations efficiently. In the context of energy clustering, data aggregation refers to the various procedures used on raw load measurements to generate the proper inputs for the clustering model based on the relative objective defined by the operator. Data aggregation not only reduces the total volume of data, but it also speeds up the discovery of valuable information. Data resampling and data grouping are the two most common data aggregation techniques used on load measurements.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

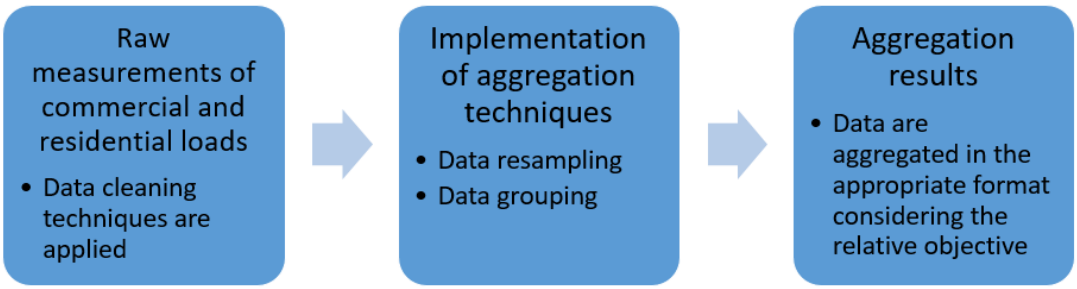

Figure 4.4: A flow diagram of the aggregation procedure

Time-series data resampling: Resampling of time series is a technique for grouping time series data by some convenient frequency. Resampling generates a unique sampling distribution based on the actual load measurements. By aggregating time-series into a lower frequency, data aggregation can help to reduce and smooth large fluctuations that occur within the load measurements. The resampling frequency is subject to the needs of the clustering objective. The graph below depicts energy consumption over a one-week period at two different frequencies. The data in the first figure corresponds to raw energy measurements, whereas the data in the second figure has been aggregated on an hourly basis. It becomes clear that the resampling in hourly resolution eliminates the abrupt variations that occur within energy consumption. The energy consumption data illustrated in this graph is collected from ITI nZEB Smart Home located within the campus of the Center for Research and Technology Hellas (CERTH) [37].

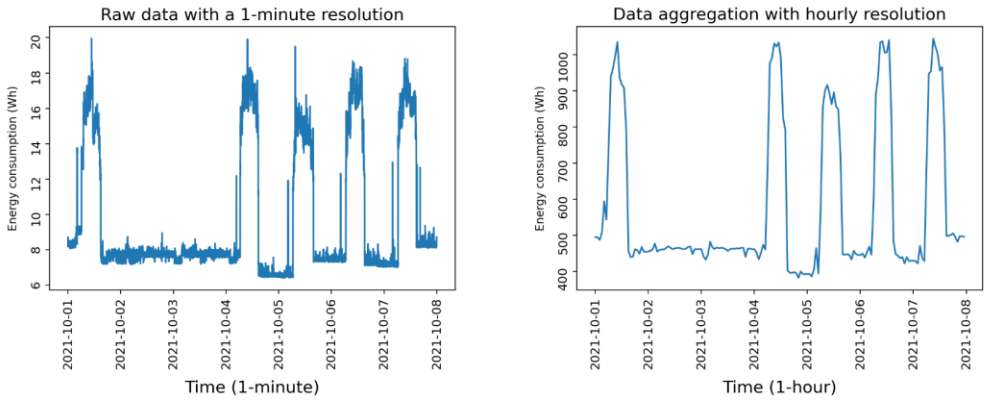

Figure 4.5: Data resampling example (1-min vs 1h data aggregation)

Time-series data grouping: There are several approaches for grouping the load consumption of a single or several users. Each approach's suitability is determined by its possible application. Considering the averaged values of different times is the most obvious grouping technique. A DSO, for example, could benefit by averaging data over a period to have a better understanding of the network's load. An indicative example of data grouping to formulate average energy profiles of three users is illustrated in figure 5.4.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

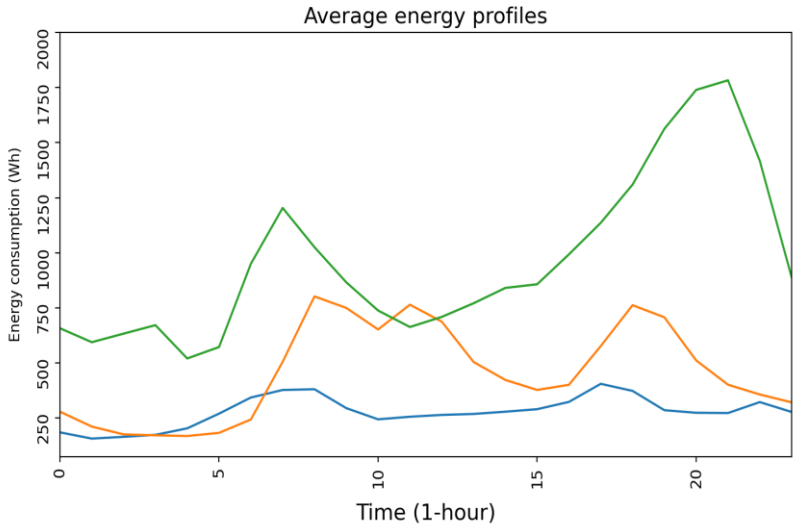

Figure 4.6: Data grouping example (Daily average energy profiles)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

5 Segmentation and Clustering module

In a smart grid system, the aggregator should be provided with meaningful information in order to make the best decisions possible when it comes to demand side management of their portfolio. Clustering will be employed in the context of IANOS to provide relevant insights directly to the aggregator and to supply information to the other iVPP modules that will enhance their functionality. The objectives are derived from IANOS use cases and the clustering process will result in the establishment of a group of users with similar profiles considering their energy consumption. Furthermore, the formulated clusters will be updated according to the application to include newly acquired data and renew the cluster centroids in specific time-intervals. Before proceeding with the clustering analysis, Table 5.1 has a detailed description of the dataset that will be used as input for the different clustering approaches. The different clustering algorithms will be tested on 69 users from the Smart-Grid Smart City Customer (SGSC) dataset [38]. It is important to mention that this data set is used, due to the lack of operational data from the relevant pilots.

Table 5.1: Description of the dataset

Name of the dataset	Smart-Grid Smart City Customer (SGSC)
Date range of the dataset	2012-03-14 00:00:00 - 2014-03-04 08:00:00
Temporal resolution of dataset	30-minutes
Number of users	69

A representative example of a user belonging to the SGSC dataset is given below.

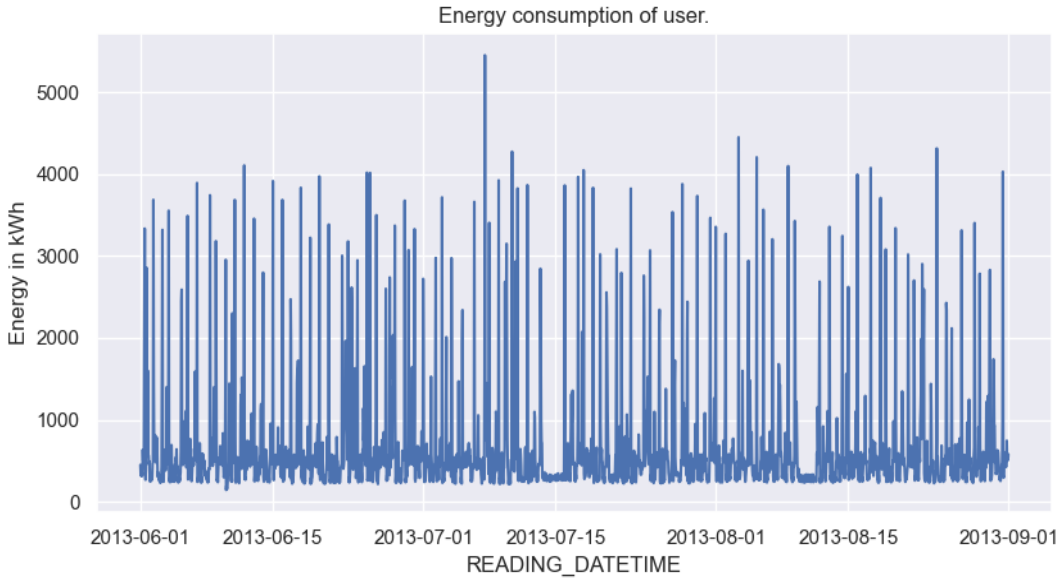

Figure 5.1: Energy Consumption of a single user

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

5.1 Data Segmentation component

The clustering procedure will be carried out via the segmentation component included within the iVPP platform, which will take into account residential and commercial loads. The IANOS segmentation component will serve as a visual information source, providing the operator with valuable insights that will aid in making decisions about users who may be able to participate in demand response schemes. Thus, a set of objectives are defined to investigate the load shapes of users from all perspectives. The objectives must be thoroughly established before the implementation of the clustering process. All the clustering objectives that will be examined within IANOS are provided in Table 5.2:

Table 5.2: Description of IANOS clustering objectives

IANOS Clustering objective	Features
Examine typical consumption patterns	Averaged daily profiles
Detect the most appropriate set of customers for demand response schemes	Peak demand periods: ● divide and estimate the average across certain time periods of the day ● standard deviation
Examine the daily variation of the consumption time series	Raw daily load profiles of the users

5.1.1 Examine Typical Consumption Patterns

An indicative figure of the energy measurements of 69 users from the dataset over a month with half-hour resolution are presented in Figure 5.5.

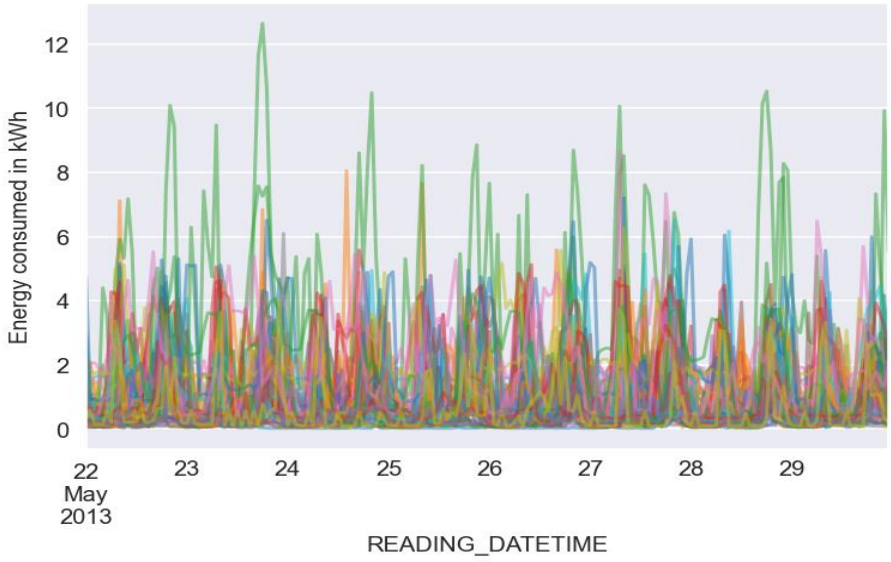

Figure 5.2: Energy measurements of 69 users over the period of a week

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

While these measurements provide some information regarding the consumption of each user, as the number of users equipped with smart meters increases, the volume of the data increases too and the knowledge discovery process becomes harder or even impossible. A first step towards assisting an energy system's stakeholders with their decision making is reducing the volume of the data that are needed in order to plan and program their actions. By first aggregating the raw time series according to certain techniques and then clustering them according to certain features, groupings of customers can be created that are represented by a cluster center. Here, the first clustering objective is to assist the aggregator with the creation of pricing mechanisms for different users according to their level of energy consumption. In order to achieve the cluster is conducted using the users according to their Hourly Daily Load (HDL). In HDL clustering, a feature vector is created for each prosumer. This feature vector contains 24 values that correspond to each of the 24 hours of a typical day. These values are calculated by averaging the observations at each hour of the day for the entire dataset.

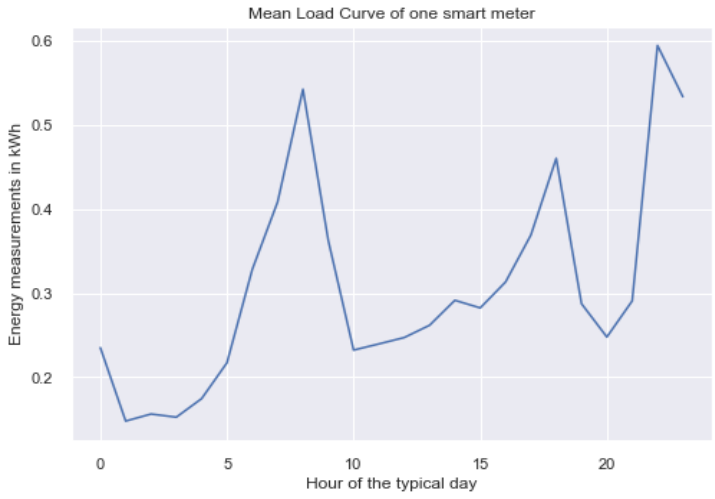

Figure 5.3: Mean Load Curve of a single user

When creating the HDL curve for each user the volume of information is reduced. Here the HDL curves for 69 smart meters are presented:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

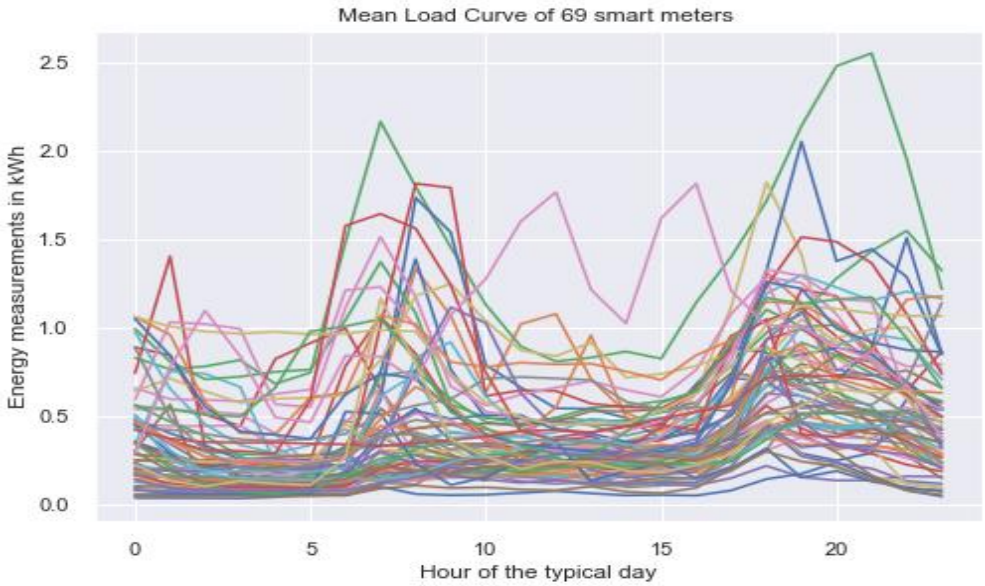

Figure 5.4: Mean Load Curves of measurements from 69 users

Using as features a vector with the 24 values for each user, we use unsupervised machine learning algorithms to cluster the users together and extract their representative centroids. By utilizing the centroids that are computed by the clustering algorithm, the aggregator can design different tariff groups for each group of customers. The first step towards this knowledge discovery process is detecting outliers.

Outlier Detection: After having selected the features, we use DBSCAN for detecting the outliers in the dataset. Next, the outliers are separated from the dataset and the rest of the data are clustered. For better visual representation, the 24 dimensions of the dataset are reduced to 2 dimensions, utilizing the Principal Component Analysis Technique:

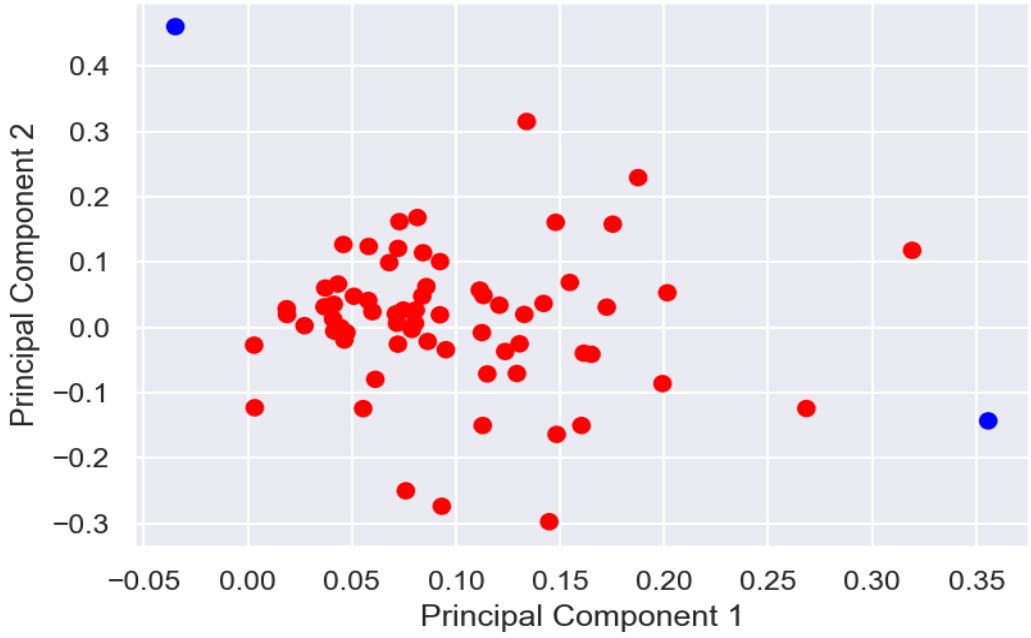

Figure 5.5: Outliers (in blue) after DBSCAN on Mean Load Curves

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

Figure 5.6: Hourly Daily Load of Outlier 1

Figure 5.7: Hourly Daily Load of Outlier 2

Two outliers were detected in the dataset and are removed from the initial dataset. Three different algorithms are used for the cluster formation, namely the k-means, the spectral and the hierarchical clustering, which are then evaluated by their silhouette score. Depending on the algorithm used, the number of clusters that will be formatted is either defined a priori or is computed by the algorithm. The number of clusters that will be used for each algorithm is decided with the silhouette score.

Table 5.3: Algorithms with silhouette scores according to number of clusters

Algorithm	Silhouette Score	Number of Clusters
k-means	0.44	2

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

	0.30	3
	0.29	4
	0.18	5
Spectral	0.46	2
	0.38	3
	0.36	4
	0.38	5
Hierarchical	0.46	2
	0.25	3
	0.251	4
	0.248	5

Hierarchical and spectral algorithms appear to be the most effective, however, the difference is negligible. Following that, a visual depiction of the clusters created by PCA is created:

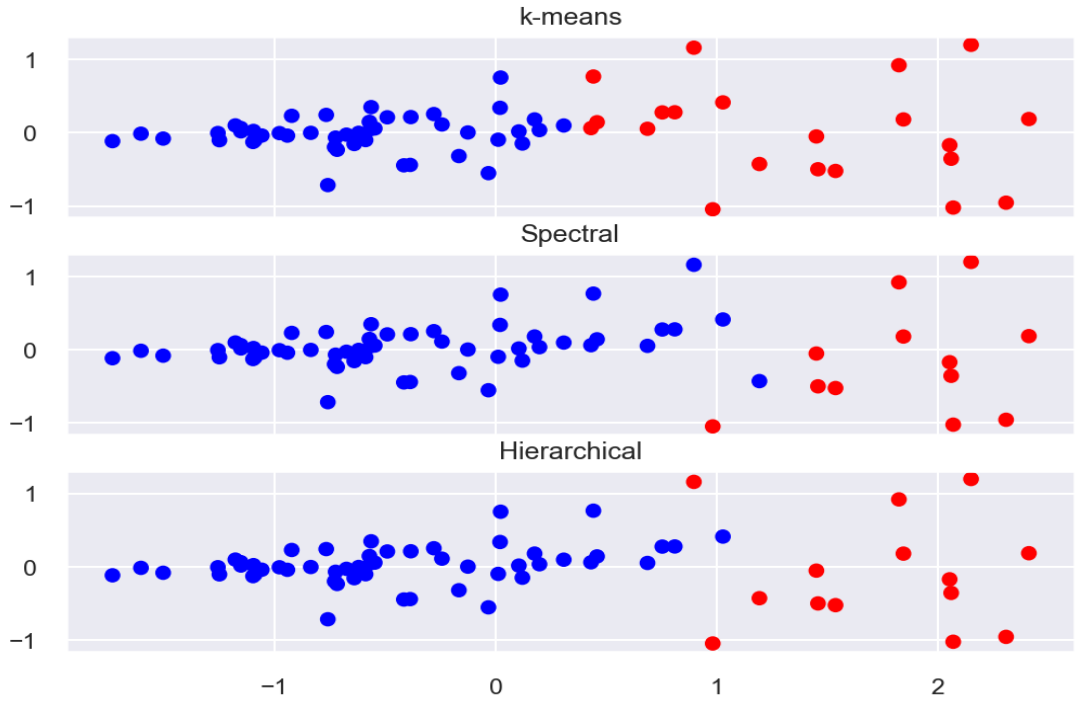

Figure 5.8: PCA Representation of the clustered dataset

Spectral and hierarchical algorithms produce similar results. While labeling the data according to their cluster assignment is important, it is also important to compute the centroids of each cluster, which is defined as the average of all the data that are contained in a cluster. The labeled data with their corresponding centroids are presented in Figure 5.9 and Figure 5.10:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

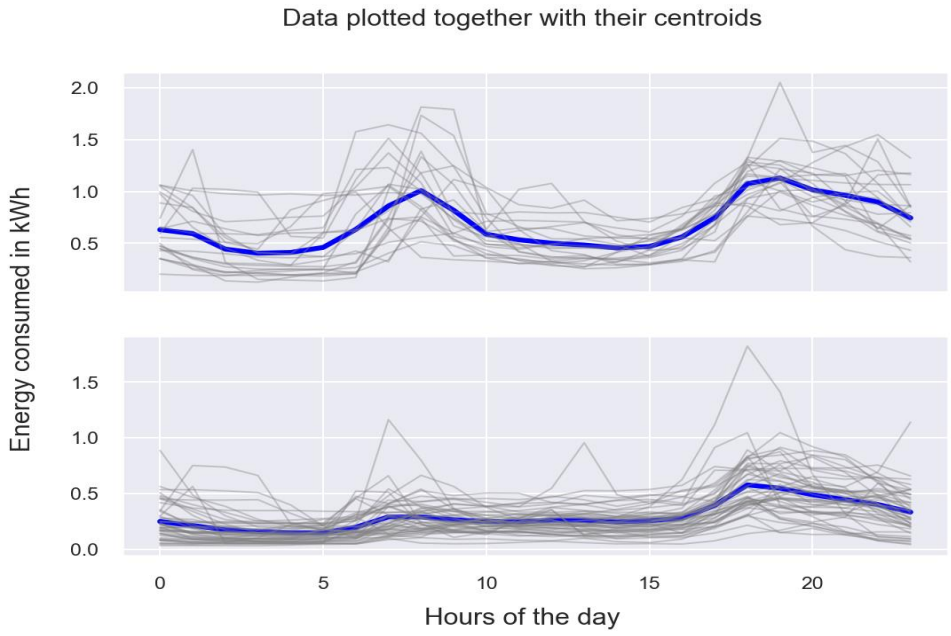

Figure 5.9: Labeled data plotted with their respective centroids according to k-means

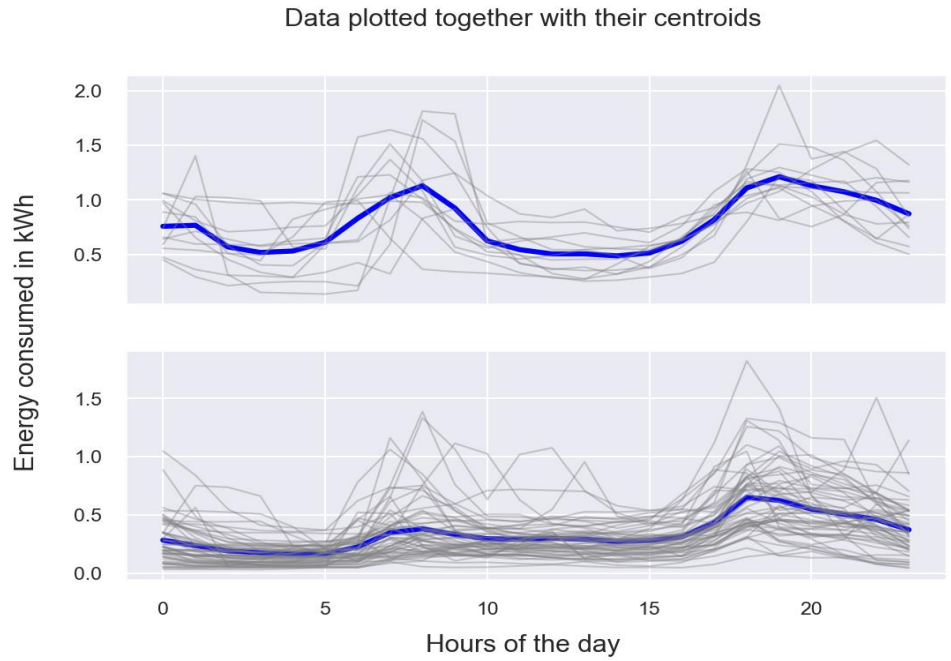

Figure 5.10: Labeled data plotted with their respective centroids according to Spectral Clustering

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

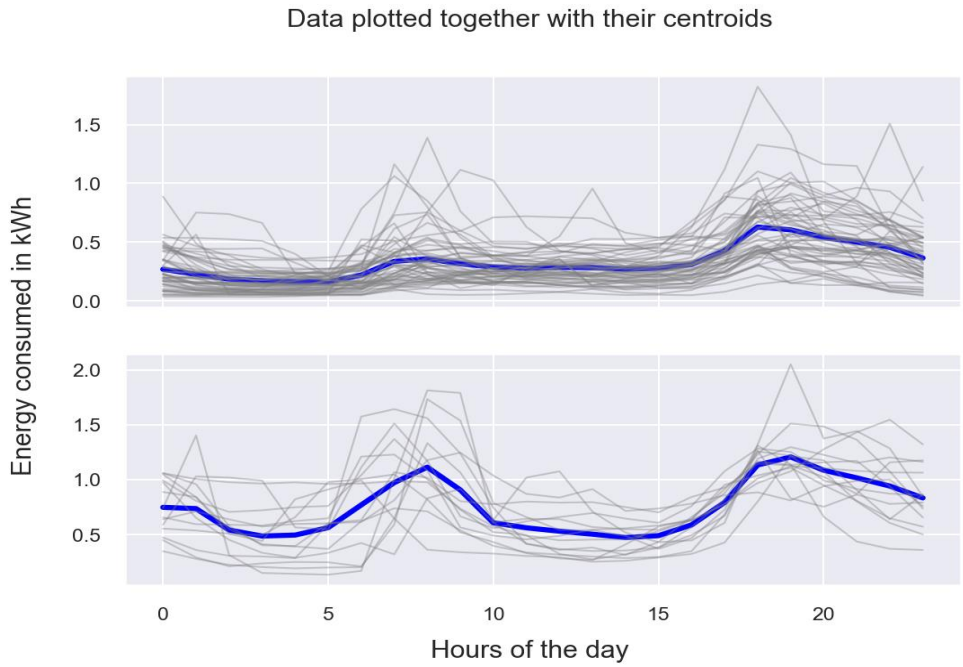

Figure 5.11: Labeled data plotted with their respective centroids according to Hierarchical Clustering

All clustering results point to the fact that two main classes of users exist within the dataset that have quite different levels of consumption and similar consumption patterns, which is to be expected when using non-normalized energy consumption as input features to the algorithm. An aggregator can potentially design different pricing programs depending on the class of the user, and assign new users to classes according to their consumption patterns. In addition, users of the high-energy consumption classes could potentially be used for peak shaving applications when energy demand is high during system peak periods.

5.1.2 Peak Demand Detection

An important aspect of load consumption is the peak periods of electricity usage. To provide the aggregator with information that assists to identify loads with similar patterns and loads appropriate for peak shaving or load shifting, the normalized average daily load loads (HDL) are clustered, according to five features. First, the raw daily curves for each load are normalized by dividing each of the 24 hours of each day with the sum of the energy consumed at that given day. Then the average hourly daily load is created for each user. The HDL are then divided in 4 periods:

- **Morning**: 07:00-10:00
- **Daytime**: 11:00-16:00
- **Evening**: 17:00-23:00
- **Night**: 00:00-6:00

For each period and each load, the mean is extracted as well as the standard deviation, which is then averaged across the four periods. Thus, five features describing the normalized Hourly Daily Load for each load are created. Based on the features three algorithms are used and evaluated for clustering, after detecting and dropping the outliers of the dataset.

Table 5.4: Algorithms with silhouette scores according to number of clusters

Algorithm	Silhouette Score	Number of Clusters

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

k-means	0.25	2
	0.28	3
	0.25	4
	0.24	5
Spectral	0.23	2
	0.26	3
	0.25	4
	0.25	5
Hierarchical	0.21	2
	0.23	3
	0.24	4
	0.19	5

The centroids created by each algorithm are presented in Figure 5.12 and Figure 5.13 respectively.

Figure 5.12: Labeled data plotted with their respective centroids according to k-means

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

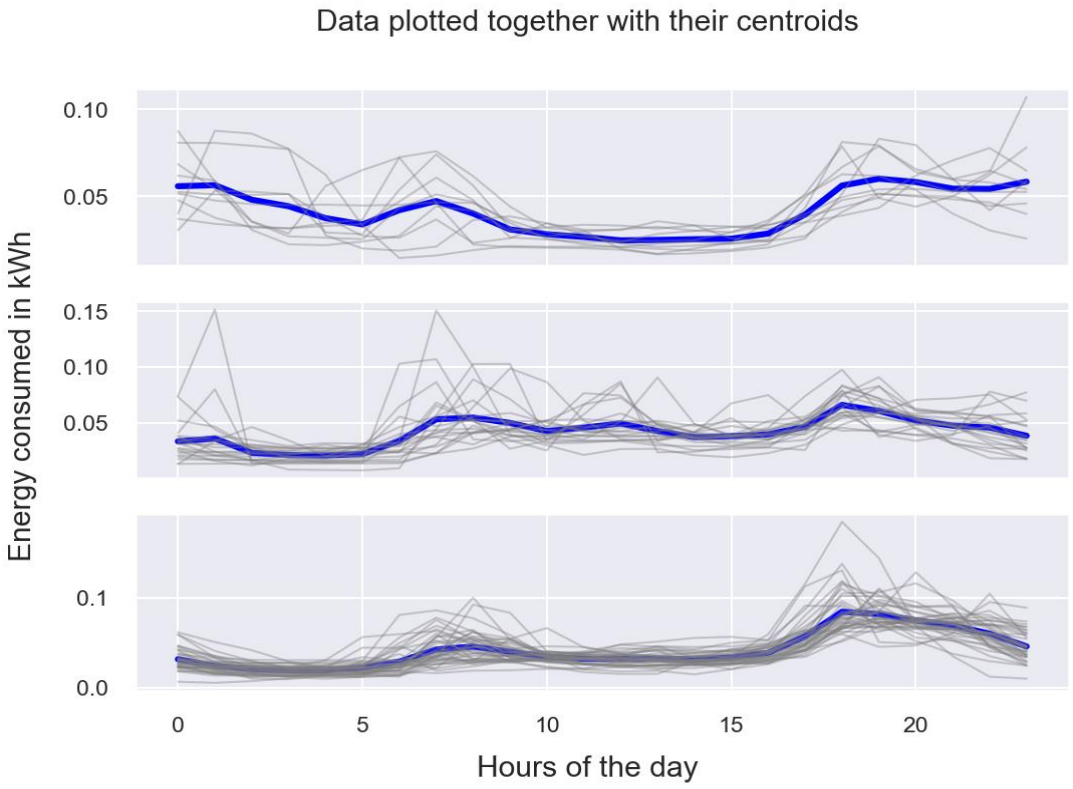

Figure 5.13: Labeled data plotted with their respective centroids according to Spectral Clustering

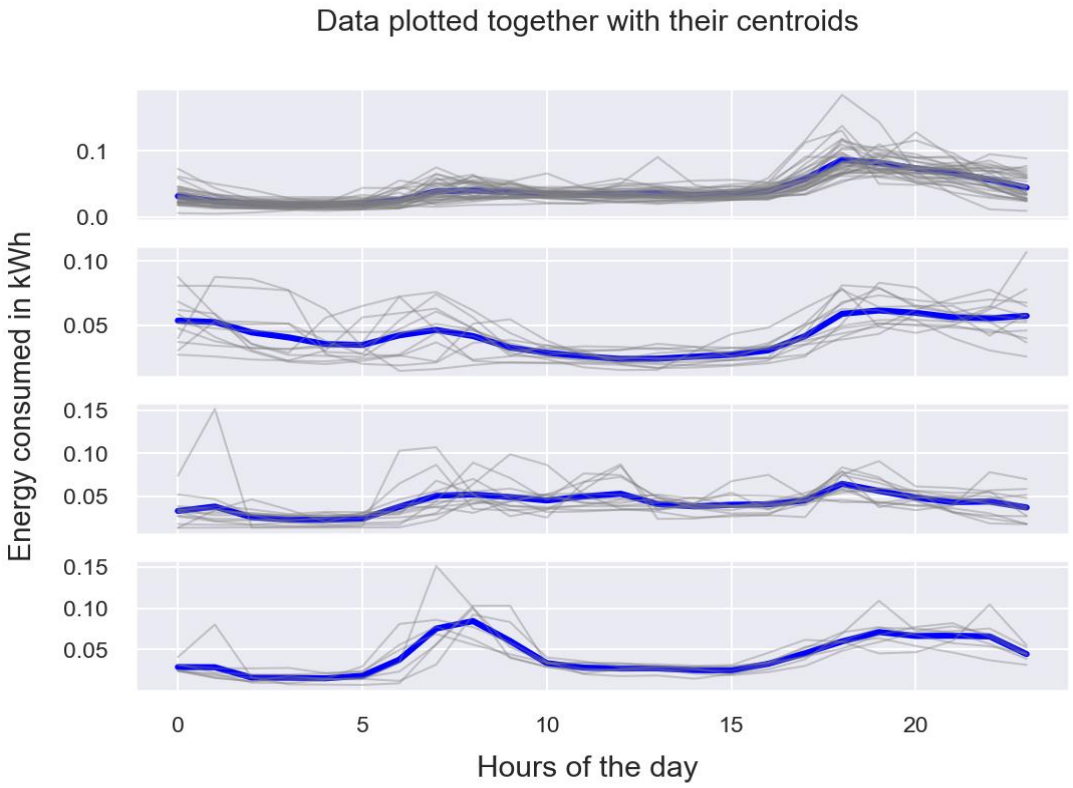

Figure 5.14: Centroids according to Hierarchical Clustering

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

The three algorithms produce significantly different clusters with different numbers and sizes. The best scoring algorithm according to silhouette score is k-means, which forms three clusters. According to the results of k-means, the rest of the analysis is carried forward. Here the cluster centroids together are presented with their means and standard deviations.

Table 5.5: Features of cluster centroids according to k-means

Cluster ID	Morning	Daytime	Evening	Night	Std deviation
0	0.032	0.0.26	0.052	0.045	0.01
1	0.038	0.037	0.064	0.023	0.009
2	0.06	0.033	0.055	0.024	0.011

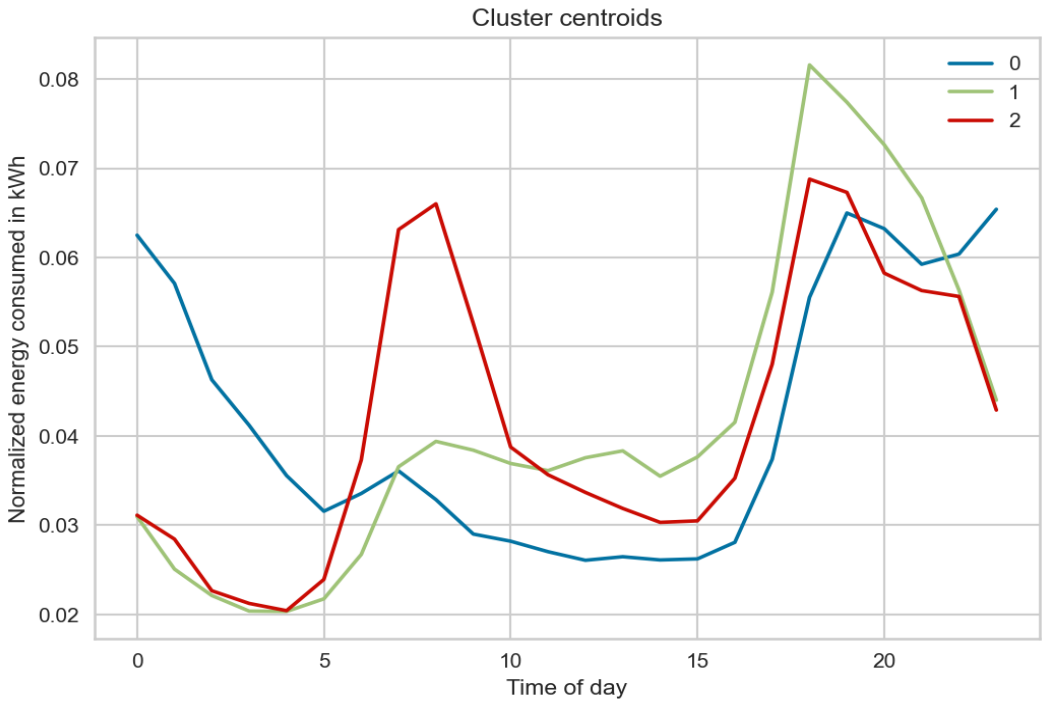

Figure 5.15: Cluster centroids according to K-Means plotted together

The analysis done above for residential load measurements provides the stakeholder with information regarding the grouping of loads to clusters according to their shape. The normalization and feature extraction add extra weight to the shape of the HDLs, revealing important information for the different peaking periods of the users. From the centroids of the two clusters formed, different peaking periods are observed.

Cluster 2 presents peaks in the morning, Cluster 1 mainly in the evening and Cluster 0 during the night. That gives very important information for the distribution of the energy consumption of different users as well as their peaking periods, which can potentially be used for the design of DR programs for peak shaving and load shifting.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

5.1.3 Daily Consumption Variation of an Individual User

Traditionally, analysis of load shapes has focused on average load shapes. Two users with similar average daily load shapes could, however, have drastically different daily electricity profiles throughout the year. As a result, a clustering analysis of the users' daily loads is carried out in order to explore the distributions of daily loads, as well as to evaluate the variability of the users' electricity usage pattern and determine whether it is consistent or varies. The cluster analysis of all users' raw daily load profiles yielded two distinct clusters, with the k-means algorithm presenting the best performance in terms of silhouette score. Cluster1 (26.4 % of the sample) has higher electricity usage throughout the day, whereas Cluster0 (73.6 % of the sample) appears to have lower electricity usage with an almost flat electricity use profile, as shown in Figure 5.16.

Table 5.6: Algorithms with silhouette scores according to number of clusters

Algorithm	Silhouette Score	Number of Clusters
k-means	0.37	2
	0.27	3
	0.21	4
	0.12	5
Spectral	0.2	2
	0.015	3
	0.019	4
	0.04	5
Hierarchical	0.36	2
	0.29	3
	0.27	4
	0.17	5

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

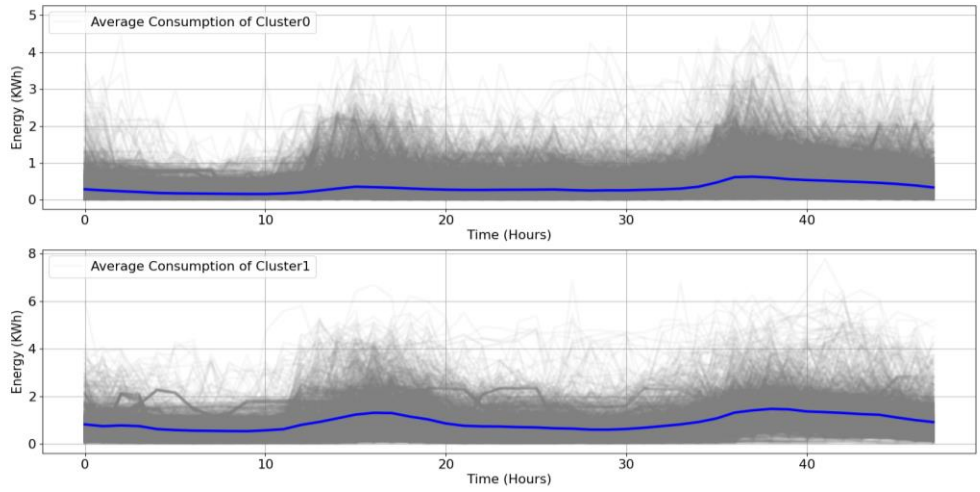

Figure 5.16: Average consumption of each Cluster along with the daily load profiles

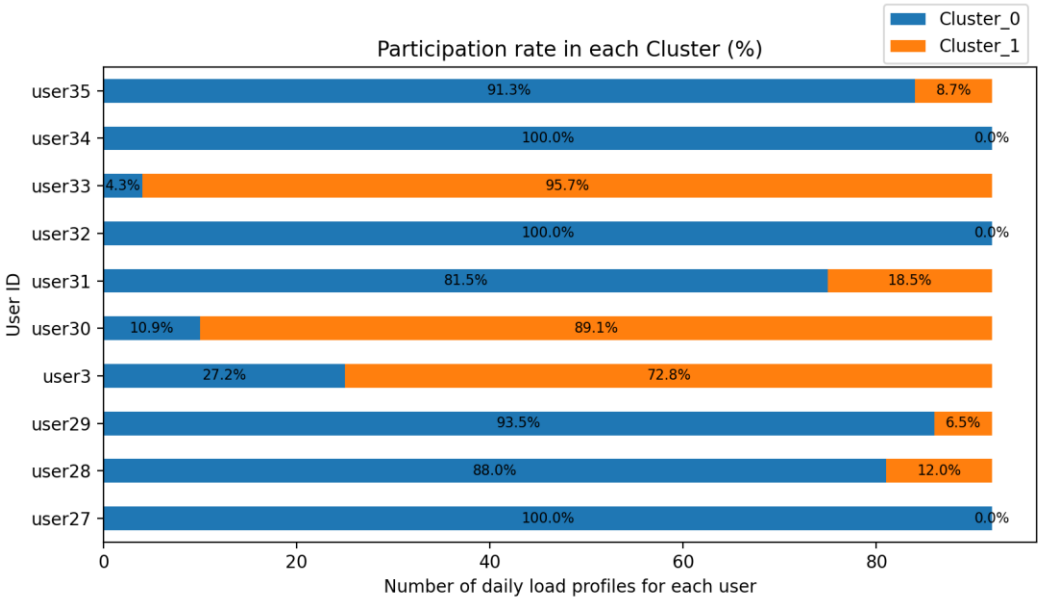

Figure 5.17: Distribution of cluster profiles for 10 randomly selected users

The distribution of cluster patterns observed for ten randomly selected users over a three-month period is depicted in Figure 5.17. It clearly demonstrates that some users adhere to a limited set of load shapes, while others are more resilient, and their daily load profiles alternate across the clusters. For example, Cluster1 describes daily load profiles for 95.7% of the observed period for user33, indicating that this user has less daily variance.

5.2 Clustering based residential load forecasting

5.2.1 Methodology description

Single-user load forecasting is a computationally inefficient method, both in terms of resources used and computational burden. Additionally, it is an approach that is not ideal for deploying the forecasting algorithm to thousands of households on a wide scale, because each new user would require a long data collection period and many data to train his own model. Therefore,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

the clustering can be utilized in order to create groups of users with similar characteristics, on which to train the forecasting models. Furthermore, new users having only few days of available data can be added to a certain cluster, leveraging the existing data from households belonging to the same cluster.

Within the IANOS context, the entire procedure will be developed and deployed in the iVPP platform. More precisely, the load measurements will be retrieved through the Enterprise Service Bus (ESB), which will interact with the established storing databases , and clustering techniques will be applied though the aggregation and segmentation component. After the formulation of the clusters, the labels of the clusters that the users have been assigned will be sent to the forecasting engine. The forecasting engine will develop the models based on the generated clusters and the load forecasts will be sent individually to the users on a daily basis. The broad overview of the load forecasting flow diagram based on clustering is shown in Figure 5.18. Regarding the clustering procedure, the feature construction contains four major characteristics that indicate temporal patterns of user behavior in terms of load consumption, effectively capturing residential users' daily routines. The 4 input features for every user are the following: 1) mean load consumption, 2) variation, 3) 10% quantile and 4) 90% quantile. Before proceeding with the clustering, a dimensional reduction is performed on the features utilizing the principal component analysis (PCA) to represent the dataset onto a lower-dimensional feature subspace, while retaining the majority of the essential information. Afterwards, the K-means algorithm is utilized in order to formulate the clusters on which the load forecasting models will be trained. Additionally, an outlier detection algorithm is performed on the results to identify and drop the users with inconsistent behavior that could affect in a negative manner the performance of load forecasting.

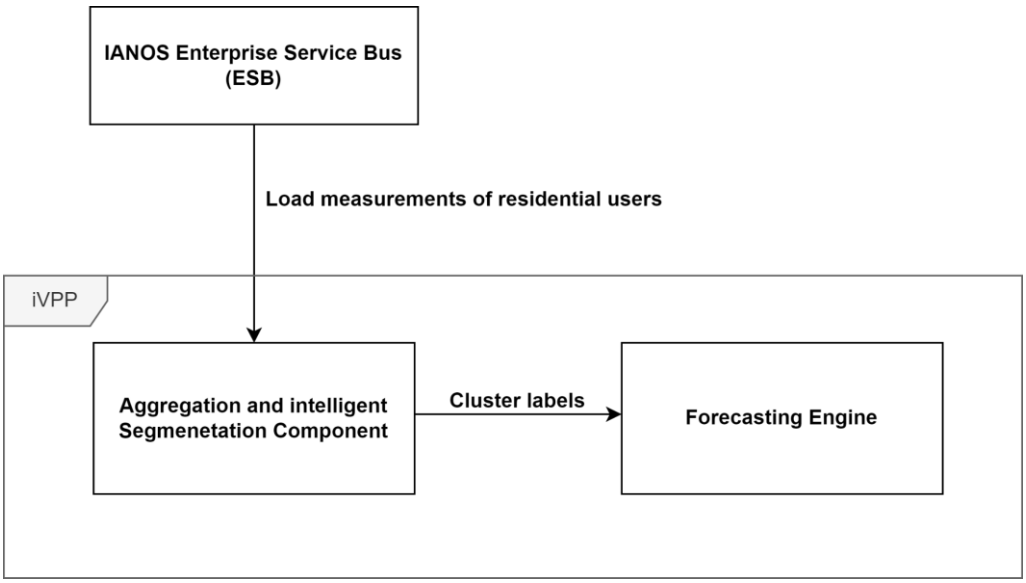

Figure 5.18: Flow diagram of residential load forecasting based on clustering

5.2.2 Results of clustering residential users

In Figure 5.19, there is a scatterplot of the clustering results using the k-means algorithm along with the users that have been identified as outliers. It is clear that two distinct groups of residential users have been formed, while 5 users have been marked as outliers. The outlier detection in terms of abnormal load consumption is conducted using the Local Outlier Factor (LOF) [39]]. Furthermore, in comparison to the other cluster, the cluster comprising the yellow data points exhibits greater coherence. In forecasting terms, this indicates that the users

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

assigned to Cluster A have similar energy consumption patterns, whereas the blue data point cluster (Cluster B) is widely dispersed throughout the space, implying that the users' load varies more. Thus, the load forecasting model of Cluster A is more likely to perform better in terms of accuracy. Finally, individual load forecasting models will be applied to each of the users considered outliers and excluded from both clusters. Nonetheless, in order to assess the high load variation of users assigned to Cluster B and to the outliers, Figure 5.20 presents the average daily load of both clusters and outliers. It is apparent that the daily average load of users in Cluster A has a consistent low variance around the overall cluster average.

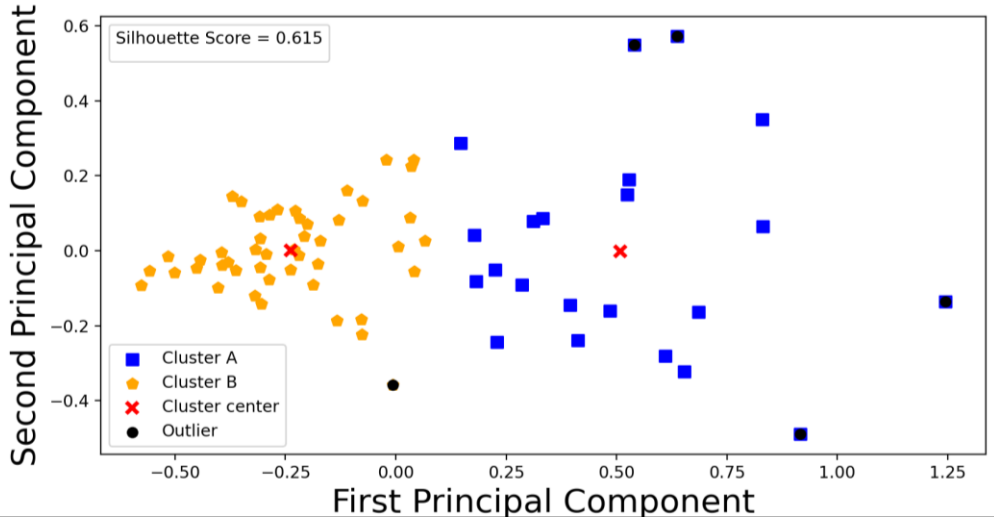

Figure 5.19: Residential users clustering

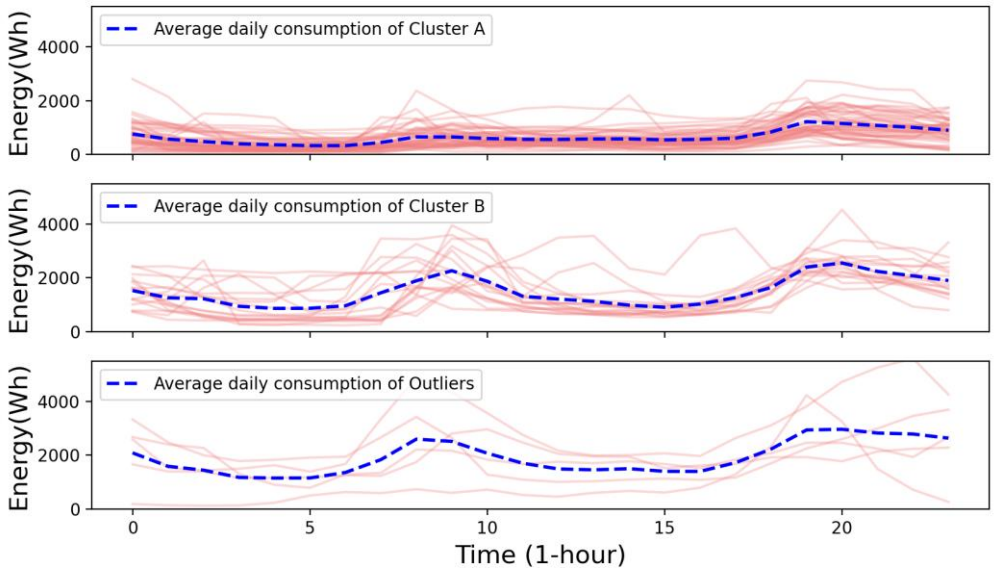

Figure 5.20: Average daily consumption of clusters and outliers

Table 5.7 provides a comprehensive description regarding the clusters that have been formulated. Cluster A is the cluster that has the most users assigned to it and has an average daily load of 15.48 kWh, while Cluster B with 35 users has an average daily load of 35 kWh.

Table 5.7: Description of the clusters

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

Cluster	No. of users	Avg. daily load (kWh)
Cluster A	46	15.48
Cluster B	18	35
Outliers	5	46

5.2.3 Classification of new residential users

After the initial clustering and the generated groups of users with similar characteristics, a classification procedure is carried out. A classifier algorithm is employed in the case of a new user to assign the user to one of the existing clusters with similar behavior in terms of energy consumption. The classifiers are developed using the same features that were used as inputs to the clustering, with the cluster labels serving as target variables. Three different classifiers have been implemented for the classification of the new users. As previously noted, the classifiers were trained using the same dataset as the clustering procedure. An 80/20 training/test split was performed for the training, resulting in 14 users who will be used to evaluate the classifier's performance. The accuracy score is estimated for the evaluation and is an indicator of the number of users who have been exactly matched with the corresponding set of labels. Three different algorithms have been utilized for the classification. The algorithms and the classification results are presented in Table 5.8. Random forest (RF) and Extreme Gradient Boosting (XGBoost) were able to correctly categorize all 14 users; however, the Support Vector Machine classifier missed two of them.

Table 5.8: Algorithms used for the classification

Classifiers	Accuracy score (%)
Random forest (RF)	100
Extreme Gradient Boosting (XGBoost)	100
Support Vector Machine (SVM)	85.7

The actual labels (blue dots) of the users and the "forecasted" labels (orange dots) that the classifier has assigned them are depicted in Figure 5.21, which was trained using the support vector machine (SVM) classifier model in this case. As can be seen, 12 users were accurately assigned to the correct cluster, whereas two users were wrongly categorized.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

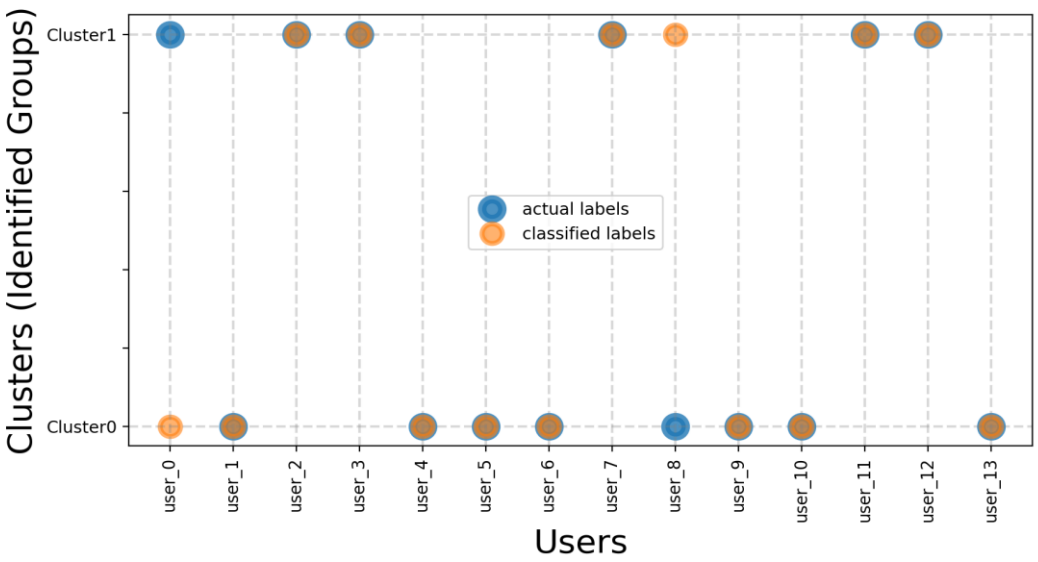

Figure 5.21 Classification results

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

6 Conclusions and future steps

This deliverable examines in detail the functionalities that the Aggregation and intelligent Segmentation component executes in the context of the IANOS project. Energy clustering will be employed as a data analysis tool to deliver valuable information about the energy behavior of the portfolio to the relevant portfolio Manager/Aggregator/Retailer. In Section 4, three useful clustering objectives have been defined and analyzed, with the corresponding results. In addition, in Section 5, a more practical application of clustering is provided. Instead of generating individual models and increasing the computation cost of the entire system, the forecasting engine will develop forecasting models based on the clusters formed by the Segmentation and Aggregation component. The next steps include the collection of pilot data and the implementation of the proposed clustering analysis on it, reporting the results on the second version of the deliverable on M32.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

7 References

[1] M. Liu, S. Yang and L. Xiatao, "Distributed MPC of aggregated heterogeneous thermostatically controlled loads in smart grid," *IEEE Transactions on Industrial Electronics ,* pp. 1120-1129, 2015.

[2] Q. CIu, "Residential appliances direct load control in real-time using cooperative game," *IEEE Transactions on Power Systems,* pp. 226-233, 2015.

[3] M. Muratori and G. Rizzoni, "Residential demand response: Dynamic energy management and time-varying electicity pricing," *IEEE Transactions on Power Systems,* pp. 1108-1117, 2015.

[4] S. Haben, C. Sigleton and P. Grindrod, "Analysis and Clustering of Residentail Customers Energy Behavioral Demand Using Smart Meter Data," *IEEE Transactions on Smart Grid,* pp. 136-144, 2016.

[5] M. Omid, A. Berry and L. O'Neil, "Clustering of residentail electricity customers using load time series," *Applied Energy,* pp. 11-24, 2019.

[6] S. Lin, F. Li, E. Tian, Y. Fu and D. Li, "Clustering Load Profiles for Demand Response Applications," *IEEE Transactions on Smart Grid ,* vol. 10, pp. 1599-1607, 2019.

[7] L. Kotzur, et al. , "Impact of different time series aggregation methods on optimal energy system design," *Renewable Energy,* pp. 474-487, 2018.

[8] G. Mavrotas, et al. , "A mathematical programming framework for energy planning in services sector buildings under uncertainty in load demand: The case of a hospital in Athens," *Energy Policy,* pp. 2415-2429, 2008.

[9] Y. Wang,et al., "Clustering of electricity consumption behavior dynamics toward big data apllications," *IEEE transactions on smart grid,* pp. 2437-2447, 2016.

[10] A. K. Jain, N. M. Murty and J. P. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR),* pp. 264-323, 1999.

[11] L. Ran, et al., "Develompent of low voltage network templates- Part I: Substation clustering and classification," *IEEE Transactions on Power Systems,* pp. 3036-3044, 2014.

[12] I. Benitez, et al, "Dynamic Clustering segmentation applied to load profiles of energy consumption from Spanish customers," *International Journal of Electrical Power and Energy Systems,* pp. 437-448, 2014.

[13] Z. Zhao, W. J. and L. Y., "User Electricity Behavior Analysis Based on K-MEans plus Clustering Algorithm," in *International Conference on Computer Technology, Electronics and Communication(ICCTEC)*, 2017.

[14] A. Rajabi,et al., "A comparative study of clustering techniques for electrical load pattern segmentation," *Renewable and Sustainable Energy Reviews,* 2020.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 957810

[15] L. Gengyuan, "Big data-informed energy efficiency assesment of China industry sectors based on K-Means clustering," *Journal of cleaner production,* pp. 304-314, 2018.

[16] G. Chicco, "Overview and performance assesment of the clustering methods for electrical load pattern grouping," *Energy,* pp. 68-80, 2012.

[17] J. Rhodes,et al., "Clustering analysis of residential electricity demand profiles," *Applied Energy,* pp. 461-471, 2014.

[18] F. Mcluoghlin, D. Aidan and C. Michael, "A clustering approach to domestic electricity load profile characterization using smart metering data," *Applied Energy,* pp. 190-199, 2015.

[19] W. Kong, Y. J. Dong, D. J. Hill, Y. Xu and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," *IEEE Transactions on Smart Grid,* vol. 10, pp. 841-851, 2019.

[20] J. MacQueen, "Some methods for classifciation and analysis of multivariete observations," in *Proceedings of the fifth Berkley sympsium on mathematical statistics and probability*, 1967.

[21] S. Archana, A. Yadav and A. Rana, "K-means with Three different distance Metrics," *International Journal of Computer Applications,* 2013.

[22] E. Martin,et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996.

[23] A. Ng, J. Michael and W. Yair, "On spectral Clustering: Analysis and an algorithm," *Advances in neural Information systems ,* 2001.

[24] "Spectral Clustering," [Online]. Available: https://en.wikipedia.org/wiki/Spectral_clustering.

[25] "Hierarchical Clustering," [Online]. Available: https://en.wikipedia.org/wiki/Hierarchical_clustering.

[26] F. Nielsen, "Hirarchical Clustering," in *Introduction to HPC with MPI for Data Science*, Springer, 2016, pp. 195-211.

[27] "Scikit-learn Hierarchical Clustering," [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering.

[28] S. Ketan Rajshekhar and N. Charles, "Cluster Quality Analysis using silhouette Score," in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 2020.

[29] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on pattern analysis and machine intelligence ,* pp. 224-227, 1979.

[30] J. W. Grzymala-Busse and W. J. Grzymala-Busse, Handling missing attribute values, Boston: Springer, 2009.

[31] C. Saranya and G. Manikandan, "A study on normalization tecniques for privacy preserving data mining," *Internation Journal of Engineering and Technology,* pp. 2701-2704, 2013.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810

[32] A. Lavin and D. Klabjan, "Clustering time-series energy data from smart meters," *Energy Efficiency,* pp. 681-689, 2015.

[33] A. Maher and F. Walling, "Smart meter data clustering using consumption indicators: responsibility factor and consumption variability," *Energy Procedia,* pp. 2236-2242, 2017 .

[34] K. A. Choksi, J. Sonal and M. P. Naran, "Feature based clustering techniques for investigation of domestic load profiles and probabilistc variation assessment: Smart Meter dataset," *Sustainable Energy, Grids and Networks ,* 2020.

[35] S. Yilmaz, J. Chambers and P. M. Kumar, "Comparison of clustering approaches for domestic electricity load profile characterization- Implications for demand side management," *Energy ,* pp. 665-677, 2019.

[36] J. Kwac, F. June and R. Rajagopal, "Household energy consumption segmentation using hourly data," *IEEE transactions on Smart Grid,* pp. 420-430, 2014.

[37] "Smart Home ITI," [Online]. Available: https://smarthome.iti.gr/.

[38] "Smart-Grid Smart-City Customer Trial Data," [Online]. Available: https://data.gov.au/data/dataset/smart-grid-smart-city-customer-trial-data.

[39] "Local Outlier Factor," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 957810