POLITECNICO
DI TORINO

POLITECNICO DI TORINO

Corso di Laurea in Energetica e Nucleare

Tesi di Laurea Magistrale

Detection and diagnosis of anomalous energy consumption patterns in buildings through a data analytics based approach

The case of Politecnico di Torino

Relatori
Prof. Alfonso Capozzoli, PhD
Eng. Marco Savino Piscitelli, PhD

Candidato
Chiosa Roberto
matricola: 262801

ANNO ACCADEMICO 2019-2020

Summary

In recent years, Smart Metering Infrastructure (SMI) has enabled the easy collection of high frequency building-related energy consumption data. Therefore, it becomes necessary to extract from meter level data as much information as possible in order to optimize building energy management, by reducing losses due to inefficiencies or anomalous behaviour of sub-systems and equipment. This paper proposes an innovative top-down Anomaly Detection and Diagnostics (ADD) methodology able to automatically detect at whole building meter-level anomalous energy consumption and then perform a diagnosis on the sub-loads responsible of that anomalous behaviour. The process consists of a multi-step procedure combining various data mining techniques. An evolutionary classification tree is firstly implemented to discover frequent and infrequent daily aggregated energy patterns opportunely abstracted through an Adaptive Symbolic Aggregate approXimation (ASAX) process. Then a post-mining analysis based on Association Rule Mining (ARM) is performed to discover the main sub-loads affecting the detected anomalous energy patterns at high meter level. The methodology is tested on metering data related to the electrical load of a transformer substation of a university campus, leading to the development of a tool useful to support the energy management with an effective characterization of energy demand at a daily scale.

Acknowledgements

I thank the BAEDA research group for inspiring and guiding me in this new world, with the conviction that the future path together can be profitable.

I express my gratitude to Living Lab of PoliTo for providing data and to Eng. Giovanni Carioni for the support in data preparation and collection.

Contents

List of Figures 7

List of Tables 9

1 Introduction 11
- 1.1 Literature review . . . 15
- 1.2 Novelty . . . 17

2 Data analysis methods 19
- 2.1 Dimensionality reduction . . . 19
  - 2.1.1 Symbolic Aggregate approXimation . . . 20
  - 2.1.2 Adaptive Symbolic Aggregate approXimation . . . 22
- 2.2 Classification . . . 25
  - 2.2.1 Recursive partitioning . . . 27
  - 2.2.2 Globally optimum evolutionary tree . . . 29
- 2.3 Clustering . . . 30
  - 2.3.1 K-means clustering . . . 30
  - 2.3.2 Hierarchical clustering . . . 31
- 2.4 Association rules mining . . . 33

3 Case study 35
- 3.1 Unlabeled . . . 39
- 3.2 Labeled . . . 39
  - 3.2.1 Print Shop . . . 39
  - 3.2.2 Mathematics department . . . 40
  - 3.2.3 Bar Ambrogio . . . 40
  - 3.2.4 Rectory . . . 41
  - 3.2.5 Refrigeration unit . . . 42
  - 3.2.6 Data Center . . . 42

3.2.7 Canteen 43

4 Methodology 49
4.1 Preprocessing 49
4.2 Time series abstraction 51
4.3 Detection at meter level data 52
4.4 Diagnosis at sub-meter level data 52

5 Results 55
5.1 Preprocessing 55
5.2 Time series abstraction 62
5.3 Detection at meter level data 66
5.4 Diagnosis at sub-meter level data 72
5.5 Simulation of application 79
5.5.1 One month simulation 79
5.5.2 Six month simulation 81

6 Conclusion 87

Acronyms List 89

Bibliography 91

Further Reading 96

List of Figures

1.1 EMIS tool classification 13

2.1 Example of SAX process 22
2.2 Example of ASAX process 24
2.3 Definition of trend feature triangle and trend angle 25
2.4 Classification process. 26
2.5 Classification tree description. 27
2.6 Example of K-means with $K = 3$. 32
2.7 Example of single link hierarchical clustering. 33

3.1 Electrical substations of PoliTo 36
3.2 Hierarchical structure of the electrical load database under study. 38
3.3 Meter-level power distribution of total loads and sub-loads 44
3.4 Box-plots of hourly electrical load 45
3.5 Box-plots of daily electrical load 46
3.6 Box-plots of monthly electrical load 47

4.1 Flow chart explaining the adopted methodology. 49
4.2 Outlier detection and handling of a positive skewed distribution. 51
4.3 Sub-meter level diagnosis methodology description. 54

5.1 Outliers identification through boxplots. 56
5.2 Carpet plot for the electrical load of Total Power 57
5.3 Carpet plot for the electrical load of Not allocated 57
5.4 Carpet plot for the electrical load of Print Shop 58
5.5 Carpet plot for the electrical load of DIMAT 58
5.6 Carpet plot for the electrical load of Bar Ambrogio 59
5.7 Carpet plot for the electrical load of Rectory 59
5.8 Carpet plot for the electrical load of Refrigeration unit2 60
5.9 Carpet plot for the electrical load of Data centre 60

5.10 Carpet plot for the electrical load of Canteen 61
5.11 Carpet plot for the external air temperature 61
5.12 CART tree for the sub-daily time window identification 62
5.13 Complexity parameter and tree size determination. 63
5.14 Adaptive breakpoints search on the aggregated total electrical load 65
5.15 ASAX representation of the total electrical load 66
5.16 Globally optimum tree for time window 1 (00:00 - 06:29). 70
5.17 Globally optimum tree for time window 2 (06:30 – 08:59). 70
5.18 Globally optimum tree for time window 3 (09:00 - 15:44). 71
5.19 Globally optimum tree for time window 4 (15:45 - 19:14). 71
5.20 Globally optimum tree for time window 5 (19:15 - 24:00). 72
5.21 ASAX carpet plot for Print shop. 74
5.22 ASAX carpet plot for DIMAT. 74
5.23 ASAX carpet plot for Bar Ambrogio. 75
5.24 ASAX carpet plot for Rectory. 75
5.25 ASAX carpet plot for Refrigeration unit. 76
5.26 ASAX carpet plot for Data centre. 76
5.27 ASAX carpet plot for Canteen. 77
5.28 Diagnosis procedure applied on node 5 of time window 2 78
5.29 Confusion matrix January 2019 83
5.30 Focus on electrical load in period 2 node 5 (January 2019) 84
5.31 ocus on electrical load in period 4 node 2 (January 2019) 85

List of Tables

2.1 Breakpoints or lookup table according to alphabet size. 22
2.2 Example of transactional database. 33

3.1 List of facilities fed by electrical substations. 37

5.1 Sub-daily time windows for total electrical power 64
5.2 Accuracy results comparison between test and validation 68
5.3 Decision rules extracted from globally optimal classifier 69
5.4 Validation accuracy results comparison between not retrained model (A) and retrained model (B) 82

Chapter 1

Introduction

In the last years, the building sector is continuously increasing its energy demand, accounting for one-third of global energy consumption [28]. According to the International Energy Agency (IEA) this trend is the result of a combination of different factors: extreme climatic events, increasing demand for building energy services and the easier access to electricity and ownership of heating and cooling appliances in emerging economies. At the actual state energy efficiency innovations and sustainable policies are not able to keep pace with the energy increase demand rate. However, sustainability concerns and energy targets set by the international community [1] enhanced researches and development efforts for energy savings.

Next to building envelope and construction improvements, higher plug-load devices efficiency and energy-efficient equipment, a consistent part of energy reduction potential hides behind energy management strategies. Malfunctioning of sensors or control logics, unexpected environment conditions, wrong settings, human or equipment-related faults is the cause of massive energy waste. Buildings are full of energy savings potential, implementation of energy management strategies and automatic control could save up to 22% of building energy consumption by 2028 [2]. This is the reason why many governments are adopting policies, like the European Energy Performance of Building Directive [32], incentivizing the installation of Building Automation Systems (BAS) and Smart Metering Infrastructure (SMI) to implement energy management strategies for reducing energy wastes and operational costs.

In this context, it is worth reminding how the adoption of Information Technology (IT) is disruptively changing and transforming the building sector. The

most interesting side of technological advancement in electronics and measurements instrumentation and its wide adoption, is the reduction of costs of sensors installation and data storage [26]. This aspect lead to a broader adoption of Advanced Metering Infrastructure (AMI) which are enabling the collection of massive amounts of data that could lead, if effectively analyzed, to significant energy savings [13]. The quantity, quality and detail of data and the successive analysis and results strongly depend on the set up of the sensors infrastructure. Those sensors measures, depending on the level of detail provided, can be classified as follows:

- *Meter level*: refers to aggregated measures at whole building level of variables such as the electrical energy for lighting, cooling, ventilation or gas consumption for heating;
- *System level*: refers to aggregated measures of a particular system such as the electrical load absorbed by a pump, boiler, fans or terminal unit;
- *Detailed-level*: refers to punctual measures of a particular variable such as temperature or flow rate.

By their nature, buildings are complex systems in which constant interactions between humans, technological systems, whether and physical phenomena are present. As a consequence, related data are heterogeneous and can be categorized as follows:

- *Climatic data*: dry bulb temperature, dew point temperature, pressure, humidity, wind speed, solar radiation, total rainfall;
- *Phisical parameters*: floor area, heated gross volume, U-value, aspect ratio, window-to-wall ratio, orientation, other thermo-physical parameters;
- *Operational data*: systems operational data of Heating, Ventilation and Air Conditioning (HVAC), indoor temperature, energy consumption, energy price, renewable energy production, indoor environmental quality parameters
- *User related data*: occupancy, number of occupants, number of ON/OFF appliances, opening/closing windows, social economic factors;
- *Time variables*: season, month, date, day of the week, the hour of the day.

Because of this heterogeneity, lack of standard storage processes and the ineffectiveness of traditional statistical analysis to process huge amounts of data,

research and development in this field could unlock the building energy savings potential.

Artificial Intelligence (AI), advanced data analytics and machine learning are the tools that have the potential to fill this gap on the technical point of view, since they are capable of handling large data, enhancing and speeding up analysis and offer new insights in data.

In the building field, Energy Management and Information Systems (EMIS) is a family of data analytics tools rapidly evolving that offers insights of energy use, building performance and control optimization, providing energy savings up to 20% [19].

A first classification of EMIS tools is formulated considering if their functionalities are enabled at meter- or system-level, those can be summarized and classified as shown in Figure 1.1. The first category of EMIS considers data measurements at a high level (e.g., data retelated to the total load or of the main sub-loads) while system-level EMIS are focused on more detailed data related to the operation of specific systems or components. Benchmarking, Energy Information Systems (EIS) and Building Automation Systems (BAS) are the more traditional tools, in this framework, the focus will be on the advanced Decision Support Systems (DSS) which are Advanced Energy Information Systems (AEIS), Fault Detection and Diagnosis (FDD) and Automated System Optimization (ASO), reported with light blue background in Figure 1.1.

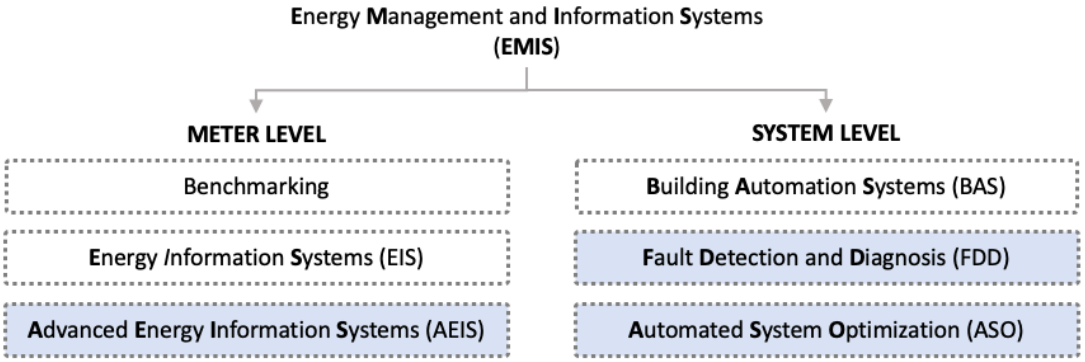


Figure 1.1: EMIS tool classification according to detail of data and detail of analysis. Adapted from [19].

- *Energy benchmarking*: consists in the analysis of annual energy data to rank the building performance among its peer. Is a useful tool for the assesment

of energy consumption but does not offer a detailed insight on how to improve the performance;

- *Building Automation System (BAS)*: they are a measuring infrastructure composed of sensors, controls and other components that acquire system level data and automatically manages and controls many complex systems such as air conditioning, heating and cooling, lighting and security systems;

- *Energy Information Systems (EIS/AEIS)*: consists of on hardware part (sensors, acquisition system, storage) used for the collection and storage of data and a software part used for the analyses and display of building-related data. The basic EIS consist of monthly bill consumption analysis and Key Performance Index (KPI) calculation to track the performance in time of a given building or a portfolio of buildings. The more advanced tools use meter-level, system-level and detailed-level data and proposes innovative machine learning algorithms to automatically analyse, interact and display data, to identify energy savings opportunities;

- *Fault Detection and Diagnostic systems (FDD)*: consists of a software that, analysing high-frequency system-level data from BAS, automates the detection of anomalous behaviour and offers a diagnosis of the potential causes. Unlike a simple alarm, FDD applies machine learning algorithms on BAS data providing a more detailed description of the fault and can lead to a significant energy saving;

- *Automated System Optimization (ASO)*: consists of a software that continuously analyzes BAS data and modifies control outputs to optimize HVAC operation, while maintaining occupant comfort. An example is the reinforcement learning approach.

The integration of innovative data analytics methods in DSS with BAS measurements can contribute to great energy savings opportunities. BAS systems continuously store a considerable amount of high-frequency real-time measurements of many variables (temperature, humidity, power, etc.), mainly used to fulfil control strategies. Moreover, it can also provide simple threshold-based alarms when measured data are out of range. However, the analytical capabilities of BASs are not enough developed for supporting users in gaining insight into measured data.

To this purpose EMIS can be employed, in particular EIS which are intended as tools focused on meter-level monitored data that are not usually integrated

with BAS. In this context extracting knowledge from meter-level data results of paramount importance considering that modern SMI make it possible to have available high frequency measurements of total electrical load and of the main sub-loads even if a BAS is not installed in a building.

This work proposes an EIS tool to perform an Anomaly Detection and Diagnostics (ADD) analysis by exploiting meter-level data in order to support the prompt detecting of possible anomalies and inefficiencies. ADD procedures are usually performed offline or on small subsets of historical data, but more and more interest is growing in creating automatic real-time techniques to analyse data. In this thesis is presented an innovative top-down ADD methodology conceived for working in streaming which allows the automatic detection of anomalies at whole building level and performs a diagnosis to evaluate the sub-load responsible for the anomalies detected.

1.1 Literature review

The importance of systematically analyse building-related data with the purpose of energy saving is evident from the massive research work that is being performed. Researches can be categorized in the following categories:

- Prediction of energy consumptions;
- Energy profiling;
- Fault detection and diagnosis;
- Anomaly detection and diagnosis;
- Benchmarking;
- Study on occupant behavior.

In this framework the focus will be just on energy profiling, fault and anomaly detection and diagnostics.

Energy profiling Energy profiling employs machine learning algorithms to perform the characterization of energy trend and patterns in time, or load profiling. Automatically detect those patterns allows enhancing not only building management strategies but even grid operation and reliability in smart city context

[15]. Energy profiling has been successfully employed in grid management, customer classification [4] and tariff definition for the energy market as reported in [9].

The objects of the analysis are usually high dimensional time series representing energy consumption either at the whole-building level and system level. The most used dimensionality reduction for time series handling is Symbolic Aggregate approXimation (SAX) which permits to convert the numeric time series into a symbolic alphabetic string in which it is possible to recognise frequent and unfrequent patterns, called respectively *motifs* and *discords*.

A top-down automated procedure was proposed by [31] to discover and filter *motifs* in whole-building energy consumption, through SAX dimensionality reduction, clustering and effective visualisation.

Fault Detection and Diagnosis FDD data mining techniques are specifically designed to recognise and detect infrequent patterns. In literature, two types of FDD approach can be identified: component level (bottom-up) or whole building level (top-down).

Bottom-up approaches are more effective to find the root cause of anomalies since the analysis is performed on component level, excluding complex relationships with other building systems or external variable. Many FDD applications have been performed on system-level in particular on chiller [17] and HVAC systems [30]. ARM is widely used as a post-mining method to discover infrequent patterns [17], [22] creating a specific system-level FDD enhancing operation and changing conventional inefficient management strategies. Association Rule Mining (ARM) is particularly suitable in analysing large database; an example is the rules extraction to reduce energy wastes in academic spaces lighting [20].

Top-down approaches allow finding components faults by analysing the whole building energy consumption. This is a very challenging task since many variables must be considered, and often it fails to capture particular patterns on system-detail data. Very few studies are performed on aggregated loads with a top-down approach for discovering component or system faults.

Anomaly Detection and Diagnosis While frequent patterns and loads are usually the focus of energy profiling, infrequent patterns or anomalies are usually filtered out. Anomaly detection as an outlier identification process is used in [3] where abnormal energy profiles are filtered out through a clustering process in order to create an accurate prediction model. Many studies were performed

to discover similarities and anomalies in consumption patterns; for example [7] introduced different similarity measures in order to rank anomalies.

In the context of ADD, techniques are specifically designed to recognize and detect infrequent patterns at meter-level. ADD methods to analyse energy consumption time series are various. A classical statistical approach is proposed in [12] where control chart is used to detect anomalies in energy consumption time series. More advanced machine learning approach is reported in [25] where joint use of SAX and ARM is employed to find deviant events of multivariate time series in a production line.

1.2 Novelty

This work proposes an EIS tool to perform an ADD analysis by exploiting meter-level data in order to support the prompt detecting of possible anomalies and inefficiencies. In this thesis is presented an innovative top-down ADD methodology conceived for working in streaming which allows the automatic detection of anomalies at whole building level and performs a diagnosis to evaluate the sub-load responsible for the anomalies detected.The field of investigation was considered very interesting since very few researches have been performed, and the potential energy savings that could derive from a robust method is relevant. The created methodology novelty can be summarized as follows:

- It allows avoiding the computational burden of analyzing each data stream from sensors since it is a multi-level approach. In fact, at a higher level, it analyses continuously exclusively the aggregated building electrical load, and only if the overall consumption does not match the expected one, it analyses the lower level sub-loads data. In this sense it is a top-down ADD approach created with the objective of analyze meter level electrical load and perform a diagnosis of sub-loads anomalies;

- It enhances classical time series mining processes by extracting from the time series multiple aggregated features. Not only information related to the mean value, but always the information about the trend (trend angle) are extracted;

- It combines multiple advanced data mining techniques (association rule mining, globally optimum classification models, classification and regression trees, adaptive symbolic approximation)

- It is an accurate, robust, scalable and automatic process can be integrated into existing energy management systems. Moreover, its multi-level structure allows reducing the number of alerts to only the relevant cases (only for higher electrical load).

The rest of the thesis is organized as follows. In Chapter 2 an overview of the data mining methods used is presented. Chapter 3 presents the case study on which the methodology is tested and results obtained. In Chapter 4 is described the methodology proposed. Finally Chapter 5 and Chapter 6 presents respectively the results obtained and a critical insight, concluding with possible improvements and further developments.

Chapter 2

Data analysis methods

The knowledge extraction from building-related data is usually performed by domain experts, which based on their experience can interpret relations among data and extract useful information about the building performance and energy use. However, the ever-increasing amount of data makes more and more challenging the actual analysis from domain experts. In particular, the traditional statistical techniques often fails to effectively analyze and exploit the great potential that lies behind the systematic analysis of those data. For this reason, the analysis of building-related data is performed through Data Mining (DM) techniques. Just like any analysis algorithm, the right choice depends on the application. In building data analytics, the most used algorithms are clustering, association rules and classification trees.

In the following sections, the data mining methods employed are reviewed under a theoretical point of view. The methods description is not intended to be exhaustive, but it is aimed to underline the usefulness in the framework of the study and building energy data exploitation.

2.1 Dimensionality reduction

Meter-level data are collected in so-called time series: a two-dimensional matrix where each row correspond to a single observation in time and is composed by one column containing the time and another containing the value of a given variable. The sampling frequency determines the time interval between two consecutive observations and for building application, it is usually in the order of minutes. The resulting high-dimensional time series is computationally expensive to store and almost unfeasible to analyse directly. Many dimensionality

reduction techniques were proposed in the literature; one of the most promising is the SAX and in particular its implementation ASAX. In the following sections those two will be presented.

2.1.1 Symbolic Aggregate approXimation

The Symbolic Aggregate approXimation (SAX) is a dimensionality reduction technique that allows time series compression while preserving its fundamental characteristics; this technique it was firstly introduced by [33]. This process discretizes the original time series in sub-sequences, each of them is then converted into alphabetic symbols through an encoding process, and finally combined into a string. The resulting string is much shorter than the original time series and enables various data mining techniques while reducing the computational cost. The SAX process is summarized in the following paragraphs.

Standardization

This process is the first and fundamental preprocessing step of the time series analysis because it allows the algorithm to focus on the structural features of the time-series instead of the amplitude-driven ones. A given time series $y(t) = \{y_1, \dots y_n\}$ of length n with mean μ and standard deviation σ is transformed into a new time series $Z(t) = \{Z_1, \dots Z_n\}$ of length n with zero mean $\mu = 0$ and unitary standard deviation $\sigma = 1$ through the equation (2.1).

$$Z(t) = \frac{y(t) - \mu}{\sigma} \tag{2.1}$$

This process allows simplifying the analysis through the use of Z-scores, which is a measure of the position of data and represents how many standard deviations it is far from the mean of a standard normal distribution $N(0,1)$. If the value is positive, the value lies above the mean; if negative it lies below. Z-scores allows to easily calculate the area under a normal Gaussian distribution and will be useful when dividing the distribution into equally probable areas.

Chunking

The standardized time series $Z(t) = \{Z_1, \dots Z_n\}$ of length n is divided into N non-overlapping sub-sequences, or chunks, $T = \{T_1, \dots T_N\}$ whose length is chosen on the specific context. Each sub-sequence is further divided into W segments called time windows $\tau = \{\tau_1, \dots \tau_W\}$. The parameter W is also called

word size. During this process, it is possible to choose time windows with equal or different length, based on user preference. The tuning of the length of time windows can be performed with different machine learning algorithms and can be useful when the time series presents within the chunk different trends.

Feature extraction

In this process an aggregated numerical feature is calculated in the generic time window τ_i and this value is taken as representative of the data contained in it. Extracted aggregated features tends to underline some aspect of the time series while losing some other information. The analyst choses which feature is the most significant and whether one or more features are needed for the purpose of the study. The most used and known is the Piecewise Aggregate Approximation (PAA) introduced by [10] which performs a constant approximation of the time series $Z(t)$ by replacing the values that fall into the same time window τ_i with their mean . This is the feature extracted in the classic SAX process.

Encoding

The encoding consists in setting an alphabet size (α) and assigning an alphabetic character to each time window, according to where the extracted numerical feature lies within a set of vertical breakpoints $\beta = \{\beta_1, \dots \beta_{\alpha-1}\}$ identified according to the feature distribution shape. These breakpoints are calculated in Z-score, according to the alphabet size and under the hypothesis that the time series can be approximated as Gaussian distribution. If so, it is possible to divide the area below the distribution into equiprobable regions, creating a breakpoints table or lookup table (see Table 2.1). Finally, the encoding can be assigned for each window τ, creating a word of length W for the chunk N. The original numerical time series $y(t)$ of length n is then transformed into a alphabetic string $Z(\alpha)$ of length $W * N$.

In Figure 2.1 an example of SAX is reported. A standarized time series $Z(t)$ with $n = 192$, shown in black, is divided into two chunks T_i and T_{i+1} of 24 h each. In this example, five time windows ($W = 5$) of equal length are identified for each chunk and the alphabet size is set to five ($\alpha = 5$), meaning that four breakpoints β are identified through the lookup Table 2.1. The Gaussian distribution is shown on the right side of the Figure in ligth blue and the SAX breakpoints in dashed blue lines. The time series is then approximated through PAA (red segments), and for each segment, the corresponding symbol is assigned. The original time series for the time window T_{i+1} is converted from a numerical

Table 2.1: Breakpoints or lookup table according to alphabet size.

β	α			
	3	4	5	6
β_1	−0.43	−0.67	−0.84	−0.97
β_2	0.43	0.00	−0.25	−0.43
β_3		0.67	0.25	0.00
β_4			0.84	0.43
β_5				0.97

vector into an alphabetic string "adecb", reducing it from a 96-dimensional object in a 4-dimensional one.

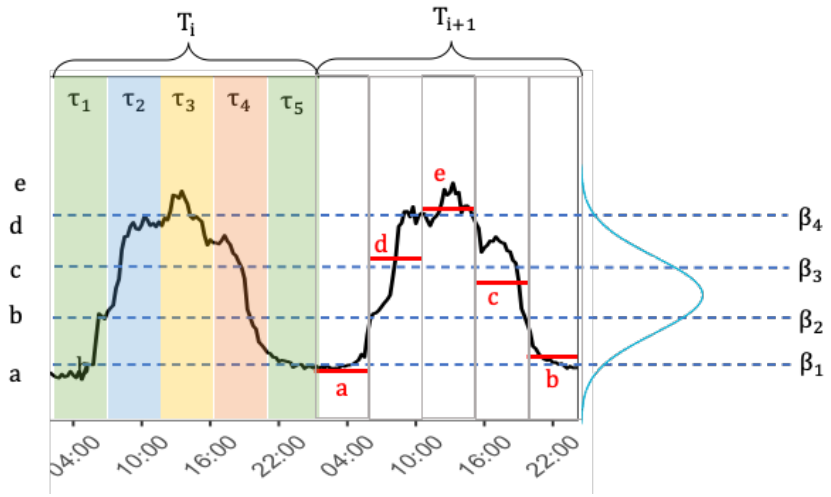


Figure 2.1: Example of SAX process applied on a standardized time series $Z(t)$. The parameters used are $T = 24\,\text{h}$, $W = 5$, $\alpha = 5$.

2.1.2 Adaptive Symbolic Aggregate approXimation

The Adaptive Symbolic Aggregate approXimation (ASAX) algorithm is an implementation of the original SAX, and was firstly introduced in [6]. The main difference is that the breakpoints identification based on the hypotheses of equally probable regions of Gaussian distribution is rejected; this permits ASAX to handle distributions different from standard normal which is the great limit of the original method.

In this algorithm no standardization is needed, and the first step is the chunking as describes in the previous paragraph. The feature extraction is performed and then follows the encoding. The key difference between this algorithm and SAX lies in the breakpoints identification. This process is handled through an adaptive method, based on K-means clustering [6]. The iterative algorithm aims to find the distribution partition (i.e. breakpoints) that minimizes the clusters total representation error, which is the objective function of the K-means.

In the following the steps of the algorithm are explained. Since it is an iterative process, the generic iteration will be labeled with the index subscript j. Is assumed that the feature extracted is the mean value for each time window through a PAA process.

Given the PAA array representation of the original time series, we denote by x_n the generic n^{th} PAA value. The starting point is the definition of the alphabet size α, which correspond to the number of clusters K, and the initial breakpoints $\beta_i^{(0)}$ with $i = \{0, \ldots, \alpha - 1\}$, $\beta_0^{(0)} = -\infty$ and $\beta_\alpha^{(0)} = +\infty$. Those breakpoints are calculated with the hypotheses of normal distribution and divide the distribution into equally probable regions; they represent the initialized starting point.

At each iteration j, the centroid $c_i^{(j)}$ between two consecutive breakpoints $[\beta_i^{(j-1)}, \beta_{i+1}^{(j-1)})$ is calculated as the center of mass of all N_i PAA points that fall between them.

$$c_i^{(j)} = \frac{1}{N_i} \sum_{x \in [\beta_i^{(j-1)}, \beta_{i+1}^{(j-1)}]} x_n \tag{2.2}$$

Then the new breakpoints $\beta_i^{(j)}$ are moved to the mean value between two consecutive centroids $c_i^{(j)}$ and $c_{i+1}^{(j)}$.

$$\beta_i^{(j)} = \frac{c_i^{(j)} + c_{i+1}^{(j)}}{2} \tag{2.3}$$

The total representation error Residual Sum of Squares (RSS) is calculated as follows:

$$RSS = \sum_{i=1}^{K} \sum_{x \in [\beta_i^{(j-1)}, \beta_{i+1}^{(j-1)}]} (x_n - c_i^{(j)})^2 \tag{2.4}$$

Then the relative error $\epsilon^{(j)}$ of the RSS is compared with a user-defined tolerance $\bar{\epsilon}$. If the condition 2.5 is satisfied the algorithm keeps running otherwise it stops.

$$\epsilon^{(j)} = \frac{RSS^{(j-1)} - RSS^{(j)}}{RSS^{(j-1)}} > \bar{\epsilon} \tag{2.5}$$

In Figure 2.1 an example of ASAX is reported. A time series $y(t)$ with $n = 192$, shown in black, is divided into two chunks T_i and T_{i+1} of 24 h each. In this example, five time windows $W = 5$) of inequal length are identified for each chunk and the alphabet size is set to five ($\alpha = 5$), meaning that four breakpoints β are identified. The time series distribution is shown on the right side of the Figure in red and the ASAX breakpoints in dashed blue lines. Looking at the distribution is easy to understand how classic SAX would result in a consistent loss of information by assuming a normal Gaussian distribution. The time series is then approximated through PAA (red segments), and for each segment, the corresponding symbol is assigned. The original time series for the time window T_{i+1} is converted from a numerical vector into an alphabetic string "abdca", reducing it from a 96-dimensional object in a 4-dimensional one.

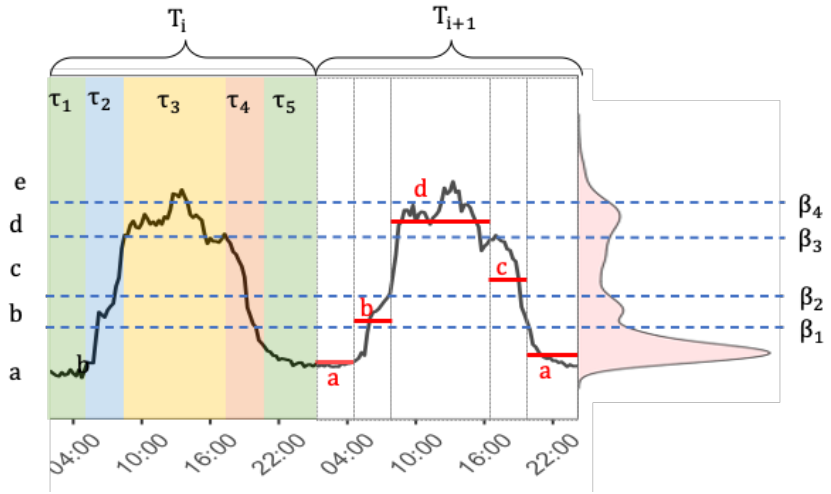


Figure 2.2: Example of ASAX process applied on a time series $y(t)$. The parameters used are $T = 24$ h, $W = 5$, $\alpha = 5$.

Focus on feature extraction

One of the steps of SAX is the choice of the aggregation feature to be extracted and encoded. As already said, this feature has to be carefully chosen, and the choice strongly depends on which aspect of the time series the analyst want to underline. The most used and known is the PAA but many other statistical features can be extracted (variance, kurtosis, skewness) not only from the time domain but even from other domains such the frequency one [21]. Others features representing essential characteristics of time series can be worth to be extracted; one

of these is the trend angle [24]. This feature is particularly effective in describing the time series trend, and it will be used in this work. Given a time series $y(t) = \{y_1, \dots y_n\}$ of length n in a given time window $\tau = \{\tau_1, \dots \tau_W\}$, defined $\Delta p(t_1)$ and $\Delta p(t_n)$ the first order distance between the initial and final point with the time series mean $\bar{y}$, it is possible to define a trend triangle and trend angle as shown as in Figure 2.3.

1. $\theta < 0$ the trend is negative;
2. $\theta < 0$ the trend is positive;
3. $\theta \approx 0$ the trend is almost stable.

In the context of building this feature could be very useful in identifying rapidly growing or decreasing electrical loads, adding a remarkable information to the analysis.

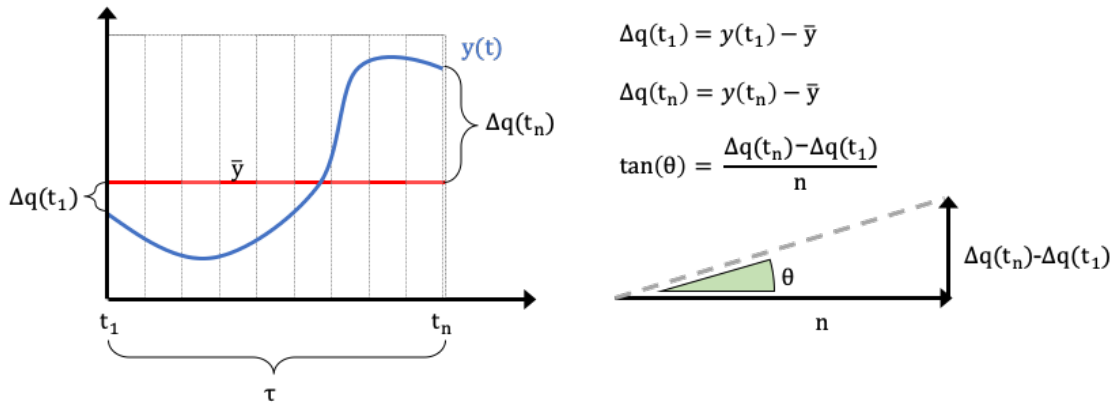


Figure 2.3: Definition of trend feature triangle and trend angle for a generic time series $y(t)$. On the left side the time series (blue) and its mean value (red) within a given time window τ, On the right side the trend triangle and the trend angle definition.

2.2 Classification

Classification is the task to assign a class label to unlabeled data instances through a classifier model, providing prediction or description of a given data set.

A general data set consists of a collection of instances or observations $D = \{d_1, \dots, d_N\}$, each of them is characterised by a set of predictor attributes x and

a target attribute or class label y. The classification model creates a relationship between the set of attributes x (input) and the class label y (output), in other words, can classify instances through the analysis of the predictive attributes. The model is created through an inductive learning algorithm using a *training set*, which is a data frame with attributes and labelled instances. Once the model is created, it is used on a *test set*, which is a data frame with attributes and unlabelled instances, in order to deduce the unknown class labels. The performance of the model can be evaluated through the comparison between the predicted labels and the real labels of the test set. A general description of the classification process is reported in Figure 2.4.

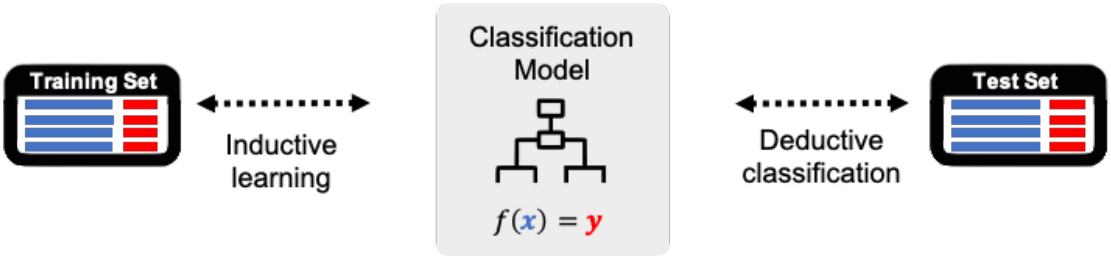


Figure 2.4: Classification process.

The tree classifier is the most commonly used classification model thanks to its understandable graphical representation, an example is shown in Figure 2.5. Depending on the type of target attribute, discrete categorical or continuous numerical, the tree is called, respectively, *classification tree* or *regression tree*. The tree consists of three kinds of nodes connected by branches:

- *Root node*: is the first node of the tree and is characterized by no incoming branches and only outgoing branches. It contains all the instances;

- *Internal node*: is characterized by one incoming branch and two outgoing branches. It contains a subset of the previous node;

- *Leaf node* (or *terminal node*): is characterized by one incoming branch and no outgoing branches. It contains a subset of the previous node, and this subset is considered satisfactory for the classification. At each leaf node is assigned a class label.

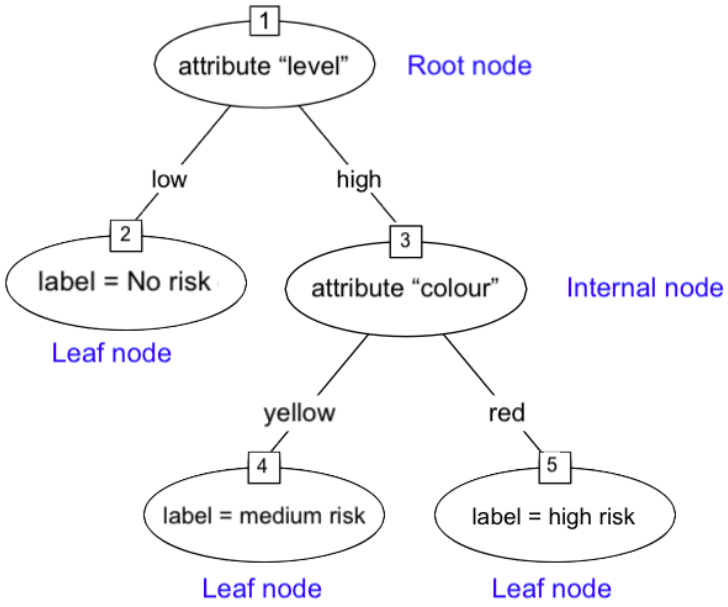


Figure 2.5: Classification tree description.

2.2.1 Recursive partitioning

The basic algorithm used to construct a decision tree is a recursive partitioning forward approach [16] which is used to create the so called Classification And Regression Trees (CART).

In the beginning, all the instances are contained in the root node. Then it is expanded by a binary split on an attribute that is chosen through an adequate splitting criterion. This process continues until a stopping criterion is satisfied. In the following paragraphs each step is described in detail.

Splitting crieterion

It is the criteria to choose the attribute test condition for the binary splitting; it decides how the instances of the parent node should be distributed into the child nodes. This criterion tends to split instances in order to create purer child nodes in which most of the instances have the same class label. This criterion tends to maximise homogeneity at each split, yielding to locally optimum split.

The impurity $I(A)$ measures how different the class labels are within the same node [29]. It can be expressed as the sum on all the classes c of a function of the

relative frequency p of instances belonging to a class i contained in node A.

$$I(A) = \sum_{i=1}^{c} f(p_{i,A}) \tag{2.6}$$

The functions, or indeces, that can be used are various, the most used are the Gini index (2.7) and the entropy (2.8). Each of them is zero if the node is pure (contains only instances from one class $p_{i,A} = 1$) and maximum if labels are equally partitioned.

$$f_{Gini} = 1 - \sum_{i=1}^{c} {p_{i,A}}^2 \tag{2.7}$$

$$f_{Entropy} = -\sum_{i=1}^{c} p_{i,A} * \log_2(p_{i,A}) \tag{2.8}$$

The variation of impurity ΔI, also known as purity-gain, between the parent and the child node is calculated to identify the best attribute condition for the split. The attribute that gives the higher impurity variation is selected.

Stopping criterion

It is the criterion chosen to stop the growth of the tree. The basic algorithm stops the growth only when the generated node has instances of the same label or the same attributes. Sometimes it is better to terminate the growth to avoid data fragmentation: when a leaf node contains a few data, and they are not enough statistically significant. Another reason for which a stopping criterion should be set is to avoid model overfitting: when the model learns the particular patterns in the test set, reducing test error, but fails to generalize or predict correctly, increasing test error. Stopping criteria are, for example, the minimum number of observation in each leaf node or the number of splits.

Next to the stopping criterion, a complexity parameter $c_p \in [0;1]$, which quantify the cost in complexity of the model when adding a new node, could be defined. By doing so, the full tree is constructed and then pruned: the higher the c_p the smallest the tree ($c_p = 1$ only root) while the lowest the c_p the largest the tree ($c_p = 0$ full tree). This parameter is calculated in the validation phase.

Validation

This phase has the goal to test the generalization performance or the ability, of the prediction model, to perform on independent data. The most used method

is the re-sampling method called k-fold Cross Validation (CV); it permits to estimate the test error and to select the appropriate level of flexibility for the model.

It divides the dataset D of size N in k folds of approximately equal size if $k = N$ this case is called Leave One Out Cross Validation (LOOCV). At each iteration, one of the folds k is selected as the test set, while the others $k - 1$ are used as train set. Once the model is trained and tested the test error is computed. This procedure is repeated for k times, and the overall error is computed as the mean of the single test errors. In this phase, is chosen the complexity parameter for which cross-validation error is minimized.

2.2.2 Globally optimum evolutionary tree

Another process that can be used to create a classification tree is the globally optimum evolutionary algorithm. The evolutionary tree algorithm is based on a stochastic algorithm that aims to construct a globally optimum classification model [18]. This process randomly initializes the root node split, then at each iteration variation operators (i.e., split, prune, major split rule mutation, minor split rule mutation, crossover) are applied. The survivor is selected, and the process repeated until stopping criterion is satisfied. The advantage of this model is that it tends to offer higher accuracy in prediction than recursive partitioning algorithms while maintaining the same interpretable tree structure.

The main peculiarity of the globally optimum classification trees lies in its stochastic nature. At each iteration the algorithm applies to the model the following variation operators:

- *Split* this operator randomly select a leaf node and assign it a random split rule that generated two child nodes;
- *Prune* this operator randomly select an internal node and prunes it removing all its child nodes from the tree;
- *Major/minor split rule mutation* this operator randomly select an internal node and modifies the splitting rule;
- *Crossover* this operator randomly select two subtrees and exchange their position.

Those variation operators are randomly applied to the model following a probability distribution set by the user.

2.3 Clustering

Clustering is the process of grouping instances of a database based on similarity within some attributes of those instances. The goal is to create groups (clusters) in which object shows remarkable similarities among them compared to objects of other clusters [5]. The similarity measure is usually defined with a mathematical formula (for example the euclidean distance), and the clustering algorithm aims to minimize this objective function. Clusterings algorithms can be categorized into *partitional* or *hierarchical*. In the first case, the observations are divided into non-overlapping subsets called clusters. The hierarchical clustering generates non-overlapping clusters and each cluster can be further divided into subclusters and so on, creating a tree structure. In the following, the partitional K-means clustering and hierarchical clustering are presented.

2.3.1 K-means clustering

K-means is a partitional prototype-based clustering. It defines the prototype as the mean point of a set of objects.

Given a dataset $D = \{x_1, \dots x_n\}$ of n instances, the user sets the number of clusters K in which the dataset has to be divided, and the initial centroid $c^{(0)} = \{c_1^{(0)}, \dots c_K^{(0)}\}$ for each cluster $C = \{C_1, \dots C_K\}$. Then each element x is assigned to the closest centroid c_i, and the points assigned to the same centroid forms a cluster C_i. The centroid of each cluster is then updated based on the points that belong to the cluster. This process continues iteratively until no points change cluster or the centroid remains the same with a certain error threshold $c^{(i-1)} \approx c^{(i)}$. In the following paragraphs each step is described in detail.

Chose initial centroid

A careful choice of the initial centroids $c^{(0)}$ and number of clusters K is the key to perform an effective clustering since the result is greatly depending on initial conditions. The centroids can be chosen by the user, can be picked randomly, chosen after multiple runs of the clustering algorithm or chosen after hierarchical clustering of a sample of point of the dataset.

Create cluster

For each element x the proximity measure to all the centroids c_i is computed, the element is then assigned to the relative cluster C_i. The proximity measure

quantifies the distance between an element and the centroid; different types of measures can be chosen regarding the type of elements to be analysed. The most widely used measure in K-means is Euclidean distance $L_2 = dist(x, c_i)$.

Update centroid

Once the proximity measure is defined the clustering algorithm has to recompute the centroid, maximizing the similarities between cluster elements by minimizing of a objective function. Given the Euclidean distance as proximity measure the Sum of the Squared Errors (SSE) can be used as objective function. It sums the error squared between an element and the closest centroid.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(x, c_i)^2 \tag{2.9}$$

Given two clusters, the best clustering is the one that has the smallest SSE because it means that the points are closer to the centroid, and this better represents the cluster. It can be demonstrated that the centroid that minimises the SSE is the mean.

In Figure 2.6 a three steps K-means clustering is performed to a simple set of instances. At first glance in the data points is possible to distinguish three natural clusters so $K = 3$ is chosen. User-defined centroids $c^{(0)}$ are represented with stars. It is possible to see how the algorithm shifts the initial centroids toward the centre of the respective cluster in the successive iterations.

2.3.2 Hierarchical clustering

As already said, hierarchical clustering consists of creating a series of nested clusters. There are two basic approaches to address to hierarchical clustering. The first is the *agglomerative clustering* that consists of starting with single point clusters and then merge the closest pair of clusters. The second is *divisive clustering* that consists of starting with one unique cluster and then split the clusters. This kind of clustering can be graphically viewed though a dendogram which shows the cluster relationships and the merging order. In the following, only the agglomerative clustering is reviewed since is the technique used in this framework.

The basic agglomerative algorithm approach consists in defining a proximity measure, compute a proximity matrix for all the instances, merge the closest clusters and update the proximity matrix until only one cluster remains. The

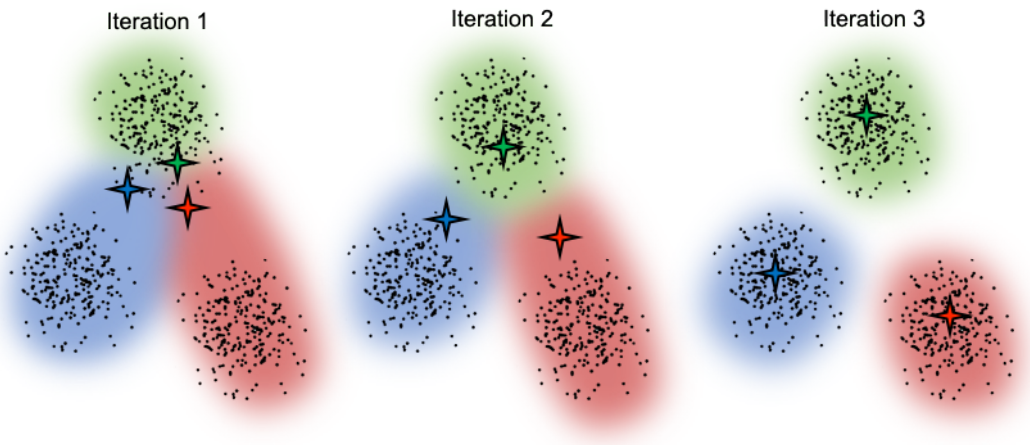


Figure 2.6: Example of K-means with $K = 3$. Adapted from [5].

key of this algorithm lies in the definition of proximity measure; there are three possibilities:

- *Single link or MIN*: the proximity is defined as the minimum distance between any of the two points of two different clusters;

- *Complete link or MAX*: the proximity is defined as the maximum distance between any of the two points of two different clusters;

- *Group average*: the proximity is defined as the average distance between all pairs of two different clusters

A simple example of single link clustering is visible in Figure 2.7. A set of six points in a 2D space are represented in the left side of the Figure. The single link agglomerative clustering defines as proximity measure the minimum distance between two points of two different clusters. The dendogram reported on the right shows that the first two points merged to create a cluster are p3-p6 and then p5-p2. The two resulting clusters are then merged, since the distance between p2 and p3 is less than the distance between any other point. Finally p4 is aggregated and followed by p1.

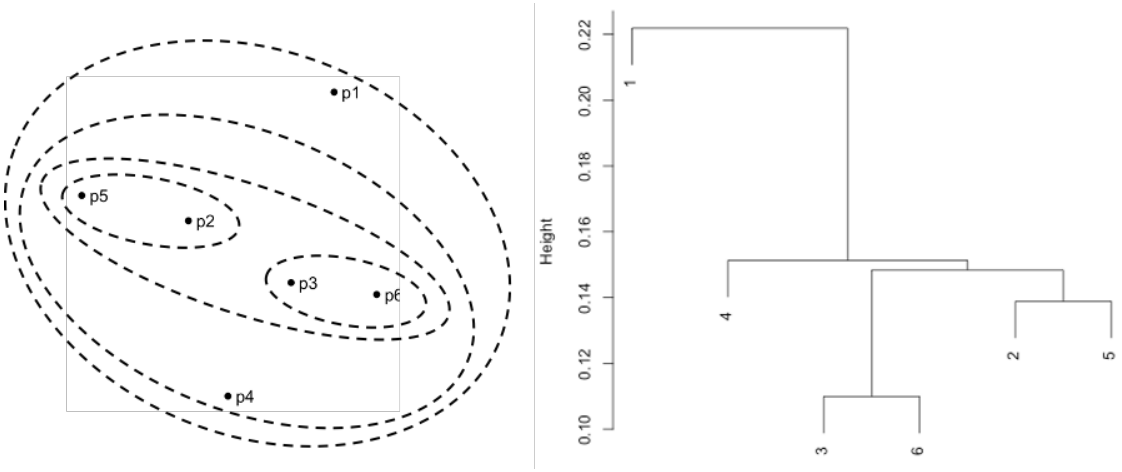


Figure 2.7: Example of single link hierarchical clustering. Adapted from [5].

2.4 Association rules mining

Association Rule Mining (ARM) is a widely used technique that allows to extract static causal relationship and correlations between attributes of a dataset; the objective is to find a group of variables (items) that frequently occur together in a database. This technique can only handle categorical variables and is computationally costly; many algorithms were created in order to optimize the task. The most used is the iterative Apriori algorithm based on frequent itemset that allows to extract static rules from a categorical transactional dataset [11]. Association rules are defined between set of items (or itemset) in the form $A \Rightarrow B$ where A is the itemset called antecedent or Left Hand Side (LHS) and B consequent Right Hand Side (RHS) and $A \cap B = \emptyset$. Rules extraction is a usually restricted to only an item in the consequent. To illustrate the concept, a small transactional database of four transactions with three categorical items $\{a, b, c\}$ is reported in Table 2.2, where an example of association rule can be $\{b, c\} \Rightarrow \{a\}$.

Table 2.2: Example of transactional database.

ID	Itemset
1	$\{a, b, c, d\}$
2	$\{a, d\}$
3	$\{d\}$
4	$\{c, d\}$

Some user-defined parameters (confidence support and lift) have to be set, in order to evaluate the significance of the obtained rule. A domain expert sets those parameters according to each particular case.

The *support* is calculated as the intersection between the antecedent A and consequence B, expressing the co-occurrence of the two events:

$$\text{supp}(A \Rightarrow B) = P(A \cap B) \tag{2.10}$$

The *confidence* is defined as the conditional probability between A and B, it gives the probability of the consequent event in all baskets containing the antecedent:

$$\text{conf}(A \Rightarrow B) = P(A|B) \tag{2.11}$$

The *lift* is the ratio between the confidence and support and gives the correlation between the conditional probability of B and the probability of B without assumptions.

$$\text{lift}(A \Rightarrow B) = \frac{P(A|B)}{P(B)} \tag{2.12}$$

When lift > 1 it means that is more probable that B is correlated with A while if lift < 1 it means negative correlations, if lift $= 1$ there is no correlation at all. This parameter is particularly important since allows to select the most interesting rules [27].

Chapter 3

Case study

The case analysed refers to the energy consumption of a medium voltage transformer station that serves a part of the main campus of Politecnico di Torino (PoliTo), one of the most important Italian technical university and is located in Turin. The central engineering campus, constructed in 1958, undergo many interventions and expansions over the year, covering in total almost 200 000 m^2. In the more recent past another area of 150 000 m^2, called Cittadella Politecnica, was added to the campus offering even more academc spaces, lecture halls, laboratories and services.

The campus is electrically fed by a loop of ten medium voltage transformer substations, which provide low voltage to the distribution system and utilities. This electrical configuration permits to reduce losses and improve performance. Each substation, identified through a letter, provides electricity to a given area of the campus. In Figure 3.1 is shown the map of the substations, while in Table 3.1 the connected facilities are listed.

Thanks to the Living Lab facility[1], this electrical network is equipped with numerous digital monitoring devices that perform real-time measurements of electrical power. Those meter level data provide information about the electrical load of a given substation. In some cases, a sub-meter level measurements permit to further detail the electrical load of a given facility.

The substation under study is the "substation C" which provides electrical energy to several facilities of the campus, for an overall served area of almost 42 000 m^2. The facilities connected to this substation are: a staff canteen (`Canteen`),

[1]This service makes available consumption monitoring data able to stimulate virtuous behavior among students and workers of Politecnico di Torino. Accessible at the following link: http://smartgreenbuilding.polito.it/panoramica/

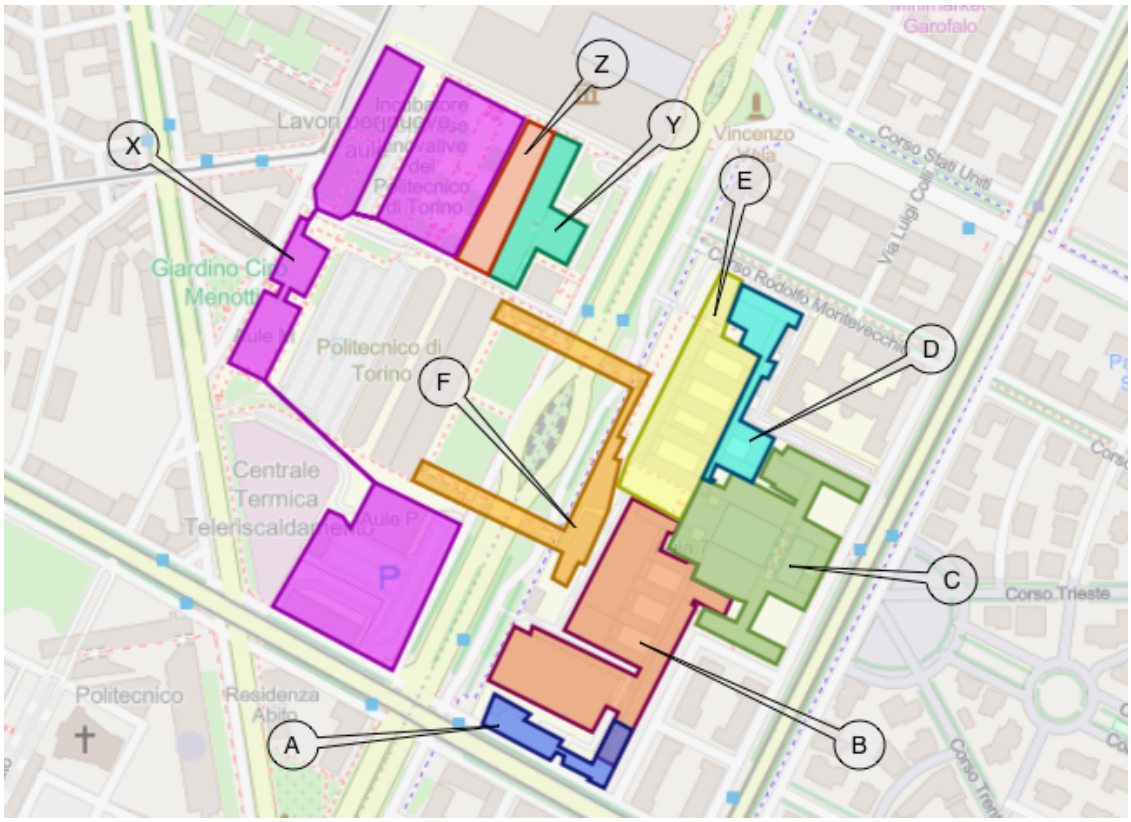


Figure 3.1: Electrical substations of PoliTo (Map data © OpenStreetMap contributors, Map layer by Esri).

the department of mathematics (`DIMAT`), the data centre (`Data centre`), a print shop (`Print shop`), a bar (`Bar Ambrogio`), the administration building (`Rectory`) and a heating/cooling mechanical room (`Refrigeration unit2`), lecture rooms and computer labs. However only some of these facilities are equipped with metering infrastructure.

Living Lab provided for the substation C a dataset almost 175 295 average power measurements (kW) related to the total electrical load and to some sub-loads. Data are available with a time-stamp of 15 minutes from 1st January 2015 to 31st December 2019, with no remarkable discontinuities. The hierarchical structure of the dataset is shown in Figure 3.2: the first level (Meter-level total load in blue) refers to the aggregated electrical load, while the second level (Meter-level sub-load in yellow) shows the breakdown among the available sub-loads. The Figure shows the breakdown of the average annual energy consumption as well, calculated on the years 2015-2019.

Table 3.1: List of facilities fed by electrical substations.

ID	Facilities
A	Department of Management and Production engineering (DIGEP), Department of Mechanical and Aerospace engineering (DIMEAS), lecture rooms.
B	Department of Energy and Nuclear engineering (DENERG), Department of Environmental and Territorial engineering (DIATI), Department of Structural and Building engineering (DISEG), Bar Denise, lecture rooms.
C	Mathematics department (DIMAT), central administration, Rectorate, central engineering library, press center, Bar "Ambrogio", staff canteen, chiller substation, data center, lecture rooms, computer labs.
D	Department of Automation and Computer Science (DAUIN), Lnguage center (CLA), Institute of Electronics and Information Engineering and Telecommunications (DET).
E	Department of Environmental and Territorial engineering (DIATI), Department of Applied Science and Technology (DISAT), lecture rooms.
F	Department of Automation and Computer Science (DAUIN) , Department of Electronics and Telecommunications (DET), secretariat..
X	lecture rooms, computer labs, offices, Start-up incubator (I3P).
Y	canteen, lecture rooms, Mario Boella institute..
Z	lecture rooms, offices.

The first level of the database provides the total electrical load of the substation. Energy consumption of those facilities that don't have a metering infrastructure, like lecture rooms and computer labs, are aggregated under a unique instance, called `Unlabeled`. This accounts for 48.85% of the total energy consumption and since it is not directly measured, cannot be assigned to a specific end use. On the other hand, facilities that have a measuring infrastructure are aggregated under the variable called `Labeled` and account for 51.24% of the total energy consumption.

Within the labeled energy it is possible to make a clear distinction on the share devoted to each sub-loads. The bar "Ambrogio" and the canteen are at disposal for students and campus personnel and account respectively for 2.75% and 15.98% of the total electrical energy consumption of substation C. The university data centre accounts for the 13.17% of the total energy consumption. The Rectory corresponds to 3.84% of energy consumption and the mathematics department (DIMAT) for 2.21%. A large share of energy consumption (12.18%) is connected to the refrigeration unit. The equipment located in the refrigeration unit room includes hot and chilled water circuits and auxiliaries such as recirculation pumps. The hot water is provided from a heat exchanger, while the chilled water is provided by two chillers of nominal electrical power of 220 kW and a rated cooling capacity of 1120 kW, and a reversible water-water heat pump, with nominal power and cooling capacity of 165 kW and 590 kW, respectively.

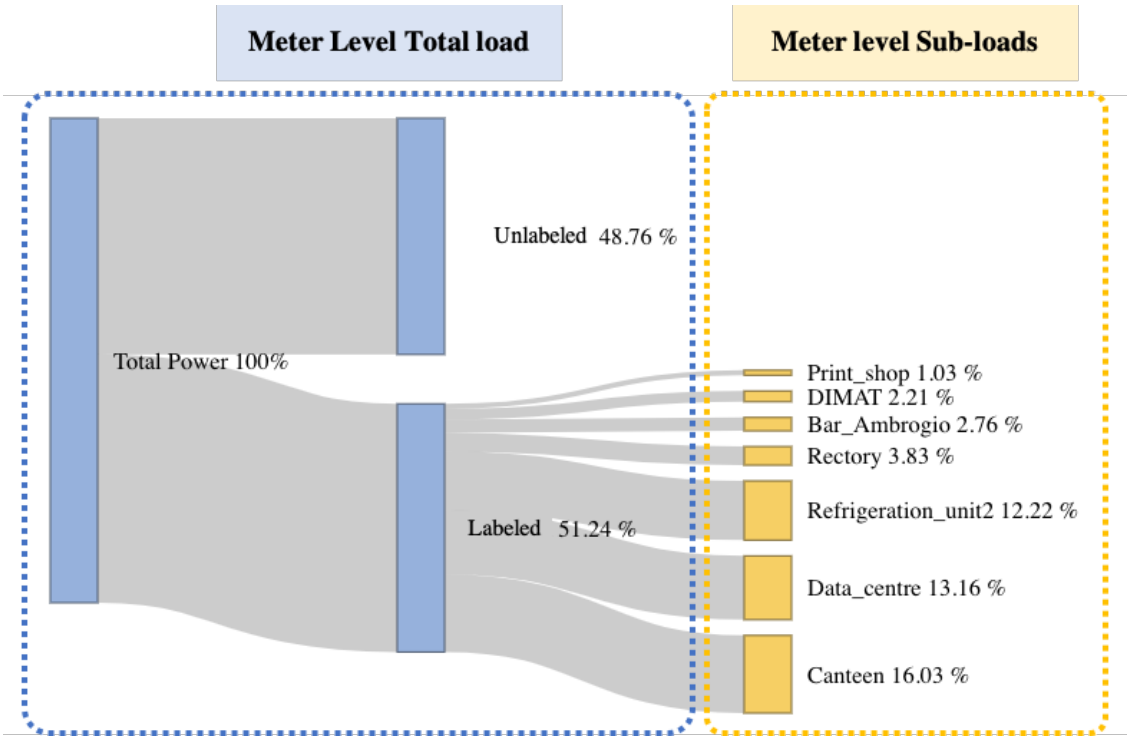


Figure 3.2: Hierarchical structure of the electrical load database under study.

During 2015-2019 equipment and energy patterns have changed, along with energy management procedures, leading to a reduction of electrical energy demand. Those changes mainly affect the reduction of unlabeled electrical consumption and refrigeration unit consumption.

An interesting analysis can be performed on the substation's operational costs associated with electricity. Costs change according to the time of use and the type of customer. In the case of PoliTo, Living Lab reports that the tariff is similar to the residential sector. Considering a mean reference value of 0.15 EUR/kWh the cost for electrical energy for the substation C in 2015 was 401 502 EUR while in 2019 was of 357 008 EUR , resulting in a reduction of 44 494.5 EUR in five years. It is evident that a continuous effort in reducing energy wastes, investing in more efficient equipment and smarter energy management procedures could abruptly reduce operating costs.

In the following sections with the use of raw data visualization of the power distributions and with the use of boxplots to identify the average hourly, weekly and monthly pattern, each database instance will be described in detail.

3.1 Unlabeled

The unlabeled load refers to an aggregated electrical load of all those facilities that are not directly measured. Services included in this load are: external and internal lighting system, circulation fans of HVAC, computers and plug loads, security systems, and elevators. Those services are at the disposal of lecture rooms, computer labs and central administration offices.

As can be seen in Figure 3.3 the power distribution has a peak at 70 kW and 270 kW , median value around 90 kW and shows a quite accentuated positive skewed tail.

On a daily base, Figure 3.4, shows a regular pattern. During night hours (21:00-5:00) the median value is $\approx$ 90 kW with low variance.

An increase of median electrical load of almost $\approx$ 160 kW from 5:00 to 10:00 takes the electrical load to a stable value of $\approx$ 250 kW.

On a weekly base, Figure 3.5, a conspicuous decrease of the median electrical load and its variance is visible during weekends, due to the weekly university break and the absence of lessons and university staff.

On a monthly base, Figure 3.6 no significant pattern is visible, the median electrical load is $\approx$ 100 kW , only in August is visible a drop of the median value and a variance reduction.

3.2 Labeled

The labeled electrical load is composed of the sum of all the meter-level sub-loads measurements. In the following paragraphs each sub-load will be described in detail.

3.2.1 Print Shop

The Print shop "Copysprinter" is a facility at disposal for student located in the first underground floor next to the university library.

The electrical load of this facility is mainly connected to printers and computers. As can be seen in Figure 3.3 the power distribution shows two peaks, one around 0 kW power absorption and the other in correspondence of $\approx$ 10 kW. The distribution shows a highly positive skewed tail.

On a daily base, Figure 3.4, shows a regular operating pattern. During closing hours (19:00-6:00) the median value is $\approx$ 0 kW meaning that all the appliances are switched off.

An increase of median electrical load from 6:00 to 9:00 takes the electrical load to a stable value of $\approx$ 6 kW. This plateau is maintained until 17:00 and then a sharp decrease until zero suggest the switching off of the appliances.

On a weekly base, Figure 3.5, a considerable decrease of the median electrical load is visible during weekends, in particular on Sundays the load is exactly zero since the print shop is completely closed.

On a monthly base, Figure 3.6 the median electrical load of $\approx$ 2 kW is almost constant, and the only sharp reduction of variance is visible in August, September and December when the university is closed due to the summer and winter holidays.

3.2.2 Mathematics department

The mathematics department (DIMAT) is located at the 3rd and 4th floor of the central building. The electrical load of this facility can be divided into lighting equipment, computers, fan coils and plug loads. As can be seen in Figure 3.3 the power distribution has a median value around 6 kW and shows a small positive skewed tail.

On a daily base, Figure 3.4, shows a regular pattern. During night hours (21:00-8:00) the median value is $\approx$ 6 kW with a low variance, an increase of median electrical load of almost $\approx$ 1.5 kW from 8:00 to 10:00 takes the electrical load to a stable value of $\approx$ 7.5 kW. This plateau is maintained until 19:00 and then a decrease until the night hours values is visible from 19:00 to 21:00.

On a weekly base, Figure 3.5, a small decrease of the median electrical load is visible during weekends, due to the weekly university break and the absence of lessons and students.

On a monthly base, Figure 3.6 the median electrical load of $\approx$ 7.5 kW is almost constant, and the only sharp decrease is visible in August when the university and all the connected services are shut down due to the summer holidays.

3.2.3 Bar Ambrogio

The bar "Ambrogio" is located at the ground floor of the central building. The electrical load of this facility is connected to all the necessary appliances used to provide a bas service to customers. Those appliances can be divided into base-load ones (refrigerators) and peak-load ones (electrical ovens, dishwasher). The total electrical load will necessary present a high variance of values and power absorption peaks. As can be seen in Figure 3.3 the power distribution has

a median value around 10 kW and shows a quite accentuated positive skewed tail.

On a daily base, Figure 3.4, shows a regular pattern. During night hours (21:00-8:00) the median value is $\approx$ 5 kW with low variance and correspond to the electrical load of appliances such as refrigerators.

An increase of median electrical load of almost $\approx$ 11 kW from 5:00 to 10:00 takes the electrical load to a stable value of $\approx$ 16 kW. This period corresponds to the opening of the bar and the coffee break. This plateau of the electrical load with high variance is maintained until 13:00 when the lunch break ends. Then a gradual decrease until the night hours values is visible from 14:00 to 21:00.

On a weekly base, Figure 3.5, a conspicuous decrease of the median electrical load is visible during weekends, due to the weekly university break and the absence of lessons and students.

On a monthly base, Figure 3.6 the median electrical load of $\approx$ 7 kW is almost constant, and the only sharp decrease of median and variance reduction is visible in August when the university and all the connected services are shut down due to the summer holidays.

3.2.4 Rectory

The Rectory contains a part of the administration offices of the university. The electrical load of this facility, as the DIMAT ones, can be divided into lighting equipment, computers, fan coils and plug loads. As can be seen in Figure 3.3 the power distribution is very similar in the shape to the DIMAT one (small positive skewed tail), but has a higher median value around 10 kW.

On a daily base, Figure 3.4, shows a regular pattern. During night hours (21:00-8:00) the median value is $\approx$ 9 kW with a low variance, an increase of median electrical load of almost $\approx$ 5 kW from 8:00 to 10:00 takes the electrical load to a stable value of $\approx$ 14 kW. This plateau is maintained until 17:00 and then a decrease until the night hours values is visible from 19:00 to 21:00.

On a weekly base, Figure 3.5, a small decrease of the median electrical load and variance reduction is visible during weekends, due to the weekly university break and the absence people working.

On a monthly base, Figure 3.6 the median electrical load of $\approx$ 11 kW is almost constant with a small increase during winter months.

3.2.5 Refrigeration unit

The so called refrigeration unit, is a cooling substation that serves the HVAC of the main building of the campus. The equipment located in this room includes hot and chilled water circuits and auxiliary water pumps used for the hot water recirculation in radiators. The hot water is provided from a heat exchanger, while the chilled water is provided by two chillers of nominal electrical power of 220 kW and a rated cooling capacity of 1120 kW, and a reversible water-water heat pump, with nominal power and cooling capacity of 165 kW and 590 kW, respectively.

As can be seen in Figure 3.3 the power distribution has an highly positively skewed distribution with a median value around $\approx$ 19 kW.

On a daily base, Figure 3.4, shows a regular pattern. During night hours (21:00-8:00) the median value is $\approx$ 0 kW with a low variance, an increase of median electrical load from 6:00 to 7:00 takes the electrical load to a stable value of $\approx$ 40 kW. This plateau is maintained until 18:00 and then a decrease until the night hours values is visible from 19:00 to 21:00.

On a weekly base, Figure 3.5, a sharp decrease of the median electrical load to 0 kW is visible during weekends, due to the weekly university break and the absence of spaces occupancy.

On a monthly base, Figure 3.6 it is possible to see that in summer the median electrical load hits its peak in July with a value of $\approx$ 100 kW. This summer behaviour is explained by the intensive use of the chillers. During winter months the electrical load is not zero because of the power absorption of the recirculation pumps.

3.2.6 Data Center

The university data centre hosts all the university servers and provides all the information technologies services to the main campus. The electrical load of this facility is mainly connected to the server base electrical load and the room chiller. This cooling system, aided by an indirect free cooling strategy, helps to avoid overheating and dusting of electronic components. This facility, for its nature, is continuously switched on and electrically fed.

As can be seen in Figure 3.3 the power distribution has an almost normal distribution with a median value around $\approx$ 36 kW.

On a daily base (Figure 3.4) and weekly base (Figure 3.5), it shows a flat pattern with median value of $\approx$ 36 kW with a low variance. This reflects the nature of the load.

On a monthly base, Figure 3.6, a seasonality increase of the median electrical load can be seen in the summer months. This is connected to the higher electrical power required from the chiller plant (especially from compressor) to cool down the data centre environment.

3.2.7 Canteen

The canteen is located at the ground floor of the central building. The electrical load of this facility is connected to all the necessary appliances used to provide a the canteen service to the university staff. Those appliances can be divided into base-load ones (refrigerators and a dedicated air handling unit) and peak-load ones (electrical ovens, dishwasher). The total electrical load will necessary present a high variance of values and power absorption peaks like the bar "Ambrogio" sub-load. As can be seen in Figure 3.3 the power distribution has a median value around 17 kW, another peak at 44 kW and shows a quite accentuated positive skewed tail.

On a daily base, Figure 3.4, shows a regular pattern. During night hours (18:00-6:00) the median value is $\approx$ 20 kW with low variance and correspond to the electrical load of appliances such as refrigerators.

An increase of median electrical load of almost $\approx$ 100 kW from 6:00 to 8:00 takes the electrical load to a stable value of $\approx$ 120 kW during the morning hours, corresponding to the dishes preparation. Another increase in the median power absorption up to $\approx$ 150 kW is present during the early afretnoon hours when the lunch is served, then a gradual decrease is visible from 14:00 to 17:00.

On a weekly base, Figure 3.5, a conspicuous decrease of the median electrical load and variance reduction is visible during weekends, due to the weekly university break and the absence of lessons and university staff.

On a monthly base, Figure 3.6 the median electrical load of $\approx$ 20 kW is almost constant, only in August is visible a drop of electrical load and its variance when the university and all the connected services are shut down due to the summer holidays.

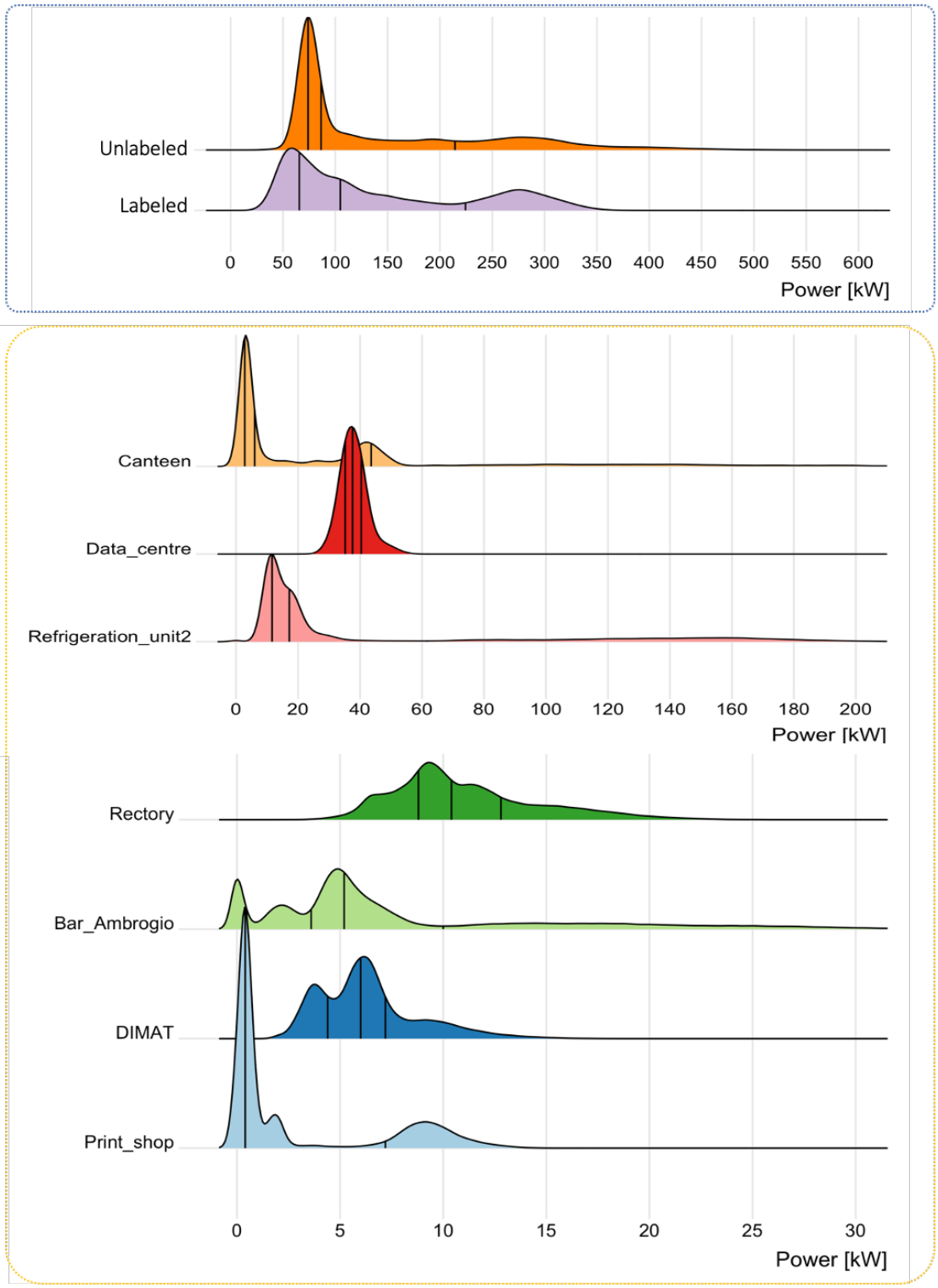


Figure 3.3: Meter-level power distributions of total loads (circled in blue) and sub-loads (circled in yellow).

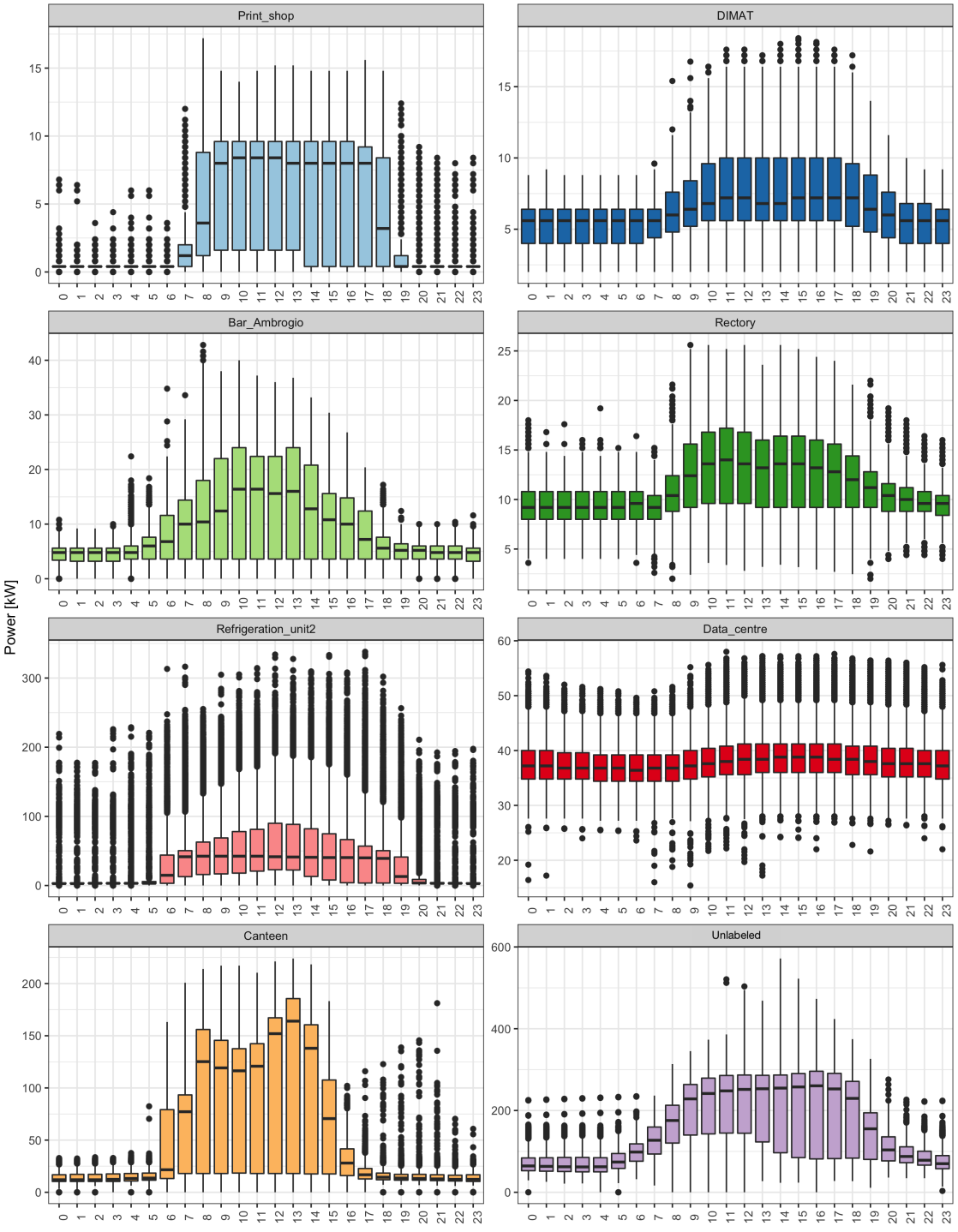


Figure 3.4: Box-plots of hourly electrical load (from 2015 to 2019) divided by sub-systems.

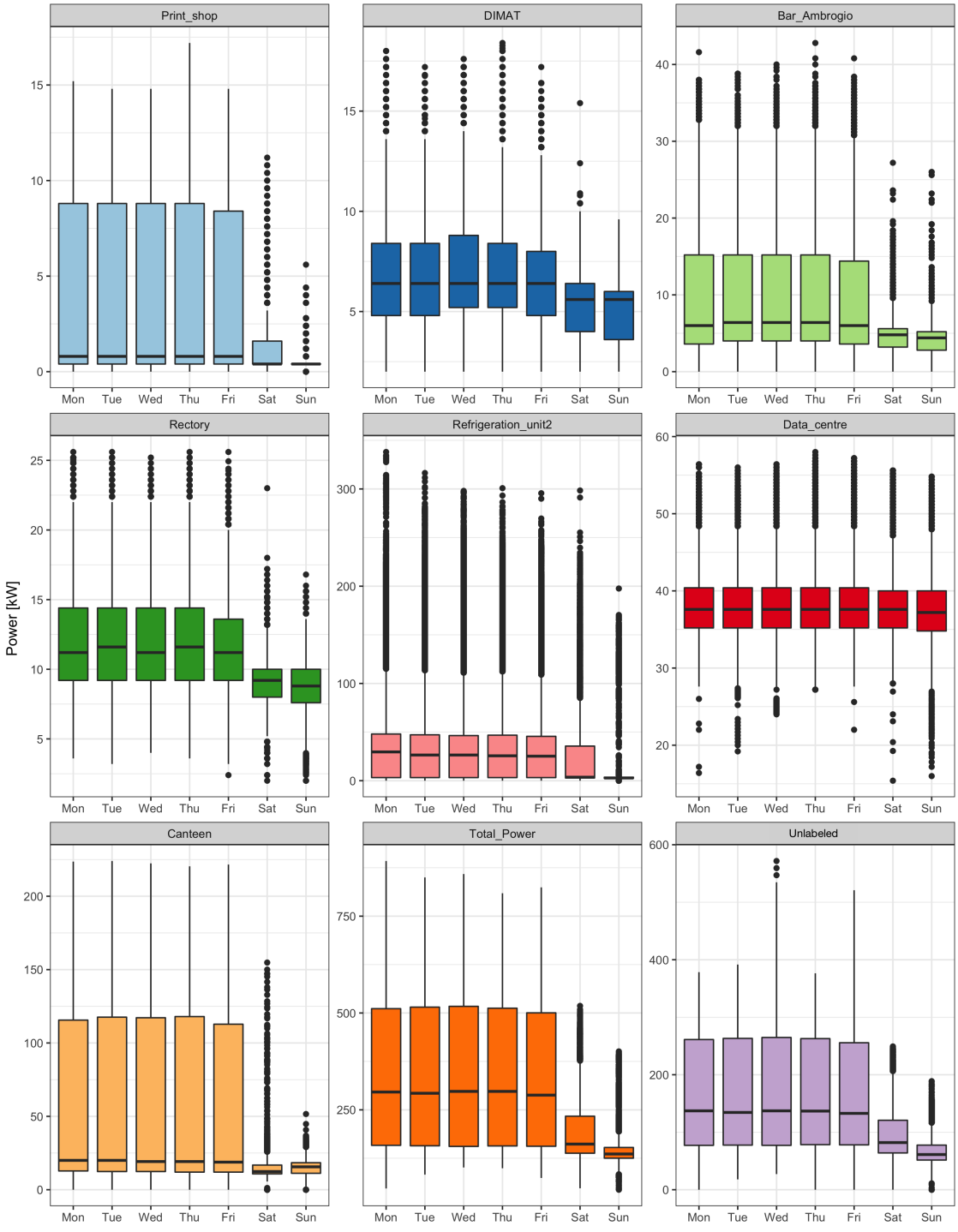


Figure 3.5: Box-plots of daily electrical load (from 2015 to 2019) divided by sub-systems.

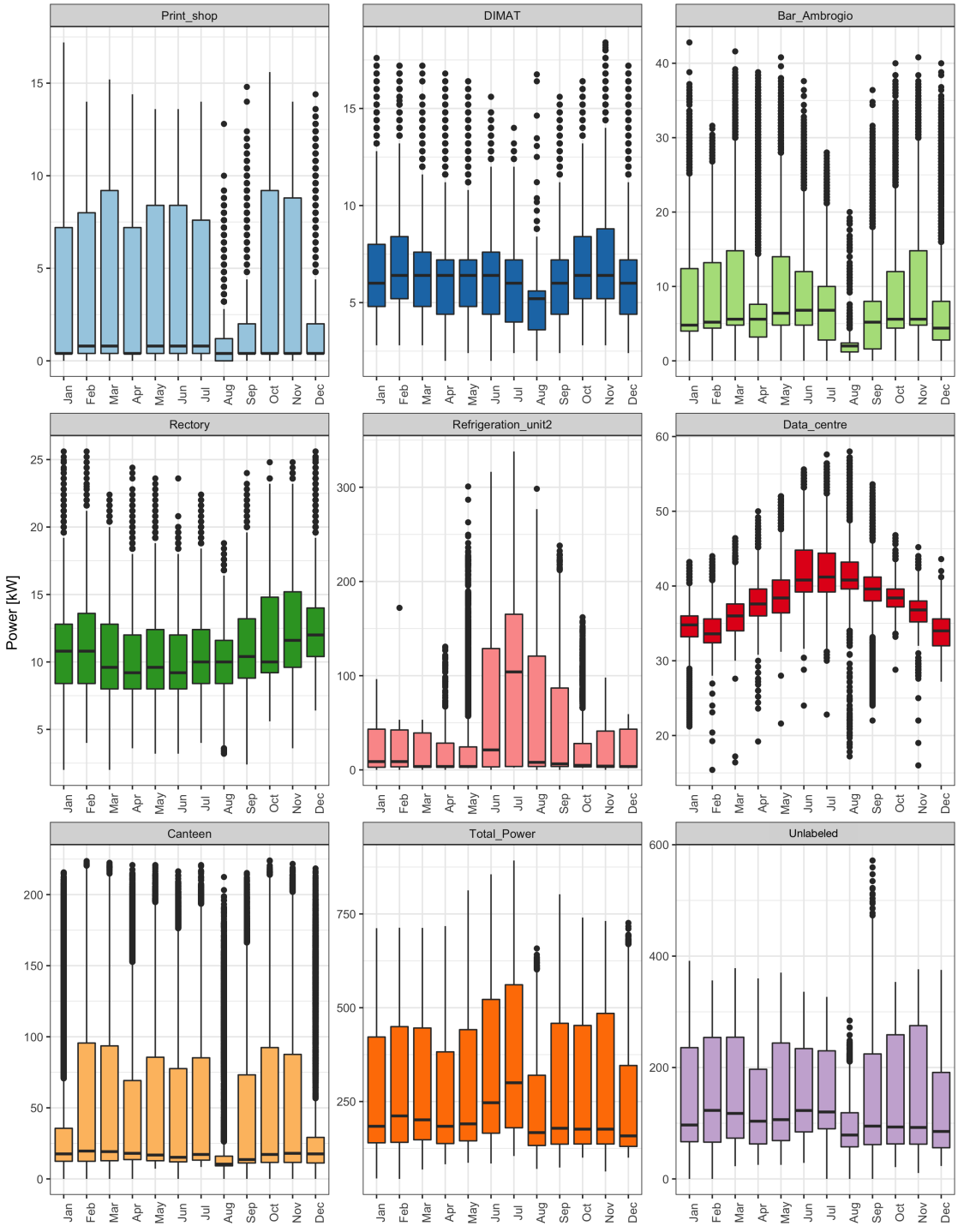


Figure 3.6: Box-plots of monthly electrical load (from 2015 to 2019) divided by sub-systems.

Chapter 4

Methodology

The proposed process aims to develop a two-level ADD methodology capable of making in a first step a high-level detection on meter level total load time series and in a second step a diagnosis on sub-loads. The methodology is based on different data mining techniques and follows the flow chart structure shown in 4.1. In the following sections, each step is described in detail.

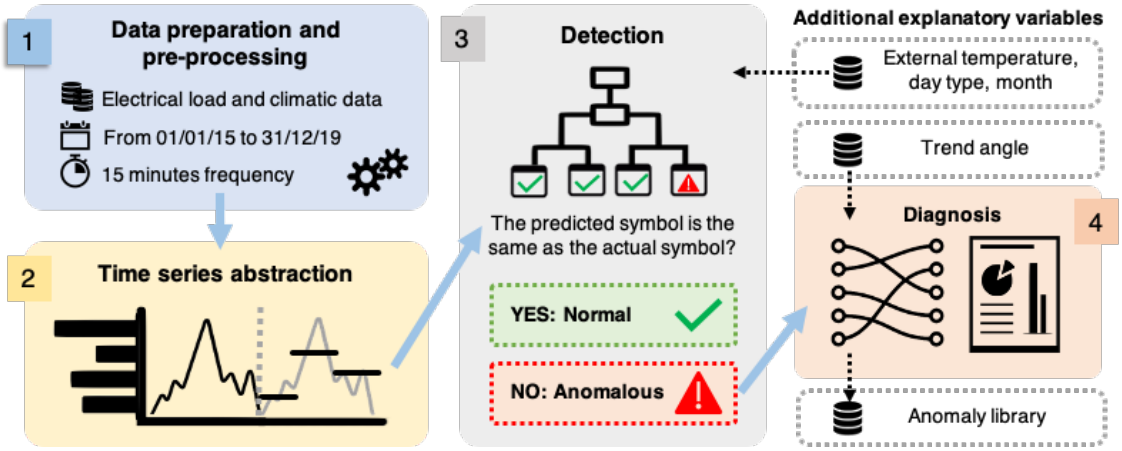


Figure 4.1: Flow chart explaining the adopted methodology.

4.1 Preprocessing

This process is crucial for the further developments of the analysis because a good pre-processing is an assurance for the robustness and accuracy of the results.

The dataset used in this study includes electrical load data (from substation C measurements) and climatic data (from PoliTo meteo station) from 1st January 2015 to 31st December 2019 with 15 min sampling frequency.

First of all, inconsistencies from the database are removed. Negative power measurements are removed a priori since they are not physically acceptable. In the case under study, all the systems are electrical loads, thus the only admitted electrical behaviour is the power absorption from the grid. Zero or nearly-zero value power measurements for loads with continuously operating systems (refrigerators, emergency lighting) were considered inconsistent and removed as well.

The second step consists of the identification and removal of outliers. In time series analysis the outliers are observations unlikely to occur given the variance of the observations of the rest of the time series [27]. They can be distinguished into punctual outliers and sequence outliers, in this phase only punctual outliers are considered. Those anomalies are usually linked to a malfunctioning of the measurement system.

An effective way to visualize variables distributions and detect outliers is the boxplot because it provides in just one visual representation a lot of quantitative information. In fact, the spacing between the parts of the boxplot indicate the degree of dispersion of data and skewness of the distribution. In particular, in the boxplot are reported:

- **First quartile** $Q_{1,(25\%)}$: the point between the smallest value and the median;
- **Second quartile or median** $Q_{2,(50\%)}$: the middle value of the dataset;
- **Third quartile** $Q_{3,(75\%)}$: the point between median and the highest value;
- **Inter-quartile distance or range** IQR: the distance between Q_2 and Q_1;
- **Minimum (*min*)** : Defined as $Q_2 - 1.5IQR$;
- **Maximum (*max*)** : Defined as $Q_3 + 1.5IQR$;
- **Outliers**: all the values that fall outside the maximum or minimum.

The first and third quartiles are joined respectively to the minimum and maximum by a line called whisker. In the following study, the boxplot used is called Turkey plot, and it is slightly different because the whisker is extended not to the maximum/minimum but to the maximum value within $1.5 * IQR$.

Moreover, boxplots allow to summarize and visualize the overall distribution and decide the tolerance band outside which outliers can be removed (see Figure 4.2). In positively skewed distributions (like electrical load) many outliers are found above the third quartile Q_3; setting a standard $1.5IQR$ band of tolerance could result in a significant loss of information. This is why only points lying outside $5IQR$ from 3rd quartile are removed. For highly skewed distribution (refrigeration unit) the threshold is set to $10IQR$. Then missing values are replaced through linear interpolation.

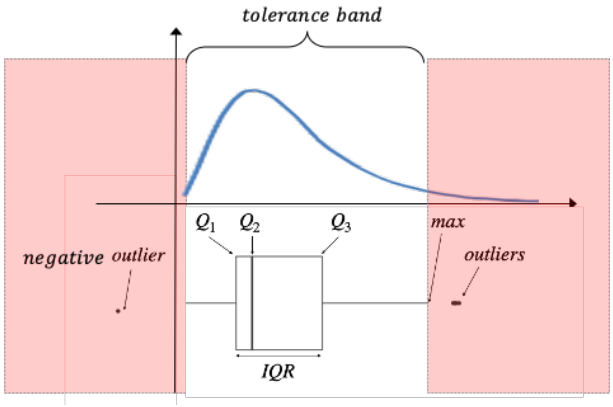


Figure 4.2: Outlier detection and handling of a positive skewed distribution.

4.2 Time series abstraction

The second step is the dimensionality reduction of the time series through the implementation of Adaptive Symbolic Aggregate approXimation (ASAX), as described in Section 2.1.2. This phase is composed of time window identification, time series reduction using PAA, breakpoints identification and encoding.

Classification And Regression Trees (CART) is used to identify unequal time window length, considering the total electrical load as a numerical target and the hours of the day as a predictive attribute. Once time windows are evaluated, the Piecewise Aggregate Approximation (PAA) approximation is performed transforming the original time series into a $N * W$ data frame. The hypotheses of equally probable regions of Gaussian distribution is rejected, and breakpoints are identified through the Adaptive Symbolic Aggregate approXimation (ASAX) procedure by choosing the appropriate alphabet size through a single linkage

hierarchical clustering process. This process is implemented through the R package NbClust [23], which allows to simultaneously evaluate the clustering quality through the use of 30 different indexes. Each index proposes a number of clusters and the optimal number is selected following a majority rule.

4.3 Detection at meter level data

Detection is performed on the aggregated electrical load of substation C, using a globally optimum classification tree. In each time window, the total electrical load symbol is predicted by the tree, which uses as explanatory variables a combination of meteorological info (as the external temperature), calendar info (day type, holiday) and energetic info (sub-loads mean electrical load).

The model is constructed through a test-train-validation process. The dataset used contains at least one year of observations 80% of data is used for training and 20% for testing. Validation is performed on another dataset, not previously employed in model construction, in order to avoid dependencies.

Given a specific time window and a particular boundary condition, the model can predict the expected symbol with high accuracy. In the resulting leaves nodes, the symbol referring to the most frequent electrical load presents a high probability of occurrence compared to all other symbols, which express infrequent behaviours. The most occurring symbol represents the “normal" operation, while the others are potential anomalies and are further investigated in the diagnosis phase.

4.4 Diagnosis at sub-meter level data

Once the classification model is created, it is possible to proceed with the post-mining phase in which the training dataset is further described in order to find historical relationships between infrequent total electrical load letters and sub-loads anomalies. The process is described in Figure 4.3.

Given the interest in detecting higher electrical load than usual, only tree’s leaves nodes that show infrequent symbols corresponding to higher electrical load are considered. Those symbols are extracted and stored in a categorical data frame. Thanks to tree’s decision rules it is possible to retrieve additional explanatory variables, from sub-loads time series, and enrich this data frame. In particular, information about the mean value and the trend angle are extracted. They are categorized through an ASAX encoder and then added to the data frame.

This data frame is then transformed in a transactional database on which Association Rule Mining (ARM) is applied. The Left Hand Side (LHS) is composed of the additional sub-loads' categorical variables, while Right Hand Side (RHS) contains only the total electrical load anomalous symbol. ARM extract a set of rules which connects the electrical load infrequent behaviour with the sub-load conditions.

Resulting rules are then reported in a scatter plot and then sorted and filtered setting appropriate interest measures parameters. Filtered rules are then stored within an anomaly library where they are ranked and clearly show which sub-loads conditions (for example high electrical load or significantly uptrend) is responsible for the aggregated electrical load behaviour. The tool gives a critical insight into the historical load behaviour and, when implemented in real-time load analysis, can give useful hints on which sub-load energy management actions are needed.

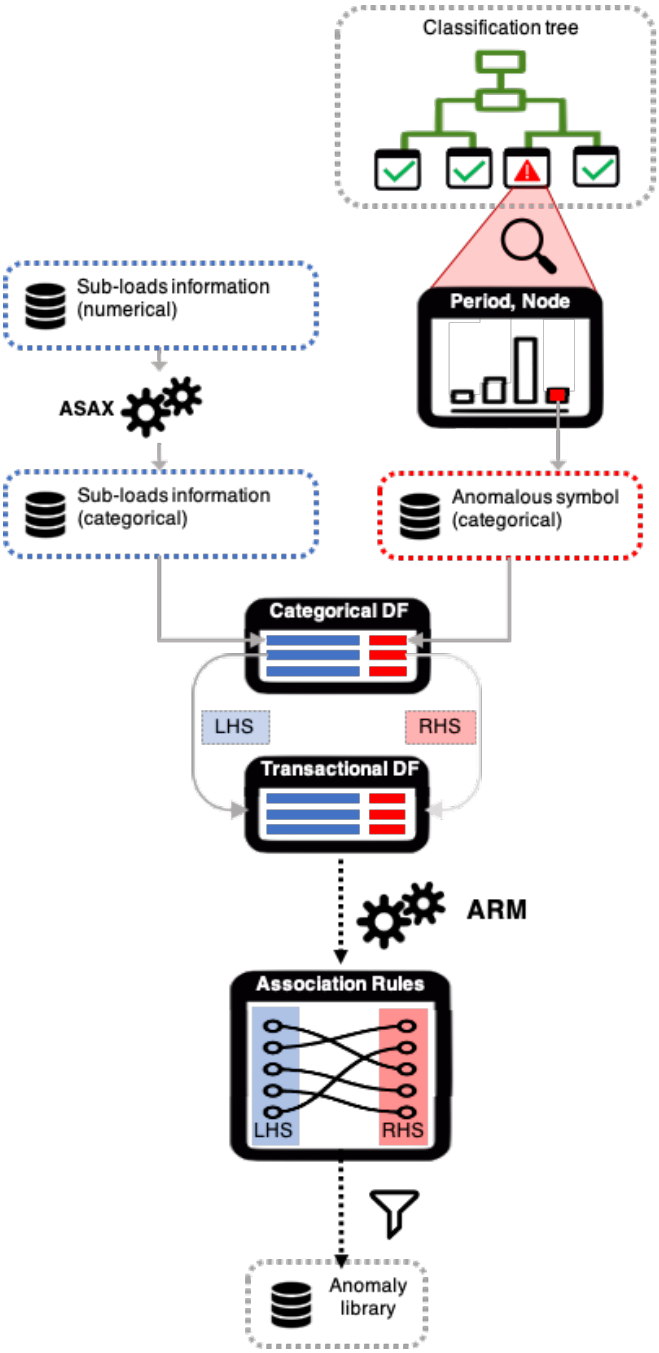


Figure 4.3: Sub-meter level diagnosis methodology description.

Chapter 5

Results

The previously described methodology was applied to the case study presented in Chapter 3. The quantitative analysis of data is performed through the open source statistical software R and results related to each stage are reported in the following sections.

5.1 Preprocessing

The pre-processing phase allowed to handle missing values and remove outliers. The procedure was applied to both electrical load and climatic dataset.

The electrical load dataset includes 175 295 observations; punctual outliers due to wrong measurements were detected (Figure 5.1), while other anomalies or missing values were not identified. Outliers were removed and filled in with linear interpolation. The total aggregated electrical load carpet plot is reported in Figure 5.2. The loads are usually turned on at 6.00 a.m. and switched off at 19.00 p.m. The electrical load increases from 8.00 a.m. until 16.00 p.m. and then starts to decrease. This hourly pattern is visible periodically every weekday, while during the weekend it can be seen a much lower electrical load. The same carpet plot representation has been constructed for all the sub-loads: Print shop in Figure 5.4, DIMAT in Figure 5.5, Bar Ambrogio in Figure 5.6, Rectory in Figure 5.7, Refrigeration unit in Figure 5.8, Data centre in Figure 5.9, Canteen in Figure 5.10, Not allocated in Figure 5.3. From those carpet plots is visible the sub-loads seasonality dependency, like in refrigeration unit2, and dependency on the academic year of student-related facilities such as canteen.

The climatic dataset contains 156 579 observations of external temperature, 42 missing values and periods of missing measurements from October-December

2015, August-November 2016 and August-September 2019.

The relative carpet plot is shown in Figure 5.11. Some extreme values of temperature were removed a priori, since unrealistic, like temperatures below −50 °C and over 50 °C. Then missing values are filled with linear interpolation.

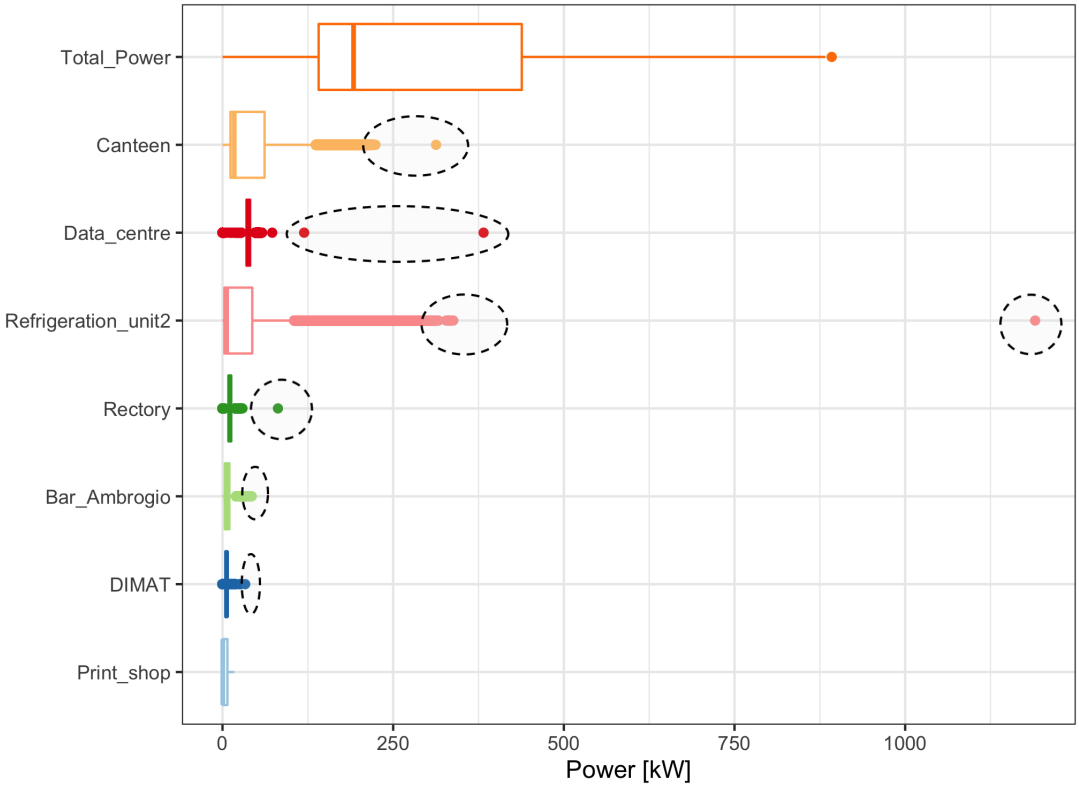


Figure 5.1: Outliers identification through boxplots.

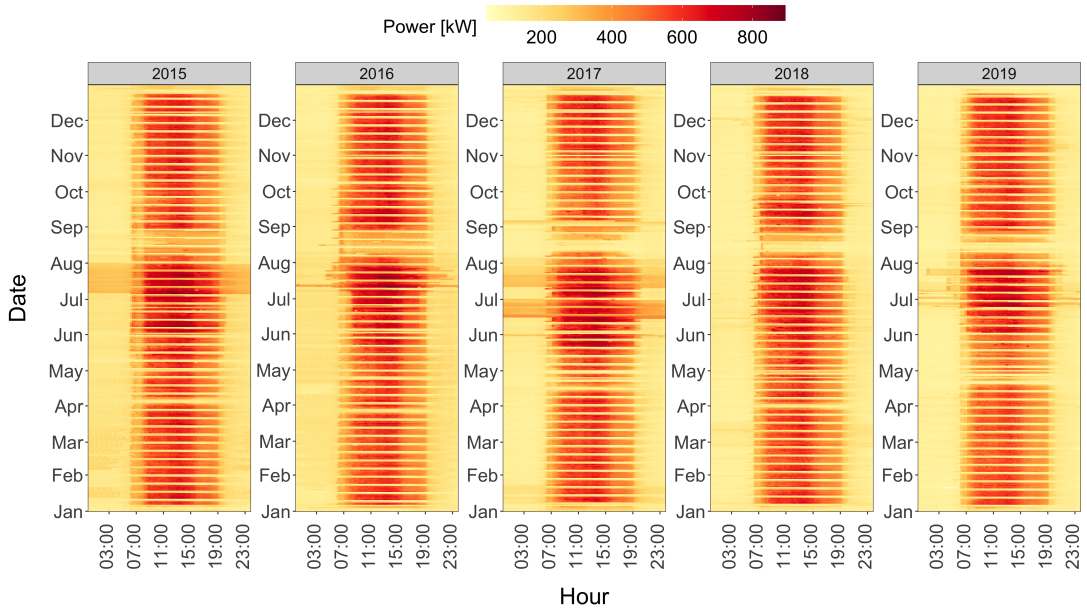


Figure 5.2: Carpet plot for the electrical load of Total Power

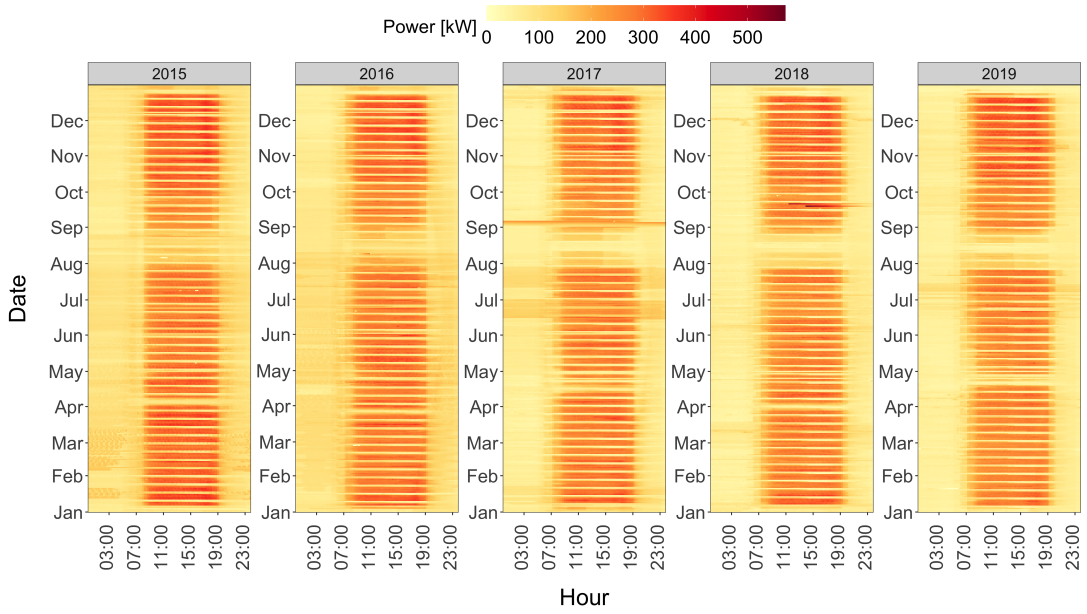


Figure 5.3: Carpet plot for the electrical load of Not allocated

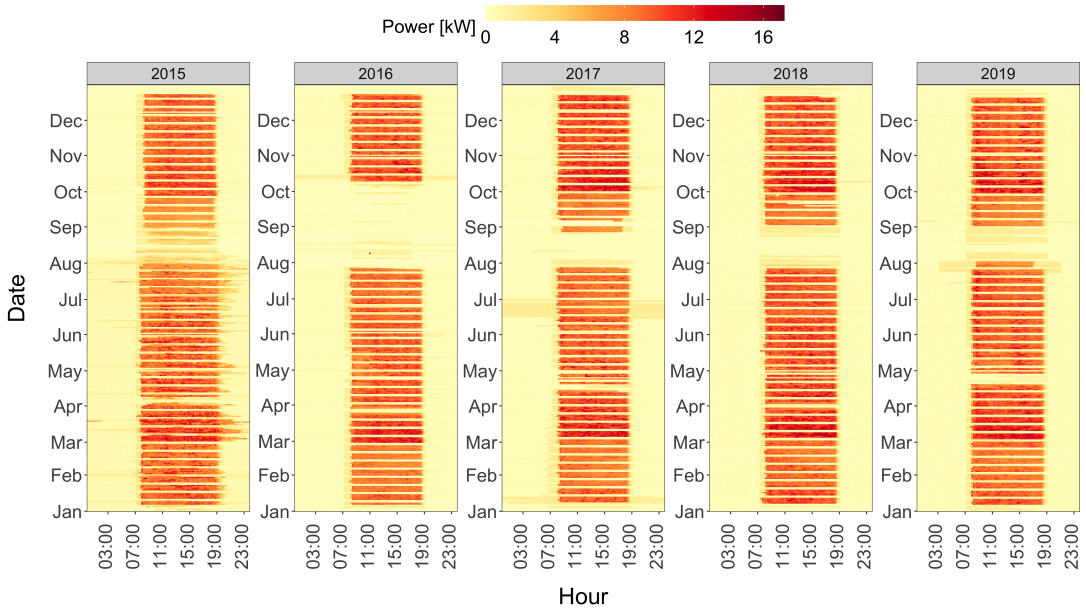


Figure 5.4: Carpet plot for the electrical load of Print Shop

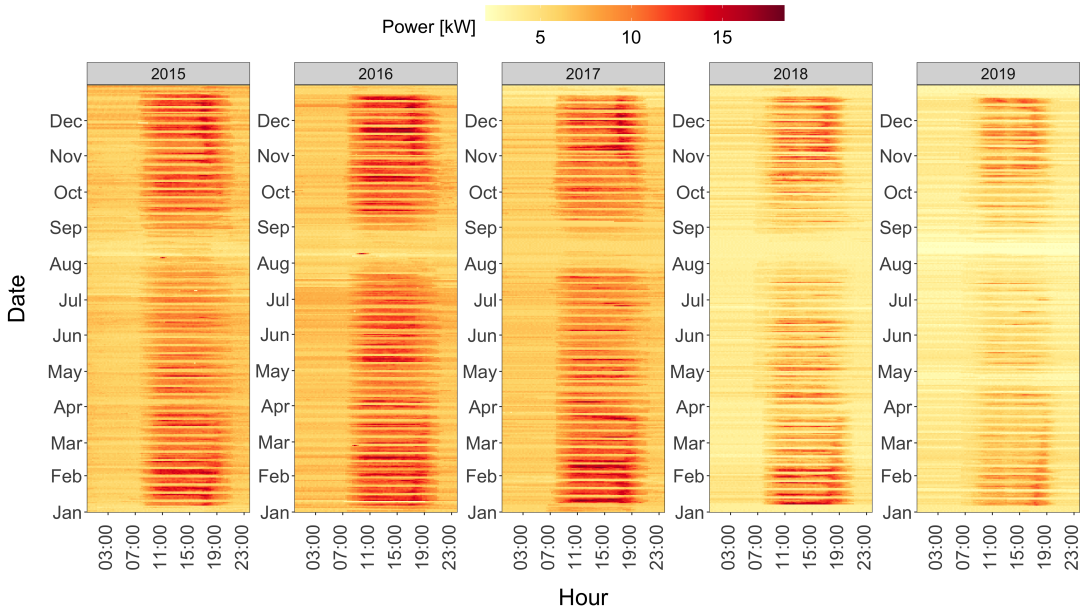


Figure 5.5: Carpet plot for the electrical load of DIMAT

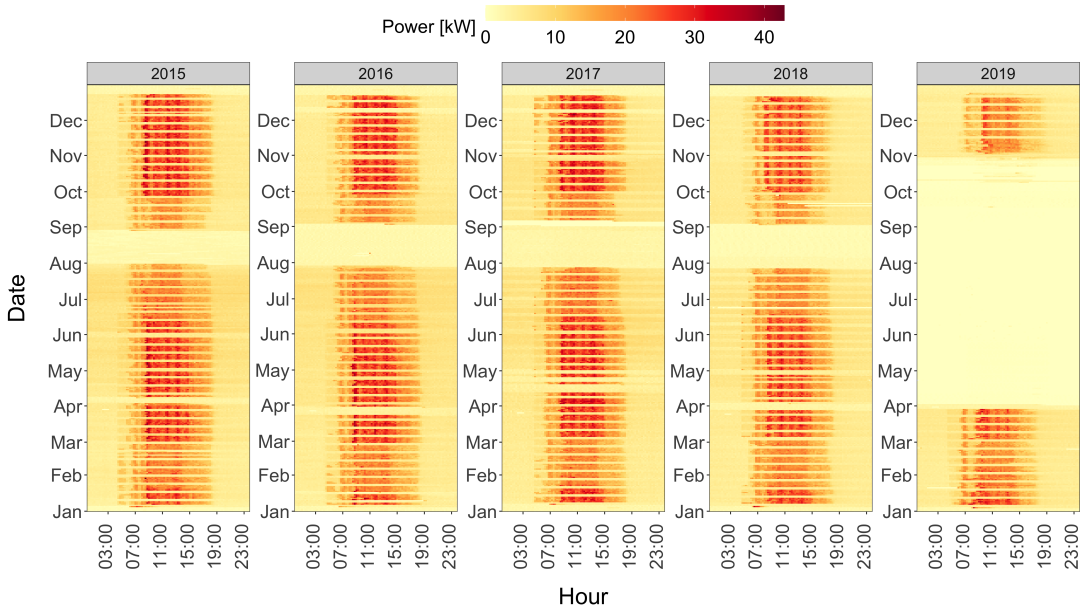


Figure 5.6: Carpet plot for the electrical load of Bar Ambrogio

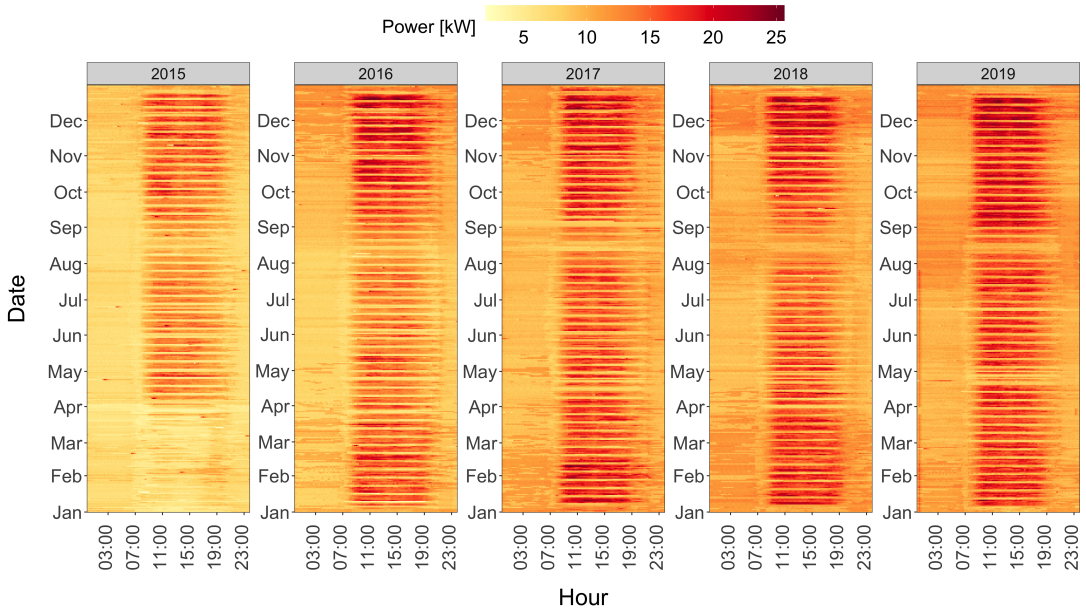


Figure 5.7: Carpet plot for the electrical load of Rectory

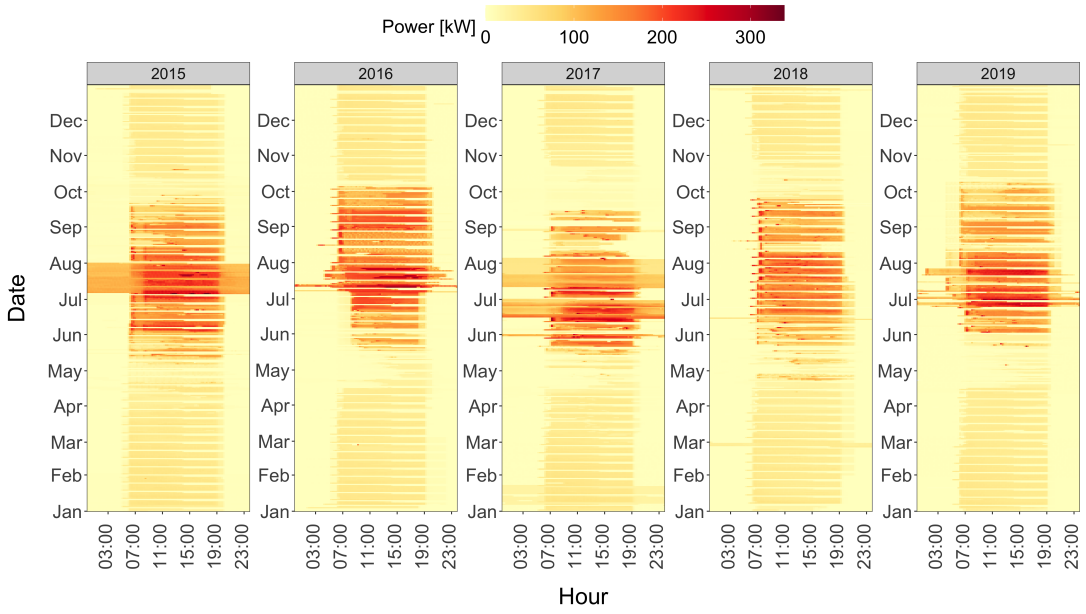


Figure 5.8: Carpet plot for the electrical load of Refrigeration unit2

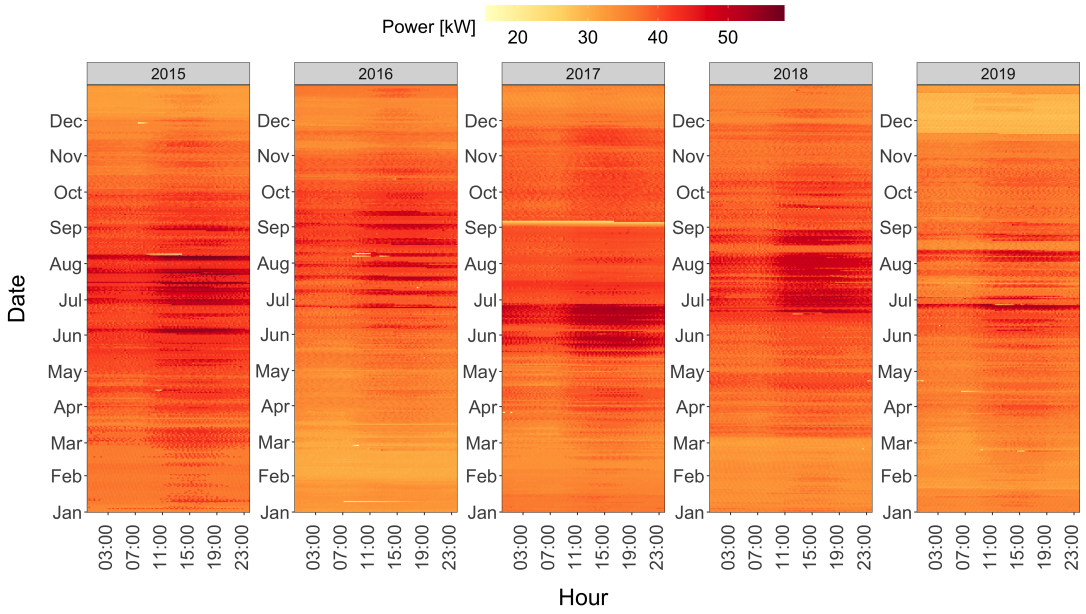


Figure 5.9: Carpet plot for the electrical load of Data centre

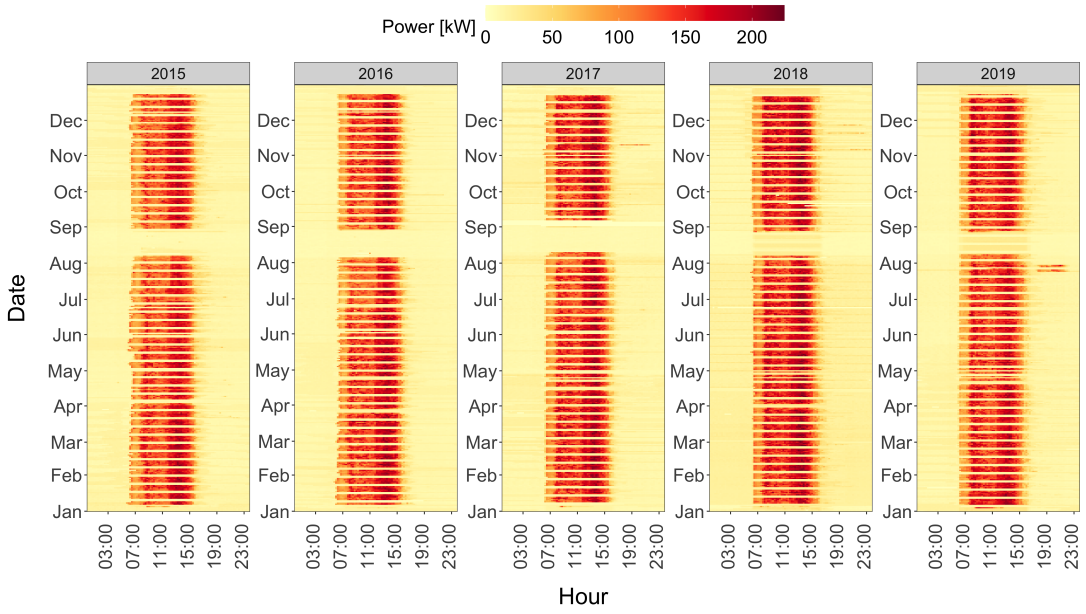


Figure 5.10: Carpet plot for the electrical load of Canteen

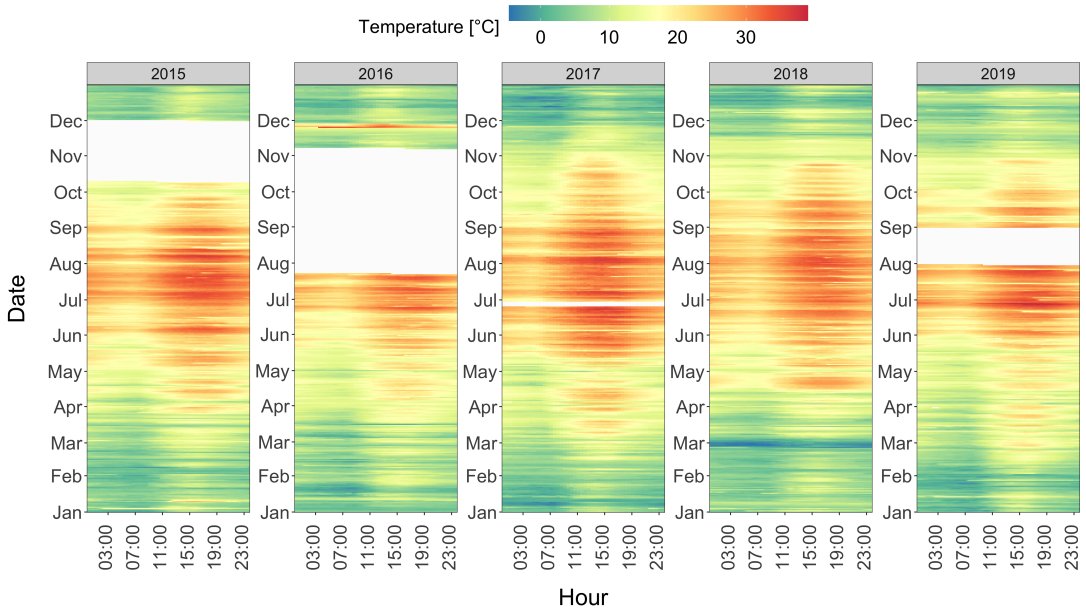


Figure 5.11: Carpet plot for the external air temperature

5.2 Time series abstraction

In order to undergo the data transformation and dimensionality reduction, the original time series of electrical load was chunked into 24 h intervals since a daily periodical pattern was observed.

The time windows of daily load profiles were evaluated though Classification And Regression Trees (CART), considering the total electrical load as a numerical target and the hours of the day as a predictive attribute. The aggregated electrical load from 2015 to 2019, was used. The resulting tree is shown in Figure 5.12.

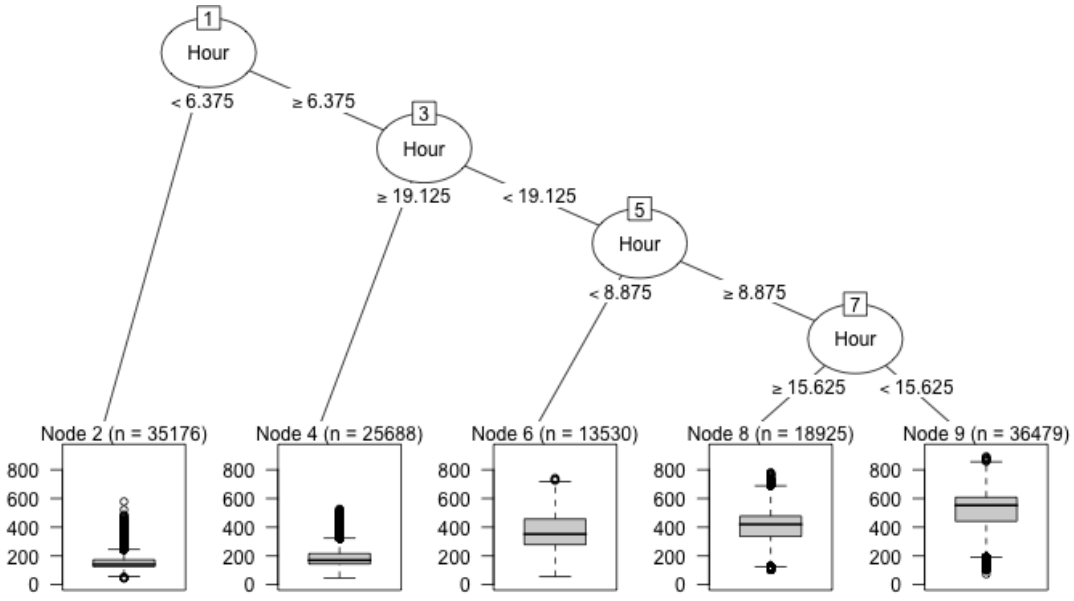


Figure 5.12: CART tree for the sub-daily time window identification

Holidays and weekends were excluded from the analysis since they usually present flat profiles with low variance and include those days in the model would have reduced the accuracy of the results. The splitting criterion adopted is based on the variance reduction around the numerical target's mean, in each leaf node. By doing so, it allows the daily pattern to be split into homogeneous consumption time windows. As a stopping criterion a minimum number of objects in the child nodes at each split was set, in order to have a time window length of 2 h, at least. The parameter expressing this condition is calculated as

follows:

$$\texttt{minbucket} = (N_{days} - N_{days,excluded}) * \frac{W_{min,length}}{W_{timestep}} \tag{5.1}$$

Where `minbucket` the minimum number of object in the child node, N_{days} the total number of daily load profiles available, $N_{days,excluded}$ the total number of daily load profiles excluded, $W_{min,length} = 120\,\text{min}$ the minimum required length of time window in minutes and $W_{timestep} = 15\,\text{min}$ the time-step corresponding to the sampling frequency. A maximum number of splits was set as an other stopping criterion ($\texttt{maxdepth} = 10$).

The regression tree automatically identifies the optimal number of windows thanks to a cost complexity pruning process. The tree grows entirely and then is pruned iteratively until the root is reached. At each step, the complexity parameter c_p was computed. This process can be seen in figure 5.13 where the complexity parameter connected to the tree size is plotted against the LOOCV error. When the tree is fully expanded c_p is zero, reducing the size (i.e. the number of leaves) c_p increases reaching the maximum when only the root is present. If the computed cross-validation error falls within one standard error ($1 - SE$ rule) the trees are statistically equivalent (below the red line), so the simplest tree (smallest size) is chosen for the model [8].

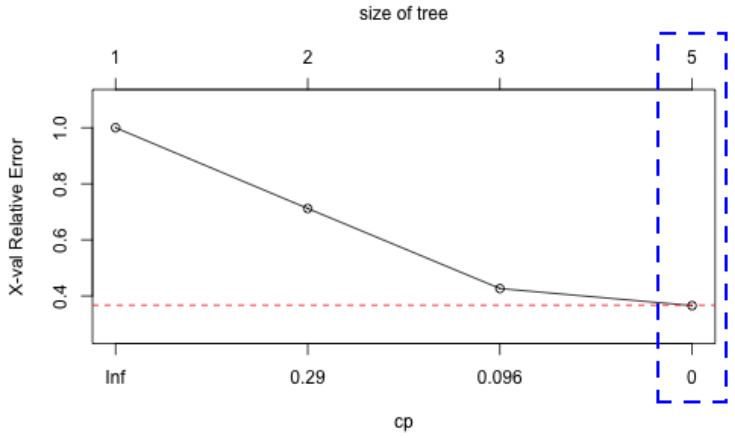


Figure 5.13: Complexity parameter and tree size determination.

The resulting tree, shown in Figure 5.12, has five leaves which correspond to five time windows, which are summarized in Table 5.1. It can be seen that the 1st and 5th time window correspond to the night hours during which the university is closed and not occupied. In contrast, the others time windows correspond to the university's operational hours. The length of the time windows is very

different from one another. This means that the tree is effective in isolating time windows in which there is an abrupt load variation (see the 2nd time window) from other behaviours.

Table 5.1: Sub-daily time windows for total electrical power

Time window	Hours	Rounded Hours	Duration
1	00:00 - 06:23	00:00 - 06:29	6 h 30 min
2	06:24 - 08:53	06:30 - 08.59	2 h 30 min
3	08:54 - 15:38	09:00 - 15:44	6 h 45 min
4	15:39 - 19:08	15:45 - 19:14	3 h 30 min
5	19:09 - 24:00	19:15 - 24:00	4 h 45 min

Once the time windows were identified and features extracted from the database, Adaptive Symbolic Aggregate approXimation (ASAX) was performed on the total electrical load time series for each time window. The time interval used is from January 2015 to December 2019.

A fundamental parameter to be set is the alphabet size (α) which determines how many characters are going to be used to for the encoding, and so the number breakpoints the algorithm needs to find. While in [8] a domain expert chooses the alphabet size, in this framework an unsupervised technique of single-link hierarchical clustering was used, setting an interval of potential values for the optima number of clusters between 3 and 8. According to the majority rule, the optimal number of clusters is 6 so the alphabet size, was assumed to be equal six ($\alpha = 6$).

The initial breakpoints, calculated under equally probability assumption, were used as initialization of ASAX iterative algorithm. As it can be seen in Figure 5.14, those breakpoints (dotted lines) are not able to divide the distributions effectively, producing narrow intervals at low values and wider intervals for higher values. The final adaptive breakpoints (solid lines) were evaluated once a tolerance of 10^{-10} on the representation error is reached.

Carpet plots were used to understand the distribution of symbols during the day and along the year, while histograms were used to visualize the occurrence frequency of symbols in each time window. In Figure 5.15(a) are reported the carpet plot and histograms referring to the encoded total electrical load time series. The figure shows that in time window 1 and 5, the most frequent symbols are "a" and "b", that corresponds to a low load during the night hours. In time window 2 and 4, corresponding to early morning and late afternoon, there is a prevalence of medium load identified with symbol "d", corresponding to the

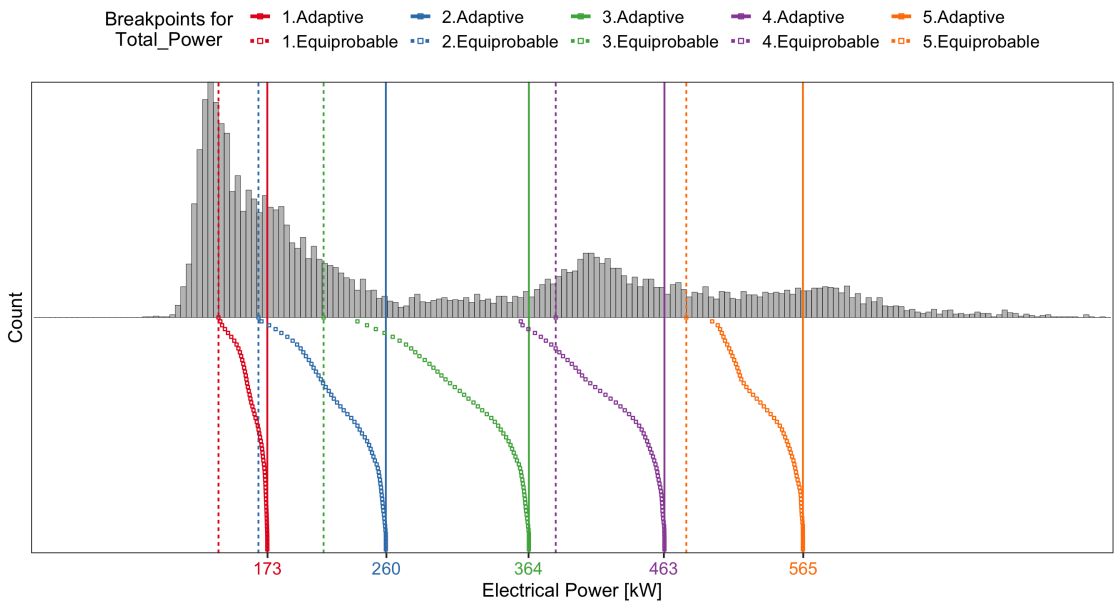
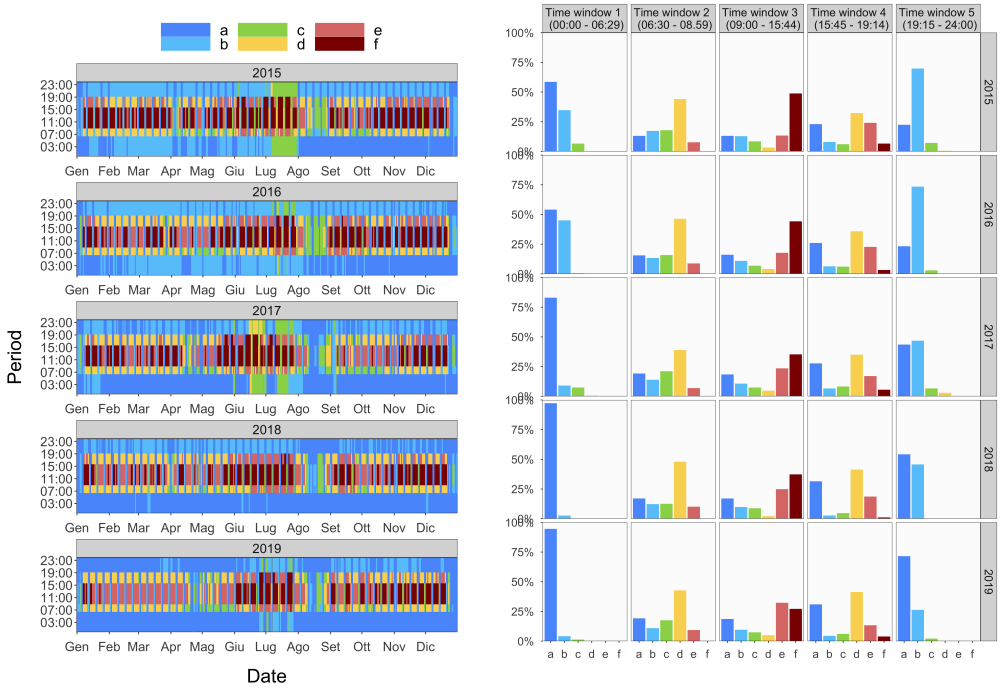


Figure 5.14: Step by step identification of adaptive breakpoints through the ASAX algorithm applied on the aggregated total electrical load.

switch on of the systems. In time window 3, the symbols "e" and "f" are the most frequent since the electrical load in the middle of the day is the highest. On a yearly base analysis, it can be seen that the load pattern has changed, with an overall trend in reducing electrical absorption. In particular in time window 1 and 5 the symbol "b" reduced its frequency in favour of the previous symbol “a”; the same trend is shown in time window 3 where symbol “f” reduced it frequency in favour of the symbol “e”.



(a) (b)

Figure 5.15: ASAX representation of the total electrical load: (a) carpet plot (b) histogram distribution of letters along the time windows and along the years

5.3 Detection at meter level data

Once the total electrical load time series is reduced in dimension and encoded the detection model can be constructed. For each time window, a globally optimum classification tree is developed in order to investigate further the dependency of the total electrical load (i.e. target variable) on boundary conditions such as the external temperature or day of the week (i.e. predictive variables). This tree aims to construct a model with very accurate decision rules, so in the leaf node, a high occurring symbol can be found. If so, the low occurrence symbols are potential anomalies for the given time window. Those anomalies will be further investigated in order to understand which sub-system can be imputed for infrequent behaviour.

Only data from 2018 was used for the model construction since in order to build a more accurate classification model that can be used as an anomaly detection tool is better to train the model on recent past data.

A process of cross-validation between test and train set is used to construct and select the model. The 2018 data was split by sampling randomly each week 80% of the days for train and 20% of days. Validation of the model is performed on data on the first month of 2019. In order to create a model that automatically learn new patterns as the building energy consumption behaviour changes, the idea is always to train and test the model on the last year of data and apply it on o a full month. Once the month ends, it is included in the training-test dataset of the new model that will be used for the following month.

Predictor attributes included in the tree are:

- The day of the week;
- Whether is a holiday or not;
- Mean external temperature of the time window (T_{air});
- Mean external temperature of the previous time window ($T_{air,pre}$);
- Mean total aggregated electrical load (in kW) of the previous time window;
- Mean electrical load (in kW) of Canteen and Refrigeration unit 2.

The choice to use as predictive values some sub-loads and not others comes from the analysis of the variance and the fraction on the total electrical load. The Canteen and the Refrigeration unit weight respectively for 12,22% and 16,03% on the total electrical load (Figure 3.2) and, among the sub-loads, they show the higher variance along 2018, this means that they present peaks and dumps of electrical load driving the overall electrical load up and down.

The maximum depth parameter of the tree was set to 6, the minimum number of observations in each node was set to 20, and the default probability setting for variation operators was assumed (20% crossover, 40% mutation and 40% split/prune).

Since the evtree algorithm and the splitting process are randomly initialized, the seed for the random number generator is set in the code in order to replicate the analysis easily.

In the following figures will be reported the classification trees for each time window, in particular in Figure 5.16 the first, in Figure 5.17 the second, in Figure 5.18 the third, in Figure 5.19 the fourth and in Figure 5.20 the fifth. In general is possible to see that those trees effectively separate in each leaf node the most

frequent symbol[1] from the others while maintaining a readable and understandable format. Decision rules extracted for each time window are reported in a easely interpretable IF⇒THEN format in Table 5.3.

Many trials were attempted in order to obtain a satisfactory trade-off between the testing accuracy and validation accuracy. The model performance results are shown in Table 5.2 where it can be seen that the mean overall accuracy in testing is around 86% while in validation 89%.

In particular, the first time window shows higher testing and validation accuracy (96.98% and 100% respectively). This is mainly due to the flat profile that the electrical load shows in this night and early morning period. In fact, the relative classification tree (Figure 5.16) is very simple: is composed of just one node in which the most frequent symbol is "a", with a relative frequency of 97.3%. The lowest training and validation accuracy can be seen in the third time window (79.45% and 58.06% respectively). Even if in the relative classification tree (Figure 5.18) the separation between frequent and infrequent symbols is well performed, the tree has difficulty in generalizing the behaviour since in this time window all the sub-loads shows very different operational patterns and electrical absorption.

As previously said, many trials were performed in order to choose the right predictive variables and to appropriately chose the test validation proportion. The training-test procedure avoids learning specific pattern allowing to create a more generalizable classifier avoiding overfitting on the validation set. The presented results are those that provide the best performance.

Table 5.2: Accuracy results comparison between test and validation

Time window	Test Accuracy [%]	Validation Accuracy [%]
00:00 - 06:29	96.89	100
06:30 - 08.59	82.19	93.55
09:00 - 15:44	79.45	58.06
15:45 - 19:14	86.30	96.77
19:15 - 24:00	86.30	96.77

[1]The symbols are represented in the figures by numbers in order to calculate more easily performance parameters in the code. However, there is a correspondence between number and symbols as follow: $1 \rightarrow a, 2 \rightarrow b, 3 \rightarrow c, 4 \rightarrow d, 5 \rightarrow e$ and $f \rightarrow 6$

Table 5.3: Decision rules extracted from globally optimal classifier

Time window	Node	Decision rule	Symbol	Accuracy
00:00 - 06:29	1	-	⇒ a	97.3%
06:30 - 08.59	2	IF Holiday = Yes	⇒ a	92.0%
	5	IF Holiday = No AND Refrigeration unit 2 < 85.84 kW AND Canteen < 96.4 kW	⇒ b	82.1%
	6	IF Holiday = No AND Refrigeration unit 2 < 85.84 kW AND Canteen >= 96.4 kW	⇒ d	89.7%
	8	IF Holiday = No AND Refrigeration unit 2 >= 85.84 kW AND Canteen < 108.44 kW	⇒ c	71.4%
	9	IF Holiday = No AND Refrigeration unit 2 >= 85.84 kW AND Canteen >= 108.44 kW	⇒ e	86.5%
09:00 - 15:44	3	IF Canteen < 54.4 kW AND Holiday = Yes	⇒ a	96.0%
	5	IF Canteen < 54.4 kW AND Holiday = No AND Total Power pre < 257.12 kW	⇒ b	76.5%
	6	IF Canteen < 54.4 kW AND Holiday = No AND Total Power pre >= 257.12 kW	⇒ c	85.0%
	8	IF Canteen >= 54.4 kW AND Canteen < 143.556 kW	⇒ e	73.9%
	10	IF Canteen >= 54.4 kW AND Canteen >= 143.556 kW AND Refrigeration unit 2 < 38.015 kW	⇒ e	86.0%
	11	IF Canteen >= 54.4 kW AND Canteen >= 143.556 kW AND Refrigeration unit 2 >= 38.015 kW	⇒ f	81.1%
15:45 - 19:14	2	Total Power pre < 388.83 kW	⇒ a	87.4%
	4	Total Power pre >= 388.83 kW AND Total Power pre < 614.074 kW	⇒ d	86.5%
	5	Total Power pre >= 388.83 kW AND Total Power pre >= 614.074 kW	⇒ d	85.4%
19:15 - 24:00	2	IF Holiday = Yes	⇒ a	96.0%
	4	IF Holiday = No AND Day Type = {6,7}	⇒ a	97.2%
	6	IF Holiday = No AND Day Type = {1,2,3,4,5} AND Canteen < 16.526 kW	⇒ a	85.5%
	7	IF Holiday = No AND Day Type = {1,2,3,4,5} AND Canteen >= 16.526 kW	⇒ b	87.6%

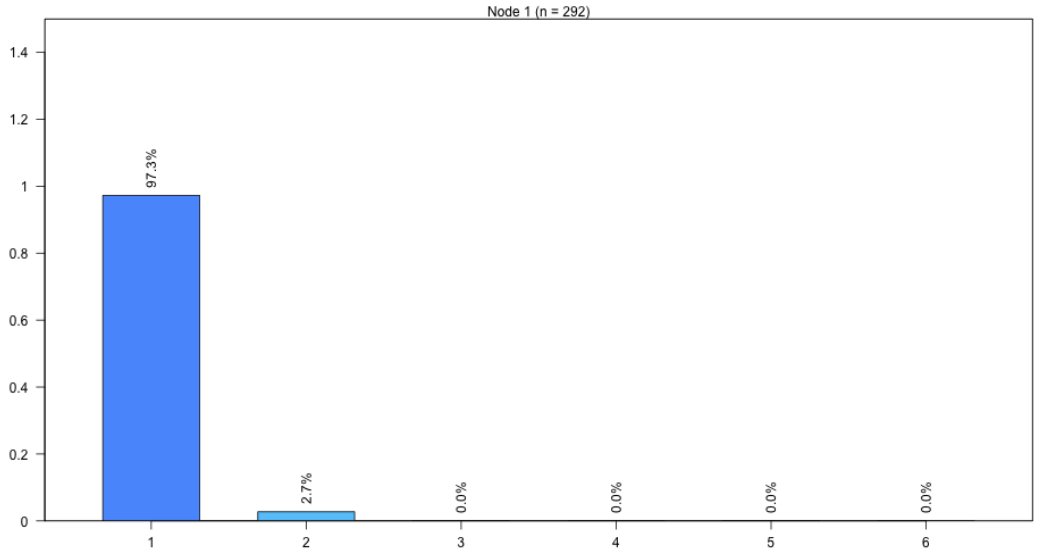


Figure 5.16: Globally optimum tree for time window 1 (00:00 - 06:29).

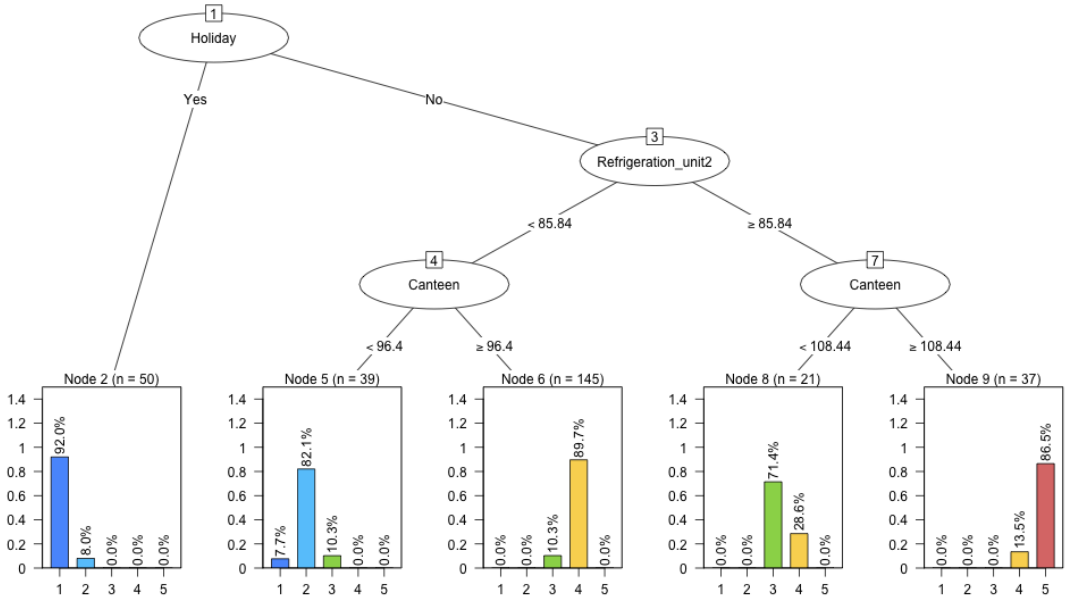


Figure 5.17: Globally optimum tree for time window 2 (06:30 – 08:59).

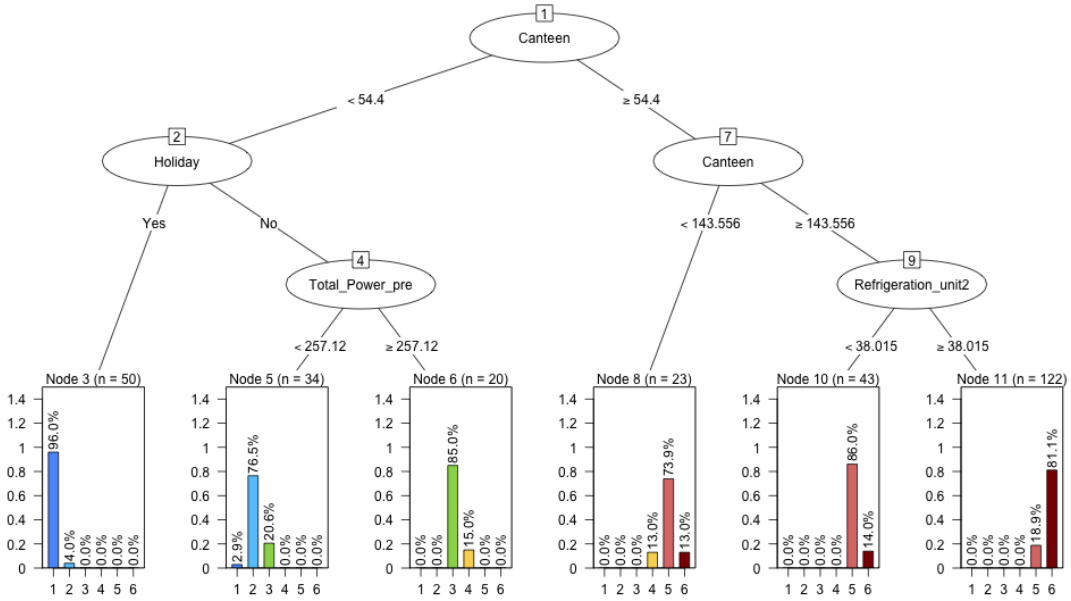


Figure 5.18: Globally optimum tree for time window 3 (09:00 - 15:44).

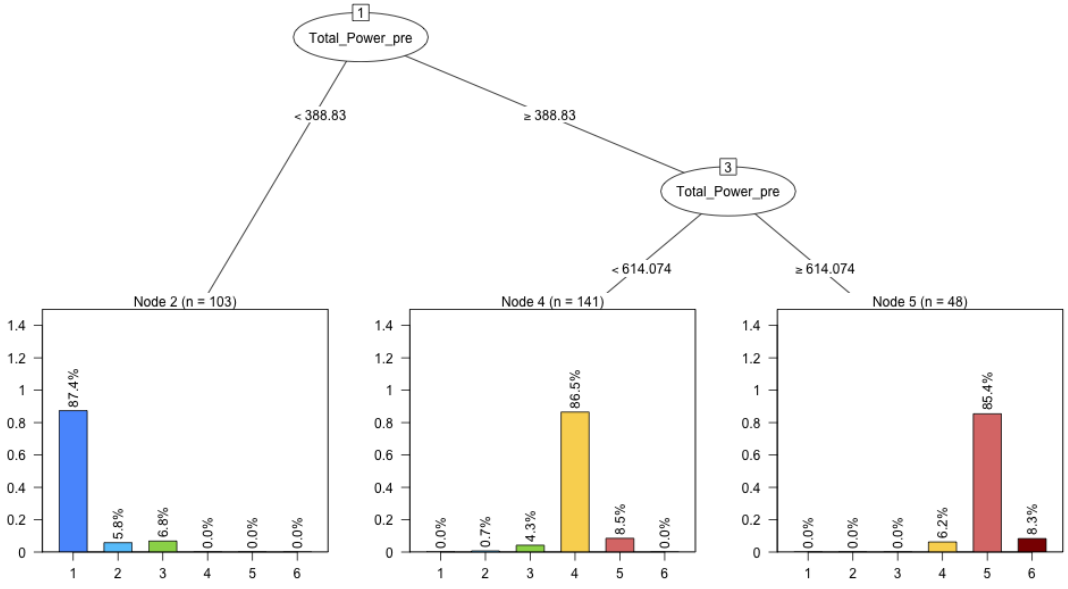


Figure 5.19: Globally optimum tree for time window 4 (15:45 - 19:14).

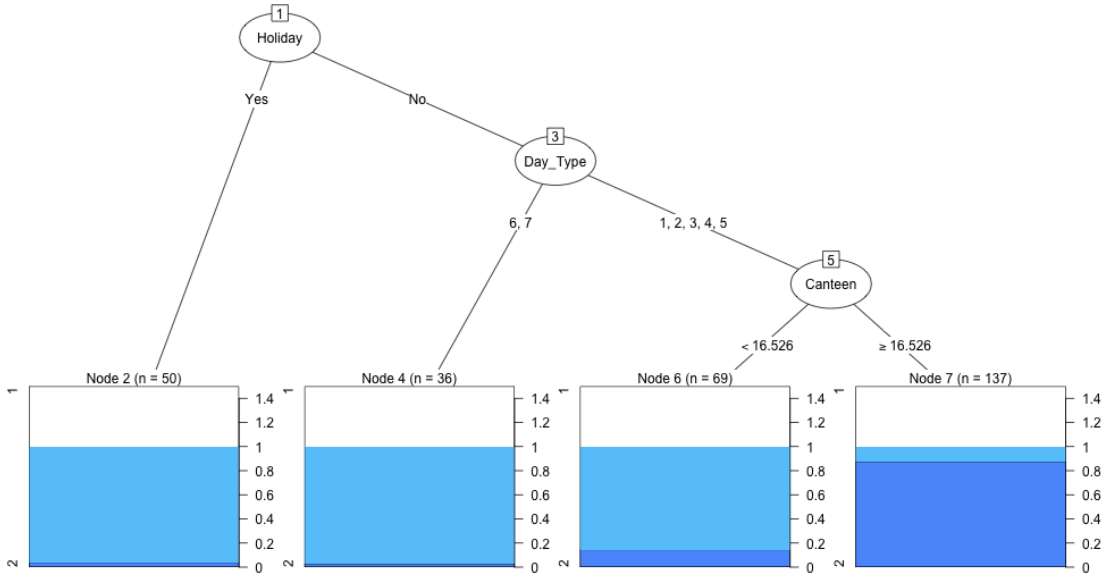


Figure 5.20: Globally optimum tree for time window 5 (19:15 - 24:00).

5.4 Diagnosis at sub-meter level data

Once the classification model is created, the subset of observations contained in each node is transformed into a transactional database which contains the categorical target variable (total electrical load symbol) and some additional explanatory variables related to the sub-loads.

To extract those additional categorical variables, sub-loads time series undergo the same time series abstraction process described for the total electrical load in Section 5.2. Using the same time window discretization as the total electrical load and the same alphabet size ($\alpha = 6$) each time series electrical load is encoded through ASAX. In order to enrich information about sub-loads, the trend angle is extracted and encoded as well. This feature allows to keep track of the trend of the time series in the given time window, in particular identifying if the load presents an increasing, decreasing or stable trend. In this case, the alphabet size was set to three ($\alpha = 3$) in order to reflect those three trends.

- $\theta >> 0$ means that the trend angle is positive and so the time series trend is increasing upward. This condition is codified with the symbol "UP" in red in figures;

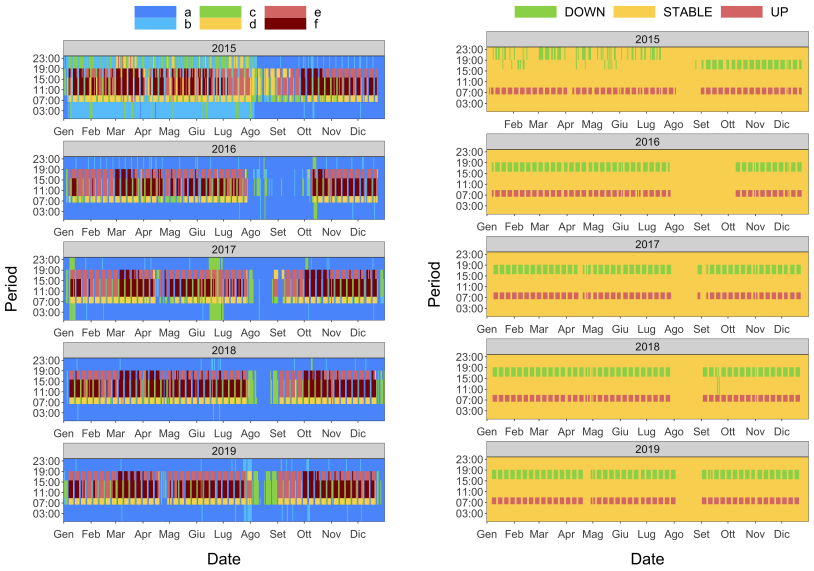
- $\theta \simeq 0$ means that the trend angle is almost zero and so the time series does not present any particular trend . This condition is codified with the symbol "STABLE" in yellow in figures;

- $\theta << 0$ means that the trend angle is negative and so the trend is decreasing downward. This condition is codified with the symbol "DOWN" in green in figures;

The initial breakpoints, calculated under equally probability assumption, were used as initialization of ASAX iterative algorithm and the final adaptive breakpoints were evaluated once a tolerance of 10^{-10} on the representation error is reached.

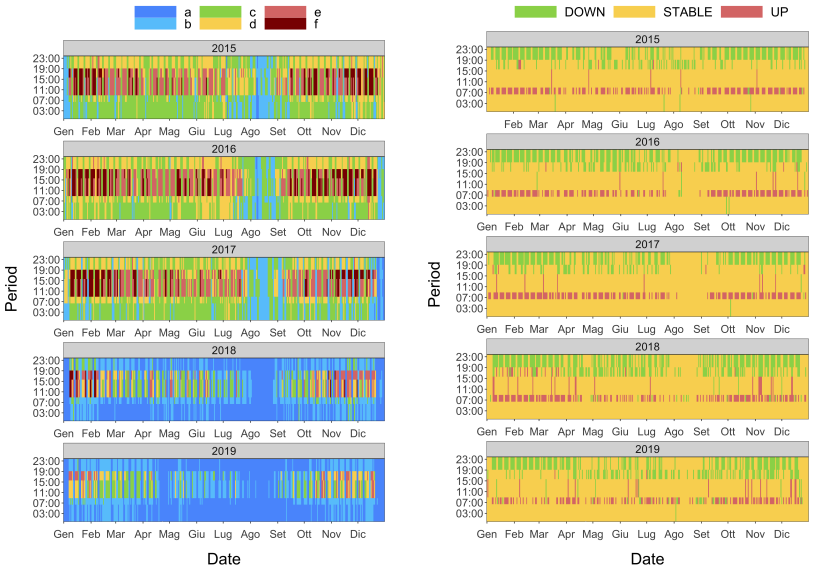
The carpet plot representation for each sub-loads has been constructed and are represented in the following figures Print shop in Figure 5.21, DIMAT in Figure 5.22, Bar Ambrogio in Figure 5.23, Rectory in Figure 5.24, Refrigeration unit in Figure 5.25, Data centre in Figure 5.26, Canteen in Figure 5.27.

In each figure is reported on the left side the encoded electrical load symbol (a) while on the right side the encoded trend angle (b). From these figures next to the electrical load information already discussed in Section 5.1 another important information is added: the trend angle. This feature permits to identify the load ON/OFF schedule and anomalies in this sense. For example is possible to identify very regular patterns of for the Print shop in which appliances are switched on in the 2^{nd} time window and switched off in the 4^{th} time window. Any behavior different from this one my result in possible anomalies or unusual schedule and thus can be corrected or avoided in phase of energy management. This feature permits to enrich the information in the following ARM phase without increasing the computational effort.



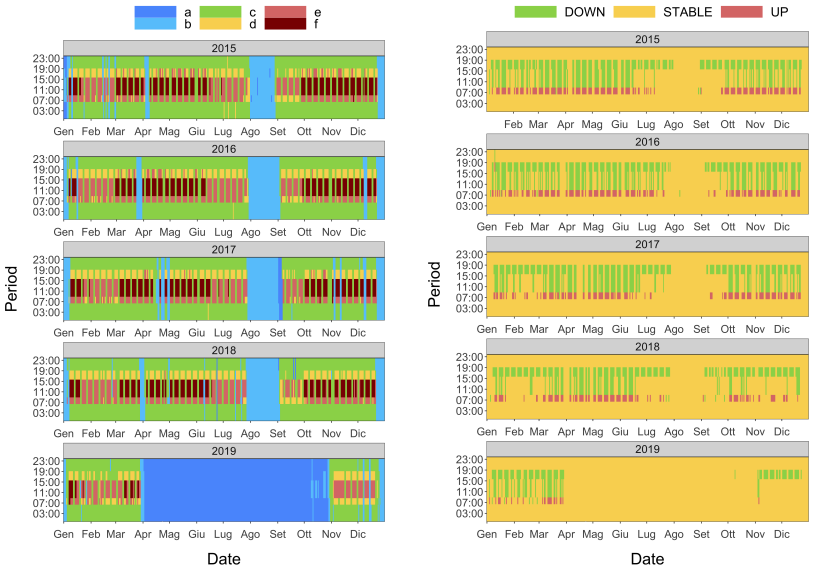
(a) (b)

Figure 5.21: ASAX carpet plot for Print shop.



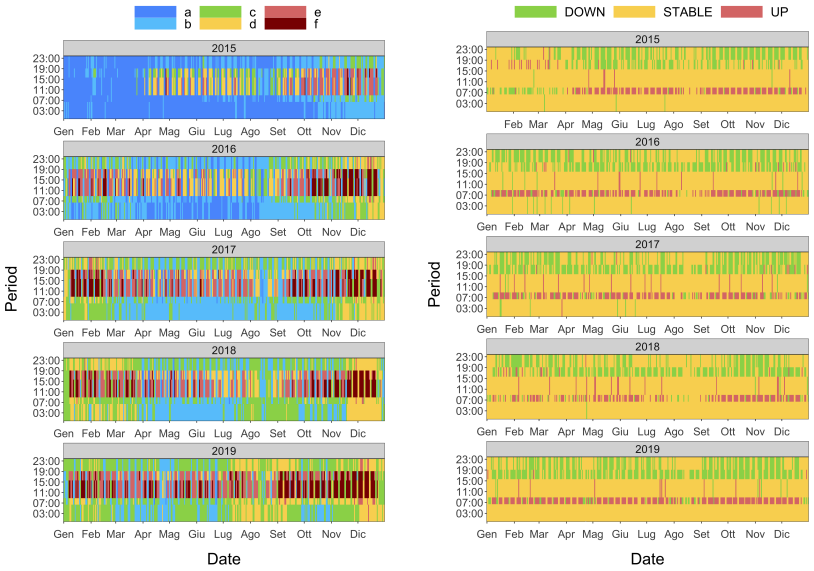
(a) (b)

Figure 5.22: ASAX carpet plot for DIMAT.



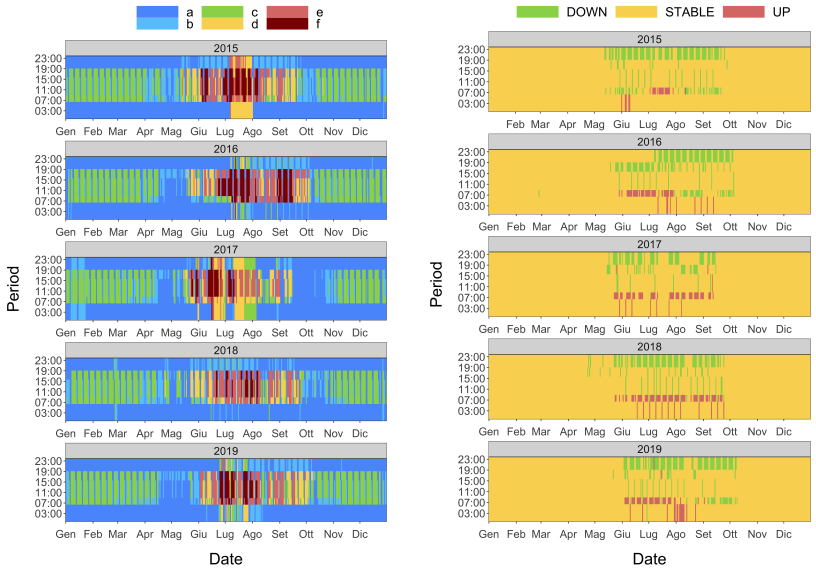
(a) (b)

Figure 5.23: ASAX carpet plot for Bar Ambrogio.



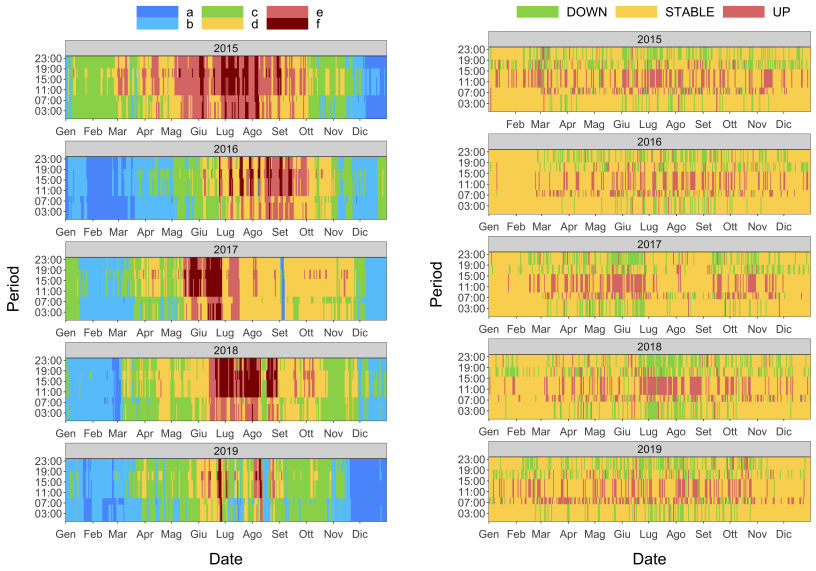
(a) (b)

Figure 5.24: ASAX carpet plot for Rectory.



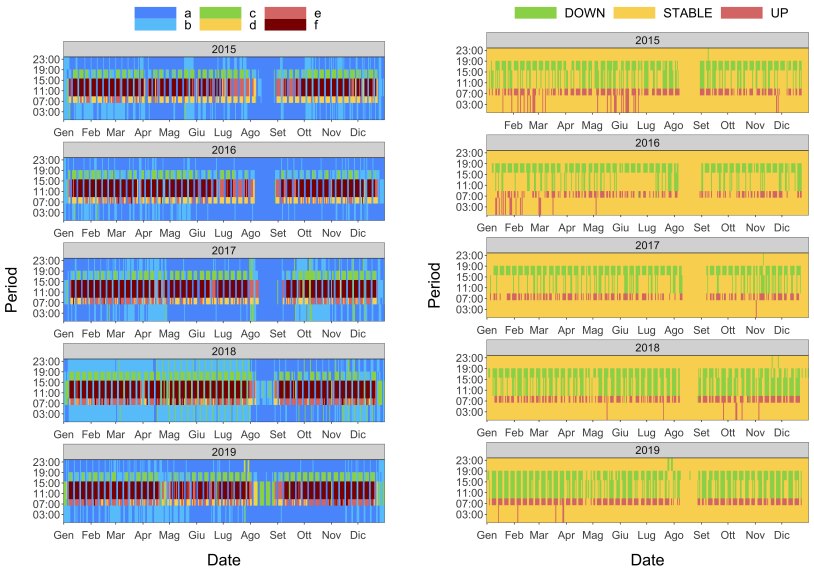
(a) (b)

Figure 5.25: ASAX carpet plot for Refrigeration unit.



(a) (b)

Figure 5.26: ASAX carpet plot for Data centre.



(a) (b)

Figure 5.27: ASAX carpet plot for Canteen.

Then Apriori ARM algorithm is applied on the transactional database using the R package arules [34]. The RHS is the anomalous total electrical load symbol, while in the LHS is composed of all possible combination of electrical load symbols and trend angles of sub-loads. The minimum and the maximum number of items in a transaction is set in order to create resulting rules with one or maximum two items in the LHS. The minimum support to mine rules is set to 0.005, and the minimum confidence is set to 0.005.

Redundant rules, equally or less predictive of a more general rule, are removed and the remaining are represented in a scatter plot. This kind of plot permits to visualize the association rules by representing on the x-axis the support, on y-axes the confidence and by coloring the points with a gradient scale representing the lift [14]. Interesting rules are those that are less frequent (i.e. low support) and have high confidence and high lift. From the scatter plot it is possible to isolate interesting rules by setting $lift > 1$ and $confidence > 0.5$.

Those rules are then stored in the anomaly library. The anomaly library is a database composed of five colums. The first column stores the LHS of the association rules, the second columns stores the RHS, the third column stores the support, the fourth the confidence, the fifth the coverage (support of the RHS)

and finally the sixth column the lift. This data frame is ordered by decreasing values of lift. LHS of those rules represent those loads that are significantly influencing the abnormal electrical load and so the anomaly detection of sub-loads can be concluded.

An example of the procedure is shown in Figure 5.28 for node 5 of the second time window. In this node the most frequent symbol is "b", and the only infrequent interesting symbol (higher power absorption) is "c", which constitutes the RHS. The transactional database used for ARM is composed of sub-loads categorical variables (electrical load symbol and trend angle). Of 338 rules extracted, 180 are redundant and 158 rules are significant. After filtering through the scatter plot, only 19 remain and they are stored in the anomaly library. In the particular case, the most frequent items in the anomaly library are Refrigeration unit 2 = d, Canteen = c and Rectory = d.

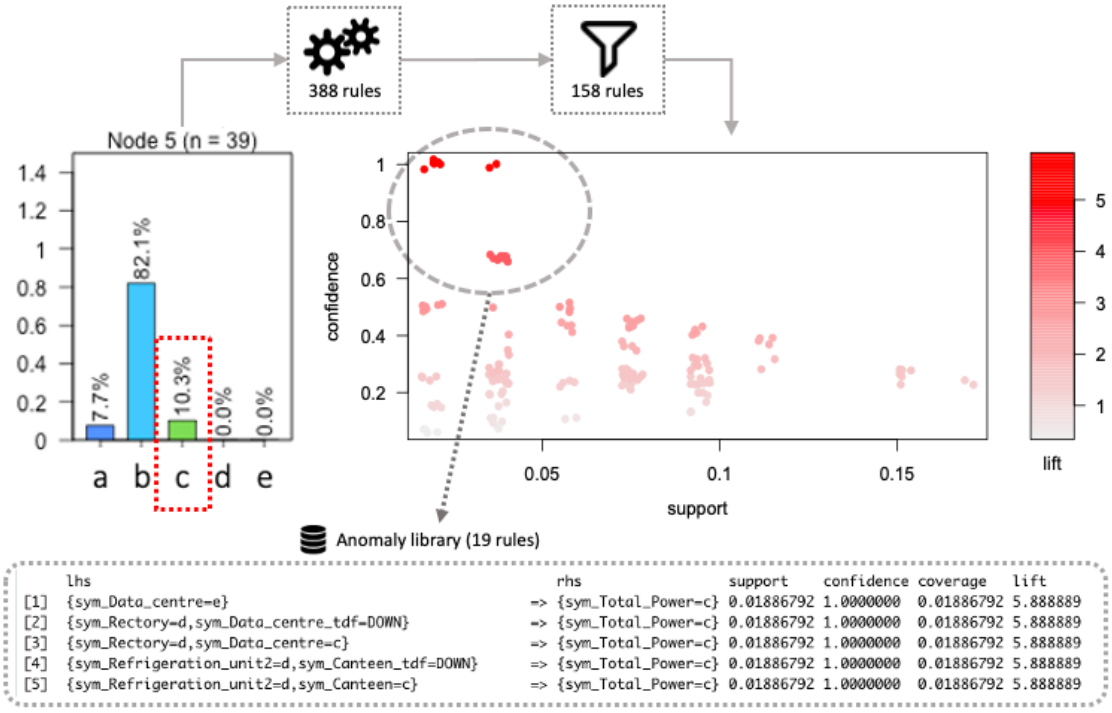


Figure 5.28: Diagnosis procedure of extracting, filtering and selecting only relevant association rules from node 5 of time window 2.

5.5 Simulation of application

The methodology is intended to be implemented in an EIS that provides real-time data analysis using meter-level data. The EIS system continuously collects data, and once a time window is concluded, the total electrical load symbol is calculated employing ASAX and compared to the one predicted by the globally optimal tree. Three possible cases can be found applying this model:

- The actual symbol is the same as the predicted symbol. This means that given the boundaries conditions the total energy consumption of that time window is as expected, no further diagnosis is requested;

- The actual symbol is different from the predicted symbol and indicates a lower electrical load than expected. This means that given the boundaries conditions the mean electrical load of that time window is lower than expected, no further diagnosis is requested since the focus of the methodology is to find infrequent behaviors that cause higher consumption;

- The actual symbol is different from the predicted symbol and indicates a higher electrical load than expected. This means that given the boundaries conditions the mean electrical load of that time window is higher than expected; a further diagnosis is needed.

In the latter case, the diagnosis is enabled. Given the boundaries conditions, the corresponding leaf node of the classification tree is identified, and the tool automatically extracts new association rules related to anomalous behavior and then compares them to the ones contained in the anomaly library. If there is a match, those items or rules robustly identify which sub-loads are connected to the anomaly.

5.5.1 One month simulation

In the first part of this simulation example, the methodology is applied on January 2019. The process of detection through the tree evolutionary tree is performed on all the time windows and results are shown in Figure 5.29. Only in the 2^{nd} and 5^{th} time windows happen that the actual symbol is different from the predicted symbol and indicates a higher electrical load than expected, respectively “c” instead of “b” and “b” instead of “a”. Both of those anomalies happened on Friday 2019-01-04.

Time window 2

Once identified the day and the time window of the anomaly the corresponding tree's leaf node is identified as well. In this case the anomalous electrical load of Friday 2019-01-04 belongs to the 2nd time window and node number 5. The diagnosis process is enabled, and new association rules are extracted and compared with the anomaly library of the respective node. In this example, there is a partial match of the rules on the following items:

- Print shop electrical load symbol "c"
- Refrigeration electrical load unit symbol "c"
- Canteen electrical load symbol "c"
- Canteen trend angle symbol "UP"

Those are the sub-loads that cause the increase of total electrical load. A further graphical analysis is conducted in Figure 5.30 to assess the validity of this conclusion. The graph shows a comparison between of the total electrical load, Refrigeration unit, Print shop and Canteen. Only 2nd time window is shown on the x-axis: in red the anomalous data related to 2019-01-04 while in green is shown the frequent "normal" behavior of the given load in the training period (from 01-01-2018 to 31-01-2018). Along with the effective electrical load (solid lines) are reported the PAA in dashed line.

We can verify that the combined effect of higher mean power absorption of these three loads and the early switch of the canteen after 8:00 a.m., testified by a high trend angle, lead to higher overall electrical load. The mean total electrical load switches from 236.28 kW (symbol "b") to 283.09 kW (symbol "c"), and it is easy to verify that those loads contribute for almost 90% of the power shift upward of the total electrical load.

Time window 4

Once identified the day and the time window of the anomaly the corresponding tree's leaf node is identified as well. In this case the anomalous electrical load of Friday 2019-01-04 belongs to the 4th time window and node number 2. The diagnosis process is enabled, and new association rules are extracted and compared with the anomaly library of the respective node. In this example, there is a partial match of the rules on the following items:

- Rectory electrical load symbol "c"

- Refrigeration unit electrical load unit symbol "c"
- Canteen electrical load symbol "b"

Those are the sub-loads that cause the increase of total electrical load. A further graphical analysis is conducted in Figure 5.31 to assess the validity of this conclusion. The graph shows a comparison between of the total electrical load, Refrigeration unit, Rectory and Canteen. Only 2nd time window is shown on the x-axis: in red the anomalous data related to 2019-01-04 while in green is shown the frequent "normal" behavior of the given load in the training period (from 01-01-2018 to 31-01-2018). Along with the effective electrical load (solid lines) are reported the PAA in dashed line.

We can verify that the combined effect of higher mean power absorption of these three loads lead to higher overall electrical load. The mean total electrical load switches from 153.34 kW (symbol "a") to 207.97 kW (symbol "b").

5.5.2 Six month simulation

In this second part of the simulation application the methodology is tested on the first six months of 2019 in order to assess the accuracy of the results as the times passes and the whole building patterns changes. The retraining process consists in successive steps:

1. Suppose that the actual date is 01/01/2019;
2. Train and test the methodology (classification tree + ARM) on the past year of data $I_{traini,test} = [01/01/2018; 01/01/2019)$;
3. Use the methodology from now to the end of the month on the interval $I_{validation} = [01/01/2019; 01/02/2019)$;
4. Once the month is concluded (Actual date 01/02/2019) retrain and retest the methodology (classification tree + ARM) on the past year of data $I_{traini,test} = [01/02/2018; 01/02/2019)$;
5. Repeat steps 3 and 4

By doing so the methodology is retrained monthly providing higher validation accuracy as shown in Table 5.4. In this table the case A refers to accuracy results of not retrained model (i.e. the model constructed only on 2018 data) while the case B refers to the monthly retrained model. It can be seen that the average validation accuracy calculated as the mean of all the accuracies of all

the time windows, in case A is 78.77% which is lower than case B where it is 82.85%, confirming that the retrain strategy is efficient in capturing the mutation of energy behaviour providing better performances.

Table 5.4: Validation accuracy results comparison between not retrained model (A) and retrained model (B)

Time window	February		March		April		May		June	
	A [%]	B [%]	A [%]	B [%]	A [%]	B [%]	A [%]	B [%]	A [%]	B [%]
00:00 - 06:29	100	100	100	100	100	100	100	100	86.67	86.67
06:30 - 08.59	100	100	96.77	96.77	60.00	53.44	22.58	22.58	66.67	90.00
09:00 - 15:44	57.14	85.71	64.52	100	50.00	76.67	77.42	58.06	60.00	76.67
15:45 - 19:14	100	100	96.77	96.77	76.67	76.67	80.65	64.52	60.00	66.67
19:15 - 24:00	89.29	89.29	77.42	77.42	90.00	90.00	96.77	100	60.00	63.33
Mean	89.29	95.00	87.10	94.19	75.33	79.36	75.48	69.03	66.67	76.67

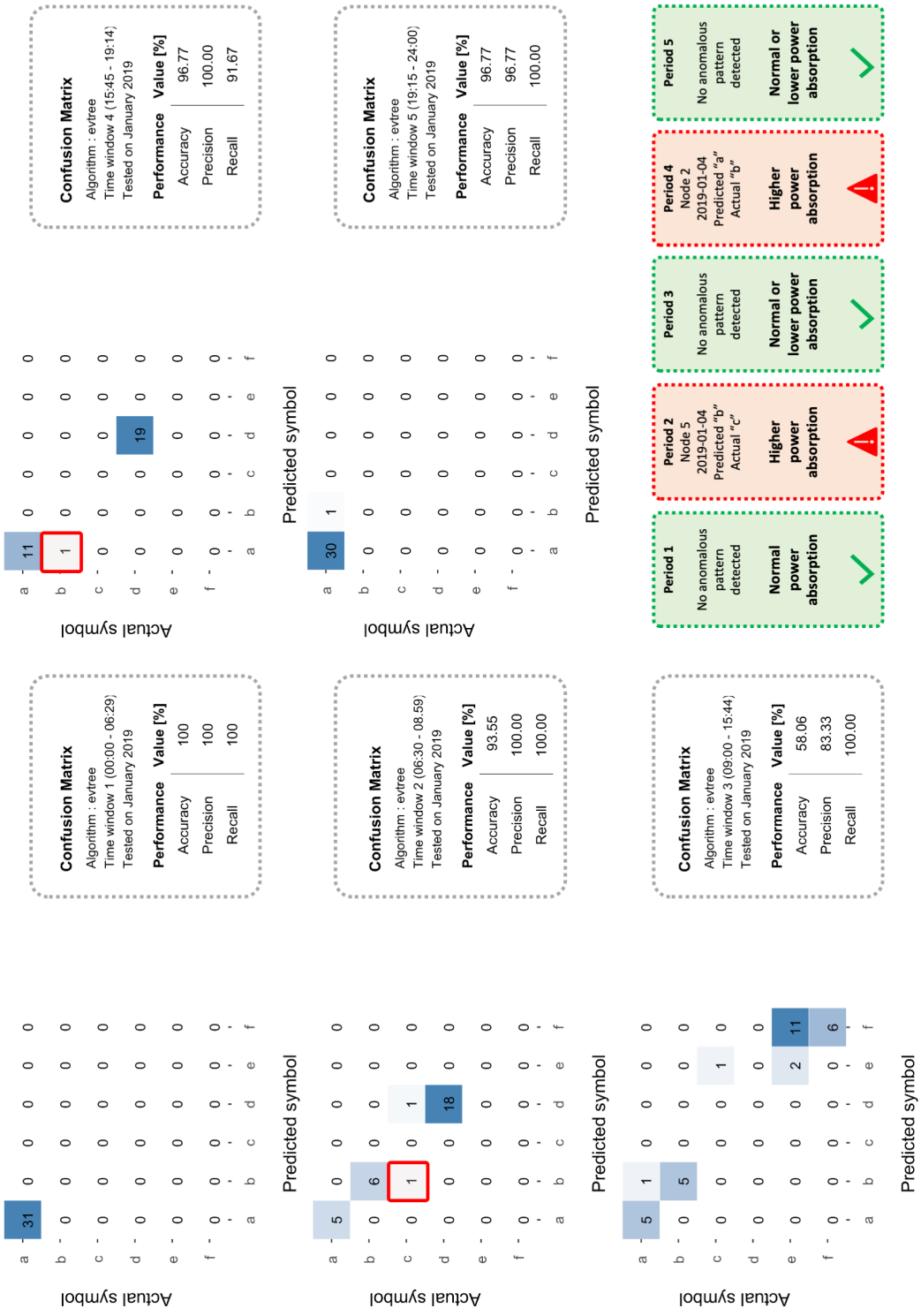


Figure 5.29: Confusion matrix for the evtree classification tree predicting January 2019 total electrical load symbol. In the red square the anomalous behavior to be investigated.

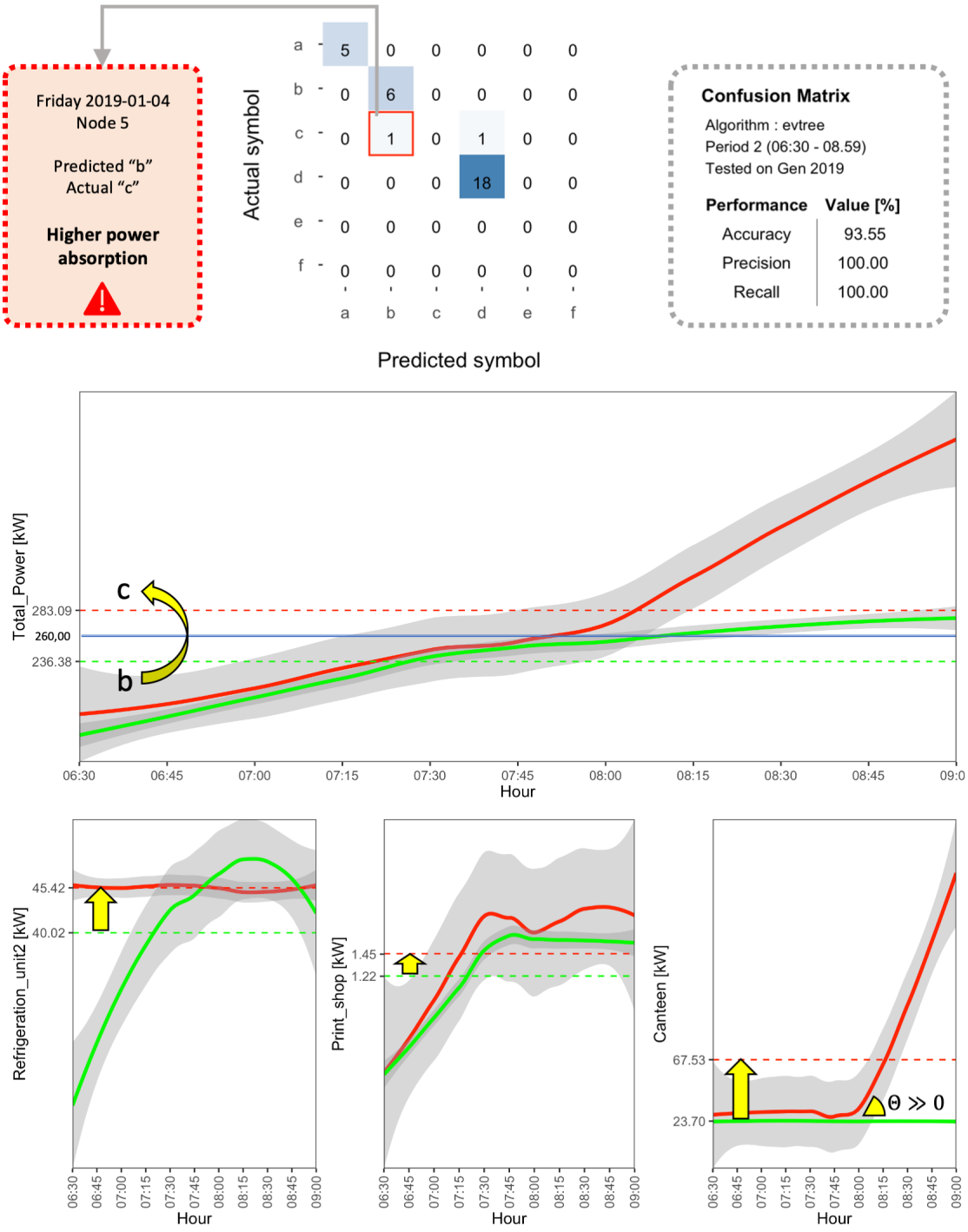


Figure 5.30: Focus on electrical load in period 2 node 5, comparison between the detected anomalous electrical load on 2019-01-04 and the normal behavior.

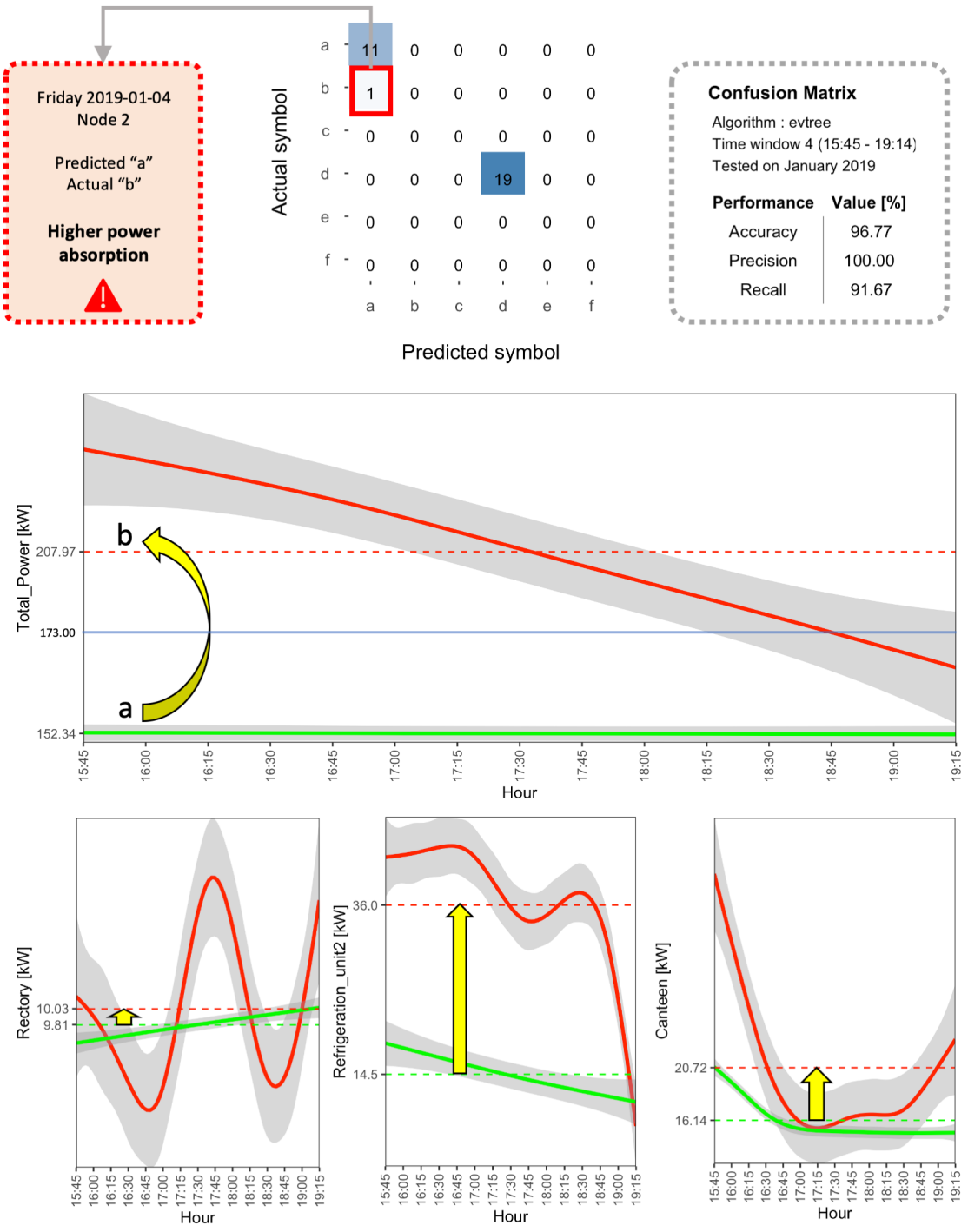


Figure 5.31: Focus on electrical load in period 4 node 2, comparison between the detected anomalous electrical load on 2019-01-04 and the normal behavior.

Chapter 6

Conclusion

This paper focused on the development of a top-down ADD methodology able to analyze meter-level electrical load data in order to detect anomalous pattern and perform a diagnostic process on sub-loads through ARM.

This methodological framework was conceived to be a highly scalable and reliable tool ready to be implemented in energy data acquisition systems that can help to promptly detect anomalies and avoid energy wastes to be prolonged over time.

In this framework, the aim was to create an automatic procedure by using as much as possible methods in which the supervised choice of parameters is limited to only the necessary ones. A right choice of parameters for a particular building could be the wrong choice for another, and an incorrect setting could cause an important loss of accuracy.

The time window size and alphabet size for the ASAX encoding are essential parameters. In [31] is reported an interesting sensitivity analysis based on these two parameters, showing that a tradeoff between window numbers and alphabet size has to be found in order to minimize the variance between patterns and resolution needed. In this thesis the time window number was chosen by using a CART and the alphabet size by a hierarchical clustering evaluation. Once those parameters are set, the ASAX encoding procedure is completely automatic. Moreover the conducted analysis shows that considering a trend angle as feature a robust sub-loads characterization can be performed without adding any computational burden.

Regarding the classification model, in this framework no sensitivity analysis was performed, the choice of the variations operators probability correspond to the default choice c20m40sp40 [18]. For a detailed sensitivity analysis on how the choice of variation operators affect the misclassification rate of the classification

tree refer to [4].

Moreover the selection of the predictive variables for the evolutionary tree needs particular attention. The overall energy consumption of a building is strongly connected to occupancy schedule, environmental conditions, thermo-physical features of the building, users behaviour. For this reason, those variables should be all included in the classification model and could help in describing infrequent but non-anomalous patterns. On the other hand, trustworthy values are difficult to retrieve or measure with continuity. For sure, the inclusion of those variables could qualitatively increase the model predictions.

A further interesting aspect of being considered is related to the data that should be used for training and how often training is needed. Is well known that building electrical load varies along the years due to the electrification of end uses and the seek of the higher performance of appliances and facilities. For this motivation, a good trade-off between re-training rate and computational effort should be performed with monthly retrain of the classification tree and an update of ARM anomaly libraries. By maintaining to one year the size of an ideal moving window the new month would be included while the corresponding month of the previous year excluded. In this way, the models would keep pace to the change of energy consumption patterns.

Moreover, the methodology could lead to the construction of "normal" knowledge database to compare future operation and identify anomalies or a database of anomalies and detect directly if those happens, and filling with new observation the database.

Finally, further developments of this work may include a real time implementation in Politecnico di Torino Energy Information Systems (EIS) and deployment in a real case application.

Acronyms List

AEIS	Advanced Energy Information Systems
ADD	Anomaly Detection and Diagnostics
AI	Artificial Intelligence
AMI	Advanced Metering Infrastructure
ARM	Association Rule Mining
ASAX	Adaptive Symbolic Aggregate approXimation
ASO	Automated System Optimization
BAS	Building Automation Systems
CART	Classification And Regression Trees
CV	Cross Validation
DM	Data Mining
DSS	Decision Support Systems
EIS	Energy Information Systems
EMIS	Energy Management and Information Systems
FDD	Fault Detection and Diagnosis
IEA	International Energy Agency
HVAC	Heating, Ventilation and Air Conditioning
KPI	Key Performance Index
IT	Information Technology

LHS Left Hand Side

LOOCV Leave One Out Cross Validation

PAA Piecewise Aggregate Approximation

PoliTo Politecnico di Torino

RHS Right Hand Side

RSS Residual Sum of Squares

SAX Symbolic Aggregate approXimation

SMI Smart Metering Infrastructure

SSE Sum of the Squared Errors

Bibliography

[33] J. Lin, E. Keogh, L. Wei, and S. Lonardi, «Experiencing SAX: a Novel Symbolic Representation of Time Series», *Cs.Gmu.Edu*, vol. 15, pp. 107–144, 2007, ISSN: 1573-756X.

[1] U. N. UN, «Transforming our world: the 2030 Agenda for Sustainable Development», General Assembly of United Nations, Tech. Rep., 2015. [Online]. Available: https://sustainabledevelopment.un.org/post2015/transformingourworld.

[2] P. Waide, J. Ure, N. Karagianni, G. Smith, and B. Bordass, «The scope for energy and CO2 savings in the EU through the use of building automation technology», *Final Report for the European Copper Institute*, 2013.

[3] C. Fan, F. Xiao, and S. Wang, «Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques», *Applied Energy*, vol. 127, pp. 1–10, 2014, ISSN: 03062619. DOI: 10.1016/j.apenergy.2014.04.016. [Online]. Available: http://dx.doi.org/10.1016/j.apenergy.2014.04.016.

[4] M. S. Piscitelli, S. Brandi, and A. Capozzoli, «Recognition and classification of typical load profiles in buildings with non-intrusive learning approach», *Applied Energy*, vol. 255, no. August, p. 113 727, 2019, ISSN: 03062619. DOI: 10.1016/j.apenergy.2019.113727. [Online]. Available: https://doi.org/10.1016/j.apenergy.2019.113727.

[5] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, «Cluster Analysis: Basic Concepts, and Algorithms», *Introduction to Data Mining*, p. 526, 2019.

[6] N. D. Pham, Q. L. Le, and T. K. Dang, «HOT aSAX: A novel adaptive symbolic representation for time series discords discovery», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5990 LNAI, no. PART 1, pp. 113–121, 2010, ISSN: 03029743. DOI: 10.1007/978-3-642-12145-6_12.

[7] H. Ren, M. Liu, Z. Li, and W. Pedrycz, «A Piecewise Aggregate pattern representation approach for anomaly detection in time series», *Knowledge-Based Systems*, vol. 135, pp. 29–39, 2017, ISSN: 09507051. DOI: 10.1016/j.knosys.2017.07.021. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2017.07.021.

[8] A. Capozzoli, M. S. Piscitelli, S. Brandi, D. Grassi, and G. Chicco, «Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings», *Energy*, vol. 157, pp. 336–352, 2018, ISSN: 03605442. DOI: 10.1016/j.energy.2018.05.127. [Online]. Available: https://doi.org/10.1016/j.energy.2018.05.127.

[9] Y. Wang, Q. Chen, C. Kang, M. Zhang, K. Wang, and Y. Zhao, «Load profiling and its application to demand response: A review», *Tsinghua Science and Technology*, vol. 20, no. 2, pp. 117–129, 2015, ISSN: 18787606. DOI: 10.1109/tst.2015.7085625.

[10] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, «Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases», *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, 2001, ISSN: 0219-1377. DOI: 10.1007/pl00011669.

[11] C. C. Aggarwal and Data, *Data Mining: The Textbook*. Springer, 2012. arXiv: arXiv:1011.1669v3.

[12] J. Zhu, Y. Shen, Z. Song, D. Zhou, Z. Zhang, and A. Kusiak, «Data-driven building load profiling and energy management», *Sustainable Cities and Society*, vol. 49, no. March, p. 101 587, 2019, ISSN: 22106707. DOI: 10.1016/j.scs.2019.101587. [Online]. Available: https://doi.org/10.1016/j.scs.2019.101587.

[13] M. S. Piscitelli, «Enhancing energy management in buildings through data analytics technologies», PhD thesis, Politecnico di Torino, 2020.

[14] M. Hahsler and S. Chelluboina, «Visualizing Association Rules: Introduction to the R-extension Package arulesViz», *R project module*, pp. 1–24, 2011. [Online]. Available: http://www.comp.nus.edu.sg/%7B~%7Dzhanghao/project/visualization/[2010]arulesViz.pdf.

[15] A. Capozzoli, M. S. Piscitelli, and S. Brandi, «Mining typical load profiles in buildings to support energy management in the smart city context», *Energy Procedia*, vol. 134, pp. 865–874, 2017, ISSN: 18766102. DOI: 10.1016/j.egypro.2017.09.545. [Online]. Available: https://doi.org/10.1016/j.egypro.2017.09.545.

[16] E. J. Atkinson and T. M. Therneau, «An Introduction to Recursive Partitioning Using the RPART Routines», *Mayo Clinic Section Biostatistics Technical Report*, vol. 61, p. 33, 2000. [Online]. Available: http://nova.wh.whoi.edu/palit/Atkinson,%20Therneau%7B%5C_%7D2000%7B%5C_%7DMayo%20Clinic%20Section%20Biostatistics%20Technical%20Report%7B%5C_%7DAn%20Introduction%20to%20Recursive%20Partitioning%20Using%20the%20RPART%20Routines.pdf.

[17] C. Fan, Y. Sun, K. Shan, F. Xiao, and J. Wang, «Discovering gradual patterns in building operations for improving building energy efficiency», *Applied Energy*, vol. 224, no. March, pp. 116–123, 2018, ISSN: 03062619. DOI: 10.1016/j.apenergy.2018.04.118. [Online]. Available: https://doi.org/10.1016/j.apenergy.2018.04.118.

[18] T. Grubinger, A. Zeileis, and K. P. Pfeiffer, «Evtree: Evolutionary learning of globally optimal classification and regression trees in R», *Journal of Statistical Software*, vol. 61, no. 1, pp. 1–29, 2014, ISSN: 15487660. DOI: 10.18637/jss.v061.i01.

[19] H. Kramer, G. Lin, J. Granderson, C. Curtin, and E. Crowe, «Synthesis of Year One Outcomes in the Smart Energy Analytics Campaign Building Technology and Urban Systems Division», no. September 2017, 2017.

[20] M. Cabrera, D. F., and H. Zareipour, «Data association mining for identifying lighting energy waste patterns in educational institutes», *Energy and Buildings*, vol. 62, pp. 210–216, 2013, ISSN: 03787788. DOI: 10.1016/j.enbuild.2013.02.049. [Online]. Available: http://dx.doi.org/10.1016/j.enbuild.2013.02.049.

[21] Y. Zhang, L. Duan, and M. Duan, «A new feature extraction approach using improved symbolic aggregate approximation for machinery intelligent diagnosis», *Measurement: Journal of the International Measurement Confederation*, vol. 133, pp. 468–478, 2019, ISSN: 02632241. DOI: 10.1016/j.measurement.2018.10.045. [Online]. Available: https://doi.org/10.1016/j.measurement.2018.10.045.

[22] C. Zhang, Y. Zhao, T. Li, and X. Zhang, «A post mining method for extracting value from massive amounts of building operation data», *Energy and Buildings*, vol. 223, 2020, ISSN: 03787788. DOI: 10.1016/j.enbuild.2020.110096.

[23] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs, «NbClust : An R Package for Determining the», *Journal of Statistical Software*, vol. 61, no. 6, pp. 1–36, 2014, ISSN: 1548-7660. DOI: 10.18637/jss.v061.i06.

[24] Y. Yu, Y. Zhu, D. Wan, H. Liu, and Q. Zhao, «A novel symbolic aggregate approximation for time series», *Advances in Intelligent Systems and Computing*, vol. 935, pp. 805–822, 2019, ISSN: 21945365. DOI: 10.1007/978-3-030-19063-7_65.

[25] H. Park and J. Y. Jung, «SAX-ARM: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining», *Expert Systems with Applications*, vol. 141, p. 112 950, 2020, ISSN: 09574174. DOI: 10.1016/j.eswa.2019.112950. [Online]. Available: https://doi.org/10.1016/j.eswa.2019.112950.

[26] A. Capozzoli, T. Cerquitelli, and M. S. Piscitelli, *Enhancing energy efficiency in buildings through innovative data analytics technologies*. 2016, pp. 353–389, ISBN: 9780128037027. DOI: 10.1016/B978-0-12-803663-1.00011-5.

[27] C. Fan, F. Xiao, H. Madsen, and D. Wang, «Temporal knowledge discovery in big BAS data for building energy management», *Energy and Buildings*, vol. 109, pp. 75–89, 2015, ISSN: 03787788. DOI: 10.1016/j.enbuild.2015.09.060. [Online]. Available: http://dx.doi.org/10.1016/j.enbuild.2015.09.060.

[28] I. E. A. IEA, *Buildings A source of enormous untapped efficiency potential*, 2020. [Online]. Available: https://www.iea.org/topics/buildings (visited on 09/07/2020).

[29] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, «Classification: Basic Concepts, and Techniques», in *Introduction to Data Mining*, 2019, p. 114.

[30] C. Zhang, Y. Zhao, and X. Zhang, «An improved association rule mining-based method for discovering abnormal operation patterns of HVAC systems», *Energy Procedia*, vol. 158, pp. 2701–2706, 2019, ISSN: 18766102. DOI: 10.1016/j.egypro.2019.02.025. [Online]. Available: https://doi.org/10.1016/j.egypro.2019.02.025.

[31] C. Miller, Z. Nagy, and A. Schlueter, «Automated daily pattern filtering of measured building performance data», *Automation in Construction*, vol. 49, no. PA, pp. 1–17, 2015, ISSN: 09265805. DOI: 10.1016/j.autcon.2014.09.004. [Online]. Available: http://dx.doi.org/10.1016/j.autcon.2014.09.004.

[32] E. U. EU, «DIRECTIVE 2010/31/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 19 May 2010 on the energy performance of buildings», *Official Journal of the European Union*, 2010, ISSN: 00327867.

[34] H. Michael, C. Buchta, B. Gruen, K. Hornik, I. Johnson, and C. Borgelt, *Mining Association Rules and Frequent Itemsets Description: Package ' arules '*, 2020. DOI: 10.18637/jss.v014.i15>.Classification/ACM.

Further Reading

[36] Z. Yu, F. Haghighat, B. C. Fung, and L. Zhou, «A novel methodology for knowledge discovery through mining associations between building operational data», *Energy and Buildings*, vol. 47, pp. 430–440, 2012, ISSN: 03787788. DOI: 10.1016/j.enbuild.2011.12.018. [Online]. Available: http://dx.doi.org/10.1016/j.enbuild.2011.12.018.

[35] S. Milborrow, «Plotting rpart trees with the rpart.plot package», pp. 1–32, 2019.

[37] E. Chiabrera, S. Brandi, and A. Capozzoli, «Development of a tool for anomaly detection and power load forecasting: the case of Politecnico di Torino», PhD thesis, 2018.

[38] E. Petrova and P. Pauwels, «Semantic Enrichment of Association Rules Discovered in Operational Building Data», August, São Paulo, Brazil, 2020.

[39] A. Reinhardt and S. Koessler, «PowerSAX: Fast motif matching in distributed power meter data using symbolic representations», *Proceedings - Conference on Local Computer Networks, LCN*, vol. 2014-Novem, no. November, pp. 531–538, 2014. DOI: 10.1109/LCNW.2014.6927699.

[40] Z. Yu, B. C. Fung, and F. Haghighat, «Extracting knowledge from building-related data - A data mining framework», *Building Simulation*, vol. 6, no. 2, pp. 207–222, 2013, ISSN: 19963599. DOI: 10.1007/s12273-013-0117-8.

[41] M. Y. Ansari, A. Ahmad, S. S. Khan, G. Bhushan, and Mainuddin, «Spatiotemporal clustering: a review», *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2381–2423, 2020, ISSN: 15737462. DOI: 10.1007/s10462-019-09736-1.

[42] D. Lee and C. C. Cheng, «Energy savings by energy management systems: A review», *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 760–777, 2016, ISSN: 18790690. DOI: 10.1016/j.rser.2015.11.067.

[43] W. Li, C. Koo, T. Hong, J. Oh, S. H. Cha, and S. Wang, «A novel operation approach for the energy efficiency improvement of the HVAC system in office spaces through real-time big data analytics», *Renewable and Sustainable Energy Reviews*, vol. 127, no. March, p. 109 885, 2020, ISSN: 18790690. DOI: 10.1016/j.rser.2020.109885. [Online]. Available: https://doi.org/10.1016/j.rser.2020.109885.

[44] J. Roth, H. A. Brown IV, and R. K. Jain, «Harnessing smart meter data for a Multitiered Energy Management Performance Indicators (MEMPI) framework: A facility manager informed approach», *Applied Energy*, vol. 276, no. October, 2020, ISSN: 03062619. DOI: 10.1016/j.apenergy.2020.115435.

[45] M. S. Piscitelli, S. Brandi, A. Capozzoli, and F. Xiao, «A data analytics-based tool for the detection and diagnosis of anomalous daily energy patterns in buildings», *Building Simulation*, pp. 1–17, 2020, ISSN: 19968744. DOI: 10.1007/s12273-020-0650-1.

[46] I. Antonopoulos, V. Robu, B. Couraud, D. Kirli, S. Norbu, A. Kiprakis, D. Flynn, S. Elizondo-Gonzalez, and S. Wattam, «Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review», *Renewable and Sustainable Energy Reviews*, vol. 130, no. May, p. 109 899, 2020, ISSN: 18790690. DOI: 10.1016/j.rser.2020.109899. [Online]. Available: https://doi.org/10.1016/j.rser.2020.109899.

[47] M. Zhuang, M. Shahidehpour, and Z. Li, «An Overview of Non-Intrusive Load Monitoring: Approaches, Business Applications, and Challenges», *2018 International Conference on Power System Technology, POWERCON 2018 - Proceedings*, no. November, pp. 4291–4299, 2019. DOI: 10.1109/POWERCON.2018.8601534.

[48] M. Butler and D. Kazakov, «SAX Discretization Does Not Guarantee Equiprobable Symbols», *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1162–1166, 2015, ISSN: 10414347. DOI: 10.1109/TKDE.2014.2382882.

[49] M. Molina-Solana, M. Ros, M. D. Ruiz, J. Gómez-Romero, and M. J. Martin-Bautista, *Data science for building energy management: A review*, 2017. DOI: 10.1016/j.rser.2016.11.132.

[50] Y. Sun, J. Li, J. Liu, B. Sun, and C. Chow, «An improvement of symbolic aggregate approximation distance measure for time series», *Neurocomputing*, vol. 138, pp. 189–198, 2014, ISSN: 18728286. DOI: 10.1016/j.neucom.2014.01.045. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2014.01.045.

[51] N. D. Pham, Q. L. Le, and T. K. Dang, «HOT aSAX: A Novel Adaptive Symbolic Representation for Time Series Discords Discovery», vol. 5990, 2010, pp. 113–121. DOI: 10.1007/978-3-642-12145-6_12. [Online]. Available: http://link.springer.com/10.1007/978-3-642-12145-6%7B%5C_%7D12.

[52] A. R. Kandakatla, V. Chandan, S. Kundu, I. Chakraborty, K. Cook, and A. Dasgupta, «Towards Trust-Augmented Visual Analytics for Data-Driven Energy Modeling Towards Trust-Augmented Visual Analytics for Data-Driven Energy Modeling», in *IEEE VIS 2020 workshop on Trust and Experience in Visual Analytics*, 2020. DOI: 10.13140/RG.2.2.24817.30560.

[53] H. Wickham and D. Seidel, «scales: Scale Functions for Visualization», *R package version 1.1.0.*, 2019. [Online]. Available: http://cran.r-project.org/package=scales.

[54] A. Zeileis and T. Hothorn, «partykit : A Toolkit for Recursive Partytioning», *Journal of Machine Learning Research*, vol. 16, no. Hothorn, pp. 3905–3909, 2014, ISSN: 15337928. [Online]. Available: http://jmlr.org/papers/v16/hothorn15a.html.

[55] S. P. Kesavan, J. K. Li, C. Ross, D. Christopher, B. Robert, and K.-l. Ma, «A Visual Analytics Framework for Reviewing Streaming Performance Data», in *IEEE Pacific Visualization Symposium (PacificVis)*, 2020, pp. 206–215, ISBN: 9781728156972.

[56] J. Nembrini, R. Sánchez, and D. Lalanne, «Discussing the Potential of BMS Data Mining to Extract Abnormal Building Behaviour Related to Occupants' Usage», *Impact: Design With All Senses*, pp. 727–736, 2020. DOI: 10.1007/978-3-030-29829-6_56.

[57] T. Hothorn, K. Hornik, C. Strobl, and A. Zeileis, «party: A Laboratory for Recursive Partytioning», *R package version 0.9-0, URL http://CRAN. R-project. org*, no. 1994, p. 37, 2015. DOI: 10.1.1.151.2872. [Online]. Available: http://party.r-forge.r-project.org/.

[58] G. Lin, M. Pritoni, Y. Chen, and J. Granderson, «Development and implementation of fault-correction algorithms in fault detection and diagnostics tools», *Energies*, vol. 13, no. 10, pp. 1–20, 2020, ISSN: 19961073. DOI: 10.3390/en13102598.

[59] Y. Yan, «Package 'MLmetrics'», *Cran*, no. 1.1.1, 2016. [Online]. Available: https://cran.r-project.org/web/packages/MLmetrics/MLmetrics.pdf.

[60] P. Esling and C. Agon, «Time-series data mining», *ACM Computing Surveys*, vol. 45, no. 1, 2012, ISSN: 03600300. DOI: 10.1145/2379776.2379788.

[61] S. Brandi, A. Capozzoli, and M. S. Piscitelli, «Mining operational data for anomaly detection in buildings», PhD thesis, Politecnico di Torino.

[62] T. Therneau, B. Atkinson, and B. Ripley, «rpart: Recursive partitioning for classification, regression and survival trees. R package version 4.1-15», 2019. [Online]. Available: https://cran.r-project.org/package=rpart.

[63] M. Hahsler, I. Johnson, T. Kliegr, and J. Kuchar, «Associative classification in R: Arc, arulesCBA, and rCBA», *R Journal*, vol. 11, no. 2, pp. 254–267, 2019, ISSN: 20734859. DOI: 10.32614/RJ-2019-048.

[64] D. T. U. DTU, *Guidelines for avoiding plagiarism and self-plagiarism in PhD thesis writing*, 2020.

[65] L. Rcpp, «Package ' jmotif ': Time Series Analysis Toolkit Based on Symbolic Aggregate Discretization, i.e. SAX», pp. 1–20, 2020.