POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Energetica e Nucleare

Tesi di Laurea Magistrale

Application of data analytics processes for the detection of anomalous energy patterns in buildings

Relatori:

Prof. Alfonso Capozzoli

Ing. Marco Savino Piscitelli

Ing. Roberto Chiosa

Candidato:

Simone Dehò

Anno Accademico 2020/2021

Acknowledgements

First of all, I would like to express my gratitude the members of BAEDA Lab, especially my supervisor Prof. Alfonso Capozzoli and my co-supervisors Ing. Marco Savino Piscitelli and Ing. Roberto Chiosa, who guided me during the internship period and the months of development of this thesis. They provided me with valuable insight both from a technical and a methodological point of view, and especially the last aspect is something I will forever be thankful for.

A special thanks goes to my family and especially my parents, who helped me in many ways during this journey, always supporting my decisions and who were always there whenever I needed to talk.

Abstract

In recent years, the technological development in virtually every sector has often made it possible to consider real-world data – thanks to the ever-growing ease in collecting and storing these information – as an increasingly valuable resource to guide experts and decision-makers in a multitude of tasks. Among these, the analysis of energy consumption in large buildings is one of the areas of research that is subject to continuous innovation and refinements, as more and more data is made available through the installation of systems that ultimately aim at reducing inefficiencies by guiding the users towards a more "energetically responsible" behavior and by detecting potentially anomalous events during building operation. While collecting and storing data has seemingly become effortless, their analysis often still requires a certain degree of expert knowledge for intervention, due to the fact that it is basically impossible to define an unanimous criteria for "correct" or "incorrect" energy behavior at a whole building-level and it is even harder to investigate the individual causes of inefficiencies at a sub-meter-level starting from aggregate data. This work proposes a methodology for anomaly detection and diagnosis in large non-residential buildings that is built upon one of the newest and most promising techniques for time series analysis, the Matrix Profile (MP). Starting from an extensive review of the existing works that have contributed to the development of the Matrix Profile, its critical issues in the research field of energy data analytics are examined and a variation of the original technique, called Contextual Matrix Profile (CMP), is adopted for analysis on daily load profiles of power demand data measured by a monitoring system connected to a Medium Voltage/Low Voltage (MV/LV) transformation cabin of a university campus (i.e., Politecnico di Torino). Conventional supervised and unsupervised learning techniques, such as clustering and regression trees, are employed for the purpose of grouping together examined days with similar power demand profiles and set up the required input parameters for the CMP, while the anomaly detection step is based on the CMP output and on the combined results of two techniques – the "elbow" method and the boxplot – in order to find out the optimal number of days to be marked as "anomalous". The root causes of unexpected behaviors in anomalous days are then investigated by defining a metric that ranks sub-loads in terms of their impact on the anomaly at a meter-level. Climatic conditions are also taken into account with the aim of providing possible explanations for the behavior of sub-loads that, during their operation, are particularly influenced by factors related to seasonality, such as external air temperature.

Summary

List of Tables

List of Figures

1. Introduction

The topic of energy saving has gained widespread popularity in the last few decades, thanks to countless studies and researches which proved that reckless exploitation of the available resources on our planet, as a consequence of technological development, would inevitably lead to natural and humanitarian disasters, some of which are beginning to manifest even at the present day. Worldwide awareness campaigns on this theme quickly became part of people's everyday lives and a global effort is being made to counteract and prevent the worrying future that has been foreseen.

In this context, the building sector is certainly among the most energy-intensive ones and growing needs to ensure occupants' comfort, especially in large buildings, go hand in hand with increasing power demand: according to [1], commercial and residential buildings, together, are responsible for 41% of primary energy consumption in the United States and 40% in the European Union. Although these data may be surprising for many who underestimate the impact of buildings' power demand, the future seems promising: the technological solutions to enable energy efficiency in buildings, such as Energy Management and Information Systems (EMIS), are rapidly evolving and being refined and more and more decision-makers are starting to appreciate their long-term benefits, even when the initial expense for installation – usually the main deterrent to their adoption – is substantial. EMIS are a family of analytics systems - acting either at a meter-level (the action is applied to the whole building) or at a system-level (the action is applied to the single component) - that include Energy Information Systems (EIS), Fault Detection and Diagnosis (FDD) and Automated System Optimization (ASO) tools [2]. EMIS comprise all the software and hardware that collect and store building data, in order to control and optimize building energy use and efficiency. The ultimate goal of these tools is to bridge the gap between expected building energy performance and real performance. This "energy gap" is usually the result of a multitude of factors, such as unexpected occupant behavior, suboptimal/wrong settings of the control system, malfunctioning or inadequate equipment/components and so on. According to [1] , EMIS can enable economic savings on the order of 10-20%. The previously mentioned EMIS all play a key role towards optimal building operation and management [2]:

- EIS comprise both hardware and software tools and their main aim is to acquire meter-level data at regular intervals, store these data and ultimately analyze them and display various kinds of information to the end users in order to guide their actions and behaviors;
- ASO software aims at optimizing Heating, Ventilation and Air Conditioning (HVAC) systems' operation, to maintain occupants' comfort while minimizing the amount of energy spent in the processes. This kind of action is possible thanks to a two-way communication with the BAS (Building Automation Systems), whose data is continuously analyzed by the ASO. Optimal set-points are then returned to the BAS, in order to fine-tune control parameters;

- FDD software also acts at a system-level; its goal is to automatically detect faults in buildings' systems and suggest possible causes for the unexpected behaviors.
The concept of FDD often goes hand in hand with that of Anomaly Detection and Diagnosis (ADD), which is mostly used to indicate the same kind of process at a larger (usually whole-building) scale. While the solutions related to the task of detection and diagnosis of anomalies at a single component-level – which are generally based on simple principles, such as rule-based ("if-then" logic) diagnostics – have already reached a certain level of maturity thanks to the huge amount of system-level data collected by Building Automation Systems, one of the main challenges that sector experts are still facing is extending such processes to the meter-level: the aggregate data, related to total building energy consumption, that is measured and collected, is in most cases not explanatory of what happens at a lower level (e.g. in a single room), thus requiring the need for measurements at sub-loads level. Even when these measurements are provided, the task of attributing an anomalous behavior at a meter-level to a specific sub-load is undoubtedly challenging and difficult to automate, since different building zones are subject to different occupational patterns, operational schedules and so on.

1.1. Structure and contribution of the thesis

Numerous artificial intelligence - based techniques for anomaly detection have emerged in the last few years [3] [4], and many of them have been applied to the buildings sector [3]. In this context, however, the exploration of one of the most recent and promising methods for the analysis of ordered series of data points, the Matrix Profile (MP) – introduced in 2016 by Yeh et al. [5]- has been limited: while the MP has been the subject of various research efforts (documented in the Literature Review section of this work) in its relatively brief life, its applications for the study of buildings consumption are still extremely limited [6]–[8]. This thesis proposes an innovative framework, based on a variation of the original MP technique, called Contextual Matrix Profile (CMP) [9], for the detection of anomalies at buildings' meter-level and their diagnosis at the sub-loads level. The methodology is applied to a case study that analyzes one year of power demand data measurements from a monitoring system connected to a Medium Voltage/Low Voltage (MV/LV) transformation cabin of the university campus of Politecnico di Torino. The rest of the work is structured as follows.
Chapter 2 presents a literature review that covers topics such as Energy Information Systems, Anomaly Detection methods and, more importantly, offers an extensive overview of the most important Matrix Profile – related works up to date. Chapter 3 focuses on the concepts of Matrix Profile, together with the desirable properties and critical issues this method presents, and Contextual Matrix Profile, explaining why this variation of the original technique has been chosen for this work. The fundamental notions for the understanding of both these methods are introduced. In Chapter 4, a description of the methodological steps followed is provided, from the definition of the input parameters for the CMP to the approach for the anomaly diagnosis at sub-loads level. Chapter 5 presents the essential information to define the case study analyzed,

highlighting the pre-processing steps necessary for the dataset to be examined; a first characterization of the electrical loads subject of this work is also performed. In Chapter 6, the results of the analysis on the case study are presented and discussed, while Chapter 7 offers closing thoughts on the whole framework introduced in this work, together with future perspectives that can be persecuted to improve the methodology for the diagnosis at sub-loads level. Finally, Chapter 8 contains the bibliographic references cited in the thesis and in Chapter 9 the remaining Figures, discussed throughout the whole work and not directly inserted between text, are collected.

2. Literature Review

In this Chapter, existing publications dealing with the main topics covered in this work are examined, starting from a brief introduction to Energy Management and Information Systems and their growing importance for the purpose of energy efficiency and then analyzing the main techniques for anomaly detection in buildings up to date. Finally, an extensive overview of Matrix Profile - related research papers is presented.

2.1. Energy Management and Information Systems

As mentioned in Chapter 1, Energy Management and Information Systems are a family of software and hardware tools that allow for significant energy savings in buildings when correctly implemented, thanks to their action in the operations of data collection, data analysis and systems control. A practical example of these beneficial effects is reported in [2], where the results of an EMIS adoption campaign in a variety of buildings with different sizes and designated uses, for a total gross floor area of over 185 square feet, are examined; different participants implemented different kinds of EMIS – e.g. EIS alone or in conjunction with FDD – with resulting median cost savings of 0.2$ per square foot per year and 5% per year. The same authors published an updated version of this report around two years later, in 2019 [10], where the median cost savings were documented to have increased respectively to 0.19$/square foot and 7%/year, with an upfront median base cost for EMIS installation of 0.03$/square foot and annual recurring costs of 0.02$/square foot for software and 0.03$/square foot for labor. This kind of example shows how early adoption of these systems can be a far-sighted decision, especially considering that technological refinements to both hardware and software tools are constantly being made, thanks also to the growing amount of open access data [11]–[13] that is available to researchers and decision-makers.
Among the different types of EMIS, the first classification presented in [2] distinguishes them based on the "metering depth"; this aspect is also considered in [14], where the relationship between the depth of sub-metering and the energy savings achieved was analyzed for a building portfolio: it was found that, as a general trend, deeper sub-metering allows for cost savings that surpass the expenses for the implementation of additional metering infrastructures.

2.2. Anomaly detection in buildings

While system-level monitoring, by the means of FDD tools, is certainly able to pinpoint the exact source of unexpected behaviors, such deep-metering analysis requires significant economic and technical efforts. On the other hand, applying anomaly detection and diagnosis processes at the whole building-level - only by analyzing aggregate data - is still difficult, for the reasons already mentioned in the previous

Chapter, even though modern EIS tools allow to easily collect and make available meter-level data. of This issue is explained very clearly in [15], where the authors present an EIS tool that allows to perform an initial meter-level anomaly detection and then sub-meter diagnosis thanks to the use of Association Rule Mining (ARM). In this work, the goals of Anomaly Detection and Diagnosis (ADD) in buildings are summarized as: identification of energy consumption patterns at meter-level that are representative of the typical day, detection of anomalous load profiles based on the difference with the typical ones and finally diagnosis of the root causes of anomaly thanks to a sub-meter investigation performed on the main sub-loads.
The critical aspects regarding the process of anomaly detection in buildings' energy consumption patterns are also the core topic of [3], which presents an extensive review of the existing artificial intelligence-based techniques for this task, together with future perspectives and research directions. The rest of this section will mostly refer to this work - which deals with the issue of detection techniques with great detail, offering precious insight for a complete understanding of the lesser known aspects of this vast topic - in order to briefly introduce each one of the main methods for anomaly detection.
According to the classification provided by the authors, anomaly detection techniques can be divided in 5 main categories: unsupervised detection techniques, supervised detection techniques, ensemble methods, feature extraction techniques and hybrid learning methods.
The goal of unsupervised detection is to extract unusual patterns without using previously known information about the data and assuming that anomalous observations represent a small portion of the total data. This kind of process therefore usually aims at modeling the behavior in normal occurrences and detecting the abnormal ones as outliers. Among unsupervised techniques, the main ones are:
- clustering, which splits observations in groups marked as "normal" or "anomalous". The most popular clustering techniques are k-means, fuzzy C-means and entropy-based methods;
- one-class classification or one-class learning (OCL), which considers initial data to be part of two groups (like clustering, normal and abnormal) and then models classification algorithms while the abnormal group can be either absent or not well-defined [16]. This makes OCL a classification problem that is particularly challenging since the training data that belongs to one of the labels can be poorly represented or not present at all. In this category one-class neural networks (OCNN), one-class support vector machines (OCSVM), one-class convolutional neural networks (OCCNN) and one-class random forests (OCRF) can be found;
- dimensionality reduction, which is a technique for classification that usually presents low computational cost due to the fact that the less significant or redundant patterns are not considered [17]. Principal component analysis (PCA), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and multiple discriminant analysis (MDA) are the main methods that belong to this class.
Supervised detection techniques requires training of the machine learning classifiers by the means of annotated datasets, where each measurement is explicitly marked as normal or as anomalous. The main barrier to the widespread adoption of these methods

is the absence of annotated datasets for power consumption measurements. The main supervised techniques for anomaly detection are:
- neural networks, circuits of neurons that solve artificial intelligence problems. In this category, for example, recurrent neural networks (RNN), convolutional neural networks (CNN) and multi-layer perceptron (MLP) can be found. This family of techniques can prove particularly useful when the labeled data is noisy or the label is not perfectly clear;
- regression techniques, which identify relationships between power variable classes with the aim of producing model parameters that allow for the prediction of the generation of anomalous power measurements, also based on previously collected abnormal data. Among regression methods, the main ones are support vector regression (SVR), autoregressive models and regression trees;
- probabilistic models, which represent one of the most important machine learning tools and are based on probabilistic relationships to build real-world models. Bayesian networks, naive Bayesian algorithms and statistical models all belong to this class;
- traditional classification, a category that groups together all the models whose aim is to detect to which power consumption category a new power measurement belongs to, with the usual training set containing both normal and abnormal samples. This last class includes k-nearest neighbors (KNN), support vector machines (SVM), decision trees and logistic regression.

Ensemble learning methods split the initial group of power observations in multiple subsets and simultaneously apply different models in order to identify abnormal behavior. To obtain definitive conclusions about an observation being normal or not, anomaly scores are then employed. In this category, the following techniques can be found:
- boosting, which is a set of meta-algorithms aimed at reducing bias and variance of unsupervised learning, where weak classifiers are substituted by strong ones. Bootstrap, gradient boosting machine (GBM) and gradient tree boosting (GTB) are all part of this subset of techniques;
- bagging or bootstrap-aggregating, also a set of meta-algorithms that have the goal of improving the accuracy and stability of weak classifiers. Bootstrap aggregation and random forests represent the main methods in this group.

Feature extraction techniques are aimed at improving anomaly detection methods' performances representing the data observations in novel spaces such as high-dimensional ones, utilizing measures and functions such as distances or densities to separate normal observations from abnormal ones and representing the consumption process through new representation structures, such as graph-based representations. This category includes:
- distance-based techniques, which detect abnormal consumption patterns by evaluating each pattern on the basis of its distance to its neighbors (denser regions correspond to a situation of normality and vice versa);
- time-series analysis, aimed at detecting anomalies based on the shape of the ordered collection of data points; such anomalies can be spikes, drops, bumps and so on. Short-term time-series (STTS) analysis and rule-based algorithms represent the main detection techniques found in this category;

- density-based methods, that follow a logic similar to that of distance-based techniques, taking into consideration density instead of distance. Local outlier factor (LOF), cluster-based local outlier factor (CBLOF) and density-based spatial clustering of applications with noise (DBSCAN) all belong to this group;
- graph-based techniques, which need data to be transformed into a graph-based structure before the analysis. The main methods that can be found in this category are graph-based anomaly detection (GBAD) algorithms and parallel graph-based outlier detection (PGBOD).

Finally, hybrid learning (or semi-supervised) techniques make use of available annotated observations belonging to the class of "normal" data, in order to construct models that are able to correctly classify a new normal observation, thus adopting a strategy that does not involve the recognition on abnormal patterns. Semi-supervised support vector machines (semi-SVM) are an example of these kinds of methods.

2.3. The Matrix Profile

This section presents an overview of the existing works in scientific literature that are related to the Matrix Profile (which will also be referred to as "MP" at times, for the sake of brevity), a data analysis technique – that, according to the classification presented in the previous section, belongs to the class of time series analysis – whose concepts are at the core of this work, and its applications and modifications/improvements throughout the years. The reader that is not familiar with this technique or with the terminology used is referred to Chapter 3 for a brief introduction to the fundamental concepts necessary for its understanding.

Since the introduction of the Matrix Profile in 2016 [5], the literature about this technique has quickly expanded, starting from what can be called the "fundamental" literature (all the papers belonging to the collection that can be found on the official Matrix Profile website [18]) to a large number of scientific papers that were published in the last few years that either make use of the MP in a certain research field or take the original method and introduce a degree of novelty to it.

This literature review is divided in three parts:

- In the first one, the reader will be given a brief overview of the evolution of the algorithms for MP computation and motif discovery, the most common task among all the Time Series All-Pairs Similarity Search (TSAPSS) applications;
- In the second one, the most interesting papers published throughout the years that deal with applications of the MP to various research fields will be reviewed;
- The third and final part explores the efforts that had been made in the past years to apply the MP to the building and energy field.

2.3.1. The evolution of the algorithms for Matrix Profile computation and motif discovery

The first algorithms for the calculation of the Matrix Profile were STAMP ("Scalable Time series Anytime Matrix Profile") and its incremental variant STAMP*I* (STAMP Incremental), introduced by Yeh et al. in [5]. While STAMP needs the entire time series for the MP to be computed, the incremental version can work with streaming data. STAMP is built upon Mueen's ultra-fast Algorithm for Similarity Search (MASS), which makes use of the sliding dot product between subsequences calculated using the Fast Fourier Transform (FFT) algorithm. The time complexity of STAMP is $O(n^2 \log n)$, while the space complexity is $O(n)$. At the time it was introduced, STAMP was significantly faster than comparable methods for TSAPSS: as an example, in [5] the authors claimed that to produce exact results on a self-join with subsequence length $m = 256$ and time series length $n = 2^{18}$, STAMP took 1.17 hours; other rival algorithms took as long as almost 51.7 hours, and several concessions were made to them in order to be able to compare them with STAMP: these numbers lets the reader appreciate how revolutionary STAMP was for the all-pairs-similarity-search task at the time of its introduction.
Shortly after STAMP, Zhu et al. [19] presented STOMP ("Scalable Time series Ordered-search Matrix Profile") and its GPU-accelerated version GPU-STOMP. The main idea behind the STOMP algorithm was that that in some domains, such as seismology, the "anytime" property is not necessary; therefore, by giving up this property the time series join can be calculated at least an order of magnitude faster than STAMP. For example, the authors claim that STAMP would take more than 20 years to produce a full and exact Matrix Profile of a seismology time series sampled at 20 Hz for about 2 months, while GPU-STOMP can execute this task in around 12 days. The main novelty that STOMP introduces is the ordered search in the phase of distance profiles evaluation, exploiting the computational dependency between consecutive distance profiles; STAMP, instead, uses random search in order to be able to provide the "anytime" property. The time complexity of STOMP is $O(n^2)$: this means achieving a speedup factor of $O(\log n)$ over STAMP, that becomes more and more important the longer the time series gets (to give the reader an idea: when dealing with thousands of data points the difference between the two algorithms is negligible; if the dataset is around the magnitude of millions of data points, STOMP can provide an order-of-magnitude speedup). To further speed up the process, the authors also introduced GPU-STOMP, which takes advantage of the processing power of the Graphic Processor Unit to perform multiple computations in parallel. Not only a single GPU can be used: for machines that contain two or more graphic devices, the process can be further parallelized. STOMP, however, does have its disadvantages and the most evident one is the lack of the anytime property: while in certain domains this property may not be fundamental, it is often desirable to be able to produce a fast-converging approximate solution (e.g. executing only 10% of the full computation) and STOMP does not allow this.
In 2018, Zhu et al. [20] introduced SCRIMP++, an algorithm for motif discovery that is an improvement of both STAMP and STOMP and takes "the best from both worlds", combining the speed of the STOMP algorithm and maintaining the anytime property of

STAMP, while being able to exploit GPUs and other High Performance Computing platforms for the purpose of calculation speedup. SCRIMP++ is an algorithm consisting of two parts, the first one called PreSCRIMP and dedicated to preprocessing operations and the second one called SCRIMP which is an $O(n^2)$ anytime algorithm. While STOMP evaluates the distance matrix (the matrix obtained by joining all the distance profiles) in a row-by-row in-order logic, the SCRIMP algorithm evaluates the diagonals of the distance matrix in a random order, allowing a fix for an undesirable property due to the nature of STOMP, which is that the motifs at the end of a time series cannot be discovered early due to the in-order computation. PreSCRIMP is needed to produce a very close approximation to the Oracle Matrix Profile (the exact MP, obtained by running the computation until 100% completion) with a significative reduction on the original $O(n^2)$ computational time, by taking advantage of a property of time series subsequences called "Consecutive Neighborhood Preserving (CNP) Property": essentially, this property guarantees that a set of consecutive subsequences will find another set of consecutive subsequences as its nearest neighbor thanks to the overlapping of consecutive subsequences. This preprocessing step fixes an issue that is intrinsic of SCRIMP, which is its dependence, in terms of performance, on the number of motifs contained in the data: the more motifs there are, the faster SCRIMP is. After running PreSCRIMP, the approximated Matrix Profile obtained is refined over and over with SCRIMP until convergence to the exact solution. Both SCRIMP and PreSCRIMP can be interrupted at any moment, thus making the anytime property valid. When comparing performance to both STAMP and STOMP, SCRIMP++ shows faster convergence in various test scenarios with respect to STAMP, while the runtimes are similar to STOMP's results.

SCAMP (SCAlable Matrix Profile) was introduced in 2019 by Zimmerman et al. [21] with the purpose of being able to perform motif discovery on extremely large datasets in domains such as seismology or astronomy. This new framework allows working with datasets that do not fit entirely into GPU memory, thanks to the use of cloud computing. The SCAMP framework can be used by a cluster (a set of computers that are inter-connected and work together to perform certain computationally intensive tasks) with a host (a "master" machine or server) and a number of workers that follow the host's orders. Workers can be, for example, CPUs or GPUs. SCAMP can be deployed on cloud platforms such as Amazon Web Services (AWS). This technique allowed the authors to perform motif search on seismic datasets with over one year of continuous earthquake data points, for a total of a quintillion (10^{30}) exact pairwise comparisons.

In 2019, Zimmerman et al. [22] presented LAMP (Learned Approximate Matrix Profile), a model able to predict, in constant time, the MP values that would be assigned to incoming subsequences when dealing with streaming data. The LAMP model is able to tackle the issue of untenability (due to the increasing time required for previously existing algorithms to compute the MP as more and more data is seen) of MP computation/update in domains such as seismology, entomology and neuroscience where the sampling rate of data is faster than the order of 1 Hz.

Also in 2019, Akbarinia and Cloez [23] introduced two algorithms for MP computation:

- The first one is called AAMP and it computes the Matrix Profile using "pure" (non-normalized) Euclidean distance; the choice of using this kind of distance measure comes from the observation that, for certain types of datasets, the z-normalization process is not always beneficial for knowledge discovery. Such datasets are, for example, those that include long subsequences of a constant value: in these cases the subsequence's standard deviation is equal to zero, which means that the z-normalized distance would become infinite.
  AAMP has time complexity of *O(n x (n-m))* and space complexity of *O(n)* and it has the desirable properties of other algorithms of being anytime, exact and incrementally maintainable. Performance evaluation shows that AAMP is significantly faster than the SCRIMP++ algorithm.
  The authors also provide an extension of this algorithm to p-Norm distance, which is defined as:

$$DP_{i,j} = \sqrt[p]{\sum_{l=0}^{m-1} \left(t_{i+l} - t_{j+l}\right)^p}$$

  The p-Norm distance is a more general case of the Euclidean distance, where p=2;
- The second one is called ACAMP and it is built upon the same logic of AAMP, but for z-normalized distance calculations. Space and time complexity are the same as AAMP and also the performance evaluation, with comparison to SCRIMP++, shows positive results especially as n increases.

To conclude this section, it is necessary to mention that the techniques for MP calculation and motif discovery are continuously evolving, together with the access to more and more processing power thanks to cloud computing and high performance computation devices, often improving previous successful techniques and algorithms: works such as those of Onwongsa and Ratanamahatana [24], Kalantar et al. [25], Romero et al. [26] and Fernandez et al. [27] are just a few examples that are well-representative of this process of continuous computational improvement.

2.3.2. Various Matrix Profile applications: an overview

Since the Matrix Profile is a recently discovered technique, the literature about it is still quite sparse if compared to other data analytics methods; nevertheless, the existing papers are worth taking a look at since they often introduce novelties to the original process. This section will cover the most interesting works regarding the Matrix Profile published from 2016 to this day.

Among the collection of the "fundamental" papers [18], many of them are dedicated to applications of the MP technique to existing problems or processes regarding time series; this overview will start from them.

In 2017, Yeh et al. [28] introduced SDTS (Scalable Dictionary learning for Time Series), an algorithm that can learn a "dictionary" (a set of shapes, each one associated with a

specific event and therefore a specific label) from weakly labeled data in real-world settings. This kind of tool helps in the process of time series data classification, where common events such as noisy labels (false positives/negatives), label slop (misalignment) and skewed class distribution in the training data set make the labeling of incoming data process challenging. The SDTS algorithm uses the Matrix Profile as its building block, from which the dictionary is derived: the key concept is that subsequences with lower MP values are repeating subsequences and must be corresponding to certain recurring events to be labeled. The authors claim that the dictionary built thanks to this process offers "superhuman" performance (a human would not be able to solve the problem by "eye") for some of the case studies analyzed.
Also in 2017, Dau and Keogh [29] first presented the concept of Annotation Vector (AV); the AV is a meta-time series that can be used for the task of motif discovery in certain domains where expert domain knowledge can be useful to "correct" the results of the raw Matrix Profile algorithm. The motivation behind this work is that, in some datasets, the MP algorithm can "prioritize" as motifs certain repeated patterns that the domain expert knows are not significant. For example, in ECG data it is often possible to find a calibration signal, that lasts for a few seconds and may be repeated after start-up due to loss of contact between the sensors and the patient: this kind of signal is artificial and consists of a saw-toothed wave almost perfectly repeated, which often results in a wrong classification of these subsequences as top-1 motifs. The AV is a time series parallel to the original one, with values from 0 to 1 that serve as "weights" to be applied to the MP in order to produce a "Corrected Matrix Profile". Wherever the domain expert knows there is a certain type of data in the time series that needs to be "damped", he will act on the AV values in that time period so that the original MP values increase in order to reduce the chances of finding motifs in that region. The AV is therefore a simple yet effective way to introduce domain expert knowledge in a domain-agnostic technique such as the Matrix Profile and can be useful in many research fields.
Another paper published in 2017 by Yeh et al. and belonging to the "fundamentals" of the Matrix Profile literature is [30]; this work introduces an algorithm, called mSTAMP (multidimensional-STAMP) for the discovery of multidimensional motifs, which are repeating patterns across certain dimensions in a group of time series that can be analyzed "in parallel" since they capture different aspects of the same phenomenon/event: an example could be the time series of body parts movements through sensors applied on the arms, on the legs and so on. The authors claim that multidimensional motifs do not involve all the dimensions available, but only a subset of them: finding out which dimensions are the interesting ones is the perhaps the most challenging part in the task of multidimensional motif discovery. When trying to find motifs in all the dimensions, the user would most likely end up with unsatisfactory results; given a time series with d dimensions, the subset of the k interesting ones can be found either by deciding a priori k and letting the algorithm find the best dimensions to include ("guided search"), by deciding a priori k and explicitly include/exclude certain dimensions ("constrained search") or by letting the algorithm find the best "natural subset" k for motif discovery ("unconstrained search"). The mSTAMP algorithm is quite complex and will not be discussed here; the key concept is that it is built on top of the

original STAMP algorithm, all the desirable properties of the Matrix Profile are still present and the basic logic for motif discovery is the same as STAMP. The authors provide various examples of successfully applying the mSTAMP algorithm to domains such as motion capture, music processing, electrical load measurement and physical activity monitoring.

In [31], Zhu et al. apply the MP technique to time series chains, which are defined as "a temporally ordered set of subsequence patterns, such that each pattern is similar to the pattern that preceded it, but the first and last patterns are arbitrarily dissimilar"; chains therefore represent an "evolution" of a system and can help predicting future events. A good example of a time series chain is represented by the evolution of the search volume of a certain keyword in web search browsers year after year and during the same period (e.g. in the month of November, searches related to "Black Friday" usually have a spike; the same is for certain brands that are usually more active during certain times in the year). Time series chains mainly consist of two types: unanchored (the interest is in finding the unconditionally longest chain) and anchored (the chain should start with a certain subsequence). The authors developed two of algorithms for time series chains recognition:

- ATSC (Anchored Time Series Chains), with time complexity of $O(n)$;
- ALLC (All-Chain set, the set of all anchored TS chains within a time series that are not included in another chain), with time complexity of $O(n)$ as well.

These algorithms are built on top of LRSTOMP, which is another algorithm introduced by the authors; it is based on STOMP and it computes the Left and Right Matrix Profiles, which are also concepts introduced in this paper (they are the MPs computed applying a nearest neighbor search only on subsequences that are, respectively, on the left or on the right of the query). The authors provide empirical evaluation results of this framework to various domains where chains are present in time series, such as hemodynamics, animal and human movements and web query volume.

The topic of time series chains is also at the core of [32], where Imamura et al. face the problem of ranking the top-k chains, introducing a measure of significance of the chain using two quality metrics: "directionality" and "graduality".

The last "fundamental" paper published in 2017 is [33], by Gharghabi et al.. In this work, the authors deal with the challenge of "unsupervised semantic segmentation", which is the division of a time series in regions that show a common internal behavior or feature (for example, a wave signal that first is a sine wave, then becomes a saw-toothed wave, then again a sine wave and so on). The authors introduce FLUSS (Fast Low-cost Unipotent Semantic Segmentation) and its variant for streaming data, FLOSS: these algorithms are built on top of the concept of Matrix Profile and especially the MP indexes play a key part in the process of semantic segmentation. The main idea is to build a "arc curve", that is a meta-time series that takes into account how many nearest neighbor arcs (arcs that connect a subsequence with its nearest neighbor) cross over each data point location. The less arcs cross a certain location, the more likely it is that in that point a regime change occurs, since logic suggests that most subsequences would have a nearest neighbor within their host regime. The experimental evaluation involving biological and mechanical time series shows promising results and proves that FLUSS is able to achieve

"better-than-human" performance in certain cases and it is robust to the only parameter to be chosen, which is the subsequence length.
Another topic, that is key for the work reported in two papers, is that of variable-length motifs; since the Matrix Profile requires the choice of the subsequence length, the motifs found as a result will be of the same length. This can be satisfactory in domains where the data structure subdivision is evident (e.g. data repeating periodically, like heartbeats), however in many cases it would be beneficial to explore more than one choice of motif length, and doing so manually results in an expensive process. VALMOD (Variable Length Motif Discovery) is an algorithm presented by Linardi et al. [34] in 2018 that is able to find motifs in a time series given a user-decided subsequence length range $[l_{min}, l_{max}]$ that is relatively small. In order to compare and rank motifs of different lengths (to find the most interesting ones, irrespective of their length) a novel length-normalized distance measure, that consists in the Euclidean distance multiplied by the square root of $1/l$, is adopted. The Pan Matrix Profile (PMP) was introduced in 2019 by Madrid et al. [35]: it is a data structure that contains all the MP information for all subsequences of all possible lengths in a large range r, eliminating the need to define the subsequence length as a parameter required from the user. The algorithm that computes the PMP is called SKIMP (Scalable Kinetoscopic Matrix Profile) and has time complexity of $O(n^2r)$ and space complexity of $O(nr)$ and allows for approximate solutions that produce satisfactory results even with a small fraction of the full convergence time. The motifs of different lengths are also ranked to provide a comparison between them in order to find the top-K length-agnostic motifs.
A "complementary" concept to variable-length motifs is that of "discords of all lengths", which is the focus of [36]: the authors claim that the effectiveness of discord discovery through the MP technique is often undermined by the sensitivity to the parameter of user choice, the subsequence length; the algorithm they developed, MERLIN, is able to solve this issue by applying a logic that is quite similar to the ones of the works cited above, where a subsequence length range is passed as input to the process.
In [37], Gharghabi et al. introduce a novel distance measure based on the Matrix Profile called MPdist, that could be seen as an alternative to the traditional distance measures, such as the Euclidean distance or the Dynamic Time Warping (DTW), commonly adopted in algorithms. The main advantages of this new distance measure over Euclidean distance or DTW are the robustness to missing values and spurious regions, the invariances to phase, order, linear trend and stutter, the possibility to compare time series of different lengths and the fast computational time which allows great scalability. MPdist is built on top of the Matrix Profile technique and it is possible to understand why when considering the way this distance measure evaluates similarity: two time series are considered similar if they share many similar subsequences under Euclidean distance, no matter how these matching subsequences are ordered; this kind of evaluation is clearly made possible thanks to the Matrix Profile and, in particular, to a newly introduced structure called "Join Matrix Profile" that evaluates the Euclidean distance of subsequences in a time series A with their nearest neighbors in another time series B "from the point of view of both time series one after the other": the join MP can be seen as an array containing the Euclidean distance for each pair in the *AB-BA*

Similarity Join, which is the set that contains pairs of each subsequence in *A* with its nearest neighbor in *B* and vice versa .

One parameter, called *k*, is required to be set beforehand and its choice is not trivial: *k* is the k^{th} smallest value in the Join MP and the MPdist is equal to this value, in order to avoid choosing the smallest or the largest values as they may be sensible to spikes (largest) or offer little discrimination between time series (smallest). Experimental evaluation on entomological and power data was performed to confirm the effectiveness of MPdist.

Thanks to MPdist, Imani et al. [38] were able to carry out their work on "time series snippets", which are defined as sequences of points in a dataset that show representative data. As Ghargabi et al. write in [37] about the work of Imani et al., "The authors argue that their definition of time series snippets is enabled by the unique properties of the MPdist; no other distance measure would work for their task". Snippets are different from time series motifs, since motifs do not take into account "coverage", which represents how many time certain sequences are repeated in the whole dataset, but only "fidelity" of conservation. However, snippets can be ranked in a way that is similar to motifs: the k^{th} snippet is the one able to explain the k^{th} most time series data. The authors introduce an algorithm called "Snippet-Finder" that is able to find the top-k time series snippets in a dataset even when the data is corrupted by many undesirable factors such as noise, wandering baseline and so on. The time complexity of the algorithm is $O(n^2 \times (n-m)/m)$ and the space complexity is $O((n-m) \times k)$, where *k* is the number of snippets.

Empirical evaluation is carried out on datasets from various domains such as medicine, human behavior, electrical power demand and biology.

In [39], Zhu et al. present an algorithm, called STUMP (Scalable Time Series Ubication Matrix Profile), with the aim of focusing only on certain periods of a time series in an automated way, which allows for significant speed-up in many research tasks. The goal of this algorithm is to produce a "meta" Matrix Profile that can be precomputed and is incrementally maintainable; this data structure can then be used to quickly compute both a standard MP including a certain region of a time series and a standard MP corresponding to a time series with a certain region excluded. A real-life case study is presented for a time series with a three-years length, showing that computing the full MP hides certain discords, related to recurring annual events in the three years, that are evident when computing the monthly MPs. The authors claim that with their algorithm all the monthly Matrix Profiles can be computed in less than a second. Overall, the approach can prove very useful when there is a need to analyze motifs/discords in a dataset specifying a query range; being able to compute all the MPs in a very fast way can allow user-interactive comparison of the MPs corresponding to various ranges and their differences in terms of motifs/discords that arise from a particular setting.

To conclude the first part of this section, the other papers belonging to the "fundamentals" collection will be briefly analyzed, all of which apply concepts related to the Matrix Profile technique to a variety of already known topics.

In [40], Yeh et al. introduce a framework for the application of Multidimensional Scaling (MDS), which is a family of techniques used for data exploration and visualization, based on the principle of Minimum Description Length (MDL) that has the concept of

"compression", in terms of bits in memory, as its core; here, the MP serves mainly as an input for one of the algorithms used during the process. An interesting new idea that emerges from this paper is that of "salient subsequences", which are subsequences that produce meaningful low-dimensional (e.g. 2D thanks to MDS) projection, due to the fact that they are the subsequences that offer better compression of the data. The authors claim that trying to explain all the subsequences would lead to unsatisfactory results, and that only salient subsequences should be considered.

The work by Zhu et al. [41] focuses on the discovery of time series motifs in the presence of missing data and introduces an algorithm, called MDMS (Motif Discovery with Missing Data) for this purpose, built on top of the Matrix Profile structure, which inherits many desirable properties from this technique such as being parameter-free (subsequence length is the only parameter) and simple, incrementally maintainable, easily parallelizable and allowing for approximate solutions; MDMS has the same space and time complexity as STOMP.

In 2019, Kamgar et al. [42] introduced the concept of "time series consensus motifs", defined as "repeated structures in sets of time series data"; consensus motifs can be imagined as "blocks" like the ones present in DNA strings, from which the name "consensus motif" is derived. The authors developed an algorithm called "**Ostinato**" for fast consensus motifs search, that is limited to a "batch" (non-anytime) version but has been proven to be robust even in the presence of noisy or spurious data and is able to find conserved motifs in groups of datasets with tens of millions of data points with a satisfactory low runtime. The idea presented in this paper also appears in [43], where the anytime version of Ostinato is introduced, alongside an algorithm that allows to detect repeated structures (not corresponding to the classic definition of motifs) in a single time series, a task that Ostinato is not able to perform.

In [44], Imani and Keogh first described "time series semantic motifs" as subsequences that share similarities in some of their parts, such as "prefixes" and "suffixes", at the beginning and at the end, respectively. An example of semantically equivalent events are a single-pump handshake and a three-pump one. The authors introduce an algorithm, called "Semantic-Motif-Finder", that is able to capture this kind of phenomena in time series data and requires a maximum "don't care" length r and a prefix/suffix length s as input parameters. The concept of "Semantic Matrix Profile" is also introduced, as a MP based on the distance between each semantic motif and its nearest neighbor.

The work by Alaee et al. [45] applies Dynamic Time Warping (DTW) to time series motif discovery. While DTW is unanimously recognized as superior to Euclidean Distance in a variety of settings, the main barrier to its diffusion for the motif discovery task is the computational cost and the difficulty to combine speed-up techniques for DTW and Matrix Profile. The authors introduce SWAMP (Scalable Warping Aware Matrix Profile), an algorithm that makes it possible to apply DTW motif discovery to large datasets; they also show that some of the motifs found with this technique cannot be found with classic Euclidean Distance.

As one can appreciate from the overview of the papers cited above, the literature that represents the "foundation" of the Matrix Profile technique is quite diverse in terms of

topics covered; the most interesting works from the rest of the literature will be covered in the rest of this section.

In [46], the topic of multidimensional motifs is further explored. The authors introduce the MUSTAMP/MUSTOMP algorithms, that are used for MP computation for AB joins (the well-known mSTAMP method for this task is only able to compute time series self-joins). These algorithms are incrementally maintainable and can also be used when the two multivariate time series are of unequal length. A top-*k* most similar time series search task among multivariate time series composed of vehicles data such as speed, latitude and longitude is then carried on: the goal is to analyze driving encounters and to find similar driving behaviors in real-world traffic environment. In order to evaluate similarity and find top-*k* similar time series, a top-*k* query algorithm is then introduced: it "compresses" the matrix profile from a vector into a scalar, in order to be able to evaluate whole time series similarity with a novel distance metric that measures similarity between two unequal-length multivariate time series. The framework, that also presents a classification and clustering step, can be generalized to consider interactive behaviors such as vehicle-pedestrian or vehicle-cyclist encounters.

In [47], Silva and Batista explore the topic of Dynamic Time Warping applied to the Matrix Profile technique, also presented in [45]. In this paper, the authors introduce the a new distance for time series comparison, called the Prefix and Suffix invariant DTW (ψ-DTW distance). Their reasoning behind this novelty is that the original MP algorithm uses the Euclidean distance, which is not well-suited for a variety of tasks where warping (which is a small distortion of the time axis that usually occurs in domains such as motion tracking, when studying subjects with different paces, or music, where tempo differences between tracks may happen) is important to consider due to the presence of nonlinear time accelerations (the authors prove, with various examples, that the ED is not able to identify the most significant motifs/discords in application domains where warping is usually required); furthermore, since working with streaming data should be fundamental for the previously listed application fields, the data may not be perfectly pre-segmented in subsequences: when using standard sliding windows techniques, spurious endpoints may occur, with negative effects on the quality of the motifs/discords obtained. The ψ-DTW distance allows matching subsequences using DTW ignoring up to r (a user-defined parameter) endpoints, so that subsequences whose length differ up to $2r$ observations can be compared. The motifs/discords derived from this technique are referred to as Elastic Motifs (ELMO)/Discords (ELD) and a new ψ-DTW MP is constructed (Elastic Matrix Profile, EMP). The authors suggest that "classic" DTW is not perfectly suitable for the task of motif/discord discovery in the domains that require warping, since DTW may not be able to correctly classify subsequences as motifs, considering them dissimilar because their endpoints have an enormous influence on the evaluation of the distance. Various case studies are presented, where ED-based motifs and discords are compared to ELMO/ELD and the preference for the latter is justified for the specific applications.

In 2018, Mirmomeni et al. [48] introduced a novelty among the MP methods, that allows for the discovery of Consecutive Repetitive Patterns (CRPs), which are particularly relevant in the domain of human activity tracking with the help of wearable sensors.

While traditional MP is able to find motifs in a dataset, no information regarding the temporal closeness (locality) of the repeats is preserved. The authors also suggest that the subsequence length, the input parameter of traditional MP, is a "weakness" of MP because of its high sensitivity to it (for example, choosing a subsequence length that is too large reduces the chance of finding similar subsequences, while choosing a value that is too small may result in most of the subsequences being assessed as similar with each other) and their proposed algorithm for finding CRPs has zero parameters to be set. A Distance Index (DI) is therefore introduced, defined as "a vector that at each point stores the distance between the index of any subsequence of length *m* of a Time Series to the index of its Nearest Neighbor". The "most repeat-sensitive DI" is also defined as "a DI that aligns to the period of the repeating pattern and stays flat for the duration of the repeat", in order to find a value of *m* that allows to find a DI that is representative of the area of CRPs. The authors prove via theorems that to achieve the most-repeat sensitive DI, "the input parameter *m* has to be set to the length of the shortest subsequence that is not repeating within the signal of repeat"; they also claim that their algorithm is able to automatically find a value of *m* (they show how automatically determine the best *m*) that performs better than "manually" selecting *m* - using domain knowledge - by 15% for a specific case study on a physiotherapy dataset; they also prove that their method is able to find the regions, in the same dataset, that present CRPs in an automated way and without a priori knowledge about them.

The work from Liu et al. [49] presents a framework "that integrates an image preprocessing technology for anomaly detection with supervised deep learning for chest CT imaging-based COVID-19 diagnosis". While the whole framework will not be described in detail here, the authors propose a novelty to the traditional MP methods, that is the MP calculated at a two-dimensional level, in order to be able to treat a group of points (pixels) in the same way as a time series. By doing so, they extend the concept of Matrix Profile to high dimensional data (in this case, two-dimensional data), which is an avenue that, at the time of the publishing of this work, had not yet been explored (the most similar concept is that of Contextual Matrix Profile [9]). The authors suggest that the alternative method to evaluate the same 2-dimensional data is to flatten a matrix of values into a vector and then treat it as a time series. The 2-dimensional MP is derived by defining segments of the "main" matrix of points, each one with a constant width *w* and height *h*, by sliding a *w x h* window. Those segments are then aggregated to form a "sparse segment set" *S* and the "sparse 2-dimensional MP" (2DM) is defined as the matrix of Euclidean Distances, that has the same size as *S*, between each segment in *S* and its Nearest Neighbor in *S*. To calculate the 2DM, the classic Euclidean Distance between one element with every other in *S* is applied. The minimum value among these distances is finally stored in the 2DM in the same position as the element in *S*. The rest of the reasoning is the same as the standard MP (larger NN distances indicate more probability of anomalies).

In [50], the authors do not introduce significant modifications to the MP methods or propose solutions to a specific problem intrinsic to the traditional Matrix Profile technique; however, the framework presented is interesting in its entirety. The work focuses on the study of correlation of product sales in order to extract temporary rules,

after discovering multivariate motifs, that can assist business managers in their work. The reasoning behind this work is that certain products are often purchased together due to a specific time of the year or event (for example, drinks and food that are more typical of summer season). In market basket analysis, it is therefore important to know which products' sales are correlated, in order to promote bundle sales. It is also useful to know when it is suitable to recommend certain products to the customers and for how long and this can be achieved by identifying multivariate motifs and evaluating their length. The approach is the following: first, the similarity between a series of product sales is studied; the relationships between products are then used to construct a similarity network of product sales; thanks to this network, different groups of products can be identified and the products sales time series in the different groups are treated as a multivariate time series. If the multi-motifs that can be found in these multivariate time series are repeating, then the temporary relationships that they represent are repeatedly occurring. In the last stage, Temporary Rules (TR) (association rules that take into account the aspect of temporality and quantity rather than probability) are generated from a multi-motifs set. One of the weaknesses of Temporary Rules that the authors report, however, is the lack of a well-defined numerical standard, which results in a difficulty in generalizing the process to other domains.

2.3.3. Matrix Profile applications in the Energy and Buildings sector

Currently, the literature regarding the applications of the Matrix Profile technique in the Energy and Buildings research field is very sparse. To the best of my knowledge, there are only three papers to this day that discuss about the potential uses of the Matrix Profile to analyze buildings' energy profiles in order to discover anomalies and other useful information.

In [8], Nichiforov et al. introduce the Matrix Profile as a powerful tool for the study of large buildings' energy consumption profiles in order to:

- Build a dataset of anomaly patterns, looking for the top discords (when do they happen? Is it possible to explain their presence in relation to well-known periods such as holidays, and so on?) in each building's energy time series;
- Create a supervised learning classification model that learns these anomaly patterns and is able to assign newly observed data to the correct type of building (each building is associated with a different dominant usage pattern).

The approach described above was applied to a reference building energy dataset, containing the energy consumption for 507 university buildings from Europe and USA for a 1-year period. The sampling rate for each dataset is 1 hour, for a total of 8760 data points. The dominant energy usage patterns were divided into 4 categories: classrooms, offices, laboratories and dormitory rooms, for a subset of 422 buildings. The MP was applied on this subset of datasets, grouped by each type of dominant usage pattern. The results shown are promising: the researchers were able to explain the various observations, such as MP values distributions and top-3 discords locations, with known events and occupation trends, and the classification model based on the MP discords'

features was able to achieve relatively high accuracy despite a significant decrease in training time with respect to the case of utilization of the whole year energy measurements in the modelling stage. The authors claim that "the approach can prove useful for exploiting complementary energy consumption patterns in a decentralized control structure towards grid balancing and economic operation".

The same authors published another paper with a similar topic [7]; this work mainly aims at proving that the Matrix Profile can be used for domain-specific information extraction from buildings' energy consumption time series. Once again, the discords from the energy profiles of a large academic buildings dataset, which comprises various dominant usages such as classrooms, laboratories and offices, are studied in relation to the time when they happen in order, for example, to try to infer something about why a discord happens at a certain time. Distributions of MP values are also examined and questions such as "How are the MP values of a certain kind of buildings distributed?", "Are the average MP values lower on a weekend or on a weekday?", " Is there a day of the week that shows interesting distributions?" are taken into consideration. The Manhattan distance, which is calculated as the sum of the absolute values of the differences of the various dimensions between two points, is also introduced to evaluate the differences in results when compared to the traditional Euclidean distance. It is demonstrated that the MP calculated with the Manhattan distance is noisier, while showing an overall trend that is similar to that of the standard MP. The authors suggest that smoothing out this noisier MP would result in a very similar distance metric profile with less computational time required. The last point the authors make is that the MP could be used for model-free load forecasting: they applied this technique to a specific building and concluded that the prediction performance is not as high as other methods commonly used for this purpose, but the lower performance is compensated by faster model selection and training, which is well-suited for real-time local control.

The last paper related to the topic of the Matrix Profile in the Energy and Buildings sector is [6]: this work introduces a method for Automated Load profile Discord Identification (ALDI) and shows its application to a large building portfolio (over 100 buildings). In this kind of framework, the Matrix Profile is used mainly in the first step, in order to obtain daily MP values for each building in the portfolio. The MP values are then grouped by typical day types and a statistical evaluation is performed to compare how individual days' MP distributions are similar (or dissimilar) against the typical days' MP distributions. Finally, the days marked as anomalous are analyzed one by one in order to gain insight about why a certain day is classified as a discord after the statistical test. One of the most interesting aspects about this work, in my opinion, is how the first simple and "domain-agnostic" Matrix Profile-based step is then followed by a "domain-expert heavy" series of analyses, after intermediate statistical tests. This approach seems promising, but also not so simple to implement in a generalized way (the authors also remark that the choice of the *p-value* for statistical tests is not trivial). Taking into consideration a building portfolio and no longer a single building also means that the discords need to be identified in a different way from the conventional ones seen in the works cited above, thus requiring analysis on the distribution of MP values in order to complete this task. The authors suggest that their work could prove particularly useful

for portfolio managers in order to evaluate multiple buildings - belonging to the same geographical region and connected with the same electrical grid and metering facility and thus having similar discord load shape patterns - in terms of discord days.

3. Methods

This Chapter covers the main relevant aspects of the data analytics techniques employed in this work. While the topic of data analysis methods is broad and there is a growing interest around it in the research community, as already mentioned in the previous chapters, the focus of this work is mostly based on the Matrix Profile technique for time series analysis and one of the methods derived from it, the Contextual Matrix Profile.
In **3.1.**, the main definitions for fully understanding what a Matrix Profile is and its desirable properties will be mentioned, as well as the most evident critical aspects that are intrinsic to it and how they have been addressed so far in the existing literature works.
In **3.2.**, the Contextual Matrix Profile is presented, together with the possible motivations for its adoption instead of the classic Matrix Profile in certain case studies and in this work.
In **3.3.**, the techniques for knowledge discovery that were exploited in sections **4.3.** and **4.4.** are briefly discussed.
Finally, **3.4.** presents the methods for anomaly detection employed in section **4.5.**.

3.1. The Matrix Profile

The Matrix Profile is a technique introduced by Yeh et al. [5] in 2016 that deals with the challenge of series analysis; that is, the analysis of ordered collections of data points. These data points can belong to various "fields" such as speech, shapes, handwriting, music [51] and so on, as long as the concept of "series" can be applied; the most common area of interest, which is also the area of the research efforts in this work, however, is time series analysis. As the name suggests, time series analysis deals with the study of points ordered in time; in the energy and buildings' research field, such data points belong to power measurements, temperature measurements and other physical quantities that are commonly recorded by monitoring devices. The main idea behind the development of the MP technique was the need for improvement in the task of "time series all-pairs-similarity-search" (shortened as "TSAPSS" and also known as "similarity join") for time series, especially in terms of time needed for computation: the longer a time series is, the longer the TSAPSS process takes to complete and the way the time scales with size mainly depends on the algorithm used. The TSAPSS problem can be summarized as: "*Given a collection of data objects, retrieve the nearest neighbor for each object.*"[5]. In order to properly understand the logic behind the MP technique, it is necessary to briefly introduce a handful of definitions and notations that are directly taken from the first "MP-related" paper ever published [5].

Definition 1: A time series T is a sequence of real-valued numbers t_i: $T = t_1, t_2, \ldots, t_n$ where n is the length of T.

In the task of TSAPSS, the focus is not on the time series as a whole; instead, the interest is in studying fragments of the time series that are called "subsequences".

Definition 2: A *subsequence* $T_{i,m}$ of a T is a continuous subset of the values from T of length m starting from position i. $T_{i,m} = t_i, t_{i+1},\ldots, t_{i+m-1}$, where $1 \le i \le n-m+1$.

It is possible to consider one subsequence at a time and evaluate its similarity, in terms of "distance", to all the other subsequences; the structure that stores this information is called "distance profile".

Definition 3: A *distance profile* D is a vector of the Euclidean distances between a given query and each subsequence in an all-subsequences set (see **Definition 5**).

In the original Matrix Profile definition, the distances between the subsequences are evaluated using the z-normalized Euclidean distance, defined in the following way [23]:

Definition 4: Let μ_i and μ_j be the mean of the values in two subsequences $T_{i,m}$ and $T_{j,m}$ respectively. Also, let σ_i and σ_j be the standard deviation of the values in $T_{i,m}$ and $T_{j,m}$ respectively.
Then, the z-normalized Euclidean distance between $T_{i,m}$ and $T_{j,m}$ is defined as:

$$DZ_{i,j} = \sqrt{\sum_{l=0}^{m-1} \left(\frac{t_{i+l} - \mu_i}{\sigma_i} - \frac{t_{j+l} - \mu_j}{\sigma_j} \right)^2}$$

If the query and the all-subsequences set belong to the same time series, the distance profile is equal to zero at the location of the query, and close to zero in its neighborhood. Such matches are defined as "trivial matches": they are not taken into consideration during the computation phase by ignoring an exclusion zone, commonly set as a m-width window ($m/2$ before the location of the query and $m/2$ after).

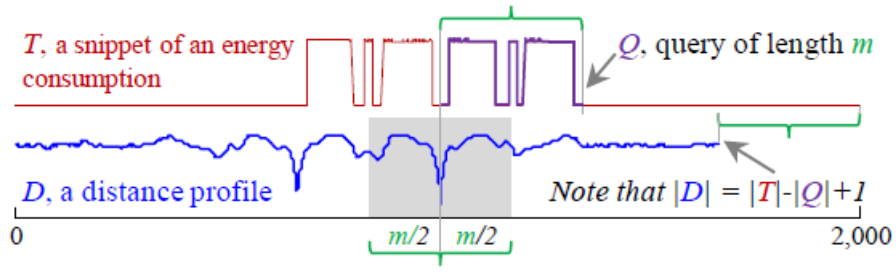

Figure 1 - A subsequence Q extracted from a time series T is used as a query to every subsequence in T. The vector of all distances is a distance profile (source: C.-C. Michael Yeh et al., "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets.")

Definition 5: An *all-subsequences set* $\boldsymbol{A}$ of a time series T is an ordered set of all possible subsequences of T obtained by sliding a window of length m across T: $\boldsymbol{A} = \{T_{1,m}, T_{2,m},\ldots, T_{n-m+1,m}\}$, where m is a user-defined subsequence length. We use $\boldsymbol{A}[i]$ to denote $T_{i,m}$.

When dealing with time series analysis, the concept of similarity between subsequences is recurring. Specifically, researchers are often interested, given a specific subsequence,

in finding the most similar subsequence in the all-subsequences set, also known as "nearest neighbor" of the given subsequence.

Definition 6: Given a subsequence $T_{i,l}$, we say that its mth best match, or Nearest Neighbor (mth NN), is $T_{j,l}$, if $T_{j,l}$ has the mth shortest distance to $T_{i,l}$, among all the subsequences of length l in T, excluding trivial matches. [34]

Definition 7: given two all-subsequences sets $\boldsymbol{A}$ and $\boldsymbol{B}$ and two subsequences $\boldsymbol{A}[i]$ and $\boldsymbol{B}[j]$, a *1NN-join function* $\theta_{1nn}(\boldsymbol{A}[i], \boldsymbol{B}[j])$ is a Boolean function which returns "true" only if $\boldsymbol{B}[j]$ is the nearest neighbor of $\boldsymbol{A}[i]$ in the set $\boldsymbol{B}$.

Definition 8: given all-subsequences sets $\boldsymbol{A}$ and $\boldsymbol{B}$, a *similarity join set* $\boldsymbol{J_{AB}}$ of $\boldsymbol{A}$ and $\boldsymbol{B}$ is a set containing pairs of each subsequence in $\boldsymbol{A}$ with its nearest neighbor in $\boldsymbol{B}$: $\boldsymbol{J_{AB}}=\{\langle \boldsymbol{A}[i], \boldsymbol{B}[j] \rangle \mid \theta_{1nn}(\boldsymbol{A}[i], \boldsymbol{B}[j])\}$. We denote this formally as $\boldsymbol{J_{AB}} = \boldsymbol{A} \bowtie_{\theta 1nn} \boldsymbol{B}$.

The definition of a Matrix Profile can finally be introduced:, the MP is a "meta time series" (a time series deriving from the original time series T) of length n-m+1.

Definition 9: A *matrix profile* (or just *profile*) P_{AB} is a vector of the Euclidean distances between each pair in $\boldsymbol{J_{AB}}$.

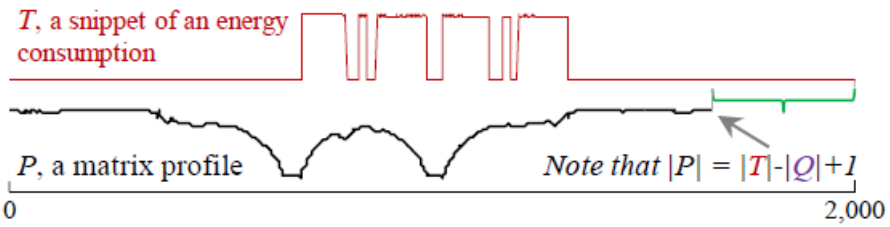

Figure 2 - A time series T and its self-join MP (source: C.-C. Michael Yeh et al., "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets.")

If a single time series is considered and it is needed to compute the Matrix Profile for that series, it is necessary to consider the "self-similarity join set".

Definition 10: A *self-similarity join set* $\boldsymbol{J_{AA}}$ is a result of similarity join of the set $\boldsymbol{A}$ with itself**.** We denote this formally as $\boldsymbol{J_{AA}} = \boldsymbol{A} \bowtie_{\theta 1nn} \boldsymbol{A}$. We denote the corresponding matrix profile or *self-similarity join profile* as P_{AA}.

The MP alone is not able to tell the user where the nearest neighbor of a subsequence is located, since it only stores information about distances. In order to know that, another meta time series called "matrix profile index" has to be introduced.

Definition 11: A *matrix profile index* I_{AB} of a similarity join set $\boldsymbol{J_{AB}}$ is a vector of integers where $I_{AB}[i] = j$ if $\{\boldsymbol{A}[i], \boldsymbol{B}[j]\} \in \boldsymbol{J_{AB}}$.

It is also important to know that the similarity join set, the Matrix Profile and the MP index are not symmetric. Therefore, $\boldsymbol{J_{AB} \neq J_{BA}}$, $P_{AB} \neq P_{BA}$, and $I_{AB} \neq I_{BA}$.

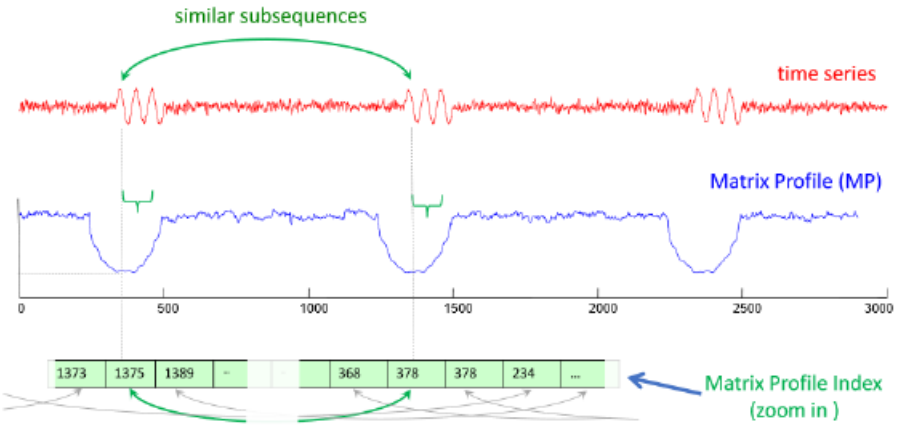

Figure 3 - A time series, its self-join MP and its MP index (source: C.-C. Michael Yeh et al., "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets.")

The last definitions that are needed are those of "time series motif" and "time series discord".
Time series motifs are subsequences that present a high degree of similarity one another.

Definition 12: $T_{a,l}$ and $T_{b,l}$ is a motif pair iff $dist(T_{a,l}, T_{b,l}) \leq dist(T_{i,l}, T_{j,l})$ $\forall i, j \in [1, 2, \ldots, n - l + 1]$, where $a \neq b$ and $i \neq j$, and *dist* is a function that computes the z-normalized Euclidean distance between the input subsequences. [34]

In contrast, time series discords are subsequences that are maximally dissimilar to their nearest neighbor.

Definition 13: A subsequence $T_{i,l}$ is a *Top-k* mth-discord if it has the kth largest distance to its mth NN, among all subsequences of length l of T. [52]

3.1.1. The desirable properties of the Matrix Profile

In the paper that introduces the concept of Matrix Profile and the first algorithms for its computation [5], the authors claim that not only this new technique is significantly faster than comparable rival methods (however, due to the novelty of the concept and to the unique features the MP presents, they also state that it was hard for them to find good baselines to enable comparison) but it also presents many desirable properties; the most important are listed below:

- It is *exact*, therefore the risk of false positives or false dismissals is completely avoided;

- It is *parameter-free*: the only parameter that has to be specified is the subsequence length, whose choice can sometimes be not trivial, depending on the expert knowledge of the domain; however, rival methods for all-pairs-similarity-search typically require more in-depth tuning of various parameters;
- It is *space efficient*: the space complexity is linearly dependent from the series length ($O(n)$), with a small constant factor;
- The results can be computed in an *anytime* way, in order to allow ultra-fast approximate solutions and real-time operations on data;
- The similarity join is *incrementally maintainable*, which means that it is possible to deal with streaming data without any kind of issue related to speed of data acquisition versus data computation;
- The method provides *full joins*, while rival TSAPSS methods often are subject to a "similarity threshold" that needs to be selected and provided beforehand;
- The *time* needed for MP computation can be *known in advance* given only the length of the time series;
- The time and space complexity *do not depend on the dimensionality*, which means that the subsequence length does not influence the performance of the MP computation;
- It is *parallelizable* (which means it is able to perform various computations at the same time) and it can leverage hardware and take full advantage of the power of multicore CPUs, GPUs, distributed systems and so on.

3.1.2. The issues of the traditional Matrix Profile technique

This section focuses on two main issues that the traditional Matrix Profile method presents and that emerged as critical aspects during the work for the case study presented in Chapter 5: the "twin freak" problem and the "z-normalization" problem.
The first issue is related to the classic definition of discord as "the subsequence that has the maximum distance from its nearest neighbor"; in real-life case studies, anomalies may happen more than once and show similar behavior, which would make a previously "isolated" point (the first occurrence of the anomaly) have a nearest neighbor with short distance between the two points. To solve this kind of issue, various paths can be taken, with the most common one being the transition from the previously cited discord definition to a more general one: the discord becomes "the subsequence that has the maximum distance from its k^{th} nearest neighbor"[53], where k is defined by the data analyst and is generally not too large (e.g. 3-5). The twin freak problem is a recurring issue in the domain of time series data analysis using the Matrix Profile, and the works of D. Duque Anton et al. [54], Dinal Herath et al. [55] and Zhang et al. [56] propose different approaches to tackle it.
In [54], which belongs to a series of papers published by the same authors on a similar topic (analysis of attacks in Industrial Process Data, considered as a framework where the Matrix Profile is part of the process), an extension to the traditional Matrix Profile technique is implemented. The authors suggest that attacks that occur multiple times

and have the same characteristics each time are not detected as attacks after the first recognition, since their behavior becomes "normal". In order to take care of this issue, they propose a solution that relies on counting the number of instances of a motif; a threshold value is set to compare the motif analyzed at each time to all other motifs and all the motifs whose distance is smaller than the threshold value are added to a list. Doing so allows to count the number of similar motifs. Therefore, the focus is more on the "number of occurrences" criteria rather than on the "minimum distance" criteria. The authors claim this kind of workaround of the twin freak problem allows them to successfully identify the periods where an attack actually took place, since in those periods the number of similar motifs was particularly low, indicating a rare behavior despite a low minimum distance.

The work by Dinal Herath et al. [55] introduces a framework, called RAMP (Real-Time Aggregated Matrix Profile), that aims at detecting anomalies in scientific workflow systems, in order to stop unwanted behaviors at an early stage and before they can possibly influence scientific discovery results. Examples of these misbehaviors may be the result of external attacks such as Denial Of Service (DOS) attacks. Without going into too much detail, such framework comprises various modules including "Anomaly Detection", that builds upon the Matrix Profile technique. The interesting modifications to the standard MP methods that are made are:

- Limiting the number of subsequences compared, in order to avoid false negatives when in presence of a repeated anomaly instance. RAMP introduces a semi-supervised model to apply MP and, in order to perform the limitation on the number of subsequences, the time series are considered only for the first $M-m+1$ subsequences, where m is the subsequence length and M is a user-set parameter whose choice is not trivial;
- Computing relative distances between subsequences instead of absolute Euclidean Distances. The authors suggest that the purpose of this modification is "to overcome the inherent bias of Euclidean Distance towards numerically larger data points".

In [56], the authors present a new primitive for time series data mining, called Localized Matrix Profile (LMP); the LMP is a tool that is well-suited for applications where the data vary statistically with time: throughout the paper, only transient systems (physical devices such as electrical motors) are considered, where multiple runs of the same "process" are evaluated, each one typically producing a Multivariate Time Series (MTS). Each time series in a MTS, representing a specific variable, comes from a sensor. A set of baseline MTS items is first produced and the LMP compares each new MTS item to the baseline set, that is composed of L items. Since different variables (sensors) may contribute in a different manner in fault detection and classification, they also introduce a vector that assigns specific "importance weights" to each sensor. In short, the LMP only compares distances of subsequences that belong to MTS items (the comparison is a join between a new item and the baseline set) with the same starting time t (the same time instance) in order to take account of the time-varying nature of the system that may otherwise lead to false positives/negatives such as a "twin freak" occurrence, while the

traditional MP computes Nearest Neighbor searching over all available subsequences involved in the MTS, no matter where they start.
The second problem, intrinsic to the traditional Matrix Profile technique, is that of z-normalization. As introduced in 3.1., the classic distance measure used to determine similarity in MP techniques is z-normalized Euclidean Distance. While the possibility of adopting other distance measures, such as Dynamic Time Warping, has already been mentioned in previous sections in this work, the z-normalization has not yet been topic of discussion.
The work of De Paepe et al. [57] mainly focuses on this aspect and goes into detail on why the z-normalization should not be blindly adopted for any application. The authors claim that the reasons behind the use of z-normalization in the original Matrix Profile algorithm are two: the first one is that the MASS algorithm, on top of which the STAMP algorithm is built, inherently calculates distances that are z-normalized; the second one is that, by applying z-normalization, the algorithm focuses on "shape-based similarity" instead of on "magnitude-based" similarity: in many domains, this is a desirable property, since it allows comparison in data with wandering baselines (Figure 4) or where recurring patterns are present but with different amplitudes.

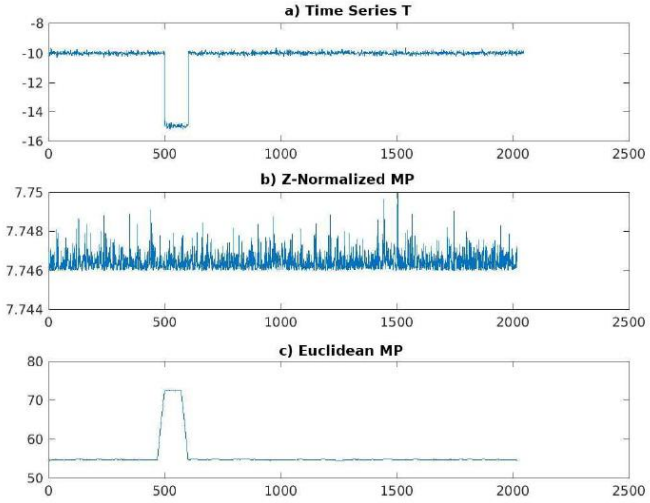

Figure 4 - A time series T (a) and its self-join z-normalized MP (b) and non z-normalized MP (c) (source: R. Akbarinia and B. Cloez, "Efficient Matrix Profile Computation Using Different Distance Functions," Jan. 2019, [Online]. Available: http://arxiv.org/abs/1901.05708)

However, as the authors suggest, this normalization has a significant downside: when flat sequences are considered, fluctuations (noise, for example) are greatly enhanced, which results in spikes in the Matrix Profile values.

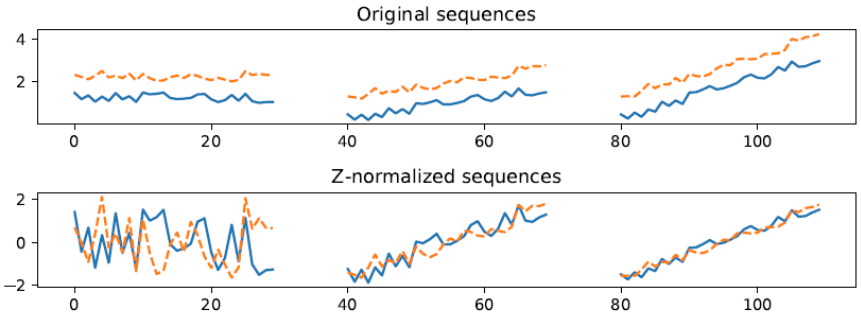

Figure 5 - Effect of z-normalization in different kinds of sequences (source: D. de Paepe, "Implications of Z-Normalization in the Matrix Profile." [Online]. Available: http://idlab.ugent.be)

Furthermore, not all the research domains benefit of the supposedly positive properties of z-normalization: for example, in the energy domain, patterns that are similar in shape but with different magnitudes are often present (for example, a building load pattern that occurs on a weekday versus one that occurs on a holiday: they may have similar shapes, but the magnitudes are not comparable) and it is often desirable to be able to distinguish between such patterns.

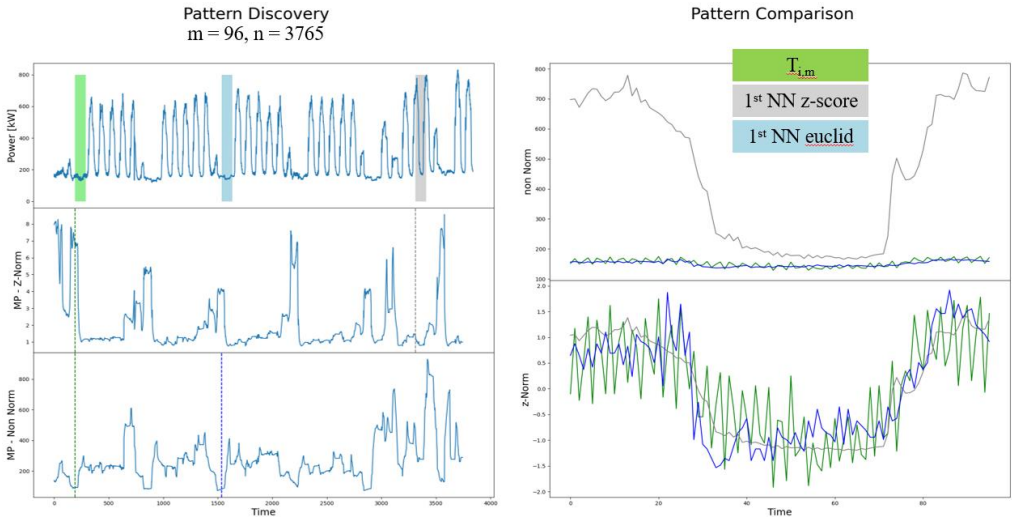

Figure 6 - An example of how z-normalization can negatively affect similarity search on power demand time series (source: BAEDA Lab, website: http://www.baeda.polito.it/)

The authors suggest a way to overcome the above mentioned issue that is based on a noise elimination technique, in order to achieve the goal of similarity between flat subsequences, no matter the presence of fluctuations in the data. This technique involves the introduction of the standard deviation of the noise: after calculating the squared distance between two subsequences using known algorithms, the squared estimate of the noise influence is then subtracted to obtain a "corrected distance".

The overall conclusion that could be obtained by taking the above mentioned issues into account is that the traditional Matrix Profile is a technique that is simple and effective especially in domains where the data in a time series can all be considered "at once" and there is no need to separate certain periods of the time series, based on an a priori domain

knowledge, during the analysis. The following section describes a technique that allows for an improvement with respect to the previously mentioned critical aspects, by focusing on patterns that differ the most from a group of similar observations rather than looking for the most unique occurrences in the whole time series.

3.2. The Contextual Matrix Profile

The 2020 paper by De Paepe et al. [9] introduces a variation of the original Matrix Profile technique, called the Contextual Matrix Profile (CMP).
The CMP can be considered as a 2D version of the Matrix Profile, that takes into account multiple matches across window regions of the time series, while the MP considers one match for each window. The authors suggest that besides enhanced data visualization, the CMP can also be used for detecting anomalies that do not correspond to the traditional definition of discord. The CMP is built on top of the same fundamental concept of the MP, that is the Distance Matrix (DM) containing the distances of all subsequences from one input time series to all subsequences from another time series; the MP is defined as column-wise minimum over the full Distance Matrix, while the CMP is defined as the minimum value across rectangular regions of the distance matrix, as shown in Figure 7. The rectangles, whose configuration is up to the user, may cover the entire Distance Matrix. Figure 8 represents examples of definitions of regions: 3 horizontal (A, B, C) and 5 vertical (1 - 5) ranges are considered and each pair of ranges from both axes results in one region of interest in the DM. The minimum value of the region is then calculated and stored in the CMP. Also, the CMP-consumer can be configured in a way that it calculates the Matrix Profile; by doing this, the CMP can be seen as a generalized version of the MP.

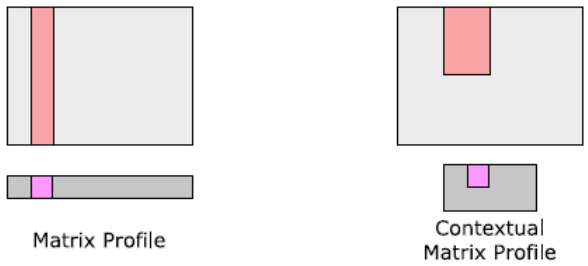

Figure 7 – Differences between how MP and CMP are created. The light grey area represents the DM. (source: D. de Paepe et al., "A generalized matrix profile framework with support for contextual series analysis," Engineering Applications of Artificial Intelligence, vol. 90, Apr. 2020, doi: 10.1016/j.engappai.2020.103487.)

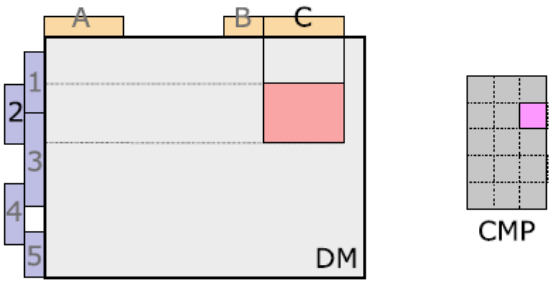

Figure 8 - Example of region definitions in the DM (source: D. de Paepe et al., "A generalized matrix profile framework with support for contextual series analysis," Engineering Applications of Artificial Intelligence, vol. 90, Apr. 2020, doi: 10.1016/j.engappai.2020.103487.)

The name "Contextual Matrix Profile" derives from the term "context", which indicates the time period - whose choice is up to the user - where each subsequence can start: this results in the possibility of comparing subsequences that are shifted in time one with respect to the other, as can be seen in the "New York taxi" example below. This aspect, coupled with the fact that the CMP can be effortlessly applied to user-defined subsets of a dataset, introduces a significant degree of expert knowledge in the process, since the choice of both the sub-groups (when needed) and especially the contexts is not trivial and different settings of these two "variables" can produce results with a quality that varies based on the user's expertise in the domain of application.

The authors suggest that the main use cases for CMP are data visualization and anomaly detection. For data visualization, the CMP can be used to gain insight about the dataset that is considered, and can be used to find patterns and deviations from them that might highlight the need for further inspection. The authors also claim that the main difference between CMP and MP in the data visualization task is that the MP is unable to provide information about the periodic nature of the data, since subsequences are compared against all others rather than in "groups" like in the CMP.

As an example of this use for data visualization, a case study on a dataset of New York Taxi passengers numbers is presented: the CMP, represented in Figure 9, that results from considering a window length of 22 hours (the remaining 2 hours of each day represent the context, which goes from 00:00 to 02:00) shows a pattern of small squares and suggests that the most common trend is 5 days with a similar behavior, followed by 2 days with different behavior (each point represents the distance between a day and another, with lower distances meaning the match between the two days is more accurate); this kind of periodic pattern represents the weekdays/weekends cycle and such information cannot be found when visualizing the MP, that only shows peaks corresponding to some holidays or other events (as represented in Figure 10).

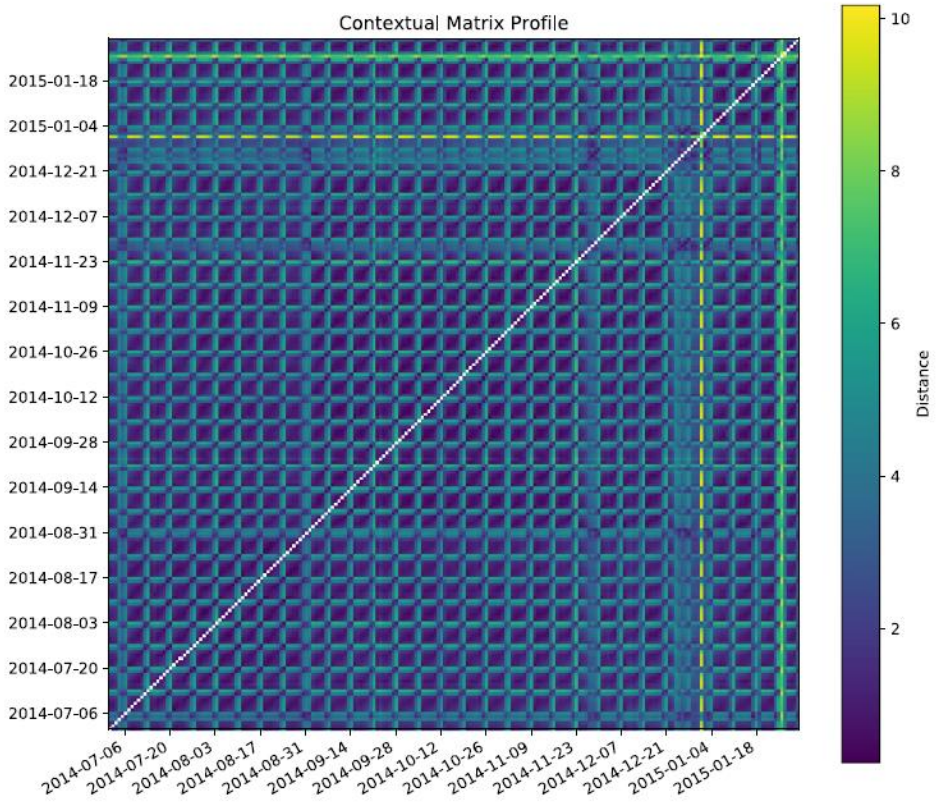

Figure 9 - The CMP for the New York Taxi dataset (source: D. de Paepe et al., "A generalized matrix profile framework with support for contextual series analysis," Engineering Applications of Artificial Intelligence, vol. 90, Apr. 2020, doi: 10.1016/j.engappai.2020.103487.)

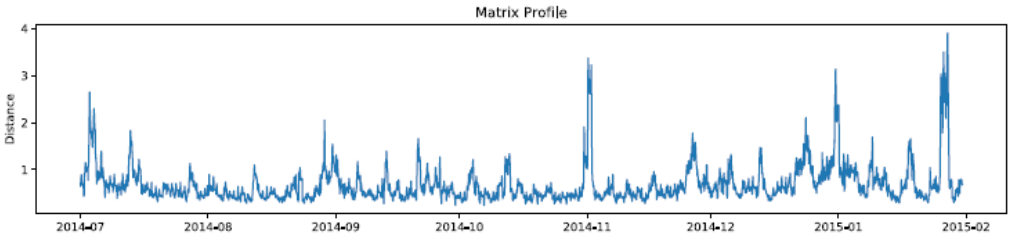

Figure 10 - The Matrix Profile for the New York Taxi dataset (source: D. de Paepe et al., "A generalized matrix profile framework with support for contextual series analysis," Engineering Applications of Artificial Intelligence, vol. 90, Apr. 2020, doi: 10.1016/j.engappai.2020.103487.)

For anomaly detection, the authors examined the same dataset and applied a custom technique in order to obtain single days' anomaly scores based on the values of the CMP at each point, distinguishing weekdays from Saturdays and Sundays. Applying the elbow method, they found a threshold to obtain the number of anomalous days in the dataset. After applying a similar reasoning to the MP (where the discords are considered anomalies), they found that the two methods returned different days as anomalous and the CMP returns anomalous days that are noticeably different from most of the reference days (Figure 11), while the MP returns various days where the "anomaly" is due to a spike or a tail with unique shape or a bump (Figure 12).

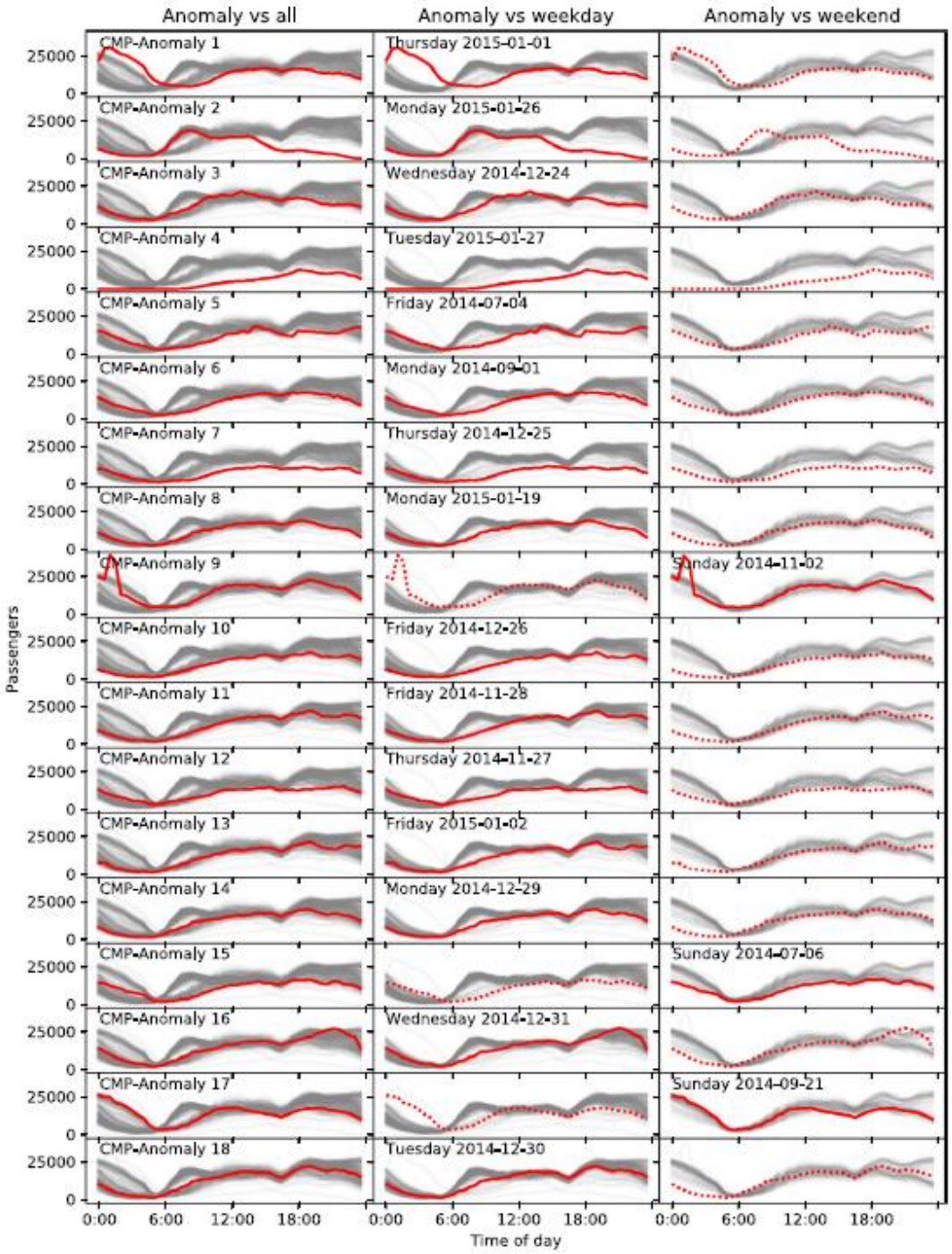

Figure 11 - The anomalous days found in the New York Taxi dataset using the CMP (source: D. de Paepe et al., "A generalized matrix profile framework with support for contextual series analysis," Engineering Applications of Artificial Intelligence, vol. 90, Apr. 2020, doi: 10.1016/j.engappai.2020.103487.)

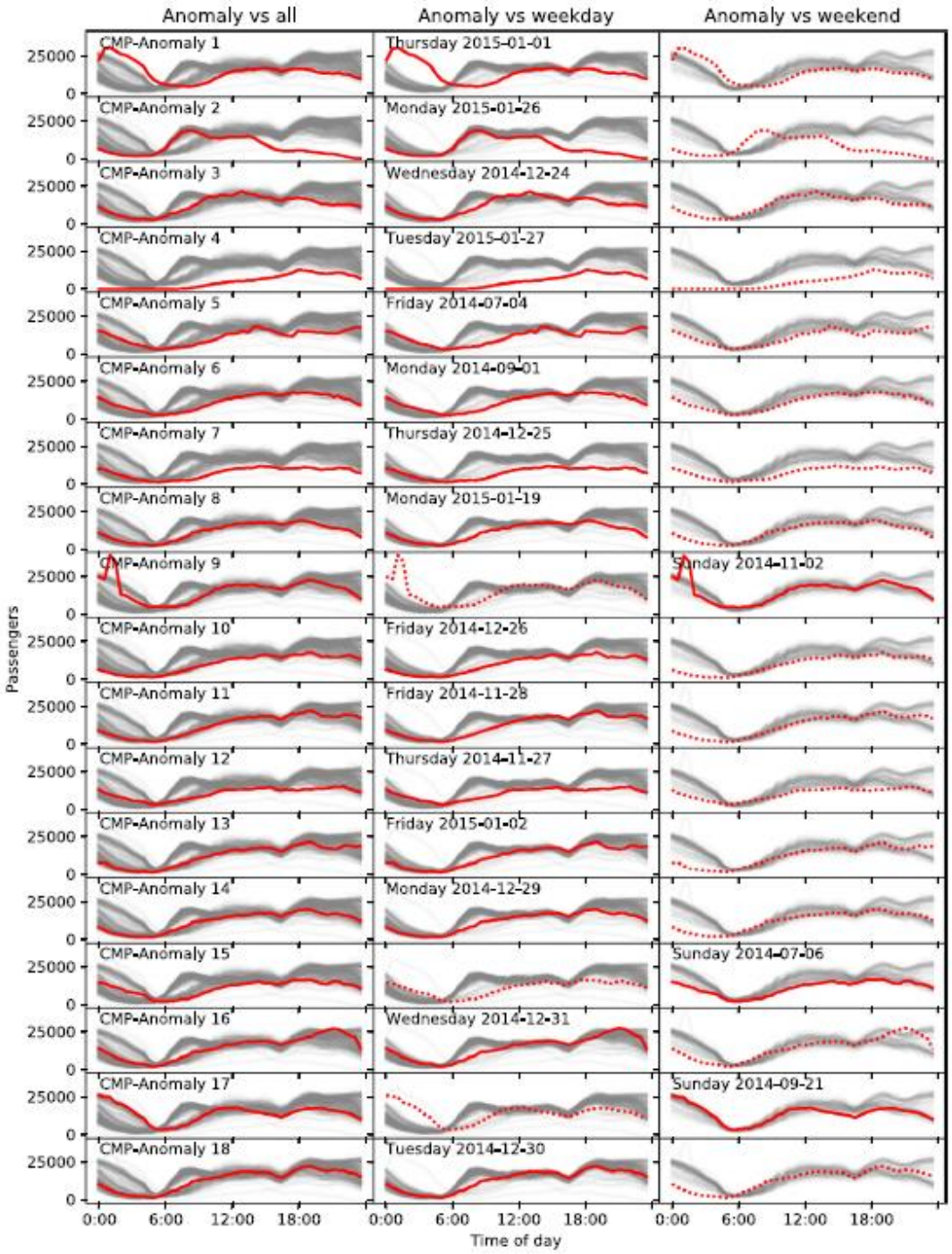

Figure 12 - The anomalous subsequences found in the New York Taxi dataset using the traditional MP (source: D. de Paepe et al., "A generalized matrix profile framework with support for contextual series analysis," Engineering Applications of Artificial Intelligence, vol. 90, Apr. 2020, doi: 10.1016/j.engappai.2020.103487.)

The authors also provide their opinion on the CMP versus the MP:
"The question arises: which of these techniques is best suited for anomaly detection? While we suspect most users will find the results of the CMP to be more insightful for this specific dataset, the general answer remains ''it depends''. Fundamentally, both techniques are searching for different things. While the Matrix Profile is looking for the most unique patterns (discords) in the series, the CMP based anomaly detection is looking for patterns that differ most from a group of reference contexts. Both approaches will have applications depending on the type of anomalies the user is interested in….
…The CMP has one other major advantage over a basic distance matrix, it allows for a (time) shift when comparing sequences, allowing us to recognize similar behavioral patterns despite them not being aligned in time. This flexibility comes at the cost of the user having to define the contexts, often having to rely on expert knowledge of the underlying process". [9]
To conclude, the Contextual Matrix Profile appears to be a suitable choice for the energy and buildings domain, where alternating weekday/weekends patterns are almost always present and therefore the twin freak and z-normalization issues, that are intrinsic to the classic Matrix Profile technique, can become troublesome in the interpretation of the results. By defining contexts and considering only subsets of the original dataset, the CMP allows to tackle the above mentioned issues, introducing expert knowledge in the "pre-processing" step of this technique; while the traditional MP only requires the definition of the subsequence length, the choice of contexts and subsets is not trivial. In the following Chapter, the framework adopted in this work is presented, starting right from this pre-processing phase.

3.3. Techniques for knowledge discovery

In this section, the data mining methods employed in the framework presented in this work are briefly introduced from a theoretical point of view, to help the reader that is unfamiliar with them understand the logic behind their adoption. The term "data mining" is quite broad and generally refers to the discovery of information and patterns from large datasets by means of Artificial Intelligence (AI) techniques, especially machine learning ones.

3.3.1. Classification and Regression Tree

Classification is the task of assigning items to one category among different ones; all the categories need to be defined prior to the classification process. According to [58], the input data for such tasks is a collection of records, each one characterized by a tuple (**x**, *y*): **x** is called the "attribute set" and *y* is known as the "special attribute" and represents the class label/category/target attribute. The term "classification" is often accompanied by the word "regression"; their meaning is somewhat similar (they indicate essentially the same process), although there is a fundamental distinction between the task of

classification and that of regression: classification assigns categorical class labels (e.g. a "name") to the items, while regression is related to continuous class labels (e.g. a power demand value).
Classification and Regression Trees (CARTs) represent the most common machine-learning algorithms to carry out a classification/regression task. They can be used both for descriptive modeling (explain what features characterize the items with a certain label) or for predictive modeling (assign class labels to a collection of unknown records) [58]. CARTs belongs to the broader family of classifiers called "decision trees", whose underlying logic is based on the splitting of items, starting from the collection of all records, into subsets containing more "homogeneous" objects (subsets with less internal "impurity", which is a measure defined by the value of specific expressions, such as "Entropy impurity measure" or "Gini index"; the reader who is interested in a more detailed explanation of how decision trees work is referred to [58]), called "nodes". The lines that connect the nodes are called "edges" or "branches". A decision tree has three types of nodes, as shown in Figure 13:

- the **"root node"**, which contains all the items in the dataset and can only have outgoing edges;
- the **"internal nodes"**, which contain homogeneous subsets and have one incoming edge and two or more outgoing edges;
- the **"leaves"** (or **"terminal nodes"**), which represent the "purest" subsets with respect to the tree settings and have one incoming edge and zero outgoing edges.

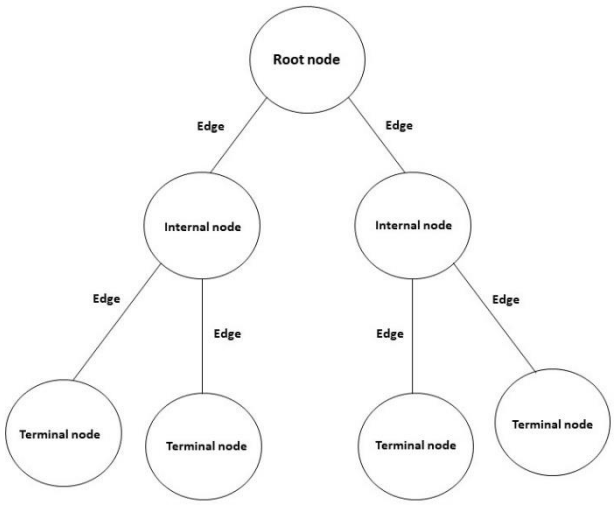

Figure 13 - Example of the structure of a decision tree

In this work, decision tree algorithms have been adopted for the purpose of predictive modeling. This task comprises two sub-phases, called "training" and "testing", which can be found in all "supervised learning techniques" (methods for knowledge discovery where a known output is assigned to unlabeled input data). Model training consists in

pairing each input with the correct output and "passing" this information to the knowledge discovery algorithm in order for it to discover the underlying relations between inputs and outputs; this makes it possible that when new data, containing only inputs, is presented to the model during the testing phase, the algorithm is able to "remember" the discovered patterns and associate an output to each input. The percentage of correct labels (accuracy) is then evaluated by comparing the newly assigned labels to the correct ones, which were not given to the algorithm as an input, and verifying how many of them are matching.

In a decision tree, the items are initially grouped together in the root node and the algorithm iteratively performs splitting on the sub-groups, based on the above mentioned criteria of minimizing impurities in each internal node. A criterion has to be set in order to stop this splitting process, to avoid model overfitting (the model would become very accurate with respect to the training set and the impurity of the terminal nodes would be equal to zero; however, a model trained this way would not be able to perform prediction successfully on any other data that is not the training set): usually, this criterion is based on parameters such as the minimum number of observations in a node for a split to be attempted, or the minimum number of observation in terminal nodes, and so on. Another way to "manipulate" (and therefore stop) the tree growth is by setting a "complexity parameter" value, which represents the minimum benefit – in terms of classification accuracy versus the computational cost – that each split must add to the tree; the greater the value of this parameter is, the more difficult is for the tree to perform a split.

The model testing and performance evaluation can be carried out by means of different techniques, such as the Holdout Method, Random Subsampling, Cross-Validation or Bootstrap [58]. In this work, Cross-Validation has been adopted since the R function "rpart" used to implement the CART defaults to this technique. In particular, a *k*-fold cross validation is performed: the data is segmented into *k* partitions, each one having equal size; during each run, a single partition is used for testing and all the others are used for training. In order to perform testing on each partition once, the run is repeated *k* times and the total error of the model results from the sum of the errors of every single run [58].

3.3.2. Hierarchical clustering

"Clustering" indicates the process of grouping together elements of a dataset that show a degree of similarity with respect to a certain attribute/characteristic/property. This similarity is usually measured by means of mathematical expressions, such as distance functions: clustering aims at minimizing the distance between the elements of a cluster, to create sub-groups that are characterized by high intra-cluster and low inter-cluster similarity.

The reasons behind a clustering operation can be various [59]. The user may want to capture the natural structure of the data in order to better understand the phenomenon that is being studied: this is called "clustering for understanding"; another useful

purpose of clustering is to pre-process the data for further analyses, for example to speed up the subsequent processes by re-organizing the dataset in subsets that are "easier to examine" for the algorithms to be applied: this is called "clustering for utility".
Clustering algorithms belong to the macro-category of unsupervised learning techniques (methods for knowledge discovery where the output is unknown and the input data is not labeled: the aim is to discover patterns and relations between the items in the dataset) and can be divided in two main groups on the basis of the logic behind the clustering process:

- **"Partitional clustering"** divides the dataset into non-overlapping subsets, so that every item falls into exactly one subset;
- **"Hierarchical clustering"** still performs the division into non-overlapping sub-groups; however, in this case, a subset can have further subdivisions (subclusters), resulting in a tree-like structure.

It is also worth mentioning that a third group of clustering techniques, containing the **"density-based"** ones, could be identified; however, in [59] it is suggested that this category could be considered part of the partitional clustering techniques macro-group.
In the framework adopted in this work, only hierarchical clustering is performed, both for utility and for understanding purposes: this process is explained in **4.3.**. The reader who is interested in further explanation on clustering techniques is referred to [59], which delves deeper into the topics that are only briefly introduced or brought up here.
Hierarchical clustering can be performed through two different approaches: the first one is "agglomerative clustering", where the items are initially treated as individual clusters and, at each step, the closest pair of clusters is merged on the basis of a measure of "cluster proximity"; the second approach is "divisive clustering", essentially consisting in the opposite of the first method: all the objects are initially together in a single cluster and, at each step, a splitting process occurs until only partitions containing single items exist. In the rest of this section, the focus will exclusively be on agglomerative clustering, due to the fact that it is the method used in this work.
The most common way to represent the results of hierarchical clustering is by means of a tree-like diagram called "dendrogram", which shows how the clusters have been split (or merged) at each step. Figure 14 shows an example of this graphical representation.

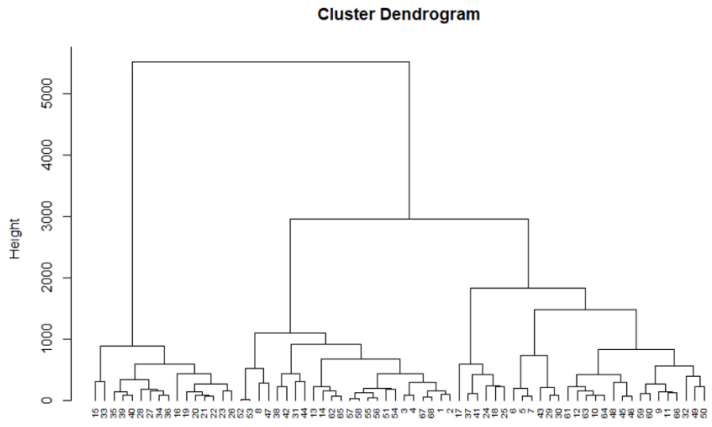

Figure 14 - Example of a dendrogram resulting from agglomerative hierarchical clustering

The typical agglomerative hierarchical clustering process can be summarized in a few steps: first, a proximity (distance) measure is defined and the matrix containing all the distances between items is computed; then, the two closest clusters are merged and the proximity matrix is recalculated; this process is repeated iteratively until only one cluster remains. This methodology is presented in detail in [59], where the most common proximity measures are also introduced; the rest of this section focuses on this aspect.
First of all, it is necessary to choose how to compute the distance between the objects in a cluster: the most common choice is to utilize Euclidean distance, calculated as the square root of the sum of the squared differences between the corresponding coordinates of the two points whose distance is being evaluated. Then, the cluster proximity is computed, based on the type of linkage method employed:

- in **"average linkage"**, cluster proximity is defined as the average pairwise distance of all pairs of points in different clusters;
- in **"complete linkage"** (or **"MAX"**), cluster proximity is calculated as the distance between the farthest two points in different clusters;
- in **"single linkage"** (or **"MIN"**), cluster proximity is defined as the distance between the closest two points in different clusters;
- finally, in **"Ward's Method"**, the proximity between clusters is defined by computing the increase in SSE ("Sum of Squared Errors" or "scatter", calculated as the sum of the squared Euclidean distances between each element and its closest centroid, for all elements in all clusters; a centroid represents a prototype object that describes the cluster, usually defined as the mean of the points in the n-dimensional space considered) that derives from merging two clusters. In this method, the SSE represents the intra-cluster variation (or variance).

Once the hierarchical clustering process has been completed and a dendrogram has been obtained, it is necessary to "cut" the dendrogram at a certain height in order to obtain a certain number of meaningful clusters: this final step represents the validation process and it is usually carried out by means of the so-called "validation metrics", or "indexes", each one calculated via a different mathematical expression prioritizing certain aspects in the data, that suggest the best number of clusters given the final dendrogram structure.

3.4. Techniques for anomaly detection

The final section of this Chapter is aimed at presenting the basic concepts regarding the two techniques for anomaly detection at meter-level that are applied to the results of the computed CMPs, as described in **4.5.** Both these methods involve graphical representations, through which the logic of the detection process can be explained.

3.4.1. Boxplot

The boxplot is a widespread and effective way of representing the distribution of a variable that allows to include a large number of information in just one simple plot. This technique is also employed to identify the data points to be marked as "anomalous" with respect to the examined distribution, as explained later in this section.
The term "boxplot" derives from the shape of the main object of graphical representation, which is effectively a box "containing" data points, as shown in Figure 15.

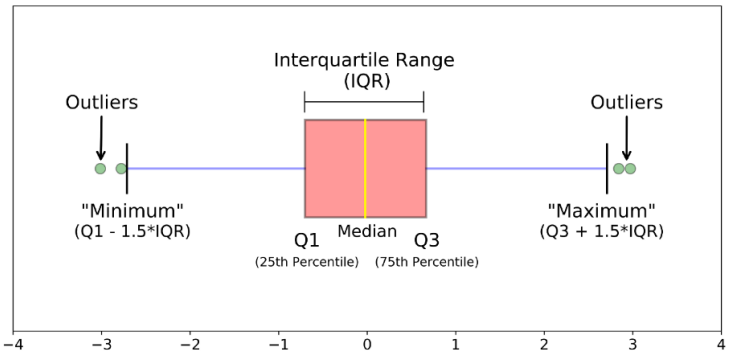

Figure 15 - Boxplot of a nearly normal distribution (source: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51)

To fully understand Figure 15, which perfectly illustrates the essential parts of a boxplot, it is necessary to introduce their meaning from a statistical point of view:

- the **Median/Q2/Second quartile/50th percentile** represents the middle value in the dataset (assuming the dataset is sorted in ascending order; this kind of hypothesis is also at the basis of the next definitions);
- the **First quartile/Q1/25th percentile** represents the middle value between the smallest number in the dataset and the median;
- the **Third quartile/Q3/75th percentile** is the middle value between the median and the highest value in the dataset;
- the **Interquartile Range (IQR)** represents the "distance" between Q1 and Q3;
- the **"whiskers"** are defined as the lines that begin at Q1/Q3 and have an extension equal to a value that is a multiple of the IQR (conventionally, this value is set to 1.5 * IQR);
- the **"Minimum"(Q1 – 1.5 IQR)** and the **"Maximum" (Q3 + 1.5 IQR)** represent the data points that are found at the end of the respective whisker's extension;
- finally, the **"outliers"** are the data points that fall outside the whiskers' extension.

These last elements, the outliers, represent the data points that can be considered abnormal with respect to the underlying distribution: they are associated to occurrences that are statistically "unlikely" and significantly differ from other data points found in the distribution. As previously mentioned, outliers are often conventionally defined as the data points that are smaller than Q1 – 1.5 * IQR or greater than Q3 + 1.5 * IQR; however, the definition of a value for a data point to be considered an outlier is not something that can be unambiguously determined and it often depends on factors such

as the phenomenon represented by the data points that are the subject of study and the properties of the distribution examined (e.g. skewness).
To conclude, boxplots are very efficient in representing the characteristics of a distribution, since the extension of the different elements (the whiskers and the parts of the box between the various quartiles) and the position and quantity of outliers can immediately give the user an idea of how the distribution analyzed compares with a normal distribution (where every boxplot part would be symmetrical), in terms of position of the median, skewness of the distribution and so on.

3.4.2. The elbow method

The elbow method is a technique that is commonly used in partitional cluster analysis to determine the optimal number of clusters in a dataset; however, it can be generalized to other applications as a means for obtaining the "best" number of objects with respect to a given statistical parameter. The main concept behind the elbow method is that of "diminishing returns", which can be described - in a generic way - as the behavior, observable in various phenomena, of decrease in marginal increase (or in marginal decrease, depending on the phenomenon examined) of a parameter of interest ("output") as more and more elements ("inputs") are taken into consideration. Usually, it is possible to identify the point (called "point of diminishing returns") where the above mentioned behavior begins to manifest and the growth (or the decrement) of the output slows down with the increase of the input: this point, knows as the "knee" or "elbow" of the curve, corresponds to a location where the "input versus output" curve clearly bends and becomes increasingly flatter.
As previously mentioned, the elbow method is often used in partitional cluster analysis: the most common implementation of this technique involves plotting a curve that represents the total intra-cluster variation (the SSE, introduced in **3.3.2.**, a parameter that should be minimized as much as possible) on the vertical axis and the number of clusters on the horizontal axis; the elbow represents the point where adding one cluster to the total number of sub-groups found does not result in a significant decrease of the total intra-cluster variation. Figure 16 presents an example of this application.

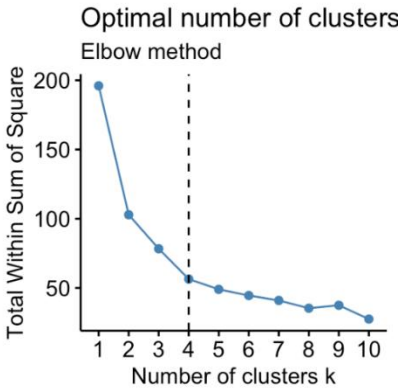

Figure 16 - Example of the elbow method applied to partitional cluster analysis (source: https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/)

Anomaly detection is a slightly less common field of employment of this method; the concept behind the use of this technique, however, is the same: the idea is to plot a curve with a “measure of anomaly” on the vertical axis and the single objects on the horizontal axis, ordered by decreasing value of the previously mentioned parameter that quantifies the abnormality of the single object. Once the elbow of the curve is found, the objects that lie on the left of the elbow are the ones that are labeled as “anomalous”.

4. Methodology

In this Chapter, the methodological steps followed in this work in order to reach the final goal, which is a diagnosis on a sub-load-level of the days marked as anomalous at a meter-level, are described, starting from the very beginning with an initial pre-processing of the dataset that is employed in the case study analyzed. Figure 17 presents a summary of the above mentioned framework.

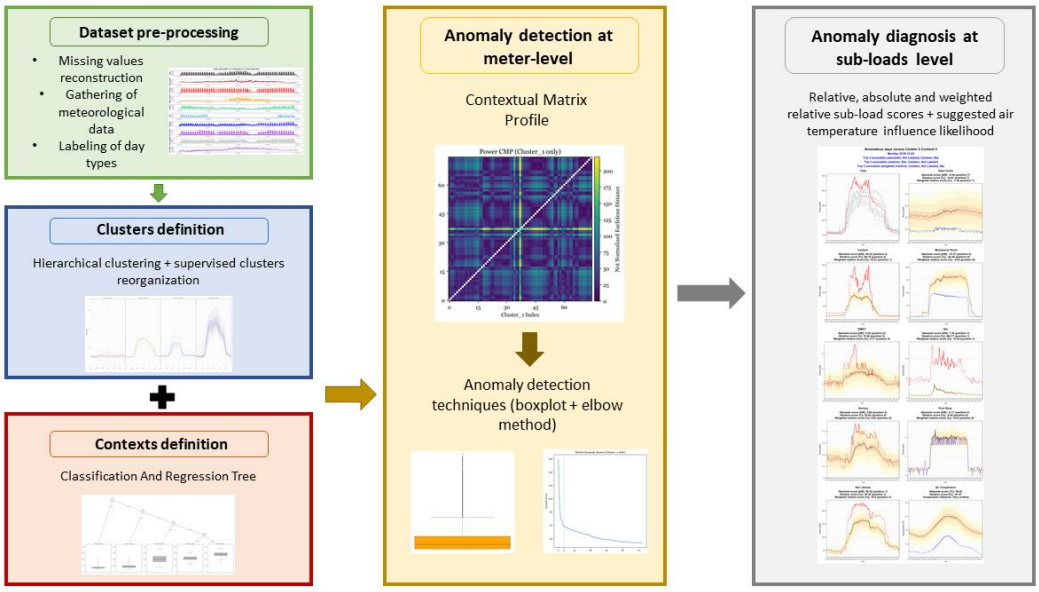

Figure 17 – Visual summary of the framework adopted in this work

4.1. Proposed framework

As previously mentioned, the adopted framework can be split into two macro-processes, the first one being anomaly detection at meter-level by means of the two techniques described in **3.4.** and applied to the results of the Contextual Matrix Profile, discussed in section **4.5.**, and the second one being the diagnosis of the anomaly at a sub-load-level by means of scores - that will be introduced in detail in section **4.6.** – whose aim is to characterize and describe the anomalous sub-loads under different points of view. Before the above mentioned macro-processes can take place, however, preliminary analyses are needed: in **4.2.**, the pre-processing operations necessary to merge all the needed information regarding the examined dataset are discussed; section **4.3.** presents the clustering operation that needs to be applied to the daily Total power demand profiles in the dataset in order to compute different CMPs for different "groups of similar days", with the ultimate goal of comparing objects that are as close as possible

one another to avoid false positives (and other undesired results) in the anomaly detection step; finally, section **4.4.** illustrates the process of definition of different time windows in a day, each one representing a distinct phase in the daily power demand behavior, such as night hours, ramp-up period and so on; this last step before actual anomaly detection essentially has the same objective as the above mentioned clustering operation on daily profiles: the idea is to compute a Contextual Matrix Profile for each combination of "group of days" + "time window" (as defined in the preliminary steps discussed in **4.3.** and **4.4.,** where conventional supervised and unsupervised learning techniques are applied), with the main aim of optimizing the comparison process and analyzing separately the different periods in a single day; this can also enable evaluation on whether a specific day is classified as anomalous during a single time window or during multiple ones.

4.2. Dataset pre-processing

The analyzed dataset is presented in Chapter 5, which delves deeper into its peculiarities. For a better understanding of this section, the reader is referred to **5.1.** for a brief presentation of the examined case study.

The pre-processing step, which is aimed at making the original "raw" dataset suitable for the following analyses, mainly consists of three phases:

1) Reconstruction of missing power demand values: in some occasions, power demand measurements may be missing due to malfunctioning of the monitoring devices or to other unexpected events. When this happens, it is possible to reconstruct the missing values using simple methods, such as linear interpolation, as long as the period with no measurements is relatively short. If this is not the case and a long period with many consecutive missing data points is present, the choice could either be to discard it completely (leaving a "hole" in the time series to avoid "fabricating" data that could be very different from the actual values whose measurements are missing) or to opt for a custom solution to fix the issue.
2) Gathering of meteorological data at each timestep: in order to be able, when possible, to explain certain behaviors in the data thanks to external factors. The meteorological data was not included in the measurements recorded by the monitoring devices; therefore, this kind of information has to be obtained via "external" sources.
3) Labeling of the dataset: this last step is aimed at classifying each day with the maximum possible level of detail, with a more general label ("Holiday" or "Weekday") and then a label that is explanatory for the kind of activity that takes place on each day (such as "Lessons", "Exams", and so on). This kind of process is particularly useful when looking for reasons for certain power demand behaviors but also for the operation of clustering of all the days in the year, dividing them in groups that show similar patterns in terms of Total Power and also present similar labels in terms of day type or daily activity.

4.3. Definition of clusters

As mentioned in **4.1.**, this step consists in separating the daily Total power demand profiles of all the days contained in the original dataset into smaller groups characterized by a high degree of similarity between the daily profiles contained in each subset. The aim of this process, together with the one presented in the next section, is to pave the way for the computation of different Contextual Matrix Profiles, each of them comparing objects that are as similar as possible one another with the main aim of avoiding false positives in the step of anomaly detection at meter-level. In fact, by comparing very different power demand profiles, such as those of Weekdays with the ones of Sundays, for example, it is very likely that one of these types of profiles will systematically be detected as abnormal (most likely the one that appears less frequently). While this is not "wrong" on a purely theoretical point of view, domain expert knowledge suggests that this kind of behavior should be avoided and the anomaly detection step should take into account the already existing differences between power demand profiles of distinct day types. This operation is executed by means of a procedure that mixes conventional unsupervised learning techniques with expert knowledge: first, a dissimilarity matrix that calculates Euclidean distances between the Total power demand profiles of each day is computed and this object is then used for agglomerative hierarchical clustering by means of the R function "hclust"; then, the results of this first unsupervised step are analyzed and an "expert knowledge-based fix" is applied if necessary, in order to define new clusters. This allows to create a new subdivision of days, starting from the one defined with hierarchical clustering, that is more representative of the most significant differences existing in daily power demand behaviors.

4.4. Definition of contexts

This phase is aimed at defining sub-daily time windows that are representative of clearly distinct behaviors in power demand at meter-level, such as night hours, ramp-up period, ramp-down period and so on. As introduced in **4.1.**, the goal of this process is to further improve the expected CMP results by comparing "smaller" objects (fractions of a day instead of the full day) that are as similar as possible, in order to reduce at a minimum the risk of the CMP producing inconclusive or unexpected results. Furthermore, the fragmentation of days opens up a new avenue for the analysis of results, making it possible to examine if a specific day was marked as abnormal only during one time window or in multiple occasions. The time windows are extracted by means of a Classification And Regression Tree (CART) implemented via the R function "rpart", using the Total power as target variable and the time as predictive variable for the tree splits. Holidays, Saturdays and Sundays are not taken into account when constructing the tree model, since their daily power demand profiles are usually flatter than typical working days' profiles and distinct functioning periods are not clearly defined; therefore, their inclusion would likely reduce the accuracy of the model.

The last step in this process is the definition of the duration of the "context". This aspect requires a bit of clarification with respect to terminology; until now, the term "context" has mostly been used as a synonym of "time window". While in the rest of this work this kind of philosophy will still be maintained, the context and the time window represent two different things: the context, according to its definition given in [9], is the time period where a subsequence can start, in order to allow comparison between subsequences that are slightly shifted in time, for a maximum shift equal to the context length. The time window, on the other hand, represents the duration of the considered subsequence. The choice that has been made in this work with regard to the duration of the context is to consider a context length equal to half of the shortest time window, rounded down to the nearest integer: this guarantees that any examined subsequence falls in the area of interest (the time window) for at least half of its length, without the risk of it beginning and also ending in the context.

4.5. Contextual Matrix Profile and anomaly detection at meter-level

Once the clusters and contexts are defined following the procedure reported in the previous sections, the Contextual Matrix Profiles for each combination of these two variables are computed and the anomalous days for each configuration are found, by applying two techniques for anomaly detection and tagging as anomalous only the days that are flagged as abnormal by both. These two techniques are the boxplot and the elbow method; both are based on the comparison between the median values of the different CMP columns, with each column containing the distances of the corresponding subsequence (each subsequence is representative of a portion, defined by the time window considered, of a day; higher distances indicate that the subsequence, and therefore the power demand profile, is less similar to the others it is being compared to) from all the others in the examined group.
The boxplot labels as outliers (and therefore abnormal) those days which fall outside of the extension of the box's whiskers, as described in **3.4.1.**; the whiskers' length is set to the "standard" value of 1.5 times the Interquartile Range (IQR). The elbow method, on the other hand, follows the logic presented in **3.4.2.**: the median values of the CMP columns are ordered from the highest to the lowest and then the so-called "elbow curve" of these values is constructed, with the median distance on the vertical axis and the subsequence index on the horizontal axis; the knee of the curve is then located and the days that lie on the left of the elbow are the ones that are flagged as abnormal. One of the most interesting differences between these two techniques is that the boxplot may sometimes not report any anomaly at all, if no outliers are present; the way the elbow method works, on the other hand, always leads to the labeling of some items as abnormal, since the knee of the elbow curve can be identified in any case, no matter the values involved. Therefore, the reason for taking into consideration both these anomaly detection methods is twofold: on the one hand, this limits the risk of erroneously labeling days as abnormal only because at least one item always has to be tagged as

anomalous when applying the elbow method; on the other hand, the more assurance of a correct detection is available, the more robust the detection process is.
This anomaly detection step is implemented in Python and the CMPs are computed thanks to the source code provided in [9], adapted for this case study.

4.6. Anomaly diagnosis at sub-loads level

This last step is aimed at identifying which sub-loads are the most responsible for a certain anomaly found at a meter-level. The diagnosis is based on the difference between the examined sub-load's profile and the "mean" (or "average") profile for that sub-load in the considered group (a «group» is defined as a combination of context and cluster settings); the mean sub-load's profile is constructed by calculating the mean power demand value for that sub-load, at each timestep, taking into consideration all the days that belong to the examined cluster. The diagnostic process takes into account two sides of the same coin, which are the "absolute" and "relative" differences of the single sub-load's profile from the mean sub-load's profile. The "absolute" difference is expressed in terms of power [kW] and represents how much more (or less) power is requested in a specific moment with respect to the amount of power that is requested in that same moment during the average group day (e.g. if the anomalous day's sub-load power demand at 03:00 is 30 kW and the average group day's sub-load power demand at the same timestep is 20 kW, the absolute difference at 03:00 is equal to 10 kW). On the other hand, the "relative" difference is expressed in terms of percentage [%] and indicates how much greater (or smaller), in a specific moment, the examined day's load is with respect to the average group day's load. (e.g. if the anomalous day's sub-load power demand at 03:00 is 30 kW and the average group day's sub-load power demand at the same timestep is 20 kW, the relative difference at 03:00 is equal to +50%).
Since both these aspects can be useful for the interpretation of results, the diagnostic process first returns these information separately and then an attempt to consider them together is performed. This results in the calculation of three "scores" for each sub-load:

1) The **"Absolute"** score: at each timestep, the difference in terms of kW between anomalous day's power demand and average group day's power demand is calculated; all these differences are then added up and their sum is divided by the number of timesteps, to obtain a difference in terms of kW for that anomalous day profile with respect to the average group's daily load.

$$\textbf{\textit{Absolute score}} = \frac{\sum_{first\ timestep}^{last\ timestep}(\textbf{\textit{Power of anomalous day}} - \textbf{\textit{Power of average group day}})}{\textbf{\textit{number of timesteps}}}$$

2) The **"Relative"** score: at each timestep, the difference in terms of kW between anomalous day's power demand and average group day's power demand is calculated; this difference is then divided by the power demand value of the average group day; all the values obtained this way at each timestep are then added up and their sum is divided by the number of timesteps, to obtain a

relative difference in terms of percentage for that anomalous day profile with respect to the average group's daily load.

$$\textbf{Relative score} = \frac{\sum_{first\ timestep}^{last\ timestep} \frac{(\textbf{Power of anomalous day} - \textbf{Power of average group day})}{\textbf{Power of average group day}}}{\textbf{number of timesteps}}$$

3) The **"Weighted Relative"** score: at each timestep, the difference in terms of kW between anomalous day's power demand and average group day's power demand is calculated; this difference is then divided by the power demand value of the average group day and the resulting value is multiplied by the weight of the considered sub-load on the Total power at that timestep; all the values obtained this way at each timestep are then added up and their sum is divided by the number of timesteps, to obtain a weighted relative difference in terms of percentage for that anomalous day profile with respect to the average group load.

$$\textbf{Weighted relative score} =$$

$$\frac{\sum_{first\ timestep}^{last\ timestep} \frac{(\textbf{Power of anomalous day} - \textbf{Power of average group day}) * \textbf{Weight of sub-load power on Total power}}{\textbf{Power of average group day}}}{\textbf{number of timesteps}}$$

This last score allows to combine both the absolute and the relative point of view, by applying a correction to the relative difference that is based on the actual "magnitude" of the anomaly.

Finally, the air temperature is taken into account by constructing a mean group day profile for this parameter and evaluating the absolute and relative differences in the same way as the sub-loads; then, a message based on the value of the Relative score is displayed, suggesting possible external air temperature influence as a "hint" of general nature: the interpretation (e.g. if there are any sub-loads affected by this factor and, if yes, which ones) is left to the user.

5. Case Study

In order to be able to test and evaluate the effectiveness of the methodology presented in the previous Chapter, a real-world dataset is studied. The analyzed dataset contains one year of power demand measurements for one of Politecnico di Torino's Medium Voltage/Low Voltage (MV/LV) transformation cabins, which serves different areas of the university campus; these zones represent the sub-loads that will be considered.
In **5.1.**, the dataset considered in this work is briefly introduced and an initial description of the electrical loads subject of this study is provided.
In **5.2.**, initial analyses – mostly by means of graphical representations – on the different sub-loads are performed, in order for the reader to be able to fully appreciate the content of the following Chapters.

5.1. Dataset description

As previously mentioned, the power demand data taken into consideration in this work refers to Politecnico di Torino's university campus, which is equipped with a loop of ten Medium Voltage/Low Voltage transformer substations that provide Low Voltage electrical power to different zones of the campus.
Politecnico di Torino is one of the most famous Italian universities for Engineering and Architecture. Its lecture rooms are located in four main building complexes in different areas of Turin; in this work, the main campus building is analyzed: the complex, opened in 1958, is located in Corso Duca degli Abruzzi 24 and mainly hosts lecture rooms, laboratories and offices for the Engineering faculty, for a total floor area of around 122000 m^2.
The power demand values examined in this case study refer to "substation C", which feeds several campus facilities, for an overall floor area of almost 42000 m^2. The power demand measurements refer to the full year of 2019, from Jan 1st to Dec 31st, sampled with a 15-minute frequency, for a total of 35040 observations.
Each observation consists of a "Total power" value and various "Sub-loads power" values, that represent how the Total power demand is split amongst the single consumers that are equipped with a monitoring device at sub-meter level. The sub-loads considered are:

- The **Data Centre**, where the university servers are located and whose electrical needs are mostly related to the servers' electrical load and a room chiller that prevents overheating in electronic devices;
- The **Canteen**, located at the ground floor of the main building, which presents loads connected to refrigerators (base-loads), ovens and dishwashers (peak-loads) and an air handling unit;
- The **Mechanical Room**, which hosts equipment for the production of chilled water as well as the circulation pumps for both hot and chilled water. The chilled water is produced by means of two chillers with nominal required power equal to 220 kW and a cooling power of 1120 kW as well as a reversible water-water

heat pump with nominal required power of 165 kW and a cooling power of 590 kW;

- The **Department of Mathematics ("DIMAT")**, which is located at the 3rd and 4th floor of the main building and requires power for lightning equipment, computers, fan coils and plug loads;
- The **Bar "Ambrogio"**, which is situated at the ground floor of the main building and presents loads that are mostly related to lightning equipment and kitchen appliances such as refrigerators, ovens, dishwashers and so on;
- The **Rectory**, which hosts administration offices and whose loads are similar to the ones found in the DIMAT;
- The **Print Shop "Copysprinter"**, which is located at the first underground floor next to the library and is equipped with various computers and printers as main power consumers;
- The **"Not Labeled"** sub-load.

The first 7 of the above mentioned sub-loads contribute to the so-called "Labeled power", which is the resulting power demand from the consumers that are equipped with a sub-meter monitoring device. However, the Total power demand is always higher than the Labeled power demand since, at every moment, additional power is requested by facilities and appliances that are not monitored at a sub-meter-level (e.g. lightning systems, HVAC components, electronic devices, plug loads, elevators, alarm systems and so on). This part of the Total power load is referred to as the "Not Labeled" load and it represents the 8th sub-load in this case study.

5.2. First dataset analyses

The first step of the analysis is a preliminary observation of the dataset, in terms of Power demand and Air Temperature values, using different techniques for data representation. The aim of this process is to extract the most evident pieces of information about the dataset, such as patterns related to seasonality or to specific days/periods during the year, even if some of them cannot be explained at a first glance.

Figure 18 represents the time series for Power demand and Air Temperature values for the full year of 2019. By examining this plot, various interesting aspects become clear; the most relevant ones are the following:

- every sub-load has its own "overall magnitude" and therefore will affect the Total power demand in a specific way: for example; the Rectory or the Print Shop never exceed 30 kW, while the Canteen or the Mechanical Room loads present values up to over 200 kW;
- the seasonality aspect of certain loads is highlighted: the most striking example of this is the Mechanical Room sub-load, which stays around 50 kW or lower during winter and spring months, and shows peaks of over 300 kW during summer months. The main "activity period" of this sub-load is therefore related to increasing cooling needs and this finds confirmation in the time series of the

external Air Temperature which "mirrors" the Mechanical Room power demand time series;

- the longer Holiday periods (Christmas, Easter and mainly Summer Closing) can be immediately identified in all the time series, which show an overall drop in power demand that is more evident in some cases (such as the Print Shop or the Canteen) and less evident in others (such as the Rectory);
- the time series related to the Bar Ambrogio load presents a long period, approximately from the beginning of April to the end of October, where the power demand is zero or close to zero: the explanation for this phenomenon lies in the fact that, during the above mentioned time period, the Bar was closed for renovation.

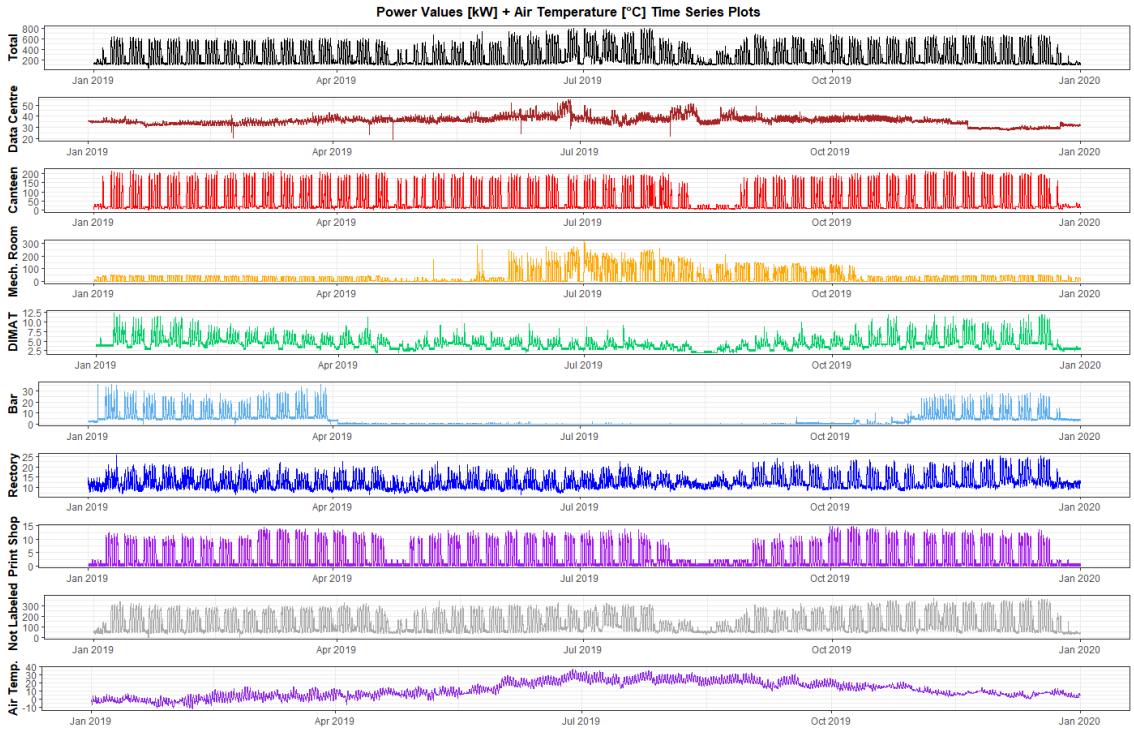

Figure 18 - Time series plots for Total power demand, all sub-loads power demand and Air Temperature

Other ways of representing this kind of data, that can lead to the discovery of other useful information, are histograms and boxplots, represented in Figure 19 and Figure 20. For example, Figure 20 clearly illustrates how specific sub-loads, such as the Canteen or the Mechanical Room, show a large number of "upper" outliers, which represent extreme behavior related to higher-than-normal power demand: this is a first hint towards the presence of "more anomalous" data in certain sub-loads, that is expected to emerge from subsequent analyses. Also, the fact that some sub-loads show large differences between the mean and the median values for power demand is an indicator of large values of skewness for the corresponding distributions, which is well represented in Figure 19. As expected, since the mean and median values are extremely similar, the distributions that more closely resemble a normal distribution are those of

the Data Centre, of the DIMAT and of the Rectory. On the opposite side, the above mentioned Canteen and Mechanical Room distributions are very distant from a normal distribution and the presence of the upper outliers, which are generally more frequent than the lower ones in all distributions, is very clearly represented by the fact that the length of the right tail of the distributions, that only comprises a small number of data points, is greater than the length of the left tail: this is also known as "positive skewness" and it can be seen that most of the sub-loads' distributions (except for the Data Centre) present this characteristic. Another hint that confirms this behavior can be found by looking at the boxplots in Figure 20, where almost all sub-loads, with the exception of the Data Centre, present a "lower half" of the box that is very narrow, indicating that the first 50% of data points is represented by a small variety of values.

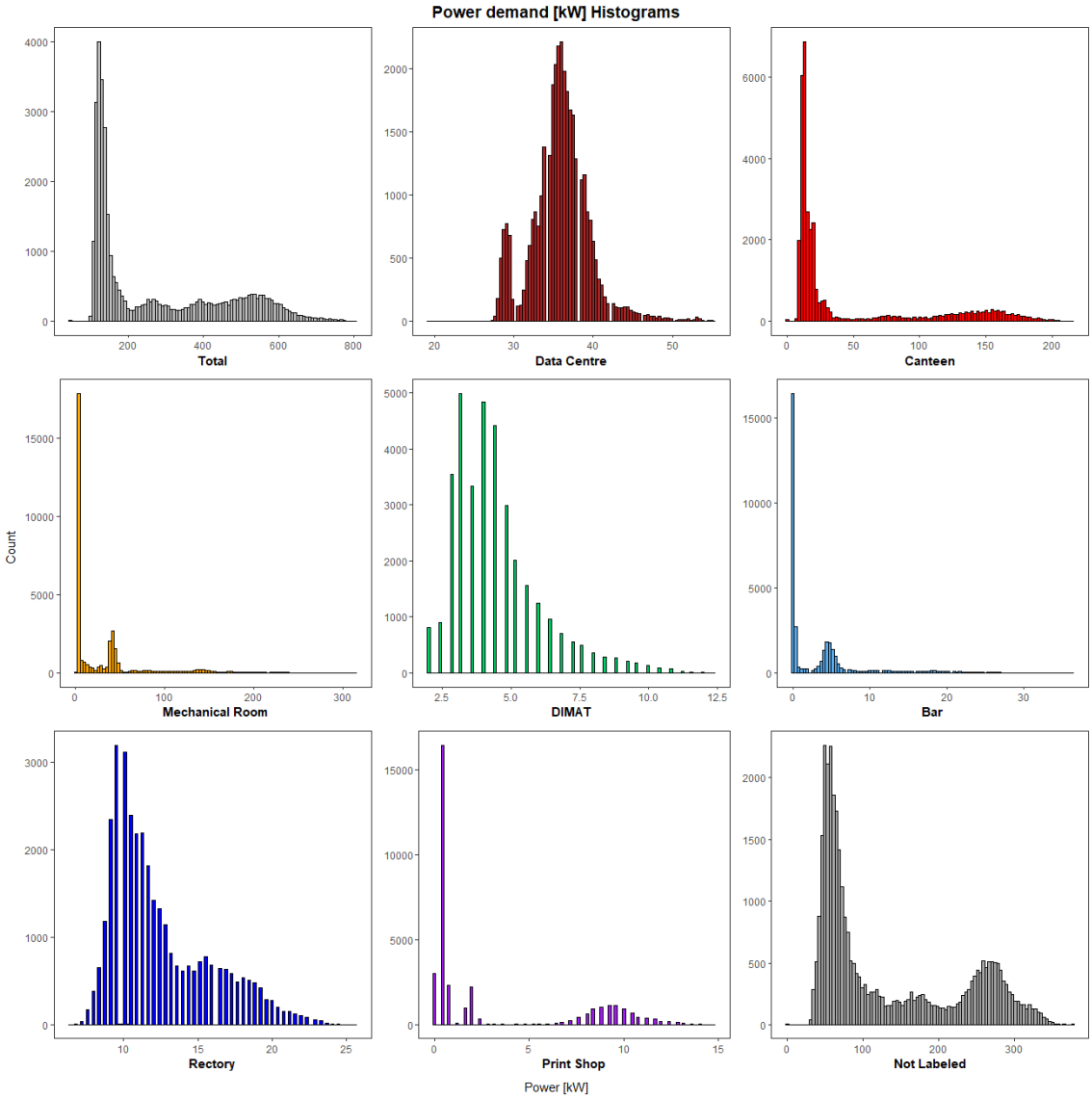

Figure 19 - Histograms of Power demand values for Total load and all the sub-loads

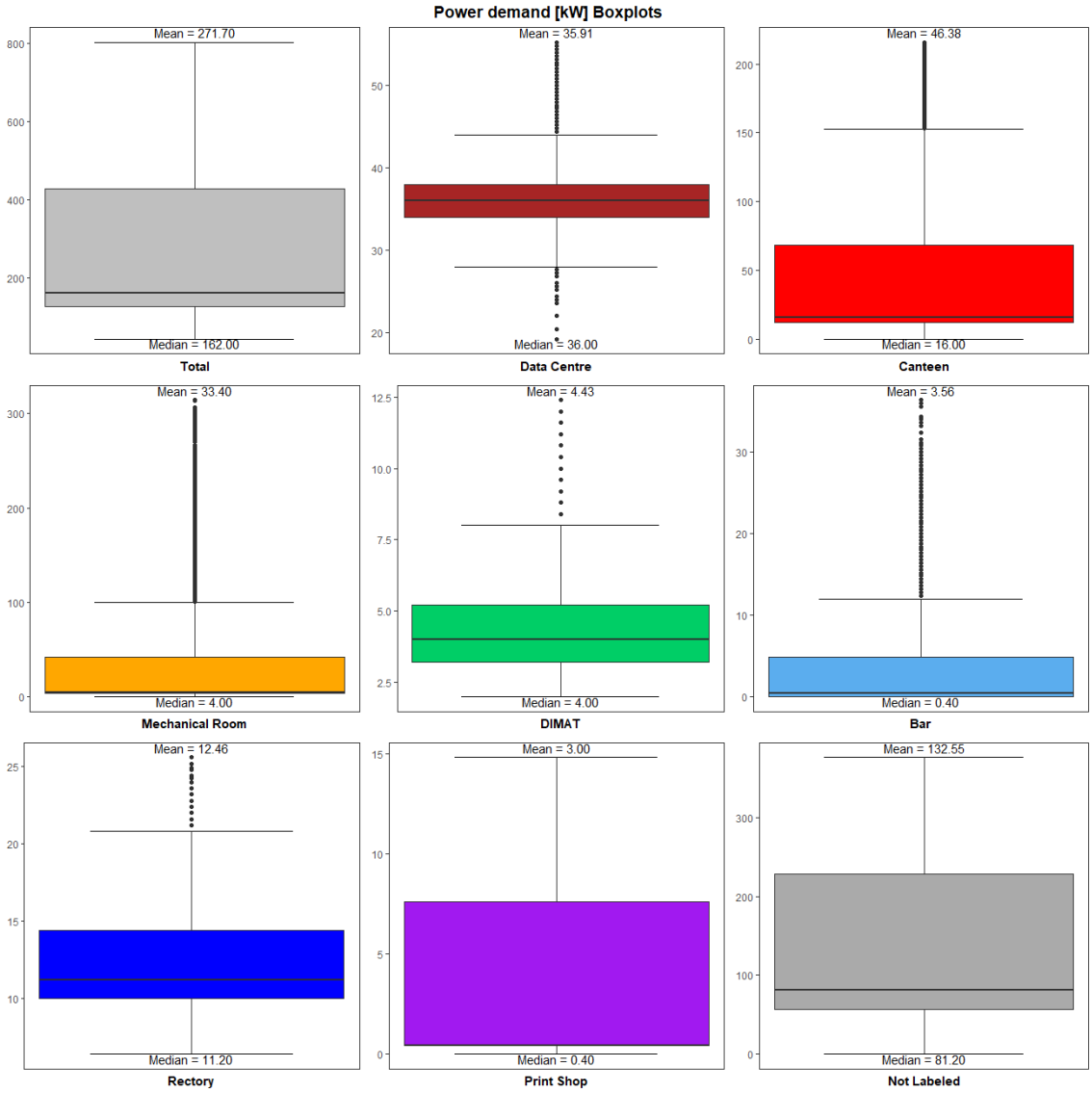

Figure 20 - Boxplots of Power demand values for Total load and all the sub-loads

Other useful information can be extracted by analyzing each sub-load on its own and taking into account the subdivision of the power demand values across the various months or day types. The rest of this section presents this process following a mostly graphical approach, with brief comments about the most important aspects that emerge from the comparison between the different representations.

Figure 21 illustrates how the **Total Power** daily patterns are similar throughout the whole year, with the exception of the summer months where the power demand is generally higher, even during morning and evening hours. The months when the main Holiday periods occur, on the contrary, show generally lower power demand values; both these aspects are reflected by the position and the extension of the boxplots in Figure 22. Figure 23 highlights two main aspects: the first one, which is the lower power demand during Holidays, is expected; the second one, however, is less obvious: the distribution for Holidays presents a large number of upper outliers, which is indicative of a situation that is far from normality (in terms of distribution) and can be explained

by the fact that "Holiday" is a broad term that aggregates various kinds of days, that belong both to "warmer" and to "cooler" months and these seasonal differences are amplified by the fact that the Holidays boxplot contains far less elements than the Weekdays boxplot, which leads to a "weaker" definition of the median value.

Moving to the **Data Centre**, Figure 21 shows the usual kind of higher power consumption behavior during summer months, however to a lesser extent when compared to the Total Power. An interesting aspect that emerges examining both Figure 21 and Figure 22 is the slight decrease in power demand during the months of November and December. The Data Centre load is apparently not influenced by the day type, as Figure 23 illustrates: this is indicative of a sort of "base-load" that is always active and can be attributed to the university servers.

The most unexpected discovery that can be done by observing the Figures in Appendix B is that the **Canteen** load shows a behavior that is almost identical, on a daily basis, during the whole year (except for Holidays): summer daily patterns are only different towards the end of July, where higher power consumption is registered during the evening hours. Another interesting fact is that there is an extremely large number of upper outliers during all months, which appear to be due to the very high power demand that typically occurs between 12:00 and 15:00 and, in some cases, also during the early morning hours: this is most likely due to the peak-load appliances present, such as ovens and dishwashers.

The **Mechanical Room** sub-load is the most striking example of load dependent on seasonality and this is clearly shown in Figure 21 and Figure 22: the power demand is very low during the first five and last three months of the year, when there is little to no need for cooling; during summer months, however, this load becomes very important, with peaks of over 300 kW in July and days of "always on" behavior, as shown in Figure 21. During Holidays, as Figure 23 illustrates, this load becomes negligible in most cases; however, the large number of outliers shows that, once again, different "kinds" of Holidays are related to different load behaviors.

The **DIMAT** sub-load, which is the smallest overall, shows a behavior of seasonality that is, though "weaker" - in terms of direct relationship - than other loads seen so far, opposite to the "standard" behavior (higher during warmer months, lower during cooler months). This is well pictured in Figure 21 and Figure 22 and it is most likely due to the fact that less loads related to electronic appliances, such as computers and plug loads, are active during summer months. A Weekday – Holiday difference between power demand values can be seen in Figure 23, although this difference is small and hardy impactful on the Total Power values due to the scale of this sub-load.

As mentioned before, the **Bar** stayed closed for a large part of 2019 and this is very evident when looking at Figure 21. What can be seen from Figure 22, however, is that the general behavior is very similar in all months of opening except for December, that shows lower values due to Christmas Holidays. Even though Figure 23 does not highlight different behaviors between Holidays and Weekdays, this kind of differentiation can be seen in Figure 21: the reason for this is mainly due to the fact that the boxplot representing Weekdays in Figure 23 includes all days of closing, which "pollute" the real statistics.

The next subject of examination is the **Rectory** sub-load; this load is comparable with the DIMAT sub-load, in the sense that both are a tiny fraction of the Total Power, the appliances responsible for power demand are similar and the month/day type variations are small. Once again, for the same reasons presented for the DIMAT sub-load, a behavior of seasonality which results in slightly higher loads during cooler months can be seen, as shown in Figure 21 and in Figure 22.

From what can be seen in Figure 23, the **Print Shop** sub-load is close to zero during Holidays, as expected. This also results in lower values during the months with longer Holiday periods, as represented in Figure 22. Furthermore, this load shows a behavior that is quite unique and that is clearly captured in Figure 21 and in Figure 22: during the months of March and October, which correspond respectively to the beginning of the lessons in the second semester and in the first semester, the power demand values are generally higher. This is very likely due to the fact that, in those months, a large number of students buy (and therefore print) books and lecture notes for the courses they will attend during the semester.

The last subject of this preliminary analysis is the **Not Labeled** power demand, which is treated as a sub-load of its own. Once again, this load shows a huge difference in terms of power values between Weekdays and Holidays (Figure 23), while the seasonality aspect is less obvious: summer months (except for August, due to the Summer closing) behave similarly to other months in terms of daily patterns, except for a period of around a week during the first half of July where both morning and evening hours show unusually higher power demand, as visible from Figure 21. This Figure and Figure 22 also illustrate how generally higher values are registered during the last three months of the year.

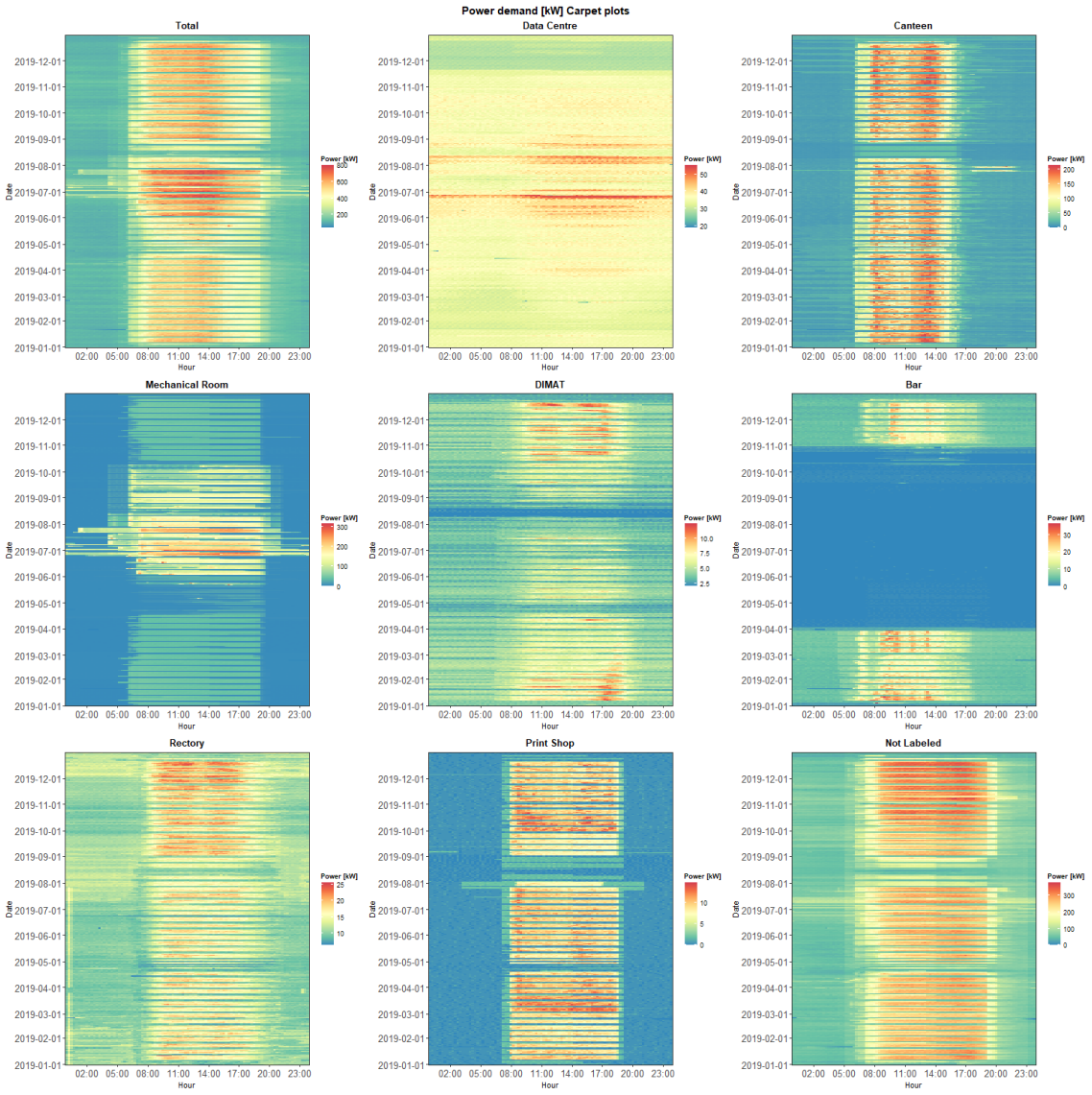

Figure 21 - Carpet plots of Power demand values for the Total load and all the sub-loads

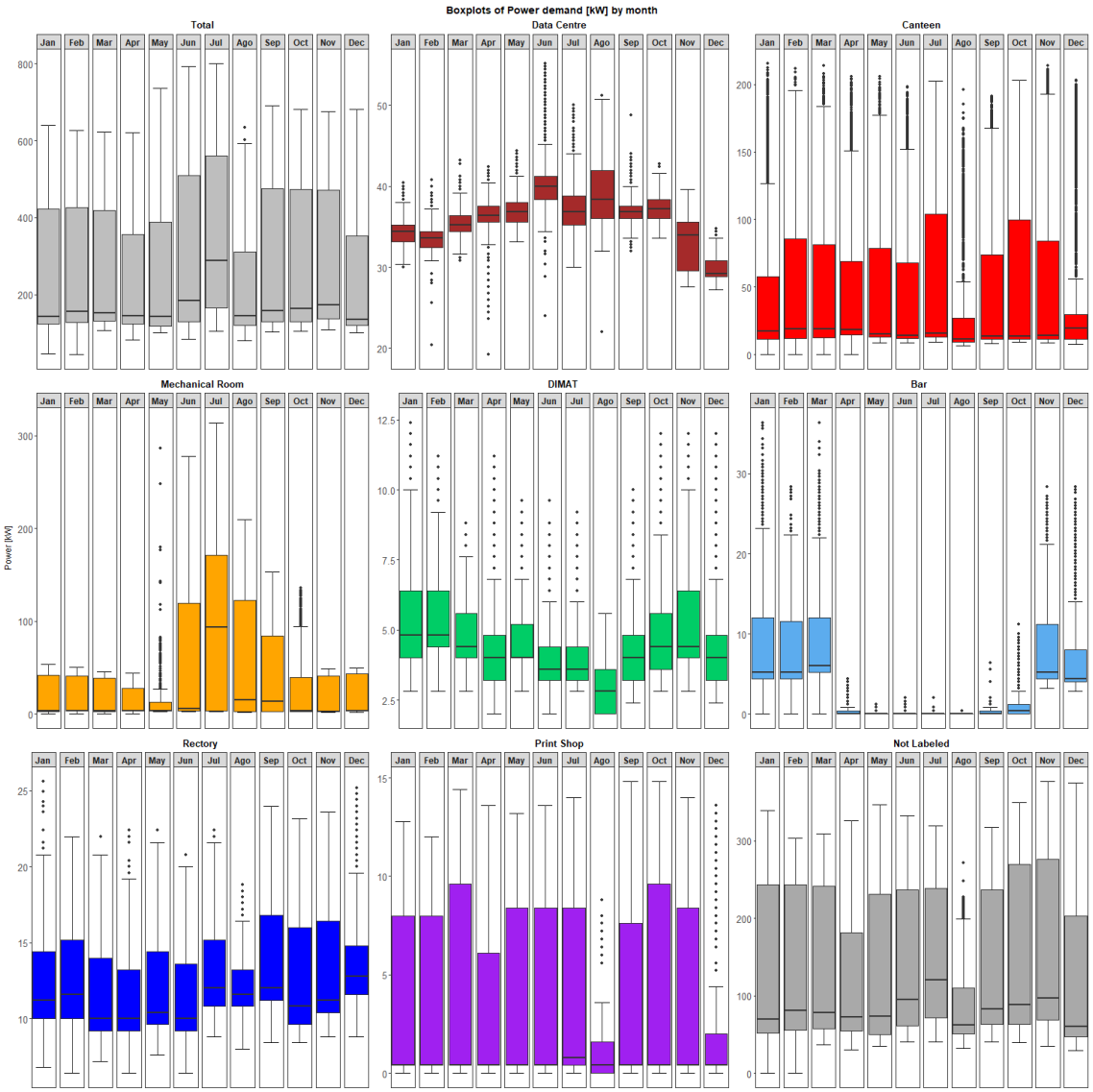

Figure 22 - Boxplots of Power demand values for the Total load and all the sub-loads in each month

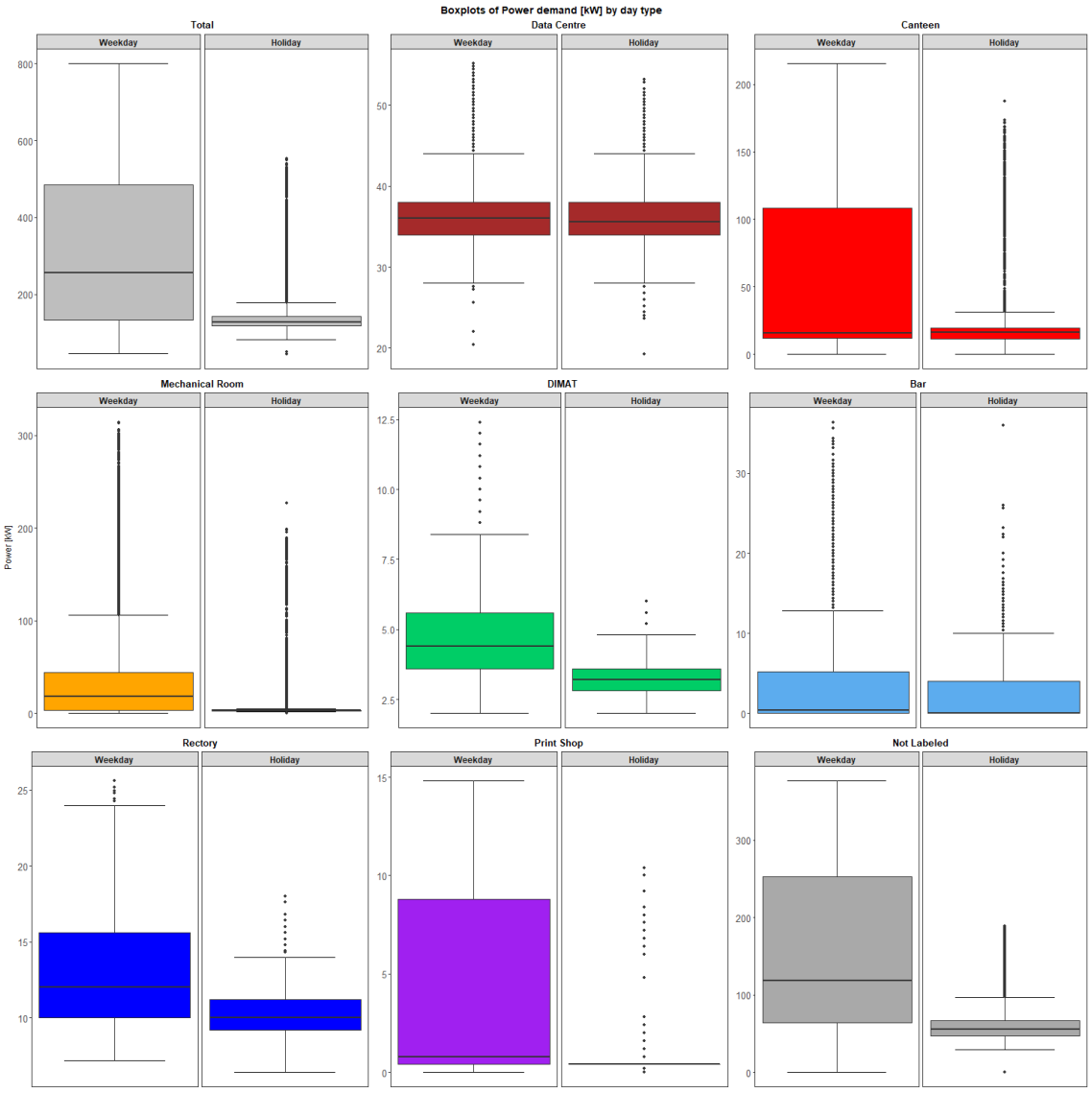

Figure 23 - Boxplots of Power demand values for the Total load and all the sub-loads in each day type

6. Results and discussion

This Chapter is aimed at presenting and discussing the results obtained by applying the methodological steps described in Chapter 4 to the case study introduced in Chapter 5. The results are organized in the same way Chapter 4 was structured, in order to better distinguish every single phase of the analysis and appreciate the contribution to the final results made at each step.

6.1. Dataset pre-processing

After performing the three operations presented in **4.2.**, the results were the following:

1) Reconstruction of missing power demand values: only one measurement was missing, at 00:00 of 2019-04-23; linear interpolation was used to fill the missing data for Total power demand and each sub-load's power demand, while the Not Labeled sub-load's power demand was obtained by subtracting the sum of the first 7 sub-loads' power demand values from the Total power demand value.
2) Meteorological data was obtained via the Solcast API [60], specifying the time period and the location of interest. The most interesting external factor for the analyses in this work is Air Temperature, since the cooling load related to the Mechanical Room is certainly dependent on the seasonality, and other loads may exhibit dependence from this variable as well. Other external factors, such as Relative Humidity or Global Horizontal Irradiance, were also originally considered; however, their relationship to any of the loads were not evident and they were discarded as possible "influencing factors" in the early stages of analysis.
3) Labeling of the dataset: the resulting subdivision, obtained by applying the described labeling process, is shown in Figure 24. A large number of different day types and daily activities can be noticed throughout the whole year, which leads to believe that this kind of annotation on each single day will be useful in later analyses to explain certain behaviors that may otherwise seem unexpected.

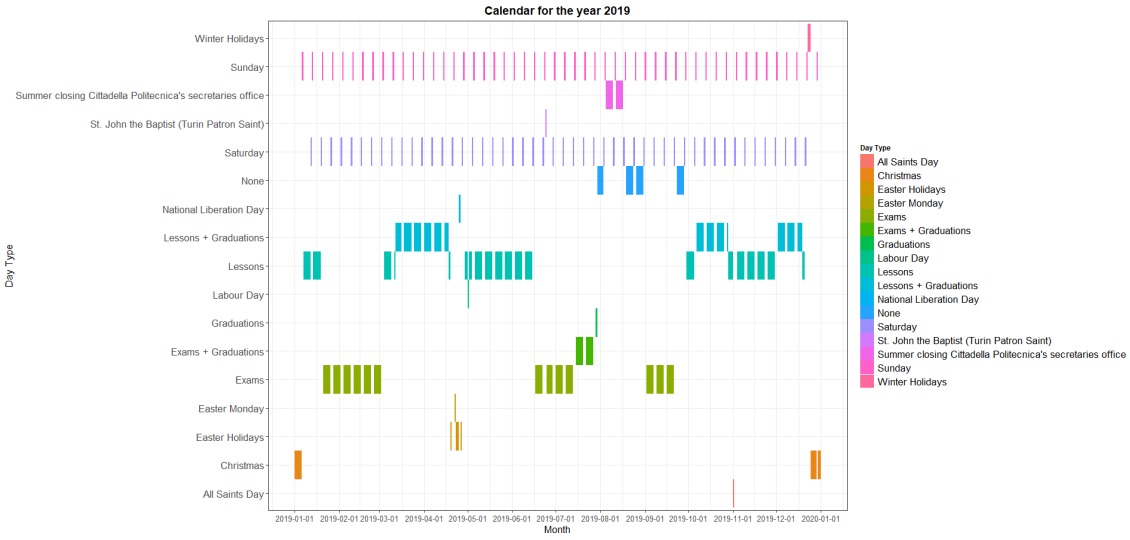

Figure 24 - Calendar for the year 2019, with all the day types and activities

6.2. Definition of clusters

The initial unsupervised hierarchical clustering step was performed various times, with different settings at each try, in order to appreciate the differences resulting from a parameter change and to evaluate the quality of the clusters found by the algorithm. Different types of distance measures were initially tested, including Euclidean, Manhattan and Minkowski distances: the results were clear in suggesting that Euclidean distance, which is the most common distance measure adopted for clustering, was also the best solution for this case; the choice of other distance measures either had little to no impact on the final subdivision of profiles or returned visibly uneven sub-groups.
Different agglomeration methods were also tested, including single linkage, average linkage, complete linkage, and Ward's criterion: this last option turned out to be the best in terms of identifying groups containing a significant amount of items and with a very distinct and well-defined profile shape.
The number of clusters to be identified was also subject of analysis. An unsupervised approach using the R function "NbClust" was initially tested: this function analyzes the results of different validation metrics (indexes) and returns as the "best" (suggested) number of clusters the most recurring one among all indexes. The minimum (3) and maximum (6) number of clusters were given to the algorithm on the basis of expert knowledge, which suggested that at least three "trivial" groups could be identified (representing respectively days of normal systems functioning e.g. Weekdays, days of "half" functioning e.g. Saturdays and days of total closing of the university campus e.g. Sundays) and that the dataset should be divided in no more than six subsets to avoid considering an extremely large number of cases when computing the Contextual Matrix Profiles as well as to prevent a phenomenon of "overfitting" for each CMP instance, so that it would be difficult to even find anomalies in each group. The algorithm returned

"4" as the suggested number of partitions: this resulted in the definition of clusters with a good homogeneity in terms of profile shape; however, the main issue was that the unsupervised process led to the creation of two clusters containing Weekdays profiles whose only difference was the power demand magnitude, as shown in Figure 25 with clusters number 3 and number 4. Once again, expert knowledge suggested that this kind of distinction is not wrong in theory, however the goal of this process is to group together days with the same daily power demand behavior, no matter the magnitude.

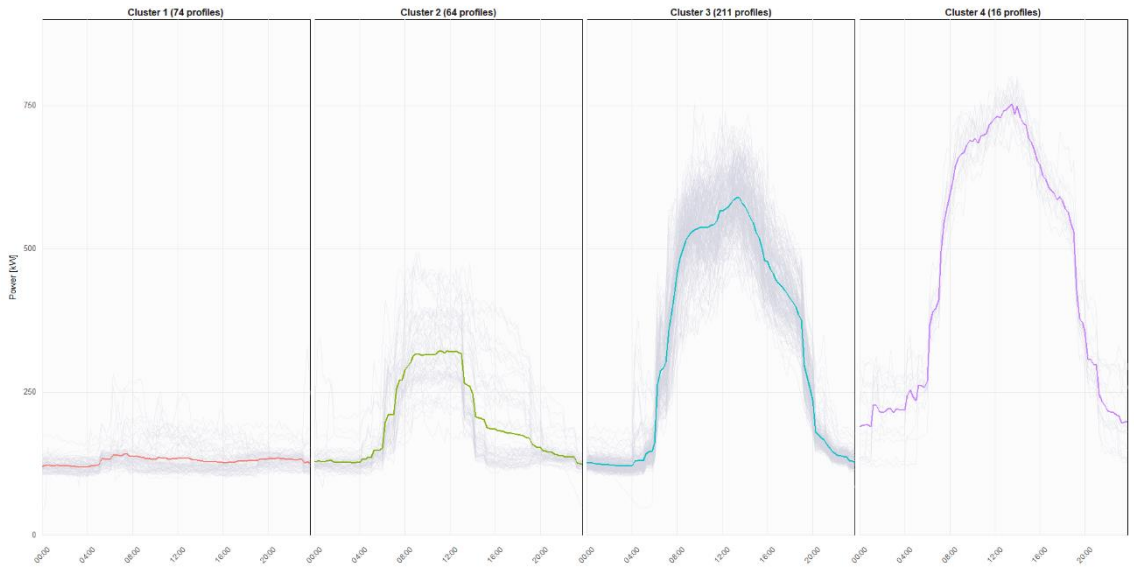

Figure 25 - Results of hierarchical clustering with clusters number set to 4

Therefore, other tries were made: a different number of clusters each time was given as an input to the clustering function and the results were evaluated: it was found that the algorithm returned the most accurate classification, shown in Figure 26, with a number of clusters set to 6. However, this result was not satisfactory due to the same issue that was found earlier: the last three clusters in Figure 26 clearly group together profiles that show the same kind of power demand behavior, belonging to a full working day, with the only difference between them being the magnitude of the power demand curve.

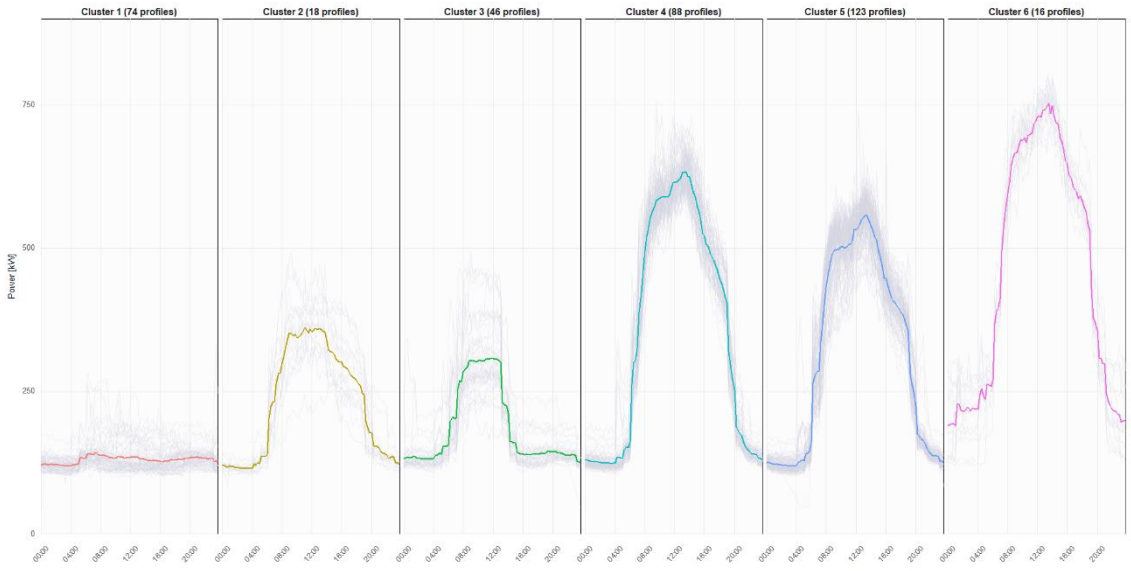

Figure 26 - Results of hierarchical clustering with clusters number set to 6

Since it was clear that, in any case, an unsupervised process would ultimately lead to this unwanted distinction, a "supervised fix", as introduced in **4.3.**, was applied: the last three clusters were merged together and some minor corrections were also applied to the rest of the clusters (all Sundays were moved into cluster 1 and all Saturdays were moved into cluster 3, for a total of one profile going from cluster 2 to cluster 1 and one profile going from cluster 2 to cluster 3). The resulting subdivision is shown in Figure 27, where 4 clusters can be identified: the first one contains all the Sundays and days of total closing of the university campus, for a total of 75 profiles; the second one is representative of the 16 days that correspond to "semi-regular" functioning: days when the campus is open but no lessons or exams take place and students are mostly not present (usually in July/August); the third one contains 47 days of "half-opening" of the campus, such as the Saturdays; the fourth one groups together all the regular working days, no matter the magnitude of the Total power demand profile, for a total of 227 profiles.

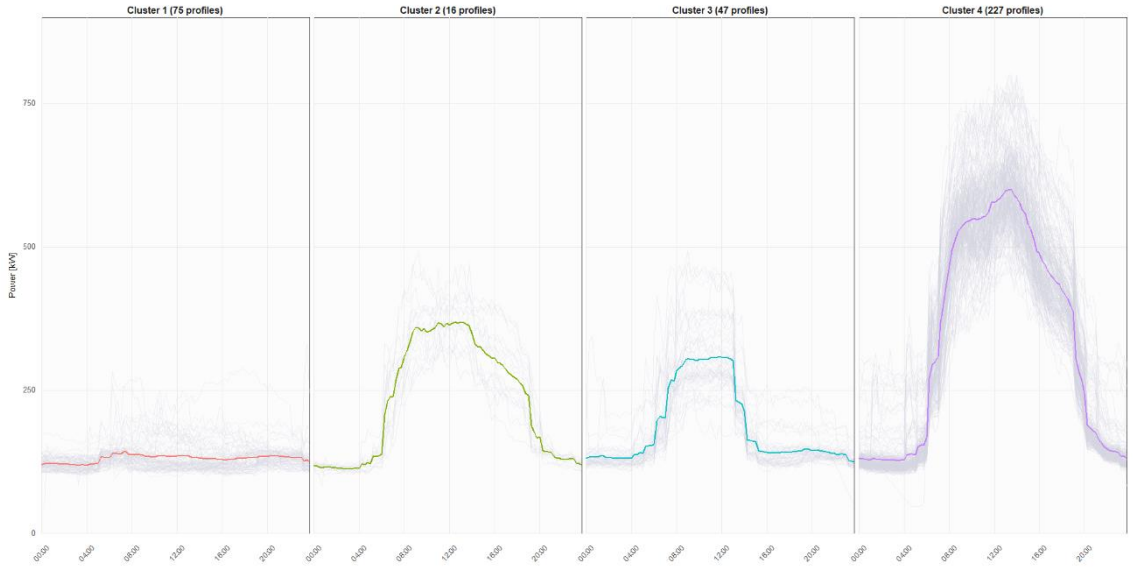

Figure 27 - Results of hierarchical clustering + supervised reorganization of clusters

6.3. Definition of contexts

The definition of time windows by means of a decision tree was performed with a logic similar to the one employed in the previous step, based on a "trial-and-error" process: the CART settings were modified at each try, one parameter at a time, in order to find the configuration leading to the best results based on expert knowledge. The ultimate goal of this procedure was to avoid defining time windows either too wide (comprising more than one type of power demand behavior, based on what is known about daily systems operation) or too narrow (resulting in a fragmentation that over-characterizes the daily power demand profile and separates the same type of behavior into two distinct windows).

First of all, the impact of the number of cross-validations was tested: no difference, in terms of the final tree structure, was found between the standard value of 10 and a higher value. Then, the complexity parameter was evaluated: as expected, increasing the value of this setting from the standard choice of 0.01 to values close to 0.1 or even higher led to a tree with less terminal nodes and less splits, with very wide time windows; the final choice was to not impose any limitation at all with regard to the tree complexity (by setting cp equal to zero, which resulted in the same structure obtained with cp equal to 0.01) and, if necessary, modify this setting at a later time. Next, Weekends and Holidays, which were initially removed when constructing the CART due to the reasons presented in **4.4.**, were re-introduced in the analysis, in order to find out their actual impact on the final tree structure: the algorithm returned splits that were extremely similar to the ones produced without considering the above mentioned days (a difference of at most half an hour in some splits), therefore the final choice was to continue with the initial settings with regard to this aspect, for "safety" reasons and also since the exclusion of those days

was motivated by expert knowledge. The maximum tree depth was also subject of analysis: however, since the tree structure obtained was generally quite simple, changing the value of this parameter had no impact on the final results (only by setting it to values of 2 or 3 led to significant changes, but common sense suggested that this kind of limitation on such simple structures made no sense).
Finally, the parameter whose modification was found to impact the final results the most was the "minbucket", which corresponds to the size of the terminal nodes of the tree. After experimenting with different values of this parameter, the best solution was to set the minimum length of the time windows to 2 hours and 30 minutes; a smaller value (e.g. 2 hours) led to an increased fragmentation of the morning hours, which meant that some of the time windows defined this way were not significant in terms of unique power demand behavior and did not represent any real change in systems operation; on the other hand, increasing this parameter's value was also found to return inaccurate results: certain periods where an actual power demand behavior change took place ended up into the same time window.
The CART that is obtained from this configuration is represented in Figure 28, where 5 time windows can be distinguished: the first one goes from 00:00 to 06:15 and represents night hours; the second one goes from 06:15 to 08:45 and captures the beginning of the ramp-up period, where various systems are switched on and the power demand sharply increases; the third one, going from 08:45 to 15:30, is representative of the end of the ramp-up period and of the hours of "peak" power demand; the fourth one goes from 15:30 to 19:00 and corresponds to the last hours of peak load and to the beginning of the ramp-down period, where systems begin to be switched off; the fifth and last one goes from 19:00 to 24:00 and captures the end of the ramp-down period and the beginning of night hours.
Finally, the context length is defined as described in **4.4.**: since the shortest time window has a duration of 2 hours and 30 minutes, the context length is chosen to be equal to 1 hour.

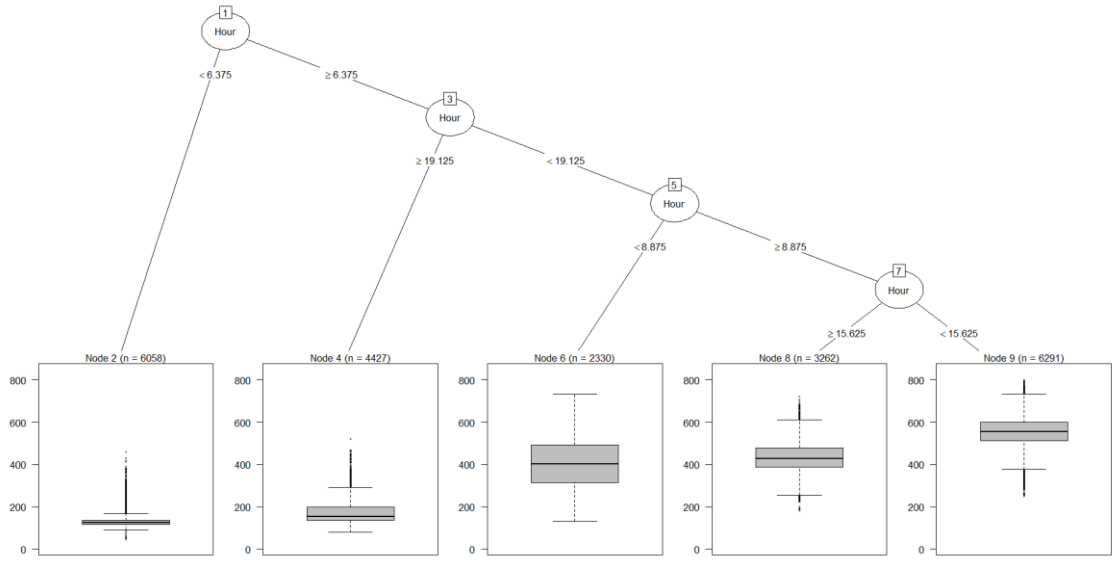

Figure 28 - Classification And Regression Tree defining the daily time windows

6.4. Contextual Matrix Profile and anomaly detection at meter-level

The first part of this step consisted in the computation of all the Contextual Matrix Profiles, one for each combination of context and cluster. An example of one of the CMPs obtained is presented in Figure 29, where at least two columns with higher median distances can be distinguished, with index numbers between 30 and 45. This kind of visualization can immediately alert the user of the existence of at least two instances that will most likely be labeled as anomalous by the detection techniques employed afterwards.

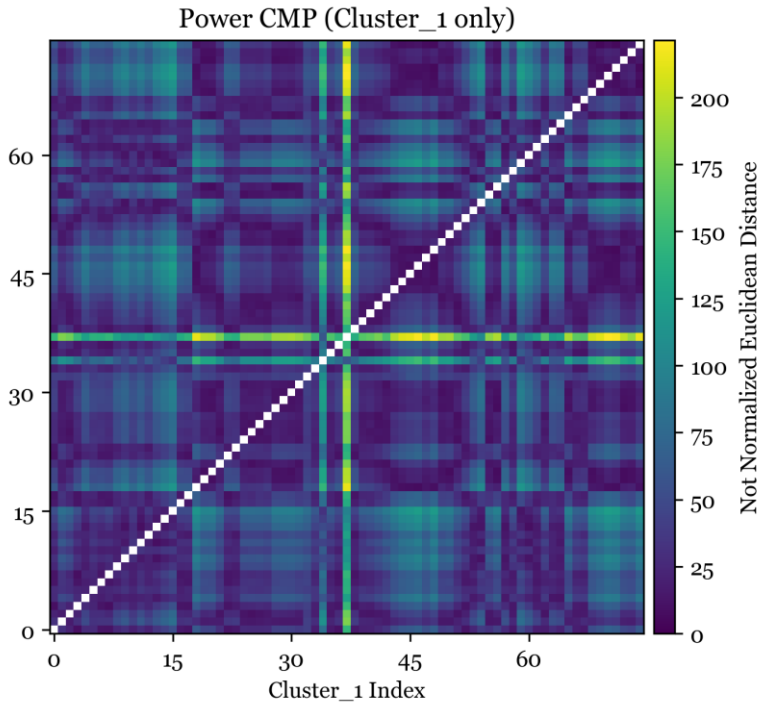

Figure 29 - The Contextual Matrix Profile for cluster number 1 + context number 1

While the nature of the items labeled as anomalous will be further discussed, with much more detail, in the next section, some considerations can be made just by looking at the list of the identified anomalies, as reported in Table 1.

First of all, the number of clusters found in **6.2.** is four and the number of contexts identified in **6.3.** is five, which leads to the computation of 20 different CMPs, one for each combination of the above mentioned "variables". However, anomalies are not found for every single examined scenario; for example, none of the abnormal instances listed in Table 1 correspond to the combination of cluster number 2 and context number 2: this is due to the fact that the two anomaly detection techniques employed have found no days that are considered abnormal for both. It can also be seen that this kind of situation has happened only for combinations involving cluster 2 (cluster 2 + context 2, cluster 2 + context 4 and cluster 2 + context 5): this is very likely explained by the small

number of profiles contained in cluster 2 (only 16), which makes it harder for the algorithm to define what is "normal" and what is "not normal" in this specific group of days; also, it is less likely to find an abnormal instance when considering only a small number of elements, due to a purely statistical reason. On the other hand, cluster 4 – which is the subset that contains the largest amount of profiles – is the one that consistently reports the highest number of anomalies, as one may have expected.
Another consideration that can be made immediately, just by looking at the list of anomalous instances, is that the majority of them corresponds to days belonging to the months of June, July or August: this is a good indicator of the fact that many anomalies will most likely be related to "events" happening during summer months; from expert knowledge of the case study analyzed, it is already possible to imagine that these "events" corresponds to an unusually high power demand from the Mechanical Room, since it is the largest season-dependent load that also increases sharply during the months with the highest cooling needs.

	Date	Context	Cluster		Date	Context	Cluster
1	2019-06-23	1	1	36	2019-06-26	3	4
2	2019-07-14	1	1	37	2019-06-27	3	4
3	2019-08-09	1	2	38	2019-07-01	3	4
4	2019-07-06	1	3	39	2019-07-09	3	4
5	2019-07-27	1	3	40	2019-07-22	3	4
6	2019-06-25	1	4	41	2019-07-23	3	4
7	2019-06-27	1	4	42	2019-07-24	3	4
8	2019-06-28	1	4	43	2019-07-25	3	4
9	2019-07-02	1	4	44	2019-07-26	3	4
10	2019-07-03	1	4	45	2019-11-10	4	1
11	2019-07-08	1	4	46	2019-07-06	4	3
12	2019-07-09	1	4	47	2019-11-09	4	3
13	2019-07-22	1	4	48	2019-06-25	4	4
14	2019-07-24	1	4	49	2019-06-26	4	4
15	2019-07-26	1	4	50	2019-07-01	4	4
16	2019-08-12	2	1	51	2019-07-08	4	4
17	2019-08-13	2	1	52	2019-07-09	4	4
18	2019-12-27	2	1	53	2019-07-22	4	4
19	2019-06-24	2	3	54	2019-07-23	4	4
20	2019-07-06	2	3	55	2019-07-24	4	4
21	2019-07-13	2	3	56	2019-07-25	4	4
22	2019-06-10	2	4	57	2019-07-26	4	4
23	2019-06-18	2	4	58	2019-06-16	5	1
24	2019-06-19	2	4	59	2019-11-10	5	1
25	2019-06-20	2	4	60	2019-07-06	5	3
26	2019-06-21	2	4	61	2019-11-09	5	3
27	2019-07-22	2	4	62	2019-06-27	5	4
28	2019-07-25	2	4	63	2019-06-28	5	4
29	2019-08-12	3	1	64	2019-07-01	5	4
30	2019-11-10	3	1	65	2019-07-02	5	4
31	2019-12-23	3	2	66	2019-07-08	5	4
32	2019-06-29	3	3	67	2019-07-22	5	4
33	2019-07-06	3	3	68	2019-07-25	5	4
34	2019-07-13	3	3	69	2019-07-26	5	4
35	2019-06-25	3	4				

Table 1 - Summary of the instances detected as anomalies

Finally, Table 2 gives an overview of how many and which contexts were labeled as abnormal in each day that presented at least one anomalous instance; the analysis of anomalies from this point of view was introduced earlier in 4.1. and 4.4. The first thing that can be noticed by looking at this summary is the fact that approximately more than half of the days listed are anomalous in more than one context, which is an indicator of periods of abnormality that, in many cases, comprise more than one type of systems' power demand behavior: when this happens, it is natural to imagine that this phenomenon is related to a power demand profile that, for most of the duration of the day, is higher than the average group day's power demand profile. Therefore, it is unlikely that the abnormality is linked to an isolated spike in power demand due, for example, to peak-load appliances being active in a short period of time; it is much more reasonable to think that, in such situations, the anomaly is related to one or more loads that are consistently higher than normal.
This kind of reasoning, that will be verified in the subsequent diagnosis phase, finds a first hint of confirmation in the fact that most of the days assigned to cluster number 4 are anomalous in more than one time window: since all the instances of this subset - which contains the regular working days - in Table 2 belong to the months of June, July or August and it was mentioned in 5.2. that the Mechanical Room load becomes negligible during Holidays and exceptionally high during summer months, it seems natural to connect these occurrences to regular working summer days, when the Mechanical Room load is higher than normal for most of the day.

Date	Cluster	Anom_Context1	Anom_Context2	Anom_Context3	Anom_Context4	Anom_Context5	Total_Contexts
2019-06-10	4	FALSE	TRUE	FALSE	FALSE	FALSE	1
2019-06-16	1	FALSE	FALSE	FALSE	FALSE	TRUE	1
2019-06-18	4	FALSE	TRUE	FALSE	FALSE	FALSE	1
2019-06-19	4	FALSE	TRUE	FALSE	FALSE	FALSE	1
2019-06-20	4	FALSE	TRUE	FALSE	FALSE	FALSE	1
2019-06-21	4	FALSE	TRUE	FALSE	FALSE	FALSE	1
2019-06-23	1	TRUE	FALSE	FALSE	FALSE	FALSE	1
2019-06-24	3	FALSE	TRUE	FALSE	FALSE	FALSE	1
2019-06-25	4	TRUE	FALSE	TRUE	TRUE	FALSE	3
2019-06-26	4	FALSE	FALSE	TRUE	TRUE	FALSE	2
2019-06-27	4	TRUE	FALSE	TRUE	FALSE	TRUE	3
2019-06-28	4	TRUE	FALSE	FALSE	FALSE	TRUE	2
2019-06-29	3	FALSE	FALSE	TRUE	FALSE	FALSE	1
2019-07-01	4	FALSE	FALSE	TRUE	TRUE	TRUE	3
2019-07-02	4	TRUE	FALSE	FALSE	FALSE	TRUE	2
2019-07-03	4	TRUE	FALSE	FALSE	FALSE	FALSE	1
2019-07-06	3	TRUE	TRUE	TRUE	TRUE	TRUE	5
2019-07-08	4	TRUE	FALSE	FALSE	TRUE	TRUE	3
2019-07-09	4	TRUE	FALSE	TRUE	TRUE	FALSE	3
2019-07-13	3	FALSE	TRUE	TRUE	FALSE	FALSE	2
2019-07-14	1	TRUE	FALSE	FALSE	FALSE	FALSE	1
2019-07-22	4	TRUE	TRUE	TRUE	TRUE	TRUE	5
2019-07-23	4	FALSE	FALSE	TRUE	TRUE	FALSE	2
2019-07-24	4	TRUE	FALSE	TRUE	TRUE	FALSE	3
2019-07-25	4	FALSE	TRUE	TRUE	TRUE	TRUE	4
2019-07-26	4	TRUE	FALSE	TRUE	TRUE	TRUE	4
2019-07-27	3	TRUE	FALSE	FALSE	FALSE	FALSE	1
2019-08-09	2	TRUE	FALSE	FALSE	FALSE	FALSE	1
2019-08-12	1	FALSE	TRUE	TRUE	FALSE	FALSE	2
2019-08-13	1	FALSE	TRUE	FALSE	FALSE	FALSE	1
2019-11-09	3	FALSE	FALSE	FALSE	TRUE	TRUE	2
2019-11-10	1	FALSE	FALSE	TRUE	TRUE	TRUE	3
2019-12-23	2	FALSE	FALSE	TRUE	FALSE	FALSE	1
2019-12-27	1	FALSE	TRUE	FALSE	FALSE	FALSE	1

Table 2 - Summary of the contexts for each anomalous day

6.5. Anomaly diagnosis at sub-loads level

The results of the diagnostic process - referred to all the groups to which the anomaly detection phase was applied - are reported, by means of graphical representations, in Appendix C. Each picture contains the diagnosis for at most two anomalous days in the examined group: in the first plot (top left) for each day, the Total power demand daily profile is drawn with a red line, while the grey lines represent the rest of the power demand profiles of the days in the cluster to which the examined day belongs; then, the subsequent 8 plots represent the sub-loads' power demand daily profiles (in blue when the Relative score is negative, in red when it is positive), together with the mean/average group daily power demand profile for that sub-load; the last plot (bottom right) provides the same kind of information for the air temperature parameter. In all the plots, the hours of the time window considered are represented by a solid line, while the remaining hours of the day are pictured with a dashed line. Moreover, "bands" corresponding to +/- %5, 15% and 25% of the value of the average group day's profile are shown with different colors (red, orange and yellow) around the mean profile itself.
The three scores introduced in **4.6.** are reported above each sub-load, while for the air temperature the third score is not calculated (since this parameter cannot be weighted following the same logic applied to the sub-loads) and it is "replaced" by a message indicating the likelihood of the influence of the external air temperature on the examined day's power demand behavior, based on the value of the Relative score; only positive Relative scores result in a message suggesting a degree of correlation between temperature and power demand, since in this case study there is a sub-load (the Mechanical Room) that is known to be related to seasonal cooling needs and no sub-loads that clearly show dependence on heating needs. A generalized version of this approach, however, should suggest external temperature influence "in both directions", whether the daily temperature is higher or lower than the average group day value of this parameter.
The analysis of the results of the diagnostic process can start from Table 3 - Table 5, which contain a summary of the sub-loads' rankings for each type of score and allow for an immediate identification of the most "dominant" loads in each case. The most recurring sub-loads in each position for each score are reported below:

Absolute: 1 - Mechanical Room; 2 - Not Labeled; 3 - Data Centre; 4 - Canteen; 5 - Print Shop; 6 - Rectory; 7 - DIMAT; 8 - Bar.
Relative: 1 - Mechanical Room; 2 - Not Labeled; 3 - Data Centre; 4 - Data Centre; 5 - Canteen/Rectory/Print Shop; 6 - Rectory; 7 - DIMAT; 8 - Bar.
Weighted Relative: 1 - Mechanical Room; 2 - Not Labeled; 3 - Data Centre; 4 - Canteen; 5 - Print Shop; 6 - Bar; 7 - Rectory; 8 - DIMAT.

The first consideration that can be made is that the most recurring sub-loads in the first four positions are the same for all three scores and their order is also (almost) always the same: this is indicative of the fact that, no matter the "point of view" considered, these sub-loads will most likely be the main culprits for the anomalous instances in this case

study. This was expected in the case of the Absolute score: the four above mentioned sub-loads are also those with higher mean and peak power demand values and therefore their fluctuations, in terms of absolute difference (expressed in kW), are naturally the most impactful among all the sub-loads; what was not obvious, however, is the fact that this phenomenon is mirrored when analyzing the sub-loads' behavior from a relative point of view, which intuitively results in the Weighted Relative score reporting an identical sub-loads' order, due to how this score is defined. This is, in some ways, reassuring for the user that wants to understand what caused a meter-level anomaly: if most of the times the three scores report the same "order of responsibility" for the sub-loads, understanding where to intervene becomes simpler and less subject to user interpretation and expert knowledge.

	Date	Context	Cluster	Sub-load 1	Sub-load 2	Sub-load 3	Sub-load 4	Sub-load 5	Sub-load 6	Sub-load 7	Sub-load 8	Temperature influence
1	2019-06-23	1	1	Not Labeled / 27.52	Data Centre / 4.12	Print Shop / 0.04	DIMAT / -0.13	Mechanical Room / -0.14	Bar / -1.72	Rectory / -2.07	Canteen / -3	Extremely likely
2	2019-07-14	1	1	Not Labeled / 43.53	Rectory / 1.28	Data Centre / 0.83	DIMAT / 0.19	Print Shop / 0.04	Mechanical Room / -0.34	Bar / -1.72	Canteen / -1.94	Extremely likely
3	2019-08-09	1	2	Mechanical Room / 13.51	Not Labeled / 6.6	Data Centre / 5.13	Rectory / 1.27	Print Shop / -0.02	DIMAT / -0.29	Bar / -0.66	Canteen / -3.67	Likely
4	2019-07-06	1	3	Not Labeled / 50.82	Mechanical Room / 33.99	Data Centre / 0.6	Canteen / 0.55	Print Shop / 0.02	Rectory / -0.38	DIMAT / -0.89	Bar / -2.04	Extremely likely
5	2019-07-27	1	3	Mechanical Room / 64.51	Data Centre / 1.54	Rectory / 1.03	Print Shop / 0.81	Canteen / 0.29	DIMAT / -0.81	Bar / -2.04	Not Labeled / -2.34	Extremely likely
6	2019-06-25	1	4	Mechanical Room / 51.52	Not Labeled / 30.74	Data Centre / 8.75	Canteen / 0.61	Print Shop / 0	Rectory / -0.12	DIMAT / -0.93	Bar / -2.33	Extremely likely
7	2019-06-27	1	4	Mechanical Room / 93.04	Not Labeled / 32.95	Data Centre / 12.4	Canteen / 0.45	Print Shop / -0.03	Rectory / -0.47	DIMAT / -0.75	Bar / -2.33	Extremely likely
8	2019-06-28	1	4	Mechanical Room / 81.46	Not Labeled / 17.03	Data Centre / 2.86	Canteen / 2.36	DIMAT / 0.36	Print Shop / -0.03	Rectory / -0.66	Bar / -2.33	Extremely likely
9	2019-07-02	1	4	Mechanical Room / 118.35	Not Labeled / 33.51	Data Centre / 1.81	DIMAT / 0.01	Print Shop / -0.01	Rectory / -0.38	Canteen / -0.7	Bar / -2.33	Extremely likely
10	2019-07-03	1	4	Mechanical Room / 72.2	Not Labeled / 50.71	Data Centre / 4.77	Print Shop / -0.01	Rectory / -0.27	Canteen / -0.38	DIMAT / -0.81	Bar / -2.33	Extremely likely
11	2019-07-08	1	4	Mechanical Room / 106.98	Not Labeled / 38.14	Data Centre / 0.78	Canteen / 0.13	Print Shop / 0	Rectory / -0.2	DIMAT / -0.99	Bar / -2.33	Extremely likely
12	2019-07-09	1	4	Not Labeled / 60.71	Mechanical Room / 30.83	Data Centre / 0.61	Canteen / 0.52	Print Shop / -0.04	Rectory / -0.17	DIMAT / -0.39	Bar / -2.33	Extremely likely
13	2019-07-22	1	4	Mechanical Room / 87.49	Not Labeled / 39.89	Data Centre / 2.51	Rectory / 2.02	Print Shop / -0.03	DIMAT / -0.05	Canteen / -0.9	Bar / -2.33	Extremely likely
14	2019-07-24	1	4	Mechanical Room / 66.77	Not Labeled / 10.83	Data Centre / 3.51	Rectory / 1.48	Print Shop / -0.03	Canteen / -0.04	DIMAT / -0.65	Bar / -2.33	Extremely likely
15	2019-07-26	1	4	Mechanical Room / 72.38	Canteen / 12.19	Not Labeled / 7.69	Data Centre / 3.63	Rectory / 1.31	Print Shop / 0.77	DIMAT / -0.08	Bar / -2.33	Extremely likely
16	2019-08-12	2	1	Mechanical Room / 63.97	Not Labeled / 19.72	Data Centre / 7.13	Canteen / 5.63	Rectory / 1.77	Print Shop / 0.38	DIMAT / -1.22	Bar / -1.76	Extremely likely
17	2019-08-13	2	1	Mechanical Room / 43.17	Canteen / 5.82	Data Centre / 2.17	Rectory / 0.57	Print Shop / -0.13	DIMAT / -1.22	Bar / -1.76	Not Labeled / -12.87	Extremely likely
18	2019-12-27	2	1	Not Labeled / 30.97	Mechanical Room / 26.51	Canteen / 5.34	Bar / 2.13	Rectory / 1.69	Print Shop / 0.59	DIMAT / -0.34	Data Centre / -3.03	Very unlikely
19	2019-06-24	2	3	Mechanical Room / 69.57	Not Labeled / 26.44	Canteen / 10.22	Data Centre / 5.62	Print Shop / 0.12	DIMAT / -0.75	Rectory / -1.93	Bar / -2.06	Extremely likely
20	2019-07-06	2	3	Mechanical Room / 82.58	Not Labeled / 48.84	Data Centre / 0.77	Canteen / 0.22	Print Shop / 0.12	Rectory / -0.22	DIMAT / -0.94	Bar / -2.09	Extremely likely
21	2019-07-13	2	3	Mechanical Room / 73.49	Not Labeled / 30.14	Canteen / 3.5	Rectory / 0.45	Print Shop / 0.06	DIMAT / -0.25	Data Centre / -0.67	Bar / -2.09	Extremely likely
22	2019-06-10	2	4	Mechanical Room / 48.36	Data Centre / 3.23	Print Shop / 0.07	DIMAT / -1.02	Rectory / -1.17	Canteen / -1.7	Not Labeled / -4.28	Bar / -4.82	Extremely likely
23	2019-06-18	2	4	Mechanical Room / 39.4	Not Labeled / 17.05	Data Centre / 4.64	Canteen / 1.4	Print Shop / 0.07	DIMAT / -0.32	Rectory / -1.15	Bar / -4.79	Extremely likely
24	2019-06-19	2	4	Mechanical Room / 41.18	Not Labeled / 15.64	Data Centre / 4.16	Print Shop / 0.44	DIMAT / -0.75	Rectory / -1.25	Bar / -4.82	Canteen / -8.15	Extremely likely
25	2019-06-20	2	4	Mechanical Room / 44.09	Not Labeled / 16.49	Data Centre / 4.27	Print Shop / 0.65	DIMAT / -0.14	Rectory / -1.25	Canteen / -2.04	Bar / -4.82	Extremely likely
26	2019-06-21	2	4	Mechanical Room / 36.6	Not Labeled / 8.94	Data Centre / 3.49	Print Shop / -0.07	DIMAT / -0.35	Rectory / -1.07	Bar / -4.82	Canteen / -6.82	Extremely likely
27	2019-07-22	2	4	Mechanical Room / 105.24	Not Labeled / 48.73	Data Centre / 2.43	Rectory / 0.21	Print Shop / 0.15	DIMAT / -0.48	Bar / -4.82	Canteen / -4.87	Extremely likely
28	2019-07-25	2	4	Mechanical Room / 118.09	Not Labeled / 36.54	Canteen / 5.61	Data Centre / 3.52	Print Shop / 1.03	Rectory / 0	DIMAT / -0.62	Bar / -4.82	Extremely likely
29	2019-08-12	3	1	Mechanical Room / 53.45	Canteen / 10.21	Data Centre / 7.95	Rectory / 2.4	Print Shop / -0.06	DIMAT / -1.21	Bar / -1.97	Not Labeled / -3.25	Very likely
30	2019-11-10	3	1	Not Labeled / 71.1	Bar / 12.8	Mechanical Room / 1.3	Rectory / 1.07	Canteen / 0.34	Data Centre / 0.33	DIMAT / 0.11	Print Shop / -0.1	Very unlikely
31	2019-12-23	3	2	Not Labeled / 58.42	Canteen / 45.22	Bar / 7.36	Rectory / 2.86	DIMAT / 0.59	Print Shop / -0.17	Data Centre / -6.94	Mechanical Room / -33.27	Very unlikely
32	2019-06-29	3	3	Mechanical Room / 94.03	Not Labeled / 3.61	Data Centre / 1.76	Print Shop / 0.12	Canteen / -0.31	Rectory / -0.48	DIMAT / -0.84	Bar / -2.37	Very likely
33	2019-07-06	3	3	Mechanical Room / 111.61	Not Labeled / 23.35	Rectory / 0.47	Canteen / 0.24	Print Shop / 0.1	Data Centre / -0.07	DIMAT / -0.81	Bar / -2.37	Very likely
34	2019-07-13	3	3	Mechanical Room / 67.43	Not Labeled / 49.22	Data Centre / 3.14	Canteen / 2.28	Rectory / 0.12	Print Shop / 0.11	DIMAT / -0.26	Bar / -2.37	Very likely
35	2019-06-25	3	4	Mechanical Room / 126.51	Not Labeled / 17.65	Data Centre / 13.98	Canteen / 2.76	Print Shop / 1.31	Rectory / -0.21	DIMAT / -1.9	Bar / -7.79	Extremely likely
36	2019-06-26	3	4	Mechanical Room / 151.12	Not Labeled / 16.45	Data Centre / 15.94	Print Shop / 0.77	DIMAT / -1.42	Rectory / -2.46	Canteen / -5.73	Bar / -7.79	Extremely likely
37	2019-06-27	3	4	Mechanical Room / 169.44	Data Centre / 6.65	Print Shop / 0.91	DIMAT / -1.25	Rectory / -1.33	Canteen / -2.17	Bar / -7.79	Not Labeled / -12.04	Extremely likely
38	2019-07-01	3	4	Mechanical Room / 190.38	Data Centre / 2.5	Print Shop / 0.87	DIMAT / -0.94	Rectory / -2.21	Canteen / -5.42	Bar / -7.79	Not Labeled / -28.52	Extremely likely
39	2019-07-09	3	4	Mechanical Room / 135.42	Not Labeled / 36.88	Print Shop / 0.36	DIMAT / -0.14	Data Centre / -0.5	Rectory / -0.63	Canteen / -0.69	Bar / -7.79	Very likely
40	2019-07-22	3	4	Mechanical Room / 140.03	Not Labeled / 15.28	Data Centre / 2.88	Canteen / 1.18	Print Shop / 0.57	Rectory / -0.98	DIMAT / -1.97	Bar / -7.79	Extremely likely
41	2019-07-23	3	4	Mechanical Room / 151.83	Not Labeled / 17.47	Data Centre / 3.28	Rectory / 0.58	Print Shop / 0.35	Canteen / 0.27	DIMAT / -1.62	Bar / -7.79	Extremely likely
42	2019-07-24	3	4	Mechanical Room / 155.16	Data Centre / 3.03	Not Labeled / 2.32	Canteen / 2.01	Print Shop / -0.01	Rectory / -1.32	DIMAT / -1.34	Bar / -7.79	Extremely likely
43	2019-07-25	3	4	Mechanical Room / 162.84	Not Labeled / 27.21	Data Centre / 3.08	Canteen / 2.24	Print Shop / -0.74	Rectory / -1.61	DIMAT / -1.87	Bar / -7.79	Extremely likely
44	2019-07-26	3	4	Mechanical Room / 157.87	Not Labeled / 12.5	Data Centre / 2.77	Print Shop / -0.96	Rectory / -1.58	DIMAT / -2.07	Canteen / -4.33	Bar / -7.79	Extremely likely
45	2019-11-10	4	1	Not Labeled / 117.85	Mechanical Room / 17.41	Bar / 11.26	Rectory / 0.86	Data Centre / 0.25	DIMAT / 0.23	Print Shop / -0.01	Canteen / -5.11	Very unlikely
46	2019-07-06	4	3	Mechanical Room / 94.1	Canteen / 2.56	Print Shop / -0.01	Rectory / -0.13	Data Centre / -0.44	DIMAT / -0.46	Bar / -2.2	Not Labeled / -2.7	Extremely likely
47	2019-11-09	4	3	Not Labeled / 93.61	Bar / 8.54	Rectory / 1.15	DIMAT / 0.05	Print Shop / -0.04	Data Centre / -0.3	Canteen / -0.68	Mechanical Room / -3.31	Very unlikely
48	2019-06-25	4	4	Mechanical Room / 146.81	Not Labeled / 32.75	Data Centre / 16.07	Canteen / 4.75	Print Shop / 1.26	Rectory / -1.36	DIMAT / -2.21	Bar / -4.74	Extremely likely
49	2019-06-26	4	4	Mechanical Room / 174.98	Data Centre / 17.17	Canteen / 2.41	Print Shop / 0.61	DIMAT / -2.21	Rectory / -2.27	Bar / -4.74	Not Labeled / -27.5	Extremely likely
50	2019-07-01	4	4	Mechanical Room / 208.47	Canteen / 4.48	Data Centre / 1.53	Print Shop / 0.31	Rectory / -0.25	DIMAT / -0.67	Bar / -4.74	Not Labeled / -31.75	Extremely likely
51	2019-07-08	4	4	Mechanical Room / 129.8	Not Labeled / 18.33	Canteen / 4.31	Rectory / -0.12	Print Shop / -0.13	Data Centre / -0.24	DIMAT / -2.29	Bar / -4.74	Extremely likely
52	2019-07-09	4	4	Mechanical Room / 128.96	Not Labeled / 32.12	Canteen / 2.39	Rectory / 0.98	Print Shop / 0.12	DIMAT / 0.07	Data Centre / -0.6	Bar / -4.74	Very likely
53	2019-07-22	4	4	Mechanical Room / 140.79	Not Labeled / 13.99	Data Centre / 3.02	Canteen / 2.54	Rectory / 0.91	Print Shop / -0.22	DIMAT / -2.52	Bar / -4.74	Extremely likely
54	2019-07-23	4	4	Mechanical Room / 151.97	Not Labeled / 15.42	Canteen / 3.42	Data Centre / 2.81	Rectory / 1.38	Print Shop / -0.24	DIMAT / -2.25	Bar / -4.74	Extremely likely
55	2019-07-24	4	4	Mechanical Room / 153.46	Canteen / 8.22	Data Centre / 2.45	Not Labeled / 1.13	Rectory / 0.05	Print Shop / -0.68	DIMAT / -2.21	Bar / -4.74	Extremely likely
56	2019-07-25	4	4	Mechanical Room / 165.21	Not Labeled / 29.17	Canteen / 26.14	Data Centre / 3.32	Rectory / 0.28	Print Shop / -1.35	DIMAT / -2.4	Bar / -4.74	Extremely likely
57	2019-07-26	4	4	Mechanical Room / 144.56	Canteen / 27.61	Data Centre / 2.12	Rectory / -0.18	Print Shop / -0.89	DIMAT / -2.65	Bar / -4.74	Not Labeled / -8.18	Extremely likely
58	2019-06-16	5	1	Not Labeled / 53.38	Data Centre / 4.43	Print Shop / 0.03	DIMAT / -0.02	Rectory / -1.27	Mechanical Room / -1.44	Bar / -1.78	Canteen / -2.41	Extremely likely
59	2019-11-10	5	1	Not Labeled / 78.24	Mechanical Room / 15.84	Bar / 4.58	Rectory / 0.73	Data Centre / 0.48	DIMAT / 0.14	Print Shop / 0	Canteen / -4	Very unlikely
60	2019-07-06	5	3	Mechanical Room / 61.49	Canteen / 2.55	Not Labeled / 0.68	Print Shop / 0.01	Data Centre / -0.12	DIMAT / -0.42	Rectory / -0.57	Bar / -2.05	Extremely likely
61	2019-11-09	5	3	Not Labeled / 65.46	Bar / 4.19	Rectory / 0.55	Print Shop / 0	DIMAT / -0.07	Data Centre / -0.27	Canteen / -0.43	Mechanical Room / -1.66	Very unlikely
62	2019-06-27	5	4	Mechanical Room / 133.16	Not Labeled / 8.23	Canteen / 3.94	Data Centre / 3.16	Print Shop / 0.07	DIMAT / -0.19	Rectory / -0.25	Bar / -2.46	Extremely likely
63	2019-06-28	5	4	Mechanical Room / 126.28	Not Labeled / 5.33	Data Centre / 2.71	Canteen / 2.27	Print Shop / 0.03	Rectory / -1.05	DIMAT / -1.7	Bar / -2.46	Extremely likely
64	2019-07-01	5	4	Mechanical Room / 141.83	Data Centre / 1.31	Canteen / 0.82	DIMAT / 0.33	Rectory / -0.03	Print Shop / -0.13	Not Labeled / -0.95	Bar / -2.46	Extremely likely
65	2019-07-02	5	4	Mechanical Room / 101.61	Not Labeled / 23.98	Data Centre / 6.99	Print Shop / 0.08	Canteen / -0.06	Rectory / -0.07	DIMAT / -1	Bar / -2.46	Extremely likely
66	2019-07-08	5	4	Mechanical Room / 74.61	Not Labeled / 41.43	Canteen / 1.7	Data Centre / 0.29	Print Shop / -0.03	Rectory / -0.38	DIMAT / -1.34	Bar / -2.46	Extremely likely
67	2019-07-22	5	4	Mechanical Room / 78.43	Not Labeled / 25.83	Data Centre / 3.32	Canteen / 0.97	Rectory / 0.7	Print Shop / -0.05	DIMAT / -1.44	Bar / -2.46	Extremely likely
68	2019-07-25	5	4	Canteen / 59.42	Mechanical Room / 56.26	Not Labeled / 23.86	Data Centre / 3.14	Rectory / 1.63	Print Shop / 0.25	DIMAT / -0.69	Bar / -2.46	Extremely likely
69	2019-07-26	5	4	Canteen / 55.7	Mechanical Room / 42.56	Data Centre / 1.89	Not Labeled / 0.91	Rectory / 0.48	Print Shop / 0.43	DIMAT / -1.44	Bar / -2.46	Extremely likely

Table 3 – Summary of the results for Absolute scores

	Date	Context	Cluster	Sub-load 1	Sub-load 2	Sub-load 3	Sub-load 4	Sub-load 5	Sub-load 6	Sub-load 7	Sub-load 8	Temperature influence
1	2019-06-23	1	1	Not Labeled / 50.9	Print Shop / 11.83	Data Centre / 11.76	Mechanical Room / 2.51	DIMAT / -4.12	Rectory / -19.7	Canteen / -20.37	Bar / -100	Extremely likely
2	2019-07-14	1	1	Not Labeled / 80.74	Rectory / 11.99	Print Shop / 11.51	DIMAT / 5.8	Data Centre / 2.33	Mechanical Room / -2.63	Canteen / -12.99	Bar / -100	Extremely likely
3	2019-08-09	1	2	Mechanical Room / 96.31	Data Centre / 14.32	Not Labeled / 12.2	Rectory / 11.05	Print Shop / -7.94	DIMAT / -9.28	Canteen / -28.36	Bar / -100	Likely
4	2019-07-06	1	3	Mechanical Room / 481.42	Not Labeled / 78.79	Print Shop / 6.84	Canteen / 4.51	Data Centre / 1.72	Rectory / -3.77	DIMAT / -22.98	Bar / -100	Extremely likely
5	2019-07-27	1	3	Mechanical Room / 941.85	Print Shop / 192.02	Rectory / 9.92	Data Centre / 4.37	Canteen / 2.93	Not Labeled / -3.86	DIMAT / -20.96	Bar / -100	Extremely likely
6	2019-06-25	1	4	Mechanical Room / 501.43	Not Labeled / 52.13	Data Centre / 24.94	Print Shop / 1.06	Canteen / 0.54	Rectory / -1.2	DIMAT / -23.94	Bar / -100	Extremely likely
7	2019-06-27	1	4	Mechanical Room / 1000.22	Not Labeled / 56.51	Data Centre / 35.32	Canteen / 0.1	Rectory / -4.59	Print Shop / -6.16	DIMAT / -19.19	Bar / -100	Extremely likely
8	2019-06-28	1	4	Mechanical Room / 872.97	Not Labeled / 28.97	Canteen / 14.65	DIMAT / 9.33	Data Centre / 8.16	Print Shop / -6.15	Rectory / -6.27	Bar / -100	Extremely likely
9	2019-07-02	1	4	Mechanical Room / 1341.79	Not Labeled / 57.32	Data Centre / 5.16	DIMAT / 0.21	Print Shop / -2.55	Rectory / -3.65	Canteen / -5.03	Bar / -100	Extremely likely
10	2019-07-03	1	4	Mechanical Room / 818.19	Not Labeled / 86.48	Data Centre / 13.55	Canteen / -1.25	Rectory / -2.64	Print Shop / -2.85	DIMAT / -20.78	Bar / -100	Extremely likely
11	2019-07-08	1	4	Mechanical Room / 1215.29	Not Labeled / 64.72	Data Centre / 2.23	Canteen / 0.63	Print Shop / 0.31	Rectory / -2.03	DIMAT / -25.53	Bar / -100	Extremely likely
12	2019-07-09	1	4	Mechanical Room / 246.82	Not Labeled / 102.66	Canteen / 2.33	Data Centre / 1.74	Rectory / -1.64	Print Shop / -9.68	DIMAT / -10.08	Bar / -100	Extremely likely
13	2019-07-22	1	4	Mechanical Room / 890.42	Not Labeled / 64.74	Rectory / 18.78	Data Centre / 7.14	DIMAT / -1.37	Canteen / -4.5	Print Shop / -6.65	Bar / -100	Extremely likely
14	2019-07-24	1	4	Mechanical Room / 650.4	Not Labeled / 17.26	Rectory / 13.65	Data Centre / 9.99	Canteen / 0.54	Print Shop / -6.12	DIMAT / -16.82	Bar / -100	Extremely likely
15	2019-07-26	1	4	Mechanical Room / 704.54	Print Shop / 170.51	Canteen / 98.34	Rectory / 12.14	Not Labeled / 11.71	Data Centre / 10.34	DIMAT / -2.17	Bar / -100	Extremely likely
16	2019-08-12	2	1	Mechanical Room / 829.37	Print Shop / 75.86	Not Labeled / 31.59	Canteen / 30.3	Data Centre / 20.6	Rectory / 17.54	DIMAT / -37.96	Bar / -100	Extremely likely
17	2019-08-13	2	1	Mechanical Room / 503.26	Canteen / 31.16	Data Centre / 6.28	Rectory / 5.66	Not Labeled / -20.61	Print Shop / -28.9	DIMAT / -37.96	Bar / -100	Extremely likely
18	2019-12-27	2	1	Mechanical Room / 294.59	Print Shop / 124.1	Bar / 120.95	Not Labeled / 50.05	Canteen / 31.19	Rectory / 16.54	Data Centre / -8.73	DIMAT / -10.67	Very unlikely
19	2019-06-24	2	3	Mechanical Room / 168.62	Canteen / 75.54	Not Labeled / 28.35	Data Centre / 16.05	Print Shop / 3.43	Rectory / -18.64	DIMAT / -19.57	Bar / -98.86	Extremely likely
20	2019-07-06	2	3	Mechanical Room / 188.46	Not Labeled / 47.44	Print Shop / 9.15	Data Centre / 2.2	Canteen / 1.79	Rectory / -2.12	DIMAT / -24.43	Bar / -100	Extremely likely
21	2019-07-13	2	3	Mechanical Room / 258.75	Not Labeled / 30.12	Canteen / 27.39	Rectory / 4.28	Print Shop / 1.66	Data Centre / -1.92	DIMAT / -6.38	Bar / -100	Extremely likely
22	2019-06-10	2	4	Mechanical Room / 114.4	Data Centre / 9.25	Not Labeled / 0.04	Canteen / -3.04	Print Shop / -9.77	Rectory / -10.11	DIMAT / -24.25	Bar / -100	Extremely likely
23	2019-06-18	2	4	Mechanical Room / 40.94	Not Labeled / 13.74	Data Centre / 13.31	Print Shop / 3.31	Canteen / 1.14	DIMAT / -7.22	Rectory / -10	Bar / -99.24	Extremely likely
24	2019-06-19	2	4	Mechanical Room / 43.59	Data Centre / 11.92	Not Labeled / 10.27	Print Shop / 0.53	Canteen / -7.86	Rectory / -11.01	DIMAT / -17.62	Bar / -100	Extremely likely
25	2019-06-20	2	4	Mechanical Room / 48.25	Not Labeled / 12.97	Data Centre / 12.23	Print Shop / 4.92	Canteen / 3.55	DIMAT / -2.73	Rectory / -10.76	Bar / -100	Extremely likely
26	2019-06-21	2	4	Mechanical Room / 36.66	Data Centre / 10.02	Not Labeled / 7.21	Canteen / -4.92	Print Shop / -6.15	DIMAT / -7.82	Rectory / -8.31	Bar / -100	Extremely likely
27	2019-07-22	2	4	Mechanical Room / 287.92	Not Labeled / 49.4	Data Centre / 6.96	Rectory / 2.76	Print Shop / -3.16	Canteen / -6.28	DIMAT / -10.64	Bar / -100	Extremely likely
28	2019-07-25	2	4	Mechanical Room / 295.34	Print Shop / 176.18	Not Labeled / 26.87	Data Centre / 10.1	Canteen / 2.71	Rectory / 0.47	DIMAT / -14.59	Bar / -100	Extremely likely
29	2019-08-12	3	1	Mechanical Room / 579.83	Canteen / 55.56	Rectory / 24.86	Data Centre / 22.24	Not Labeled / -6.16	Print Shop / -12.31	DIMAT / -37.73	Bar / -100	Very likely
30	2019-11-10	3	1	Bar / 635.09	Not Labeled / 130.51	Mechanical Room / 15.42	Rectory / 10.98	DIMAT / 3.51	Canteen / 1.89	Data Centre / 0.92	Print Shop / -19.85	Very unlikely
31	2019-12-23	3	2	Bar / 464.71	Canteen / 69.18	Not Labeled / 39.38	Rectory / 20.65	DIMAT / 15.89	Print Shop / -8.42	Data Centre / -18.67	Mechanical Room / -42.44	Very unlikely
32	2019-06-29	3	3	Mechanical Room / 166.6	Print Shop / 6	Data Centre / 4.86	Not Labeled / 1.33	Canteen / -1.48	Rectory / -4.5	DIMAT / -22.29	Bar / -100	Very likely
33	2019-07-06	3	3	Mechanical Room / 342.63	Not Labeled / 14.51	Rectory / 4.39	Canteen / 2.15	Print Shop / 2.13	Data Centre / -0.16	DIMAT / -21.6	Bar / -100	Very likely
34	2019-07-13	3	3	Mechanical Room / 123.4	Not Labeled / 37.67	Canteen / 17.76	Data Centre / 8.52	Print Shop / 2.87	Rectory / 1.26	DIMAT / -6.98	Bar / -100	Very likely
35	2019-06-25	3	4	Mechanical Room / 173.85	Data Centre / 38.28	Print Shop / 13.37	Not Labeled / 6.6	Canteen / 1.79	Rectory / -1.66	DIMAT / -31.13	Bar / -100	Extremely likely
36	2019-06-26	3	4	Mechanical Room / 208.13	Data Centre / 43.7	Print Shop / 8.45	Not Labeled / 6.98	Canteen / -3.68	Rectory / -14.37	DIMAT / -23.25	Bar / -100	Extremely likely
37	2019-06-27	3	4	Mechanical Room / 233.72	Data Centre / 18.57	Print Shop / 10.5	Canteen / -1.09	Not Labeled / -4.68	Rectory / -7.64	DIMAT / -20.61	Bar / -100	Extremely likely
38	2019-07-01	3	4	Mechanical Room / 260.24	Print Shop / 9.81	Data Centre / 6.9	Canteen / -3.97	Not Labeled / -10.97	Rectory / -12.74	DIMAT / -15.55	Bar / -100	Extremely likely
39	2019-07-09	3	4	Mechanical Room / 187.51	Not Labeled / 14.68	Print Shop / 3.53	Canteen / -0.53	Data Centre / -1.36	DIMAT / -2.61	Rectory / -3.88	Bar / -100	Very likely
40	2019-07-22	3	4	Mechanical Room / 193.32	Data Centre / 7.91	Not Labeled / 6.1	Print Shop / 5.88	Canteen / 0.71	Rectory / -5.68	DIMAT / -32.27	Bar / -100	Extremely likely
41	2019-07-23	3	4	Mechanical Room / 209.58	Data Centre / 9.01	Not Labeled / 7.13	Print Shop / 3.63	Rectory / 3.13	Canteen / 0.32	DIMAT / -26.54	Bar / -100	Extremely likely
42	2019-07-24	3	4	Mechanical Room / 214.72	Data Centre / 8.35	Canteen / 1.93	Not Labeled / 1.32	Print Shop / -0.64	Rectory / -7.42	DIMAT / -21.59	Bar / -100	Extremely likely
43	2019-07-25	3	4	Mechanical Room / 225.44	Not Labeled / 11	Data Centre / 8.47	Canteen / 1.82	Print Shop / -7.87	Rectory / -9.25	DIMAT / -30.22	Bar / -100	Extremely likely
44	2019-07-26	3	4	Mechanical Room / 218.32	Data Centre / 7.63	Not Labeled / 5.25	Canteen / -2.79	Rectory / -8.95	Print Shop / -9.98	DIMAT / -33.94	Bar / -100	Extremely likely
45	2019-11-10	4	1	Bar / 576.92	Mechanical Room / 218.89	Not Labeled / 214.61	Rectory / 8.96	DIMAT / 7.12	Data Centre / 0.7	Print Shop / -2.08	Canteen / -31	Very unlikely
46	2019-07-06	4	3	Mechanical Room / 1648.7	Canteen / 20.2	Data Centre / -1.2	Rectory / -1.34	Print Shop / -3.7	Not Labeled / -3.77	DIMAT / -12.76	Bar / -100	Extremely likely
47	2019-11-09	4	3	Bar / 385.85	Not Labeled / 129.36	Rectory / 11.76	DIMAT / 1.38	Data Centre / -0.8	Canteen / -5.1	Print Shop / -10.34	Mechanical Room / -47.87	Very unlikely
48	2019-06-25	4	4	Mechanical Room / 196.18	Data Centre / 43.52	Print Shop / 12.99	Not Labeled / 12.24	Canteen / 12.18	Rectory / -8.37	DIMAT / -34.15	Bar / -100	Extremely likely
49	2019-06-26	4	4	Mechanical Room / 233.88	Data Centre / 46.48	Print Shop / 6.58	Canteen / 5.19	Not Labeled / -10.16	Rectory / -13.81	DIMAT / -34.11	Bar / -100	Extremely likely
50	2019-07-01	4	4	Mechanical Room / 278.47	Canteen / 17.08	Data Centre / 4.12	Print Shop / 2.78	Rectory / -1.37	DIMAT / -9.69	Not Labeled / -11.66	Bar / -100	Extremely likely
51	2019-07-08	4	4	Mechanical Room / 173.34	Canteen / 14.11	Not Labeled / 6.89	Data Centre / -0.65	Rectory / -1.06	Print Shop / -1.84	DIMAT / -35.35	Bar / -100	Extremely likely
52	2019-07-09	4	4	Mechanical Room / 172.24	Not Labeled / 11.93	Canteen / 6.38	Rectory / 5.79	DIMAT / 0.98	Print Shop / 0.41	Data Centre / -1.62	Bar / -100	Very likely
53	2019-07-22	4	4	Mechanical Room / 187.7	Canteen / 11.15	Data Centre / 8.18	Not Labeled / 5.22	Rectory / 5.15	Print Shop / -2.27	DIMAT / -38.96	Bar / -100	Extremely likely
54	2019-07-23	4	4	Mechanical Room / 202.58	Canteen / 10.31	Rectory / 8.46	Data Centre / 7.61	Not Labeled / 5.79	Print Shop / -3.73	DIMAT / -34.71	Bar / -100	Extremely likely
55	2019-07-24	4	4	Mechanical Room / 204.79	Canteen / 21.61	Data Centre / 6.64	Rectory / 0.55	Not Labeled / 0.48	Print Shop / -8.53	DIMAT / -34.09	Bar / -100	Extremely likely
56	2019-07-25	4	4	Mechanical Room / 220.74	Canteen / 130.89	Not Labeled / 10.9	Data Centre / 8.98	Rectory / 2.43	Print Shop / -16.9	DIMAT / -36.81	Bar / -100	Extremely likely
57	2019-07-26	4	4	Mechanical Room / 192.17	Canteen / 142.23	Data Centre / 5.73	Rectory / -0.8	Not Labeled / -3.19	Print Shop / -8.92	DIMAT / -40.83	Bar / -100	Extremely likely
58	2019-06-16	5	1	Not Labeled / 86.49	Data Centre / 12.39	Print Shop / 8.34	DIMAT / -0.77	Rectory / -12.35	Canteen / -16.23	Mechanical Room / -23.49	Bar / -100	Extremely likely
59	2019-11-10	5	1	Mechanical Room / 356.59	Bar / 253.55	Not Labeled / 126.54	Rectory / 7.19	DIMAT / 4.62	Data Centre / 1.35	Print Shop / -1.26	Canteen / -26.88	Very unlikely
60	2019-07-06	5	3	Mechanical Room / 1339.61	Canteen / 21.1	Print Shop / 4.37	Not Labeled / 1.15	Data Centre / -0.31	Rectory / -5.55	DIMAT / -12.27	Bar / -100	Extremely likely
61	2019-11-09	5	3	Bar / 202.66	Not Labeled / 89.23	Rectory / 5.41	Data Centre / -0.72	Print Shop / -1.43	DIMAT / -2.04	Canteen / -3.58	Mechanical Room / -37.02	Very unlikely
62	2019-06-27	5	4	Mechanical Room / 1132.88	Canteen / 25.62	Not Labeled / 15.41	Data Centre / 8.74	Rectory / -1.71	DIMAT / -2.22	Print Shop / -5.35	Bar / -100	Extremely likely
63	2019-06-28	5	4	Mechanical Room / 1045.46	Not Labeled / 17.39	Canteen / 13.18	Data Centre / 7.5	Print Shop / -5.19	Rectory / -8.28	DIMAT / -35.01	Bar / -100	Extremely likely
64	2019-07-01	5	4	Mechanical Room / 1192.45	Not Labeled / 6.94	DIMAT / 4.59	Canteen / 4.53	Data Centre / 3.63	Rectory / -0.26	Print Shop / -6.65	Bar / -100	Extremely likely
65	2019-07-02	5	4	Mechanical Room / 928.14	Not Labeled / 33.31	Data Centre / 19.35	Canteen / 0.24	Rectory / -0.49	Print Shop / -3.54	DIMAT / -21.82	Bar / -100	Extremely likely
66	2019-07-08	5	4	Mechanical Room / 518.39	Not Labeled / 48.8	Canteen / 11.15	Data Centre / 0.81	Rectory / -3.02	Print Shop / -5.27	DIMAT / -26.72	Bar / -100	Extremely likely
67	2019-07-22	5	4	Mechanical Room / 492.06	Not Labeled / 22.71	Data Centre / 9.21	Rectory / 6.76	Canteen / 6.29	Print Shop / -6.21	DIMAT / -29.17	Bar / -100	Extremely likely
68	2019-07-25	5	4	Canteen / 377.61	Mechanical Room / 152.54	Print Shop / 93.63	Not Labeled / 19.19	Rectory / 13.53	Data Centre / 8.71	DIMAT / -12.79	Bar / -100	Extremely likely
69	2019-07-26	5	4	Canteen / 343.57	Mechanical Room / 121.08	Print Shop / 101.83	Data Centre / 5.23	Rectory / 4.46	Not Labeled / 3.94	DIMAT / -29.23	Bar / -100	Extremely likely

Table 4 – Summary of the results for Relative scores

	Date	Context	Cluster	Sub-load 1	Sub-load 2	Sub-load 3	Sub-load 4	Sub-load 5	Sub-load 6	Sub-load 7	Sub-load 8	Temperature influence
1	2019-06-23	1	1	Not Labeled / 28.12	Data Centre / 3.11	Print Shop / 0.09	Mechanical Room / 0.06	Bar / 0	DIMAT / -0.08	Rectory / -1.1	Canteen / -1.52	Extremely likely
2	2019-07-14	1	1	Not Labeled / 47.73	Rectory / 0.88	Data Centre / 0.54	DIMAT / 0.13	Print Shop / 0.08	Bar / 0	Mechanical Room / -0.05	Canteen / -0.95	Extremely likely
3	2019-08-09	1	2	Mechanical Room / 27.12	Not Labeled / 4.83	Data Centre / 4.4	Rectory / 1.04	Print Shop / 0.13	Bar / 0	DIMAT / -0.19	Canteen / -1.82	Likely
4	2019-07-06	1	3	Mechanical Room / 175.59	Not Labeled / 42.68	Data Centre / 0.31	Canteen / 0.3	Print Shop / 0.04	Bar / 0	Rectory / -0.19	DIMAT / -0.32	Extremely likely
5	2019-07-27	1	3	Mechanical Room / 393.46	Print Shop / 1.79	Data Centre / 0.83	Rectory / 0.55	Canteen / 0.26	Bar / 0	DIMAT / -0.33	Not Labeled / -1.2	Extremely likely
6	2019-06-25	1	4	Mechanical Room / 193.92	Not Labeled / 21.61	Data Centre / 5.15	Canteen / 0.39	Print Shop / 0.03	Bar / 0	Rectory / -0.04	DIMAT / -0.32	Extremely likely
7	2019-06-27	1	4	Mechanical Room / 375.94	Not Labeled / 19.43	Data Centre / 6.23	Canteen / 0.18	Print Shop / 0.02	Bar / 0	Rectory / -0.15	DIMAT / -0.22	Extremely likely
8	2019-06-28	1	4	Mechanical Room / 334.22	Not Labeled / 9.41	Data Centre / 1.3	Canteen / 1.2	DIMAT / 0.17	Print Shop / 0.03	Bar / 0	Rectory / -0.25	Extremely likely
9	2019-07-02	1	4	Mechanical Room / 612.98	Not Labeled / 18.55	Data Centre / 0.66	Print Shop / 0.02	DIMAT / 0.01	Bar / 0	Rectory / -0.12	Canteen / -0.2	Extremely likely
10	2019-07-03	1	4	Mechanical Room / 262.95	Not Labeled / 36.61	Data Centre / 2.1	Print Shop / 0.01	Bar / 0	Canteen / -0.06	Rectory / -0.1	DIMAT / -0.24	Extremely likely
11	2019-07-08	1	4	Mechanical Room / 525.85	Not Labeled / 22.71	Data Centre / 0.31	Canteen / 0.08	Print Shop / 0.02	Bar / 0	Rectory / -0.05	DIMAT / -0.26	Extremely likely
12	2019-07-09	1	4	Mechanical Room / 80.15	Not Labeled / 57.32	Data Centre / 0.32	Canteen / 0.26	Print Shop / 0.02	Bar / 0	Rectory / -0.1	DIMAT / -0.16	Extremely likely
13	2019-07-22	1	4	Mechanical Room / 367.72	Not Labeled / 25.49	Data Centre / 1.07	Rectory / 1.05	Print Shop / 0.02	Bar / 0	DIMAT / -0.01	Canteen / -0.18	Extremely likely
14	2019-07-24	1	4	Mechanical Room / 276.5	Not Labeled / 5.65	Data Centre / 1.88	Rectory / 0.92	Canteen / 0.1	Print Shop / 0.02	Bar / 0	DIMAT / -0.26	Extremely likely
15	2019-07-26	1	4	Mechanical Room / 297.62	Canteen / 12.22	Not Labeled / 3.76	Data Centre / 1.79	Print Shop / 1.31	Rectory / 0.68	Bar / 0	DIMAT / -0.03	Extremely likely
16	2019-08-12	2	1	Mechanical Room / 251.89	Not Labeled / 11.48	Data Centre / 3.8	Canteen / 3.8	Rectory / 0.93	Print Shop / 0.81	Bar / 0	DIMAT / -0.33	Extremely likely
17	2019-08-13	2	1	Mechanical Room / 174.37	Canteen / 5.5	Data Centre / 1.35	Rectory / 0.35	Print Shop / 0.02	Bar / 0	DIMAT / -0.46	Not Labeled / -5.86	Extremely likely
18	2019-12-27	2	1	Mechanical Room / 58.84	Not Labeled / 29.75	Canteen / 4.43	Bar / 2.55	Rectory / 1.26	Print Shop / 0.87	DIMAT / -0.17	Data Centre / -1.56	Very unlikely
19	2019-06-24	2	3	Mechanical Room / 78.26	Not Labeled / 13.19	Canteen / 5.75	Data Centre / 2.07	Print Shop / 0.05	Bar / -0.01	DIMAT / -0.2	Rectory / -0.51	Extremely likely
20	2019-07-06	2	3	Mechanical Room / 64.23	Not Labeled / 22.77	Data Centre / 0.28	Canteen / 0.07	Print Shop / 0.04	Bar / 0	Rectory / -0.08	DIMAT / -0.21	Extremely likely
21	2019-07-13	2	3	Mechanical Room / 96.29	Not Labeled / 12.97	Canteen / 1.71	Rectory / 0.17	Print Shop / 0.04	Bar / 0	DIMAT / -0.07	Data Centre / -0.21	Extremely likely
22	2019-06-10	2	4	Mechanical Room / 51.41	Data Centre / 1.16	Not Labeled / 0.44	Print Shop / 0.03	Bar / 0	DIMAT / -0.26	Rectory / -0.33	Canteen / -0.39	Extremely likely
23	2019-06-18	2	4	Mechanical Room / 23.45	Not Labeled / 6.06	Data Centre / 1.81	Canteen / 0.84	Print Shop / 0.03	Bar / -0.01	DIMAT / -0.07	Rectory / -0.37	Extremely likely
24	2019-06-19	2	4	Mechanical Room / 25.21	Not Labeled / 4.1	Data Centre / 1.61	Print Shop / 0.1	Bar / 0	DIMAT / -0.23	Rectory / -0.42	Canteen / -1.24	Extremely likely
25	2019-06-20	2	4	Mechanical Room / 26.62	Not Labeled / 5.52	Data Centre / 1.58	Canteen / 1.26	Print Shop / 0.18	Bar / 0	DIMAT / 0	Rectory / -0.41	Extremely likely
26	2019-06-21	2	4	Mechanical Room / 22.61	Not Labeled / 3.01	Data Centre / 1.4	Print Shop / 0.04	Bar / 0	DIMAT / -0.07	Rectory / -0.31	Canteen / -1.02	Extremely likely
27	2019-07-22	2	4	Mechanical Room / 94.53	Not Labeled / 21.01	Data Centre / 0.56	Rectory / 0.11	Print Shop / 0.04	Bar / 0	DIMAT / -0.07	Canteen / -0.76	Extremely likely
28	2019-07-25	2	4	Mechanical Room / 107.15	Not Labeled / 9.97	Print Shop / 1.09	Data Centre / 0.91	Canteen / 0.9	Rectory / 0.08	Bar / 0	DIMAT / -0.13	Extremely likely
29	2019-08-12	3	1	Mechanical Room / 182.7	Canteen / 7.92	Data Centre / 4.89	Rectory / 1.55	Print Shop / 0.22	Bar / 0	DIMAT / -0.38	Not Labeled / -1.08	Very likely
30	2019-11-10	3	1	Not Labeled / 75.33	Bar / 47.29	Mechanical Room / 4.82	Canteen / 0.6	Rectory / 0.56	Data Centre / 0.17	DIMAT / 0.06	Print Shop / -0.01	Very unlikely
31	2019-12-23	3	2	Canteen / 19.53	Not Labeled / 19.5	Bar / 10.24	Rectory / 0.83	DIMAT / 0.17	Print Shop / -0.03	Data Centre / -1.36	Mechanical Room / -4.54	Very unlikely
32	2019-06-29	3	3	Mechanical Room / 68.73	Not Labeled / 0.66	Data Centre / 0.55	Canteen / 0.06	Print Shop / 0.05	Bar / 0	Rectory / -0.12	DIMAT / -0.2	Very likely
33	2019-07-06	3	3	Mechanical Room / 141.87	Not Labeled / 6.41	Canteen / 0.15	Rectory / 0.12	Print Shop / 0.03	Bar / 0	Data Centre / -0.04	DIMAT / -0.16	Very likely
34	2019-07-13	3	3	Mechanical Room / 38.78	Not Labeled / 20.12	Canteen / 1.13	Data Centre / 1.1	Rectory / 0.05	Print Shop / 0.03	Bar / 0	DIMAT / -0.06	Very likely
35	2019-06-25	3	4	Mechanical Room / 49.7	Data Centre / 2.75	Not Labeled / 2.69	Canteen / 0.53	Print Shop / 0.22	Bar / 0	Rectory / -0.03	DIMAT / -0.18	Extremely likely
36	2019-06-26	3	4	Mechanical Room / 65.21	Data Centre / 3.2	Not Labeled / 2.67	Print Shop / 0.12	Bar / 0	DIMAT / -0.14	Rectory / -0.29	Canteen / -0.58	Extremely likely
37	2019-06-27	3	4	Mechanical Room / 80.66	Data Centre / 1.51	Print Shop / 0.16	Bar / 0	DIMAT / -0.14	Rectory / -0.17	Canteen / -0.19	Not Labeled / -1.61	Extremely likely
38	2019-07-01	3	4	Mechanical Room / 100.01	Data Centre / 0.4	Print Shop / 0.15	Bar / 0	DIMAT / -0.11	Rectory / -0.26	Canteen / -0.74	Not Labeled / -3.57	Extremely likely
39	2019-07-09	3	4	Mechanical Room / 54.76	Not Labeled / 5.99	Print Shop / 0.07	Bar / 0	Canteen / -0.01	DIMAT / -0.01	Rectory / -0.06	Data Centre / -0.07	Very likely
40	2019-07-22	3	4	Mechanical Room / 58.84	Not Labeled / 2.37	Data Centre / 0.45	Canteen / 0.29	Print Shop / 0.11	Bar / 0	Rectory / -0.11	DIMAT / -0.18	Extremely likely
41	2019-07-23	3	4	Mechanical Room / 65.89	Not Labeled / 2.7	Data Centre / 0.51	Canteen / 0.2	Rectory / 0.09	Print Shop / 0.06	Bar / 0	DIMAT / -0.16	Extremely likely
42	2019-07-24	3	4	Mechanical Room / 69.72	Data Centre / 0.48	Not Labeled / 0.44	Canteen / 0.41	Print Shop / 0.01	Bar / 0	DIMAT / -0.14	Rectory / -0.17	Extremely likely
43	2019-07-25	3	4	Mechanical Room / 72.43	Not Labeled / 4.17	Canteen / 0.52	Data Centre / 0.47	Bar / 0	Print Shop / -0.08	DIMAT / -0.17	Rectory / -0.2	Extremely likely
44	2019-07-26	3	4	Mechanical Room / 71.35	Not Labeled / 1.98	Data Centre / 0.43	Bar / 0	Print Shop / -0.11	DIMAT / -0.19	Rectory / -0.19	Canteen / -0.38	Extremely likely
45	2019-11-10	4	1	Not Labeled / 135.65	Bar / 30.62	Mechanical Room / 20.34	Rectory / 0.35	Data Centre / 0.1	DIMAT / 0.09	Print Shop / 0.03	Canteen / -1.26	Very unlikely
46	2019-07-06	4	3	Mechanical Room / 710.27	Canteen / 1.34	Print Shop / 0.03	Bar / 0	Rectory / -0.05	DIMAT / -0.16	Data Centre / -0.18	Not Labeled / -1.04	Extremely likely
47	2019-11-09	4	3	Not Labeled / 88.25	Bar / 17.88	Rectory / 0.54	DIMAT / 0.03	Print Shop / 0.02	Data Centre / -0.11	Canteen / -0.25	Mechanical Room / -0.7	Very unlikely
48	2019-06-25	4	4	Mechanical Room / 66.15	Not Labeled / 5.63	Data Centre / 3.52	Canteen / 0.88	Print Shop / 0.23	Bar / 0	Rectory / -0.19	DIMAT / -0.22	Extremely likely
49	2019-06-26	4	4	Mechanical Room / 94.08	Data Centre / 4.05	Canteen / 0.56	Print Shop / 0.11	Bar / 0	DIMAT / -0.23	Rectory / -0.31	Not Labeled / -3.94	Extremely likely
50	2019-07-01	4	4	Mechanical Room / 122.72	Canteen / 0.97	Data Centre / 0.24	Print Shop / 0.06	Bar / 0	DIMAT / -0.02	Rectory / -0.02	Not Labeled / -4.27	Extremely likely
51	2019-07-08	4	4	Mechanical Room / 58.15	Not Labeled / 3.26	Canteen / 1.03	Bar / 0	Print Shop / -0.02	Rectory / -0.02	Data Centre / -0.04	DIMAT / -0.24	Extremely likely
52	2019-07-09	4	4	Mechanical Room / 56.32	Not Labeled / 5.76	Canteen / 0.51	Rectory / 0.18	Print Shop / 0.03	DIMAT / 0.02	Bar / 0	Data Centre / -0.08	Very likely
53	2019-07-22	4	4	Mechanical Room / 65.32	Not Labeled / 2.4	Canteen / 0.65	Data Centre / 0.54	Rectory / 0.16	Bar / 0	Print Shop / -0.02	DIMAT / -0.25	Extremely likely
54	2019-07-23	4	4	Mechanical Room / 72.48	Not Labeled / 2.63	Canteen / 0.7	Data Centre / 0.48	Rectory / 0.25	Bar / 0	Print Shop / -0.03	DIMAT / -0.23	Extremely likely
55	2019-07-24	4	4	Mechanical Room / 75.06	Canteen / 1.61	Data Centre / 0.43	Not Labeled / 0.2	Rectory / 0.02	Bar / 0	Print Shop / -0.09	DIMAT / -0.23	Extremely likely
56	2019-07-25	4	4	Mechanical Room / 77.67	Canteen / 16.57	Not Labeled / 4.78	Data Centre / 0.54	Rectory / 0.07	Bar / 0	Print Shop / -0.16	DIMAT / -0.22	Extremely likely
57	2019-07-26	4	4	Mechanical Room / 67.78	Canteen / 21.5	Data Centre / 0.36	Bar / 0	Rectory / 0	Print Shop / -0.12	DIMAT / -0.25	Not Labeled / -1.32	Extremely likely
58	2019-06-16	5	1	Not Labeled / 54.41	Data Centre / 2.73	Print Shop / 0.02	Bar / 0	DIMAT / -0.01	Mechanical Room / -0.43	Rectory / -0.6	Canteen / -1.07	Extremely likely
59	2019-11-10	5	1	Not Labeled / 79.34	Mechanical Room / 35.66	Bar / 7.58	Rectory / 0.36	Data Centre / 0.24	DIMAT / 0.08	Print Shop / 0.06	Canteen / -1.35	Very unlikely
60	2019-07-06	5	3	Mechanical Room / 479.28	Canteen / 1.59	Not Labeled / 0.9	Print Shop / 0.06	Bar / 0	Data Centre / -0.03	DIMAT / -0.19	Rectory / -0.28	Extremely likely
61	2019-11-09	5	3	Not Labeled / 60.75	Bar / 6.24	Rectory / 0.26	Print Shop / 0.05	DIMAT / -0.04	Data Centre / -0.1	Canteen / -0.22	Mechanical Room / -0.46	Very unlikely
62	2019-06-27	5	4	Mechanical Room / 489.08	Not Labeled / 4.92	Canteen / 1.4	Data Centre / 0.97	Print Shop / 0.03	Bar / 0	DIMAT / 0	Rectory / -0.04	Extremely likely
63	2019-06-28	5	4	Mechanical Room / 432.36	Not Labeled / 6.05	Data Centre / 0.85	Canteen / 0.76	Print Shop / 0.01	Bar / 0	Rectory / -0.24	DIMAT / -0.3	Extremely likely
64	2019-07-01	5	4	Mechanical Room / 551.36	Not Labeled / 1.94	Data Centre / 0.46	Canteen / 0.2	DIMAT / 0.07	Bar / 0	Rectory / 0	Print Shop / 0	Extremely likely
65	2019-07-02	5	4	Mechanical Room / 338.03	Not Labeled / 13.21	Data Centre / 2.39	Canteen / 0.06	Print Shop / 0.03	Bar / 0	Rectory / -0.01	DIMAT / -0.22	Extremely likely
66	2019-07-08	5	4	Mechanical Room / 130.65	Not Labeled / 24.94	Canteen / 0.67	Data Centre / 0.13	Print Shop / 0.02	Bar / 0	Rectory / -0.09	DIMAT / -0.25	Extremely likely
67	2019-07-22	5	4	Mechanical Room / 152.25	Not Labeled / 10.55	Data Centre / 1.35	Rectory / 0.5	Canteen / 0.38	Print Shop / 0.01	Bar / 0	DIMAT / -0.32	Extremely likely
68	2019-07-25	5	4	Canteen / 91.35	Mechanical Room / 43.32	Not Labeled / 7.79	Data Centre / 1.28	Rectory / 0.59	Print Shop / 0.47	Bar / 0	DIMAT / -0.13	Extremely likely
69	2019-07-26	5	4	Canteen / 90.53	Mechanical Room / 32.62	Not Labeled / 1.29	Data Centre / 0.68	Print Shop / 0.5	Rectory / 0.33	Bar / 0	DIMAT / -0.35	Extremely likely

Table 5 – Summary of the results for Weighted Relative scores

Looking at the individual sub-loads in Table 3 - Table 7, it can be seen that the one that is consistently found in the first position - no matter the kind of score considered - is the Mechanical Room, as mentioned above; as anticipated in the previous section, this is very likely due to the nature of the days that are present in the list of the anomalous instances, since they mostly represent full working day belonging to summer months, as well as to the large power demand share of this sub-load with respect to the Total when the cooling needs are at their highest during the year. The next sub-load appearing more frequently in the top spot for all three scores, but mainly for the Absolute and Weighted Relative ones, is the Not Labeled load; once again, this sub-load is usually the highest (when there is no demand for cooling) or the second highest (during summer months) in terms of power demand, therefore its influence on the anomaly at a meter-level is often very important: this can be seen in Table 3 - Table 7, that show the impact of this load especially when the anomalous instances belong to non-summer months.

A peculiar aspect, which emerges from the analysis of Table 5 - Table 7, is that the Bar sub-load is the second most frequent in the first position for Relative scores: this is certainly due to the fact that, in all clusters, most of the days belong to the period when this facility was closed, which results in the average group day' power demand profile being almost flat and constantly close to 0 kW. Therefore, an anomalous day in which the bar was regularly opened is automatically labeled as an unexpected occurrence, especially in relative terms (since the Bar power demand is generally small and this results in a lower ranking of this sub-load from the absolute and weighted relative points of view). Also, the Bar appears as the most frequent sub-load in the 8th position both for the Absolute and the Relative scores but it is never ranked 8th for the Weighted Relative score; this can be explained by the fact that this last score multiplies the Relative score by the weight of the sub-load with respect to the Total power demand: if the considered instance belongs to a day when the Bar was closed, the weight of the sub-load is null for the whole day and therefore the Weighted Relative score also becomes equal to zero; the fact that zero is never the lowest Weighted Relative score indicates that, in all of the examined instances, at least one sub-load (other than the Bar), has a negative Relative score. This can be confirmed by looking at the column named "Sub-load 7" in Table 4.
With regard to the air temperature, the diagnostic process suggests a strong possibility of the influence of this parameter on the daily power demand behavior in almost all of the days listed as anomalous, except for the few ones that belong to the months of November or December, as shown in Table 3 - Table 5: this was expected, given the previous considerations about the temporal location of most of the anomalous instances. This is also reflected by the fact that during the above mentioned non-summer abnormal occurrences, the Mechanical Room appears only once (on 2019-12-27, anomaly number 18 following the numbering in Table 1) as the top-1 load in terms of Weighted Relative score, once again confirming the strong dependence of this sub-load on cooling needs due to exceptionally high external air temperature.

	Sub-load	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8
1	Data Centre	0	9	38	8	5	5	3	1
2	Canteen	2	9	12	19	4	6	8	9
3	Mechanical Room	55	7	1	0	1	2	0	3
4	DIMAT	0	0	0	10	8	18	33	0
5	Bar	0	3	3	1	0	1	15	46
6	Rectory	0	1	4	16	19	23	6	0
7	Print Shop	0	0	7	13	32	14	2	1
8	Not Labeled	12	40	4	2	0	0	2	9

Table 6 - Summary of the number of times each sub-load appeared in a specific position in the Absolute score ranking

	Sub-load	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8
1	Data Centre	0	14	20	17	8	6	4	0
2	Canteen	2	14	9	16	15	6	5	2
3	Mechanical Room	59	3	1	1	0	1	1	3
4	DIMAT	0	0	1	5	7	8	47	1
5	Bar	5	1	1	0	0	0	0	62
6	Rectory	0	1	8	14	15	26	5	0
7	Print Shop	0	8	13	7	15	19	6	1
8	Not Labeled	3	28	16	9	9	3	1	0

Table 8 - Summary of the number of times each sub-load appeared in a specific position in the Relative score ranking

	Sub-load	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8
1	Data Centre	0	9	34	12	2	4	4	4
2	Canteen	3	9	13	20	5	1	4	14
3	Mechanical Room	58	3	2	1	0	1	1	3
4	DIMAT	0	0	0	2	8	13	17	29
5	Bar	0	4	2	9	10	33	11	0
6	Rectory	0	1	2	14	12	6	26	8
7	Print Shop	0	1	7	10	32	11	6	2
8	Not Labeled	8	42	9	1	0	0	0	9

Table 7 - Summary of the number of times each sub-load appeared in a specific position in the Weighted Relative score ranking

The observations made so far are quite general and they are aimed at understanding, when possible, the main trends and characteristics in the anomalous instances considered as a whole, in order to perform the analysis of the single occurrences with more awareness with respect to those that present differences from the most common patterns in terms of sub-loads ranking.

Therefore, moving on to the analysis of the single anomalous occurrences and given what has been discovered so far, it makes the most sense to initially focus on those few instances that do not belong to summer months: in particular, the day 2019-11-09 that is abnormal in the fourth and fifth context (anomalies number 47 and 61), the day 2019-11-10 which is anomalous during the third, fourth and fifth context (anomalies number 30, 45 and 59), the day 2019-12-23 that is anomalous during the third context (anomaly number 31) and the day 2019-12-27 which shows abnormality during the second context (anomaly number 18). Looking at Figure 30 - right and Figure 31 - right, it can be seen that the most dominant sub-loads during both context 4 and context 5 for the day 2019-11-09 are the Bar and the Not Labeled; the same can be said for the following day 2019-11-10, when these two sub-loads are still dominant in terms of being the most anomalous ones, with the addition of the Mechanical Room in this case (Figure 32 – right, Figure 33, Figure 34 - right). These two days are quite unique, with respect the all the other days in the analyzed dataset, in terms of general trend of the daily Total power demand profile

as well. The fact that the Bar and the Not Labeled sub-loads are constantly abnormal in these instances, also considering that these two days are respectively a Saturday and a Sunday, suggest an anomalous amount of people present in the campus during this two-days window, since the power demand of the Bar is usually related to human activities and needs. In fact, this hypothesis can be considered realistic and these anomalous occurrences are almost certainly due to the fact that the analyzed weekend is the one when Politecnico di Torino's "Festival della Tecnologia", which is a biennial event where technology-related exhibitions and conferences take place, was held in 2019. Figure 35 illustrates the sub-loads' behavior during the third time window on 2019-12-23; the three most abnormal sub-loads are the Not Labeled, the Canteen and the Bar, while all the other sub-loads, except for the DIMAT and the Rectory which are slightly higher than usual, behave normally. Once again, the fact that the 3 most anomalous sub-loads are those that are usually related with human presence is an indicator that during this day, which is the first one of Christmas Holidays for students and belongs to the cluster where days of semi-regular functioning are grouped, the university campus was still active, perhaps occupied by staff that was working on the last days before the actual closing of all the university facilities for the Christmas period. The same, however, cannot be said for the day 2019-12-27, represented in Figure 36, which, based on the calendar for the year, was a day of full closing for the university campus: in this case, it can be seen that the most anomalous sub-loads are the Mechanical Room and the Not Labeled ones, whose sharp growth is almost identical during the time window considered, while generally all the other sub-loads are slightly higher than their corresponding average group days' power demand profiles. It is not easy to explain why this kind of behavior took place, especially since the day considered does not seem to show anything unusual in terms of external air temperature, however the fact that the Not Labeled load's profile is basically mirrored by that of the Mechanical Room can point to a possible unexpected activation of certain HVAC systems/appliances, some of which may also have belonged to the Not Labeled load.

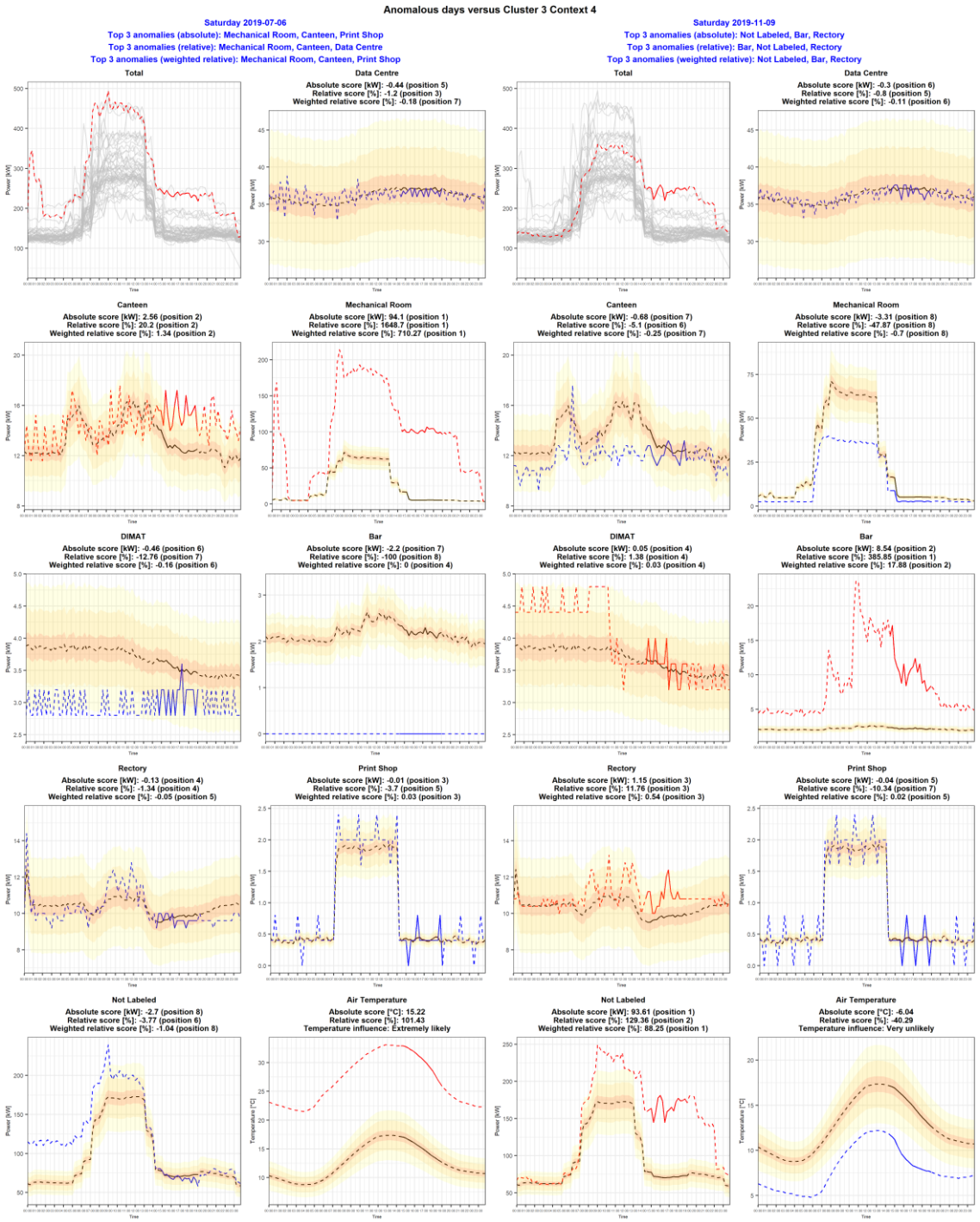

Figure 30 - Anomaly diagnosis for cluster number 3 + context number 4

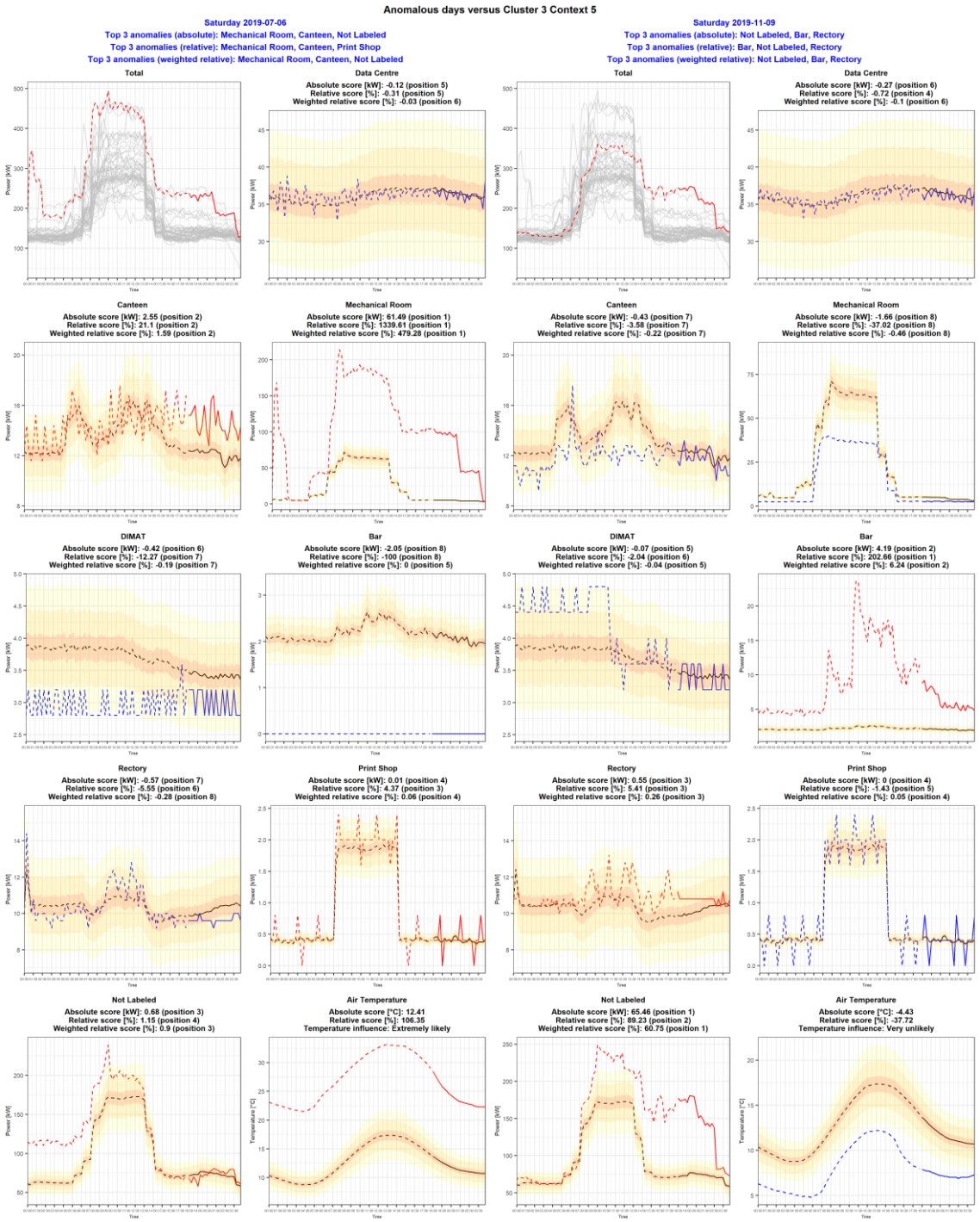

Figure 31 - Anomaly diagnosis for cluster number 3 + context number 5

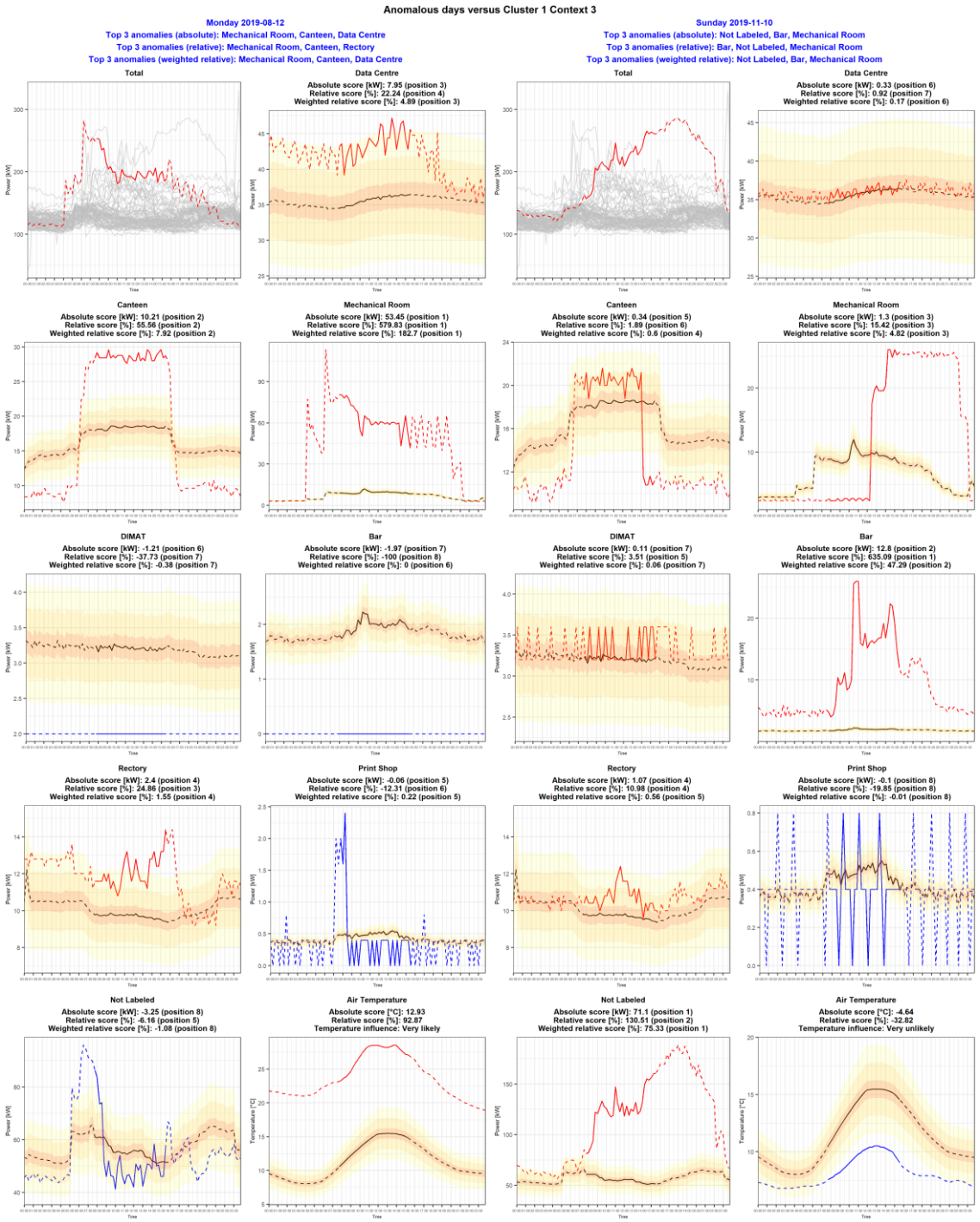

Figure 32 - Anomaly diagnosis for cluster number 1 + context number 3

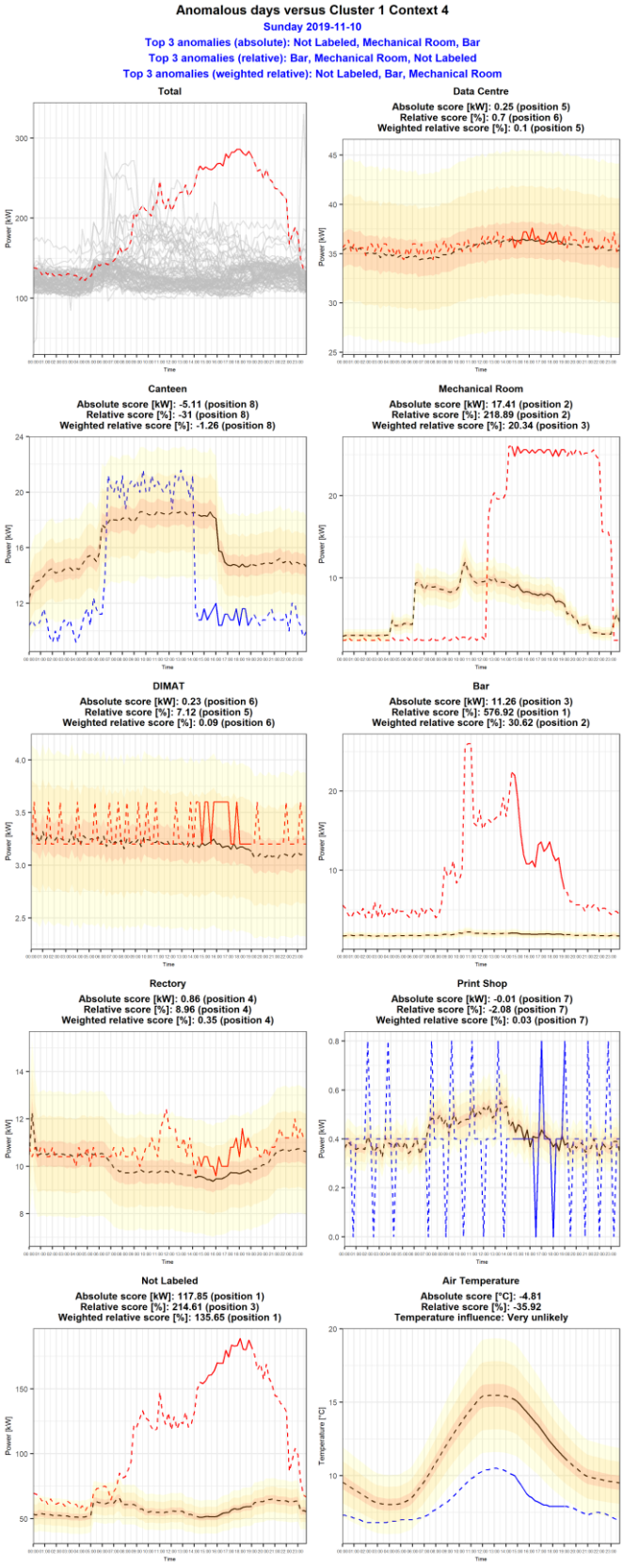

Figure 33 - Anomaly diagnosis for cluster number 1 + context number 4

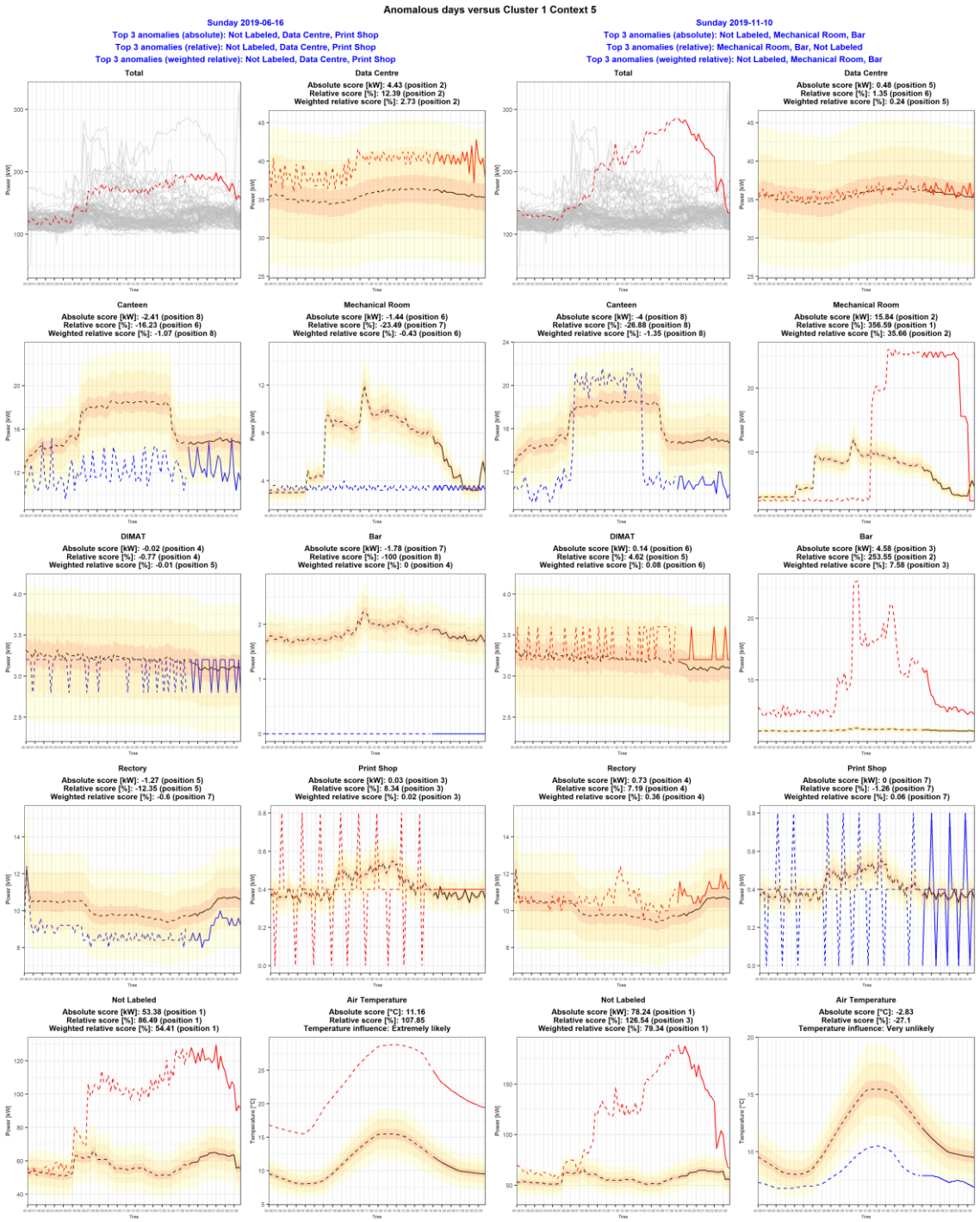

Figure 34 - Anomaly diagnosis for cluster number 1 + context number 5

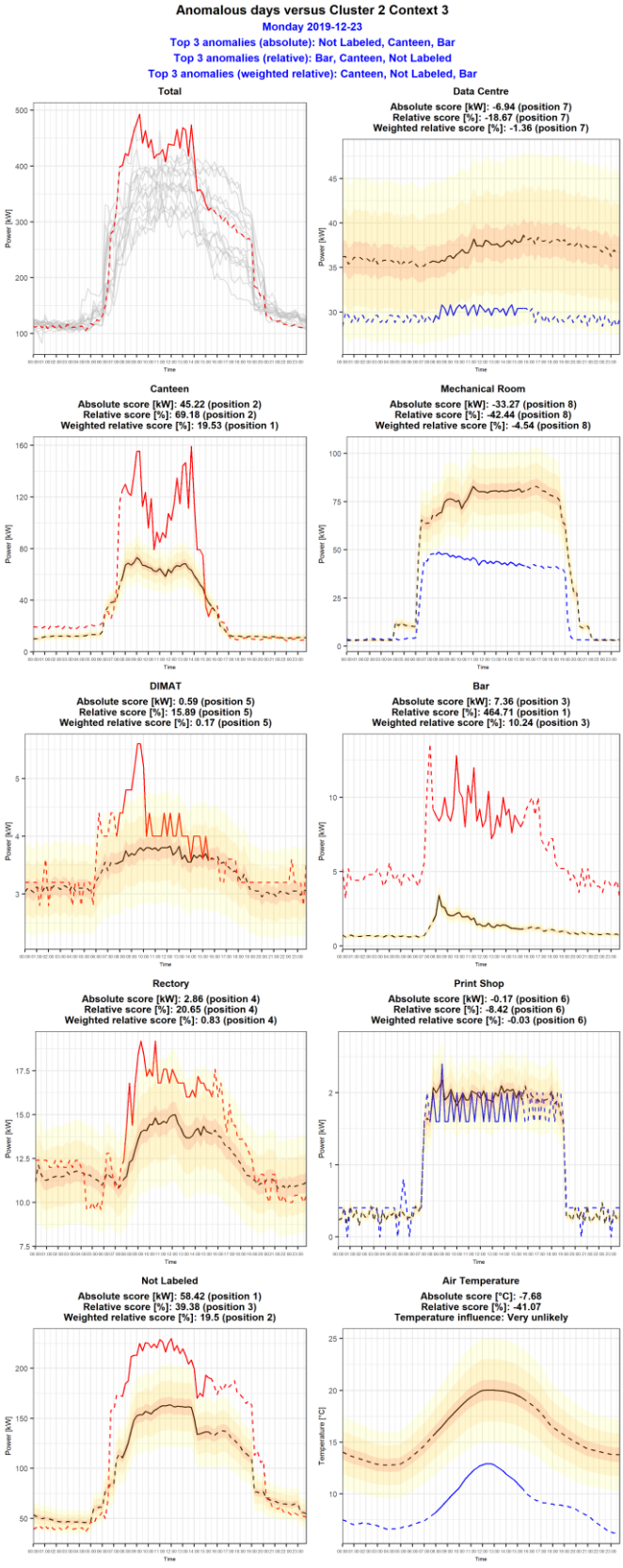

Figure 35 - Anomaly diagnosis for cluster number 2 + context number 3

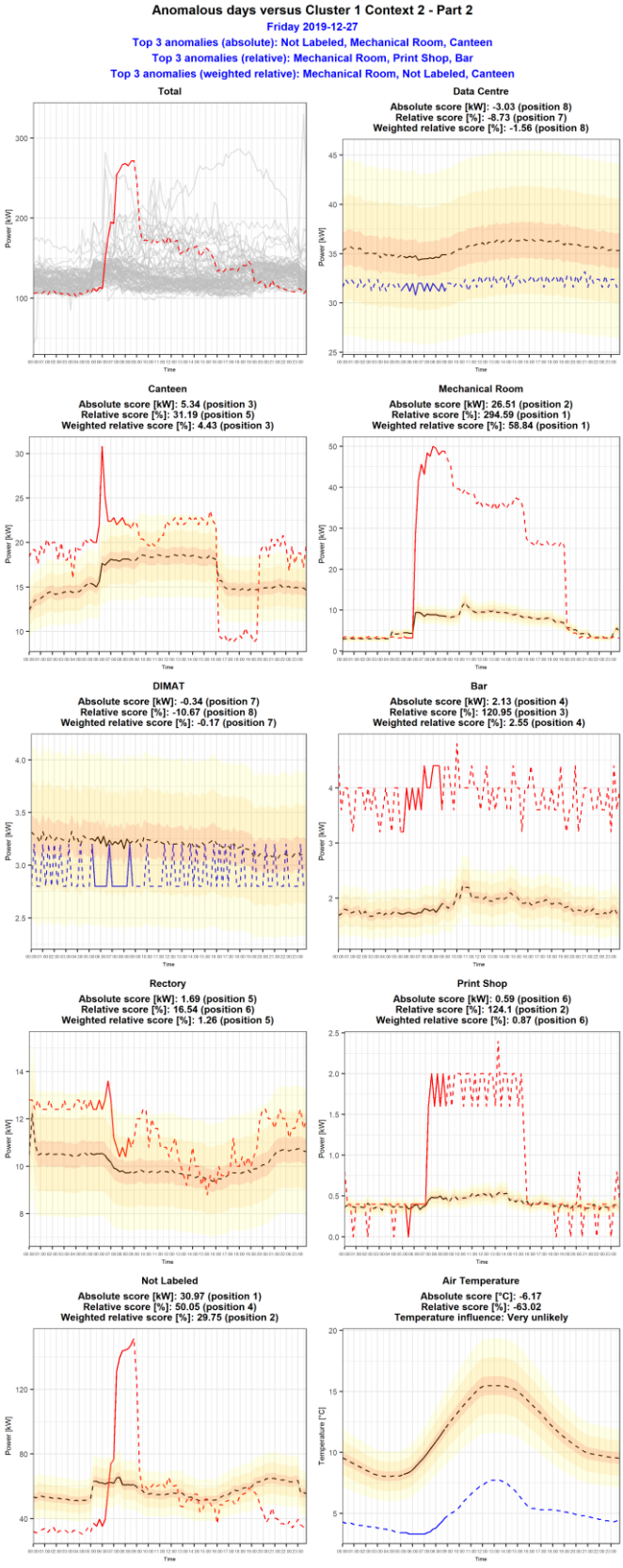

Figure 36 - Anomaly diagnosis for cluster number 1 + context number 2, part 2

Next, it is worth examining those occurrences that belong to summer months, but in which the Mechanical Room is not the most abnormal sub-load: this kind of behavior can be seen on 2019-06-23 and 2019-07-14 during the first context (anomalies number 1 and 2) and on 2019-06-16, 2019-07-25 and 2019-07-26 during the fifth context (anomalies number 58, 68 and 69); the first three instances belong to the cluster of days of full closing of the university campus, while the last two are located in the cluster of regular working days. Looking at Figure 37, it can be seen that the two abnormal days show an overall behavior that is almost identical: the profile is flat and matches the shape of all other profiles in the cluster and the instances are labeled as anomalous due to the magnitude of the power demand, which is consistently higher than normal by about 25-50 kW during the time window considered (and also slightly higher than most of the other profiles in the cluster during the following contexts, especially in the case of anomaly number 2). When analyzing the sub-loads, it is clear for both occurrences that the anomaly is mainly related to the power demand of the Not Labeled sub-load: the remaining sub-loads show a behavior that is identical or very close to that of the average group day's profile (with maybe the only exception being the Data Centre which is slightly higher than normal on 2019-06-23; however, this is still negligible when compared to the contribution to the anomaly that can be attributed to the Not Labeled load). As for anomaly number 18, in the occasions when the Not Labeled load is the most responsible for an unexpected behavior, it is hard to explain why that specific anomaly took place, since the abnormal contribution can be due to a large number of systems/appliances that are not individually monitored. Moreover, in this case, unlike anomaly number 18, there is no clear discrepancy with respect to normal behavior in any of the remaining sub-loads, which makes it even harder to formulate an hypothesis regarding the possible culprit; a reasonable guess would be the possible activation of cooling systems that do not belong to the Mechanical Room, given the high external air temperature. Identical considerations can be made for anomaly number 58: Figure 34 - left highlights how the Not Labeled sub-load is clearly the most anomalous, with the Data Centre being slightly higher than normal and the remaining sub-loads showing no signs of abnormality. This situation perfectly "mirrors" (in terms of temporal location during the day) that of anomaly number 1, although it can be seen that, in this case, the Total power demand profile was coherent with the majority of the profiles found in the cluster especially during the first time window: as the day went on, the "degree of abnormality" increased. The fact that the Not Labeled load is around 50 kW higher than normal is, once again, not explainable with the available data; however, a guess similar to that of the two previously analyzed instances can be made. Finally, anomalies number 68 and number 69 can be examined, in Figure 38; they belong to two consecutive regular working days in which the sub-loads' behavior is almost identical and the most evident abnormality is related to the Canteen load, which is regular during the day until approximately 18:00, when a sudden rise to power demand values of almost 150 kW happens, with the highest peak around 20:00 – 21:00. Looking at the other sub-loads, no clear correlation of this abnormality with any of them can be found: the Mechanical Room sub-load's power demand profile is also very different from its average group day's profile, but the reasons for this have already been explained and this behavior is

coherent with that of the other summer days in cluster number 4; the remaining sub-loads show no abnormalities, except maybe for the Not Labeled load on 2019-07-25, which is slightly higher than normal, and for the Print Shop load during both days, which seems to stay active for longer than usual (until around 21:00): however, these differences with respect to the average profiles are minor if compared to the very "strange" behavior of the Canteen power demand profile. It is not easy to explain why these anomalies occurred, especially since there is no mention on the calendar to any "special event" taking place at the university campus on these two days. Given the intended use of the room, it is not unreasonable to think that some sort of dinner event may have been held during the two evenings, perhaps to celebrate the end of the academic year and the upcoming summer Holidays period, however this is purely a guess that is impossible to confirm without further information in this regard. Nevertheless, the diagnostic process was able to highlight these exceptionally unusual occurrences which, at a meter-level, do not result in a Total power demand profile that is particularly different from all the others labeled as anomalous during the summer months: although it can be seen that the two days are characterized by a high Total power demand especially during the last hours, this very striking difference at a sub-load level is not equally "eye-catching" at a meter-level.

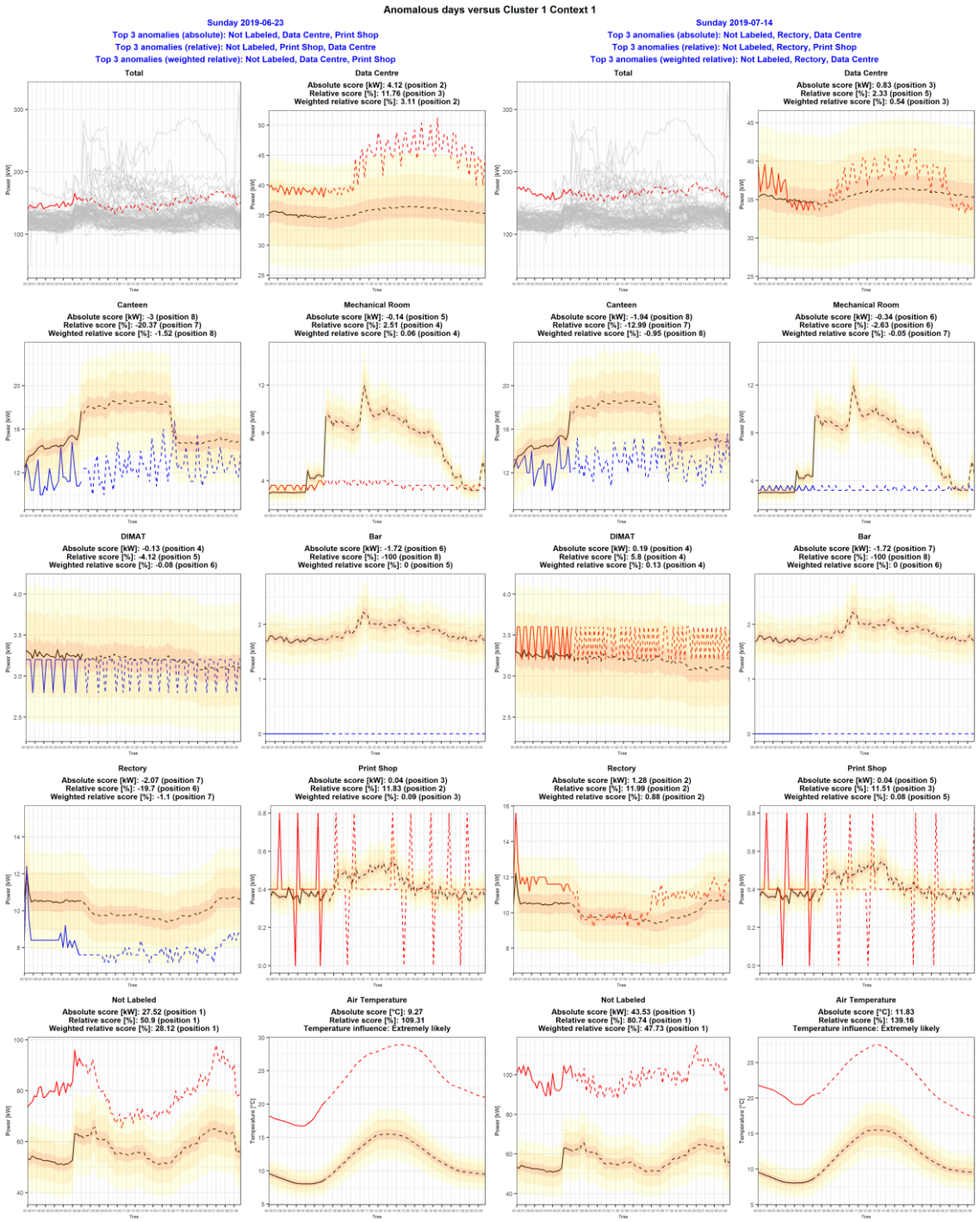

Figure 37 - Anomaly diagnosis for cluster number 1 + context number 1

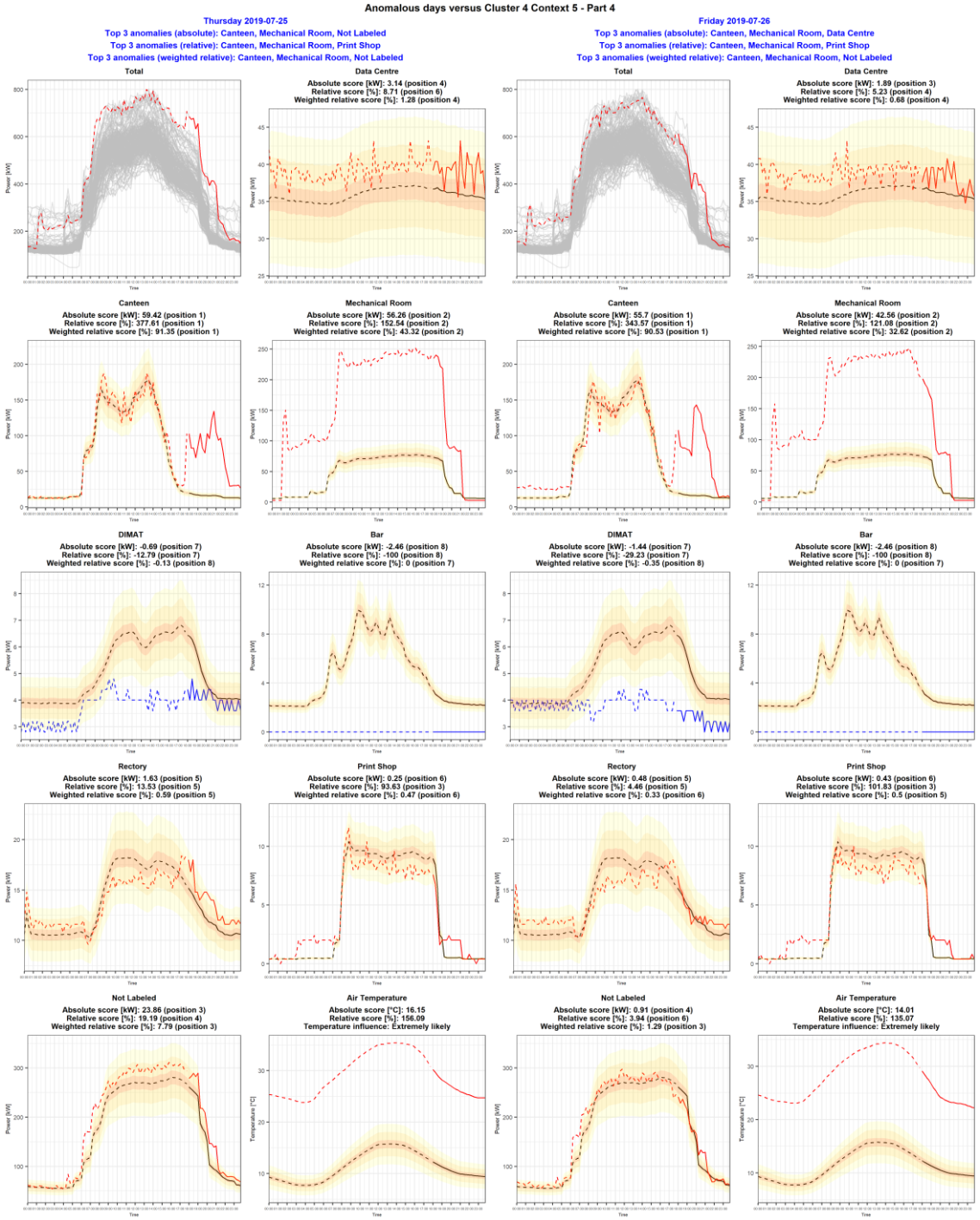

Figure 38 - Anomaly diagnosis for cluster number 4 + context number 5, part 4

This analysis of single anomalous instances concludes with the examination of those occurrences that belong to the cluster of Holidays, but in which the Mechanical Room is the prevailing sub-load in terms of responsibility on the anomaly; the behavior that has just been described is counter-intuitive, given what has been previously mentioned about this sub-load and the fact that its power demand is usually small during Holidays, even if they belong to summer months: it is therefore worth investigating these instances – that correspond to the second and third context on 2019-08-12 (anomalies number 16 and 29) and to the second context on 2019-08-13 (anomaly number 17) – in detail. Looking at Figure 39 and Figure 32 - left, it is evident that almost every single sub-load's profile is very different, in terms of shape and sometimes also magnitude, from the average group day's power demand profile; in particular, it can be seen that in all three instances the shapes of the power demand profiles of the Canteen and of the Mechanical Room resemble those of a working day with reduced activity (with a morning peak between 06:00 and 07:00 in the case of the Mechanical Room), the Data Centre's and the Rectory's profiles are also slightly higher than normal (especially on 2019-08-12), the Print Shop shows signs of activity on 2019-08-12 around 07:00 - 09:00, the Not Labeled sub-load has a peak on 2019-08-12 between 05:00 and 09:00 and then its power demand values become even slightly smaller than those of the mean group day's profile during the third context on 2019-08-12 and during the second context on the following day. All of these behaviors are quite unusual and they seem to point towards the hypothesis that these two days, originally classified as days of full closing of the university campus due to summer Holidays, may in reality be days of semi-regular functioning or at least days in which some sort of systems activity took place, with shapes and magnitudes of the power demand profiles for the various sub-loads and for the Total power that are a middle ground between those of the days in clusters number 2/3 and those of the days in cluster number 1. In this case, the detected anomaly may be attributed to a "misclassification" of the days considered, due to the fact that there is no cluster that accurately represents the behavior of these profiles: they are unique and a subset containing only those two profiles, maybe together with those of 2019-11-09, 2019-11-10 and 2019-12-27 (whose characteristics have been discussed earlier in this section) would probably not make much sense in terms of identifying behaviors that are repeated during the whole year.

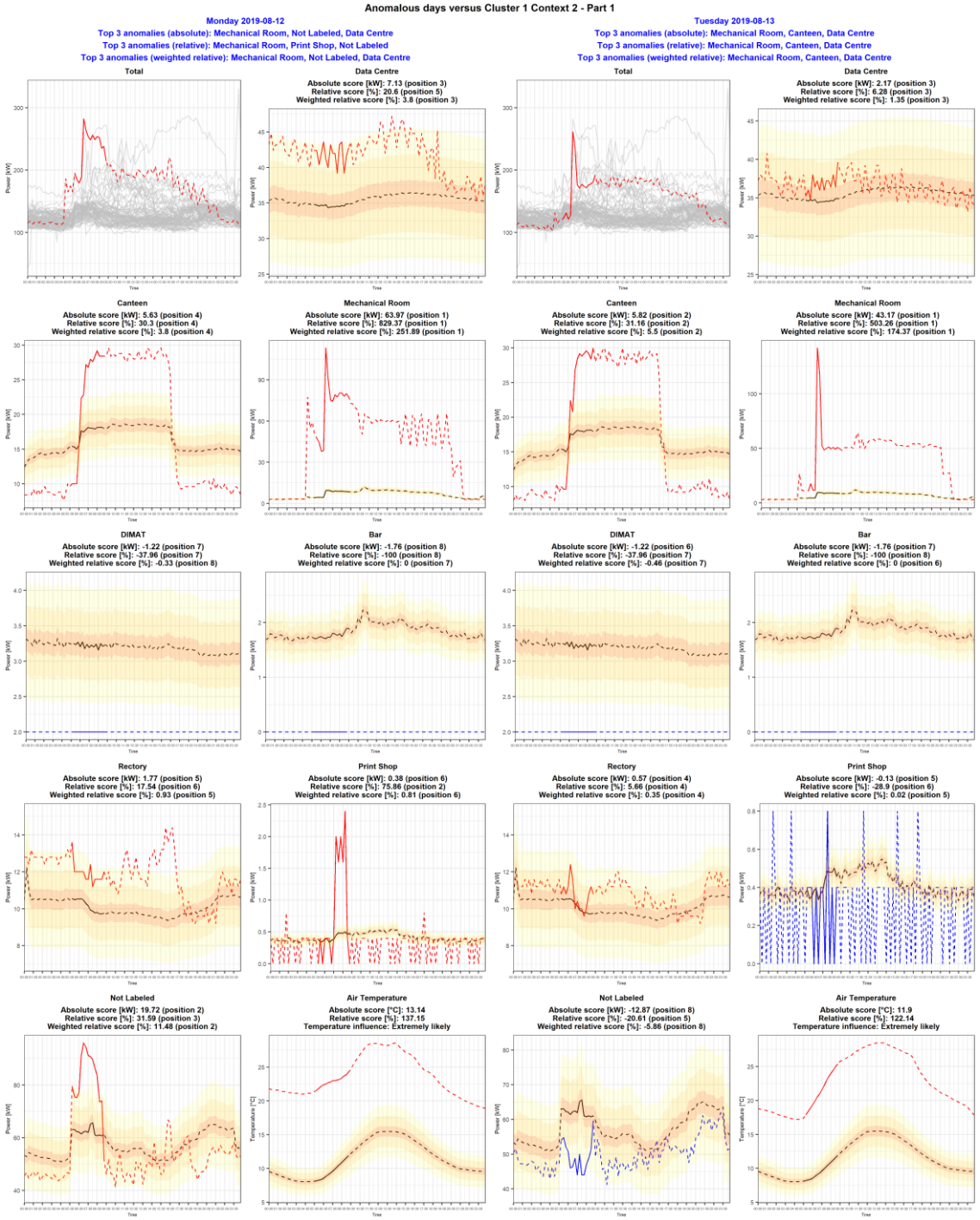

Figure 39 - Anomaly diagnosis for cluster number 1 + context number 2, part 1

To conclude this section, it is necessary to talk about the remaining instances, that include all summer days with regular or semi-regular systems' functioning, in which the most anomalous sub-load is the Mechanical Room; it is not necessary to analyze these occurrences one by one (they are reported in the Appendix, in Chapter 9), since almost all of them show the same general behavior: the Mechanical Room is always the most evidently abnormal sub-load, while the remaining loads' power demand profiles do not differ greatly from the average group day's power demand profiles; in many cases, the second and third most anomalous sub-loads are respectively the Not Labeled load and the Data Centre load: this is due to the fact that, in the same way the Mechanical Room's load is strongly correlated to high external air temperature and consequent cooling needs, part of the power demand of the two above mentioned sub-loads derives from cooling equipment and appliances (the room chiller for the university servers in the Data Centre and the various unlabeled HVAC systems included in the Not Labeled load).
In some rare occasions, the Canteen load is also slightly higher than normal: for example, on 2019-06-24 in the second context (anomaly number 19, Figure 40 - left) and on 2019-07-13 during the second (anomaly number 21, Figure 41) and third (anomaly number 34, Figure 42) contexts. In the first case, this behavior might be due to the day type: 2019-06-24 is the day when the St. Patron of Turin is celebrated and it is possible that certain Canteen systems followed a regular work day schedule, since this is what the profile shape suggests. In the second and third cases, the anomaly seems to be related to an actual unexpected behavior: a spike of about 10 kW can be seen, maybe due to a very short erroneous activation of an appliance.
These last considerations also confirm what was introduced in the previous section about the results of Table 2 and the hypothesis that many days were labeled as anomalous during different time windows due to a load being higher than normal for most of the duration of the day. Although all these Mechanical Room-related occurrences are marked as anomalies by the detection process, the subsequent diagnostic step unveils their real nature and allows the user to understand the common cause behind the abnormal power demand at meter-level: this also finds confirmation in the message suggesting high air temperature influence likelihood on the behavior of the sub-loads interested by this external factor. It is therefore possible, thanks to the analysis of the results of the diagnostic process, for the user to consider these instances as something that belongs to the "realm of normality", given the above described repeating behavior that all of them have in common and the fact that increased cooling needs during warmer periods are anything but unexpected.

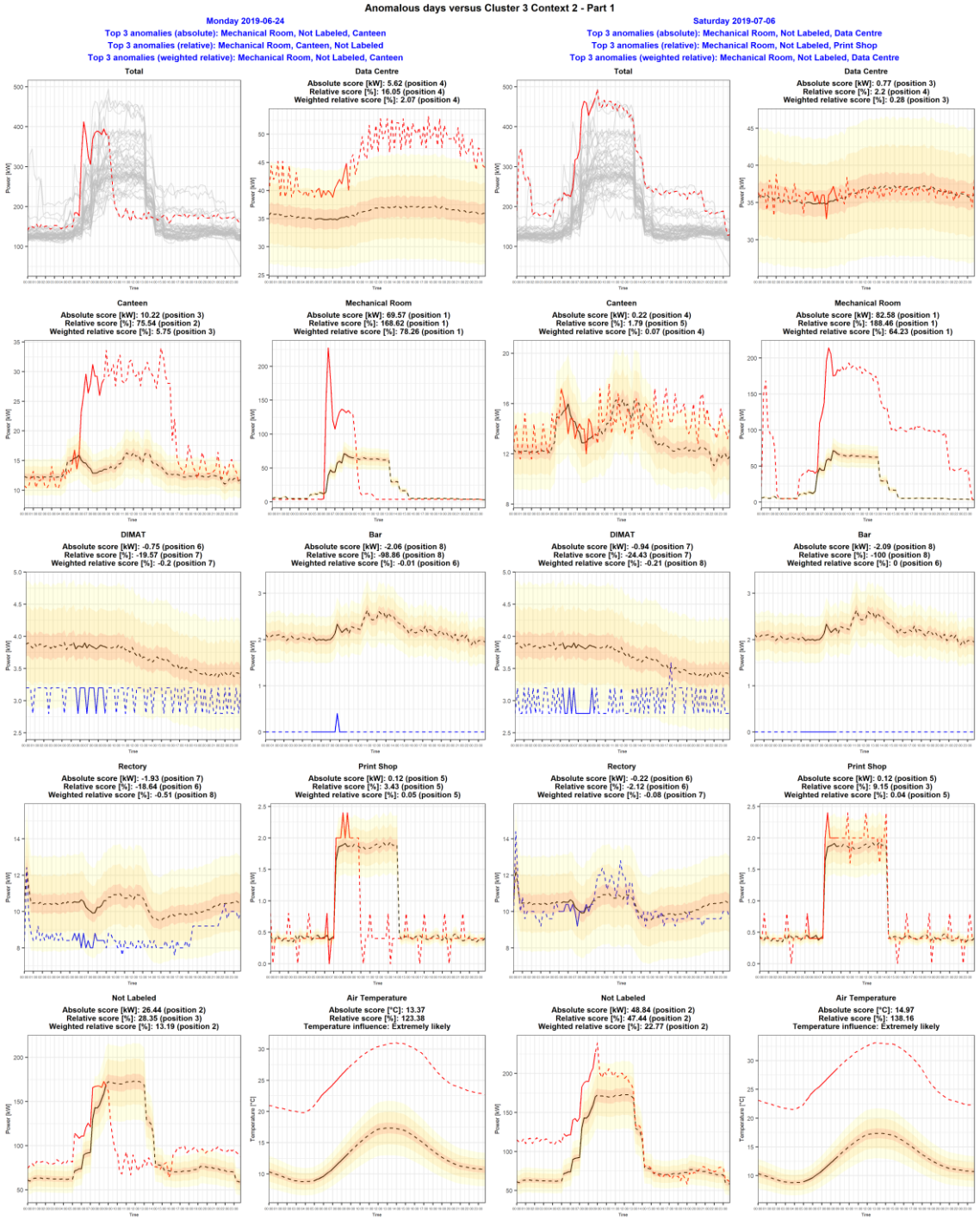

Figure 40 - Anomaly diagnosis for cluster number 3 + context number 2, part 1

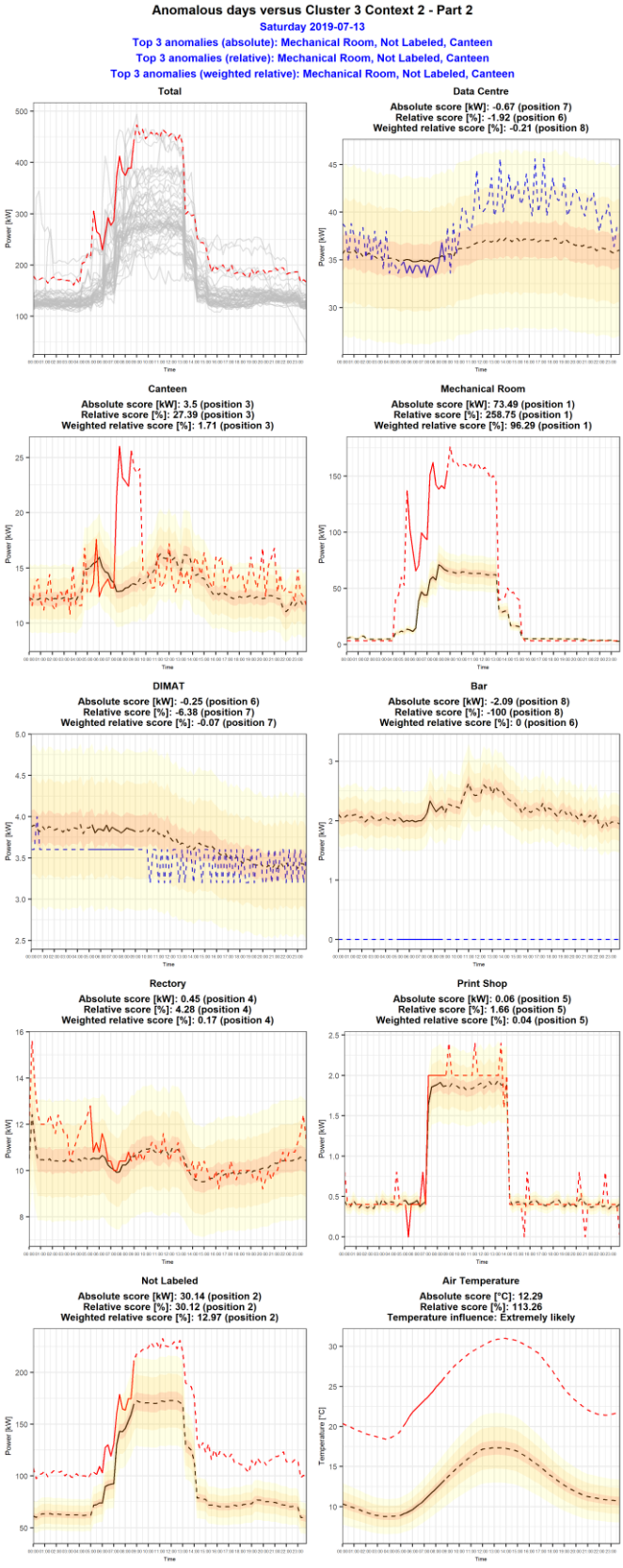

Figure 41 - Anomaly diagnosis for cluster number 3 + context number 2, part 2

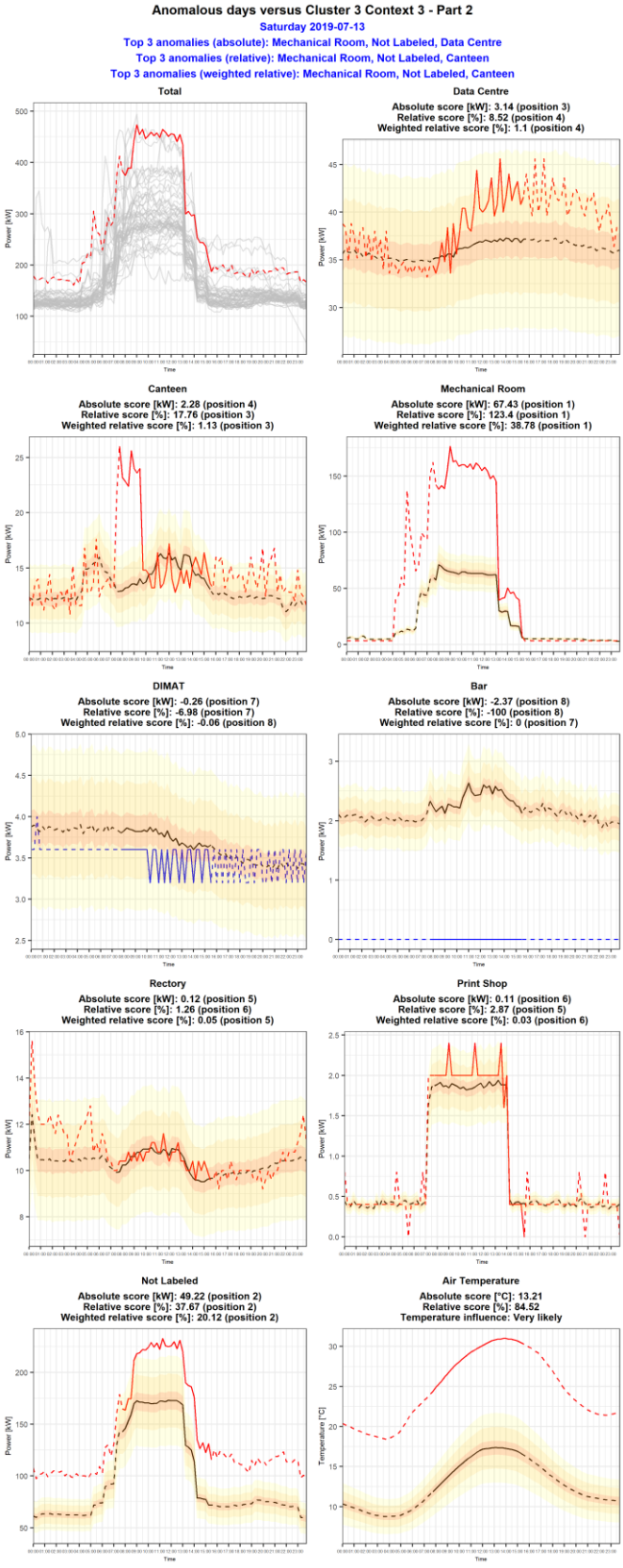

Figure 42 - Anomaly diagnosis for cluster number 3 + context number 3, part 2

One might then argue that a more accurate definition of clusters (similar to the one represented in Figure 26), aimed especially at separating summer working days with inherently higher power demand from the rest of the working days, might solve this issue of instances detected as abnormal being "false positives" by preventing the process from detecting them as anomalous from the beginning. This is certainly a viable solution: however, in the domain of anomaly detection, the expression "better to be safe than sorry" is often valuable. An experienced user is easily able to recognize a "false alarm" such as the one described above when looking at the data and discarding an anomalous instance thanks to expert knowledge takes just a small amount of time; on the other hand, introducing an excessive amount of "guidance" in a process that should be as automatic as possible comes with the risk of losing interesting, or sometimes even critical, information. This is something that can be directly seen in this case study: although the Mechanical Room-related occurrences were all analyzed together for the reasons mentioned above, there is an interesting aspect that emerges from the comparison between some of them, especially those that include night hours. In fact, the anomalies belonging to cluster number 4 and contexts number 1 and 5 show a Total power demand behavior (which is strictly related to that of the Mechanical Room, due to the entity of the load) that is sometimes quite different. A clear example of this is given by anomalies number 64 or 65 (Figure 43) versus anomaly number 67 (Figure 44 - right): in the first two instances, the Total power demand profile is consistently flat (or almost flat) and high during the beginning of the night hours, indicating that the Mechanical Room is still active - with values of power demand over 100 kW - even when the cooling needs are supposed to be reduced thanks to the absence of people and to lower night temperatures; in the last case, on the other hand, the Mechanical Room load decreases gradually approaching midnight and this is reflected in the behavior of the Total power demand curve, which follows a similar path of slow decrease to base-load values. Given that the two first two instances are characterized by external air temperature values which are basically identical to those that can be found in the third occurrence and there is no other evident difference between the examined items, more intense cooling activity on the first two nights is apparently not justifiable with the information that is available and therefore it is not unreasonable to think that the measurement of high Mechanical Room power demand values at night may be related to certain cooling loads being left active from daytime building operation or to other cases of poor practice. Had the process automatically created a cluster containing only the Mechanical Room-related instances prior to the detection phase, there would have been a decent chance of no anomalies being reported as a result, due to the high degree of similarity of the profiles: therefore, the above described peculiarity may have been significantly harder to notice.

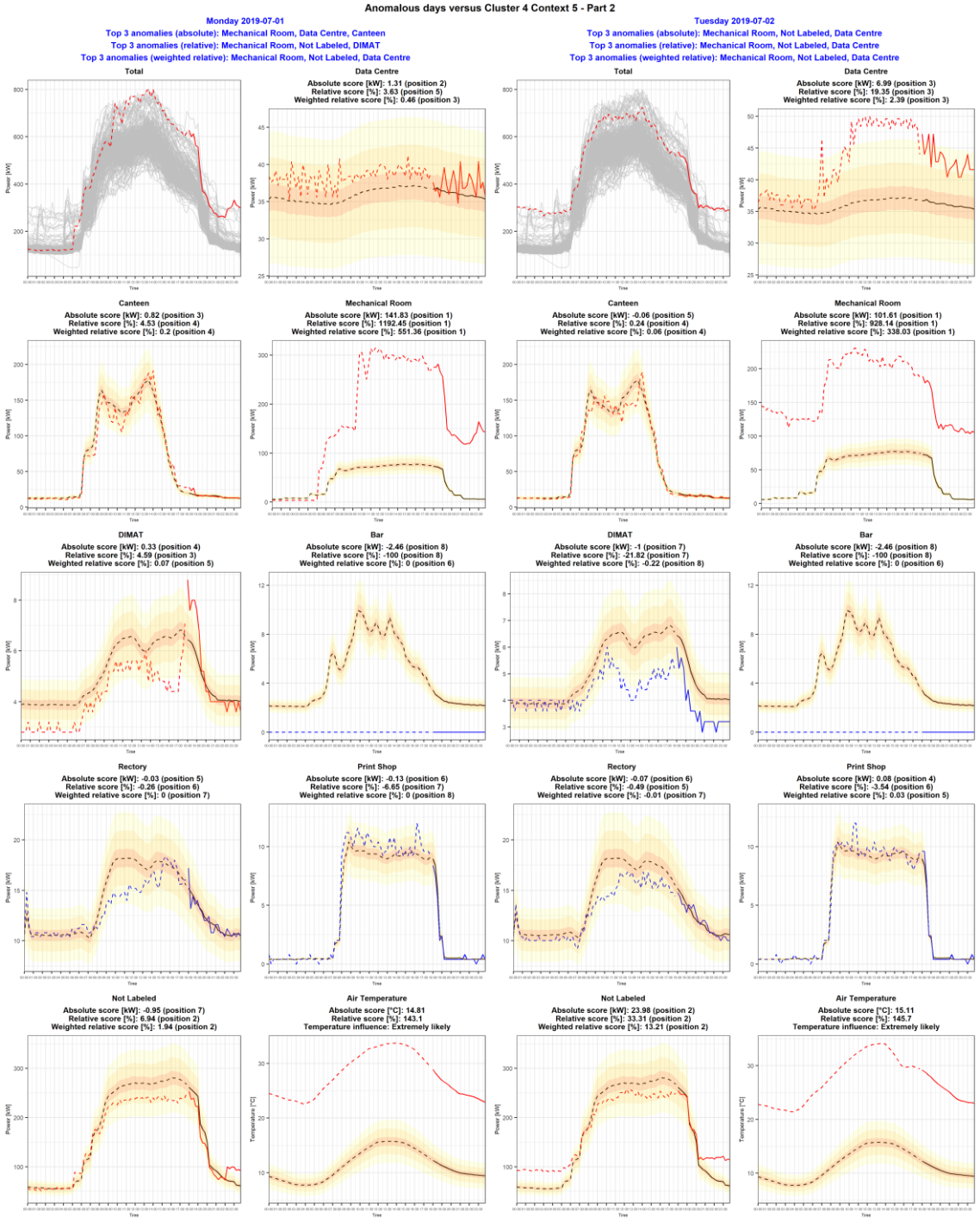

Figure 43 - Anomaly diagnosis for cluster number 4 + context number 5, part 2

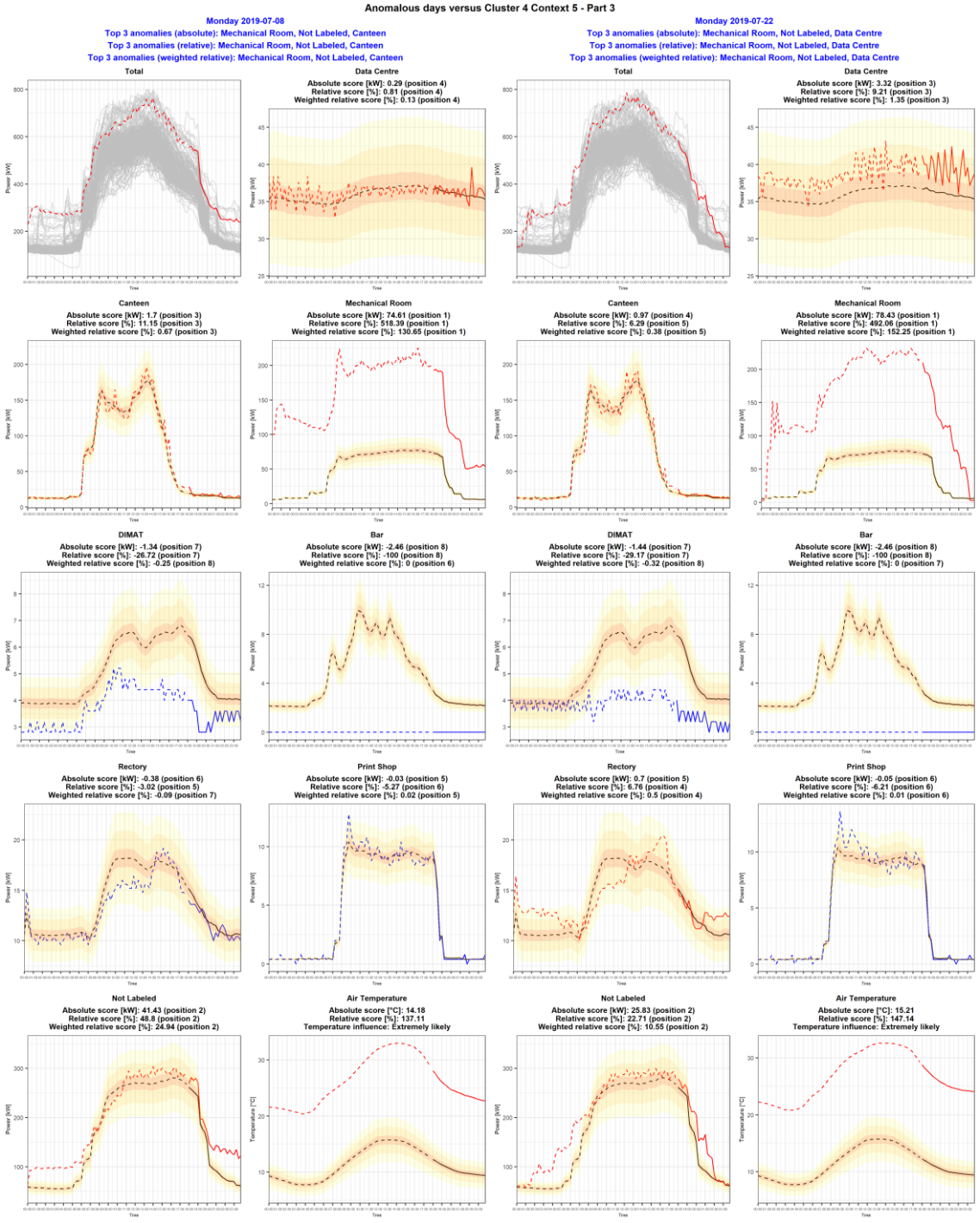

Figure 44 - Anomaly diagnosis for cluster number 4 + context number 5, part 3

7. Conclusions and future work

This work was aimed at the creation of a framework for the analysis of anomalous power demand patterns in large buildings, consisting of an initial anomaly detection phase performed at meter-level and a subsequent anomaly diagnosis phase at sub-meter-level whose goal was to identify the sub-loads that were mainly responsible for the unexpected behaviors.

The anomaly detection process was based on the Contextual Matrix Profile (CMP), a technique for time series analysis introduced by De Paepe et al. in 2020 [9]. The CMP, which constitutes a variation of the original Matrix Profile (MP) presented by Yeh et al. in 2016 [5], was chosen in this work for its flexibility with respect to the definition of "anomaly". When applying the CMP algorithm, the research of the anomaly is not focused on the most unique patterns in the whole time series, which is the logic employed in traditional MP; instead, this technique compares patterns that start in the same time period (the "context"), while allowing for temporal shifts between the considered subsequences. However, the CMP requires a certain amount of pre-processing for its correct functioning, which is left to the user's knowledge in the application domain.

In the framework adopted in this work, the first step of this pre-processing phase was aimed at the creation of groups of days that showed similar behaviors in terms of daily Total power demand's patterns, to avoid comparisons between occurrences belonging to different day types, such as Holidays versus regular working days. This was obtained by means of a combination of agglomerative hierarchical clustering techniques with a successive "supervised" manual reallocation of days that, on the basis of expert knowledge, were assigned to the incorrect cluster.

The goal of the second step of the pre-processing phase was to identify daily time windows of interest, in order to analyze the power demand profiles of portions of days rather than those of the full days. The identified time windows correspond to parts of the day that show unique behavior in terms of systems operation and power demand, such as the night hours, the ramp-up/ramp-down periods and so on. This step was carried out using a Classification and Regression Tree (CART).

For each combination of the two above mentioned parameters - time windows (or "contexts") and groups (or "clusters") - a CMP was calculated and the anomaly detection phase, aimed at identifying the abnormal days in terms of Total power demand for each CMP produced, was performed by means of two conventional techniques, the boxplot and the elbow method, both based on the comparison between the median values of the different columns in the CMP.

The last stage of the presented framework was the diagnosis of the anomalous instances at a sub-load-level, where three metrics for ranking the sub-loads' power demand profiles, for each anomalous occurrence, in terms of distance from the mean profiles of each group were introduced, with the aim of evaluating each sub-load both from an "absolute" (to quantify how much different, in terms of kW, the power demand profile of the examined day is from the mean profile of the group) and a "relative" (to quantify

how much different, in terms of percentage, the power demand profile of the examined day is from the mean profile of the group) point of view. This kind of analysis is also performed on the external air temperature, to suggest potential correlation between the value of this parameter and the behavior of those sub-loads whose power demand is influenced by seasonality.

The results of this diagnostic process were discussed and some individual instances were analyzed in detail; the main takeaway from this final step was that, although various occurrences labeled as anomalous actually presented abnormalities in the sub-loads' behaviors, many others were sort of "false positives": the reasons for this were discussed and possible modifications to the method employed in this work were taken into consideration, analyzing the main critical aspects connected to them. An important conclusion that was obtained at the end of these analyses is that the accuracy in the definition of the different subsets of days is directly reflected in the quality of the results of the anomaly detection phase; however, an overdetailed characterization of the different day types may also cause undesired effects, such as an increased difficulty in retrieving certain peculiarities in power demand behavior. It is therefore key to the success of the whole process to find a good compromise, both in terms of clusters definition and of contexts definition, that allows for an anomaly detection step that does not lead to the loss of potentially valuable insight during the subsequent diagnostic phase.

Future works may focus on the characterization of the facilities, systems and appliances that were not monitored at a sub-meter-level, which in this work were grouped together in the "Not Labeled" sub-load. Since the diagnostic process often classifies this sub-load as one of the most responsible for the anomaly both from the absolute and from the relative point of view, further inspection on which specific system behaves in an unexpected way should provide even more interesting information to the user. Since not being able to clearly identify the main culprits behind an anomalous instance when the Not Labeled sub-load is dominant has been a recurring issue during the analysis of the results of the diagnostic process, a deeper level of monitoring should be able to provide explanations to behaviors that are otherwise not fully understandable with just the raw power demand data. Given that many contributors to the "Not Labeled" macro-category are single devices – such as elevators, alarm systems and so on – their monitoring may either be performed by means of system-level sensors (the more expensive solution but also the more reliable one) or with an approach similar to that of Non-intrusive Load Monitoring (NILM), that aims at extracting information about the electricity consumption of individual appliances from the analysis of aggregate voltage and/or current data.

8. Bibliography

[1] J. Granderson, G. Lin, D. Blum, J. Page, M. Spears, and M. A. Piette, "Integrating diagnostics and model-based optimization," *Energy and Buildings*, vol. 182, pp. 187–195, Jan. 2019, doi: 10.1016/j.enbuild.2018.10.015.

[2] H. Kramer, G. Lin, J. Granderson, C. Curtin, E. Crowe, and A. Jiron, "Synthesis of Year One Outcomes in the," 2017.

[3] Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira, "Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives," *Applied Energy*, vol. 287. Elsevier Ltd, Apr. 01, 2021. doi: 10.1016/j.apenergy.2021.116601.

[4] L. Erhan *et al.*, "Smart Anomaly Detection in Sensor Systems: A Multi-Perspective Review," Oct. 2020, doi: 10.1016/j.inffus.2020.10.001.

[5] C.-C. Michael Yeh *et al.*, "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets."

[6] J. Y. Park, E. Wilson, A. Parker, and Z. Nagy, "The good, the bad, and the ugly: Data-driven load profile discord identification in a large building portfolio," *Energy and Buildings*, vol. 215, May 2020, doi: 10.1016/j.enbuild.2020.109892.

[7] C. Nichiforov, I. Stancu, I. Stamatescu, and G. Stamatescu, "Information Extraction Approach for Energy Time Series Modelling," in *2020 24th International Conference on System Theory, Control and Computing, ICSTCC 2020 - Proceedings*, Oct. 2020, pp. 886–891. doi: 10.1109/ICSTCC50638.2020.9259635.

[8] C. Nichiforov, G. Stamatescu, I. Stamatescu, and I. F˘, *Learning Dominant Usage from Anomaly Patterns in Building Energy Traces*. 2020. doi: 10.0/Linux-x86_64.

[9] D. de Paepe *et al.*, "A generalized matrix profile framework with support for contextual series analysis," *Engineering Applications of Artificial Intelligence*, vol. 90, Apr. 2020, doi: 10.1016/j.engappai.2020.103487.

[10] H. Kramer, G. Lin, C. Curtin, E. Crowe, and J. Granderson, "Building analytics and monitoring-based commissioning: industry practice, costs, and savings," *Energy Efficiency*, vol. 13, no. 3, pp. 537–549, Mar. 2020, doi: 10.1007/s12053-019-09790-2.

[11] C. Miller and F. Meggers, "The Building Data Genome Project: An open, public data set from non-residential building electrical meters," in *Energy Procedia*, 2017, vol. 122, pp. 439–444. doi: 10.1016/j.egypro.2017.07.400.

[12] C. Miller *et al.*, "The Building Data Genome Project 2, energy meter data from the ASHRAE Great Energy Predictor III competition," *Scientific Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1038/s41597-020-00712-x.

[13] B. Nastasi, M. Manfren, and M. Noussan, "Open data and energy analytics," *Energies*, vol. 13, no. 9. MDPI AG, May 01, 2020. doi: 10.3390/en13092334.

[14] Z. (John) Zhai and A. Salazar, "Assessing the implications of submetering with energy analytics to building energy savings," *Energy and Built Environment*, vol. 1, no. 1, pp. 27–35, Jan. 2020, doi: 10.1016/j.enbenv.2019.08.002.

[15] R. Chiosa, M. S. Piscitelli, and A. Capozzoli, "A data analytics-based energy information system (EIS) tool to perform meter-level anomaly detection and diagnosis in buildings," *Energies*, vol. 14, no. 1, Jan. 2021, doi: 10.3390/en14010237.

[16] Institute of Electrical and Electronics Engineers and IEEE Geoscience and Remote Sensing Society, *2016 IEEE International Geoscience & Remote Sensing Symposium : proceedings : July 10-15, 2016, Beijing, China*.

[17] T. Huang, H. Sethu, and N. Kandasamy, "A New Approach to Dimensionality Reduction for Anomaly Detection in Data Traffic," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 651–665, Sep. 2016, doi: 10.1109/TNSM.2016.2597125.

[18] "UCR Matrix Profile Page." https://www.cs.ucr.edu/~eamonn/MatrixProfile.html (accessed Oct. 18, 2021).

[19] Y. Zhu *et al.*, "Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins."

[20] Y. Zhu, C.-C. M. Yeh, Z. Zimmerman, K. Kamgar, and E. Keogh, "Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds."

[21] Z. Zimmerman *et al.*, "Matrix Profile XIV: Scaling Time Series Motif Discovery with GPUs to Break a Quintillion Pairwise Comparisons a Day and beyond," in *SoCC 2019 - Proceedings of the ACM Symposium on Cloud Computing*, Nov. 2019, pp. 74–86. doi: 10.1145/3357223.3362721.

[22] Z. Zimmerman *et al.*, "Matrix Profile XVIII: Time Series Mining in the Face of Fast Moving Streams using a Learned Approximate Matrix Profile."

[23] R. Akbarinia and B. Cloez, “Efficient Matrix Profile Computation Using Different Distance Functions,” Jan. 2019, [Online]. Available: http://arxiv.org/abs/1901.05708

[24] C. Onwongsa and C. Ratanamahatana, “An enhanced time series motif discovery using approximated matrix profile,” in *ACM International Conference Proceeding Series*, Aug. 2020, pp. 180–189. doi: 10.1145/3421558.3421586.

[25] A. Kalantar, Z. Zimmerman, and P. Brisk, “FA-LAMP: FPGA-Accelerated Learned Approximate Matrix Profile for Time Series Similarity Prediction,” in *Proceedings - 29th IEEE International Symposium on Field-Programmable Custom Computing Machines, FCCM 2021*, May 2021, pp. 40–49. doi: 10.1109/FCCM51124.2021.00013.

[26] J. C. Romero, A. Vilches, A. Rodríguez, A. Navarro, and R. Asenjo, “ScrimpCo: scalable matrix profile on commodity heterogeneous processors,” *Journal of Supercomputing*, vol. 76, no. 11, pp. 9189–9210, Nov. 2020, doi: 10.1007/s11227-020-03199-w.

[27] I. Fernandez, A. Villegas, E. Gutierrez, and O. Plata, “Accelerating time series motif discovery in the Intel Xeon Phi KNL processor,” *Journal of Supercomputing*, vol. 75, no. 11, pp. 7053–7075, Nov. 2019, doi: 10.1007/s11227-019-02923-5.

[28] C.-C. M. Yeh, N. Kavantzas, and E. Keogh, “Matrix Profile IV: Using Weakly Labeled Time Series to Predict Outcomes,” 2150.

[29] H. A. Dau and E. Keogh, “Matrix profile V: A generic technique to incorporate domain knowledge into motif discovery,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2017, vol. Part F129685, pp. 125–134. doi: 10.1145/3097983.3097993.

[30] C.-C. M. Yeh, N. Kavantzas, and E. Keogh, “Matrix Profile VI: Meaningful Multidimensional Motif Discovery.”

[31] Y. Zhu, M. Imamura, D. Nikovski, and E. Keogh, “Matrix Profile VII: Time Series Chains: A New Primitive for Time Series Data Mining.”

[32] M. Imamura, T. Nakamura, and E. Keogh, “Matrix Profile XXI: A Geometric Approach to Time Series Chains Improves Robustness,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2020, pp. 1114–1122. doi: 10.1145/3394486.3403164.

[33] S. Gharghabi, Y. Ding, C.-C. M. Yeh, K. Kamgar, L. Ulanova, and E. Keogh, “Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels.”

[34] M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh, "Matrix profile goes MAD: variable-length motif and discord discovery in data series," *Data Mining and Knowledge Discovery*, vol. 34, no. 4, pp. 1022–1071, Jul. 2020, doi: 10.1007/s10618-020-00685-w.

[35] F. Madrid, S. Imani, R. Mercer, Z. Zimmerman, N. Shakibay, and E. Keogh, "Matrix Profile XX: Finding and Visualizing Time Series Motifs of All Lengths using the Matrix Profile."

[36] T. Nakamura, M. Imamura, R. Mercer, and E. Keogh, "MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives."

[37] S. Gharghabi, S. Imani, A. Bagnall, A. Darvishzadeh, and E. Keogh, "An Ultra-Fast Time Series Distance Measure to allow Data Mining in more Complex Real-World Deployments."

[38] S. Imani, F. Madrid, W. Ding, S. Crouter, and E. Keogh, "Matrix Profile XIII: Time Series Snippets: A New Primitive for Time Series Data Mining."

[39] Y. Zhu, C. C. M. Yeh, Z. Zimmerman, and E. Keogh, "Matrix profile XVII: Indexing the matrix profile to allow arbitrary range queries," in *Proceedings - International Conference on Data Engineering*, Apr. 2020, vol. 2020-April, pp. 1846–1849. doi: 10.1109/ICDE48307.2020.00185.

[40] C.-C. M. Yeh, H. van Herle, and E. Keogh, "Matrix Profile III: The Matrix Profile Allows Visualization of Salient Subsequences in Massive Time Series."

[41] Y. Zhu, A. Mueen, and E. Keogh, "Admissible Time Series Motif Discovery with Missing Data."

[42] K. Kamgar, S. Gharghabi, and E. Keogh, "Matrix Profile XV: Exploiting Time Series Consensus Motifs to Find Structure in Time Series Sets."

[43] D. de Paepe and S. van Hoecke, "Mining recurring patterns in real-valued time series using the radius profile," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, Nov. 2020, vol. 2020-November, pp. 984–989. doi: 10.1109/ICDM50108.2020.00113.

[44] S. Imani and E. Keogh, "XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE Matrix Profile XIX: Time Series Semantic Motifs: A New Primitive for Finding Higher-Level Structure in Time Series."

[45] S. Alaee, R. Mercer, K. Kamgar, and E. Keogh, "Matrix Profile XXII: Exact Discovery of Time Series Motifs under DTW."

[46] American Automatic Control Council. and Institute of Electrical and Electronics Engineers., *2020 American Control Conference (ACC).*

[47] D. Furtado Silva and G. E. A. P. A. Batista, "Elastic Time Series Motifs and Discords," in *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, Jan. 2019, pp. 237–242. doi: 10.1109/ICMLA.2018.00042.

[48] R. Goebel, W. Wahlster, and J. Siekmann, "Lecture Notes in Artificial Intelligence 11013 Subseries of Lecture Notes in Computer Science LNAI Series Editors LNAI Founding Series Editor." [Online]. Available: http://www.springer.com/series/1244

[49] Q. Liu, C. K. Leung, and P. Hu, "A Two-Dimensional Sparse Matrix Profile DenseNet for COVID-19 Diagnosis Using Chest CT Images," *IEEE Access*, vol. 8, pp. 213718–213728, 2020, doi: 10.1109/ACCESS.2020.3040245.

[50] H. Li, Y. J. Wu, S. Zhang, and J. Zou, "Temporary rules of retail product sales time series based on the matrix profile," *Journal of Retailing and Consumer Services*, vol. 60, May 2021, doi: 10.1016/j.jretconser.2020.102431.

[51] E. K. Li, W. X. Xi, S.-H. Lee, and M. Vlachos, "LB_Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures," 2006.

[52] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently finding the most unusual time series subsequence," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2005, pp. 8–15. doi: 10.1109/ICDM.2005.79.

[53] E. Keogh and A. Mueen, "Time Series Data Mining Using the Matrix Profile: A Unifying View of Motif Discovery, Anomaly Detection, Segmentation, Classification, Clustering and Similarity Joins." [Online]. Available: www.cs.ucr.edu/~eamonn/MatrixProfile.html

[54] S. D. D. Anton, A. P. Lohfink, C. Garth, and H. D. Schotten, "Security in process: Detecting attacks in industrial process data," Nov. 2019. doi: 10.1145/3360664.3360669.

[55] C. Baru, Institute of Electrical and Electronics Engineers, and IEEE Computer Society, *2019 IEEE International Conference on Big Data : proceedings : Dec 9 - Dec 12, 2019, Los Angeles, CA, USA*.

[56] IEEE Reliability Society and Institute of Electrical and Electronics Engineers, *2019 IEEE International Conference on Prognostics and Health Management (ICPHM).*

[57] D. de Paepe, "Implications of Z-Normalization in the Matrix Profile." [Online]. Available: http://idlab.ugent.be

[58] "4 Classification: Basic Concepts, Decision Trees, and Model Evaluation."

[59] “8 Cluster Analysis: Basic Concepts and Algorithms.”

[60] “Solcast API Toolkit.” https://toolkit.solcast.com.au/historical (accessed Oct. 18, 2021).

9. Appendix

This last section contains the pictures that represent the results of the anomaly diagnosis phase that were not examined in detail in Chapter 6: they are all instances that belong to days of regular or semi-regular functioning where the Mechanical Room sub-load is the main responsible for the abnormality, due to the cooling needs related to the high external temperature.

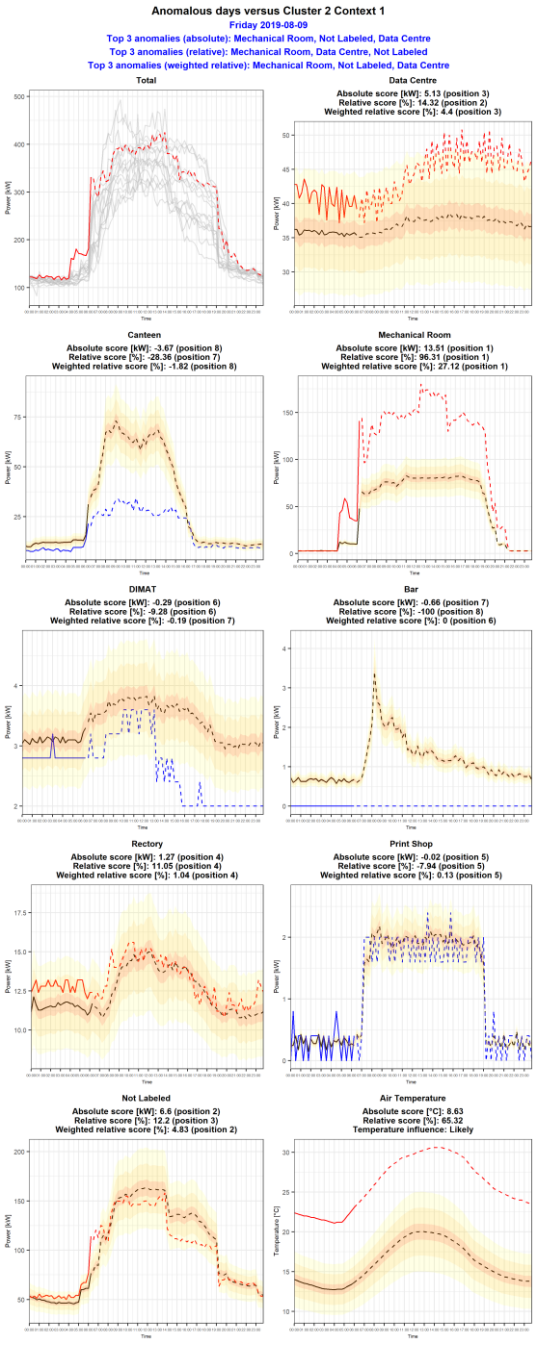

Figure A. 1 - Anomaly diagnosis for cluster number 2 + context number 1

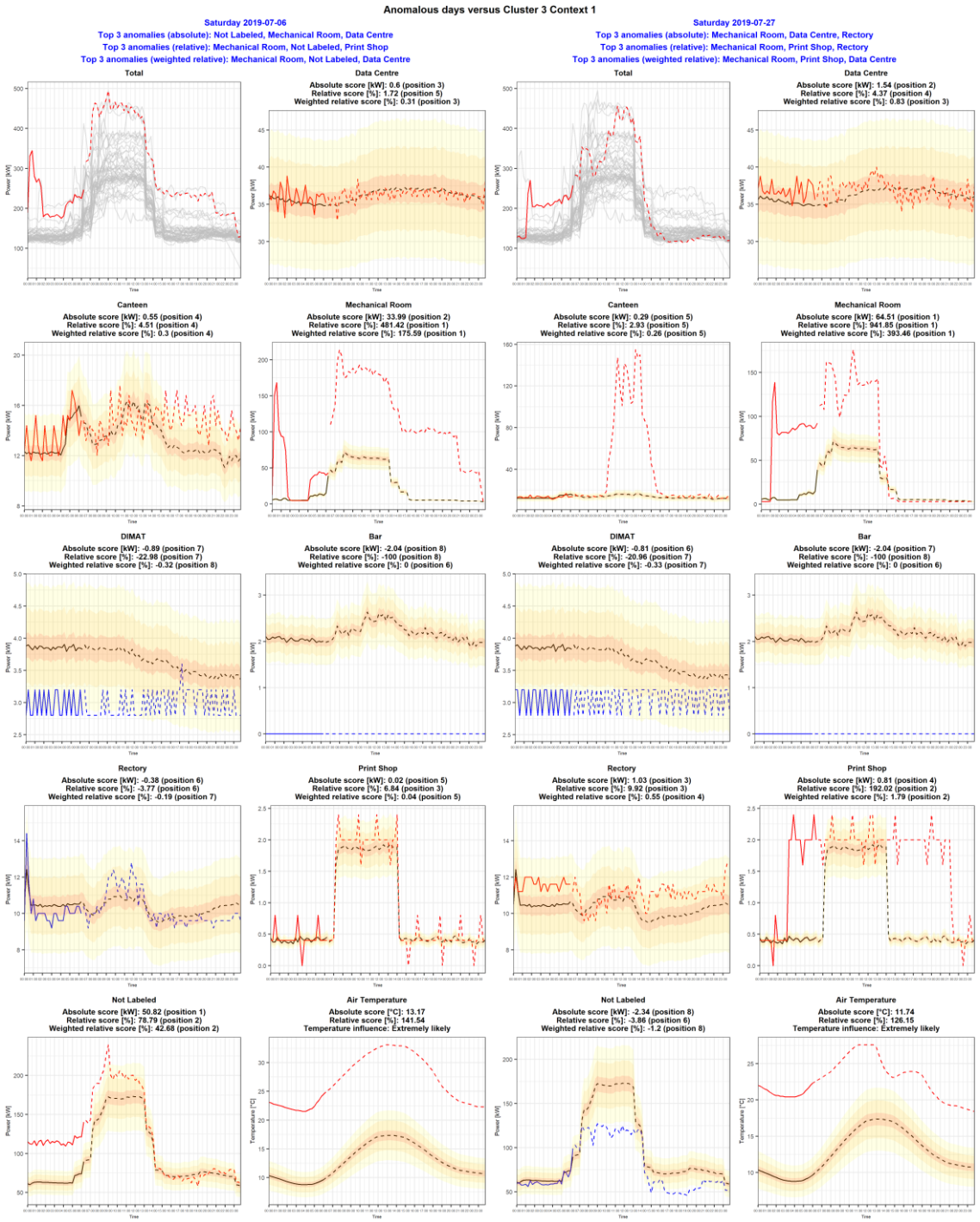

Figure A. 2 - Anomaly diagnosis for cluster number 3 + context number 1

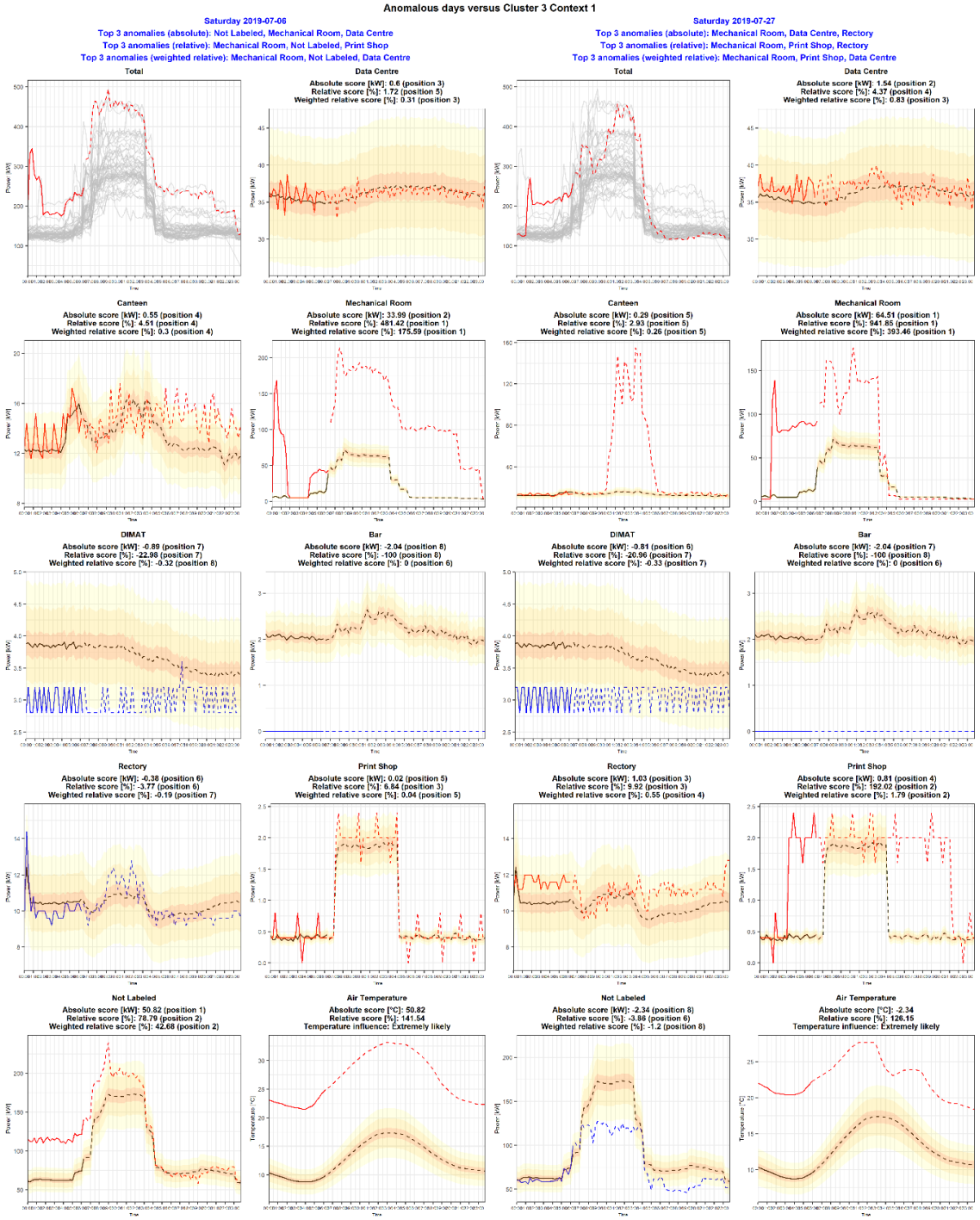

Figure A. 3 - Anomaly diagnosis for cluster number 3 + context number 1

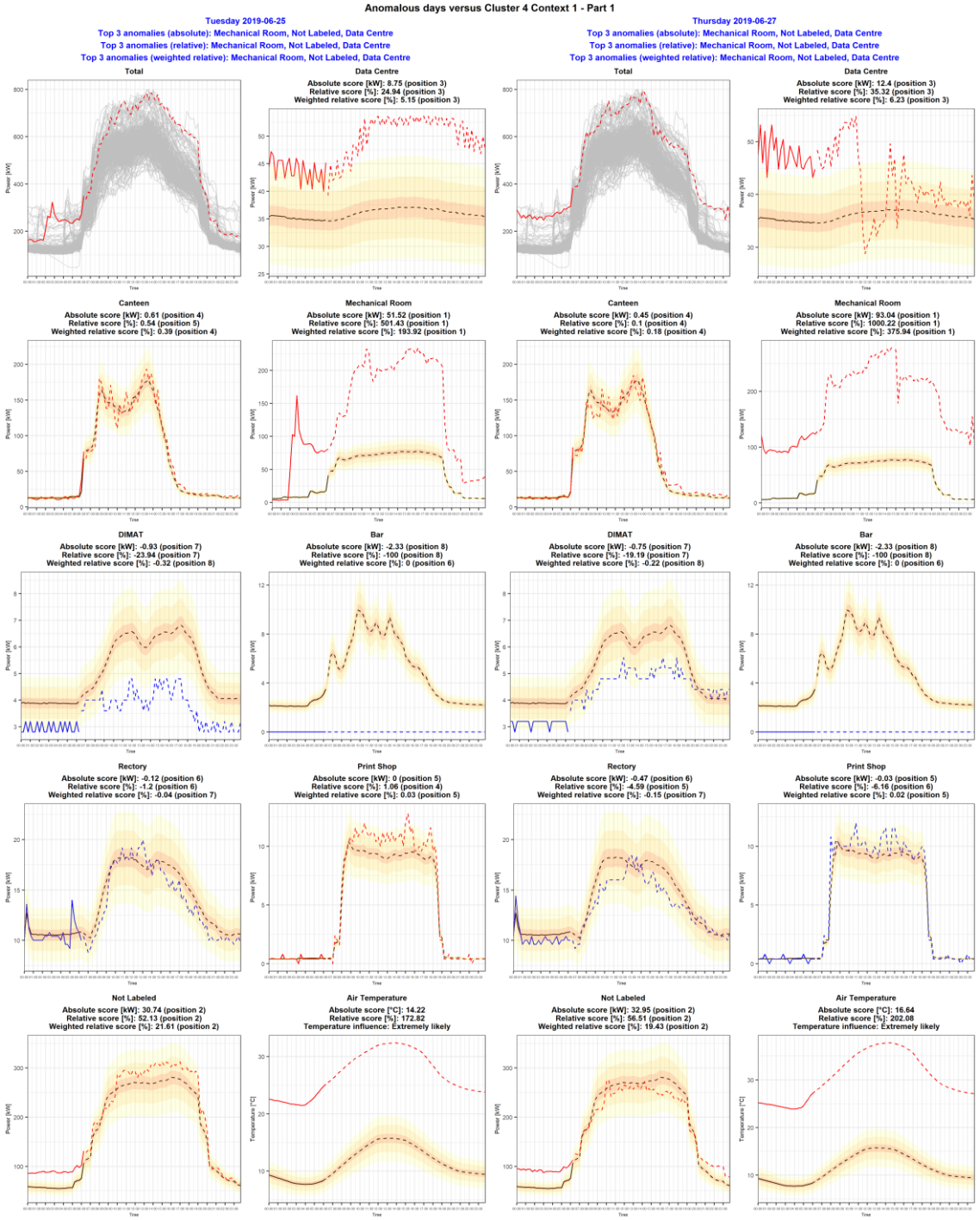

Figure A. 4 - Anomaly diagnosis for cluster number 4 + context number 1, part 1

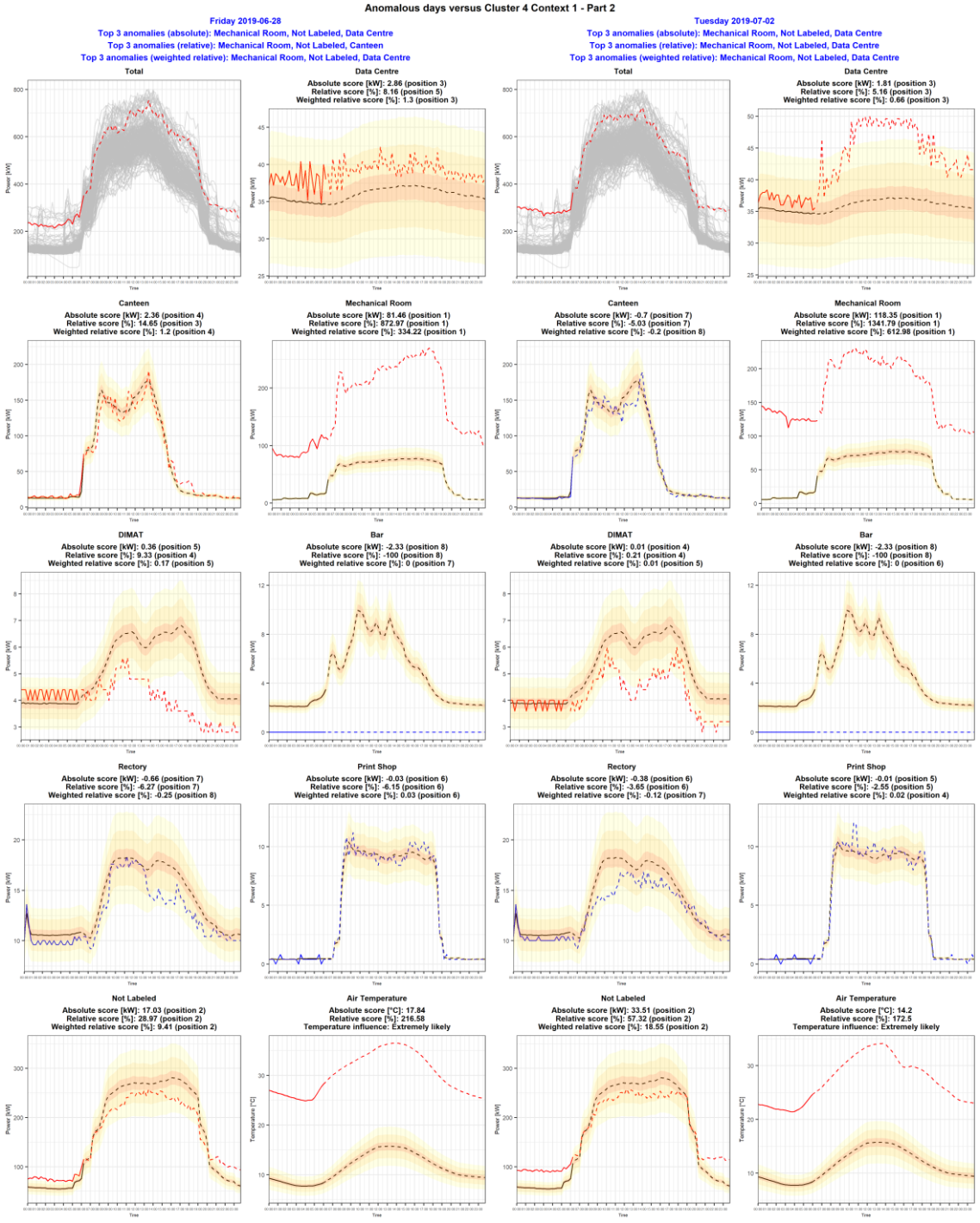

Figure A. 5 - Anomaly diagnosis for cluster number 4 + context number 1, part 2

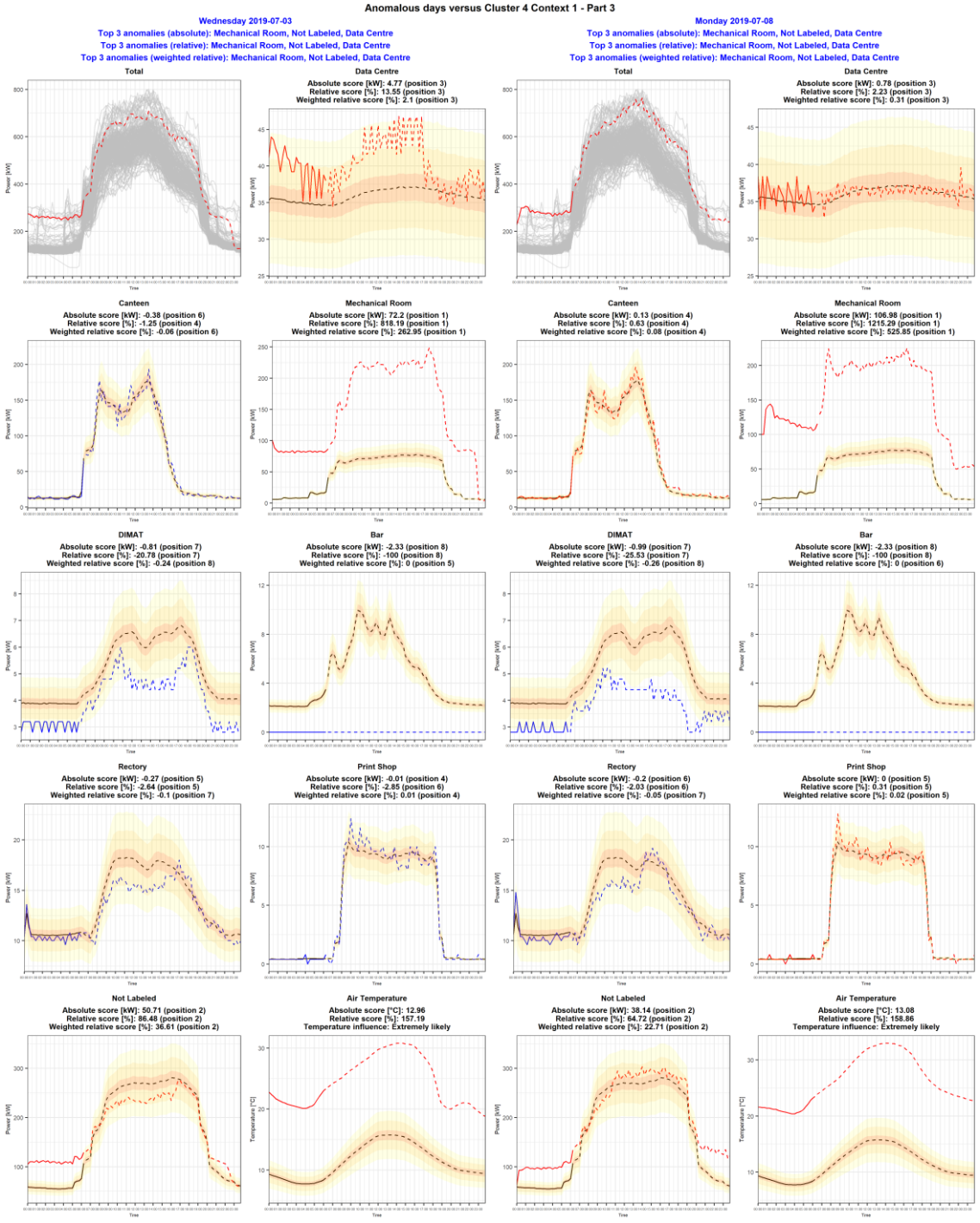

Figure A. 6 - Anomaly diagnosis for cluster number 4 + context number 1, part 3

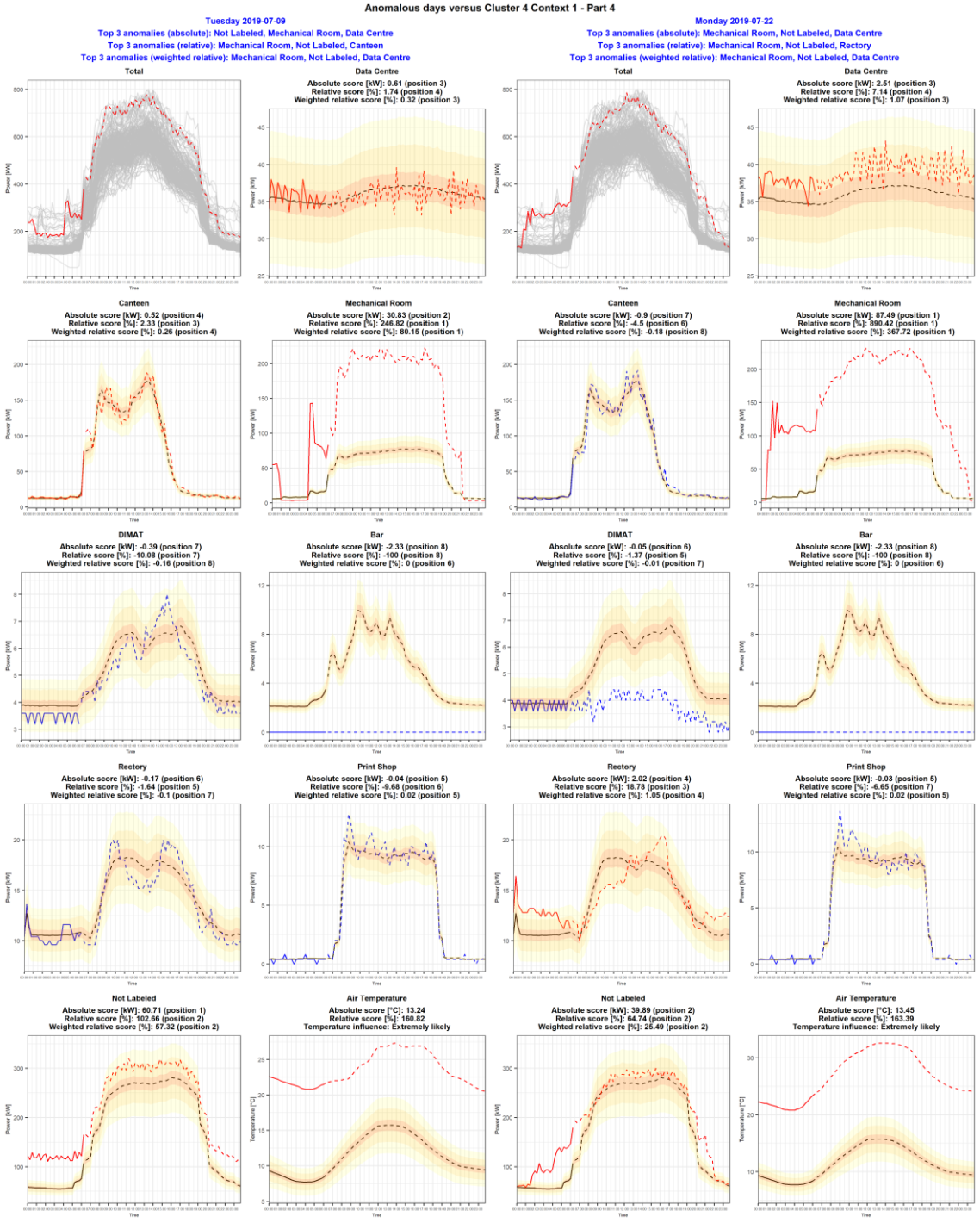

Figure A. 7 - Anomaly diagnosis for cluster number 4 + context number 1, part 4

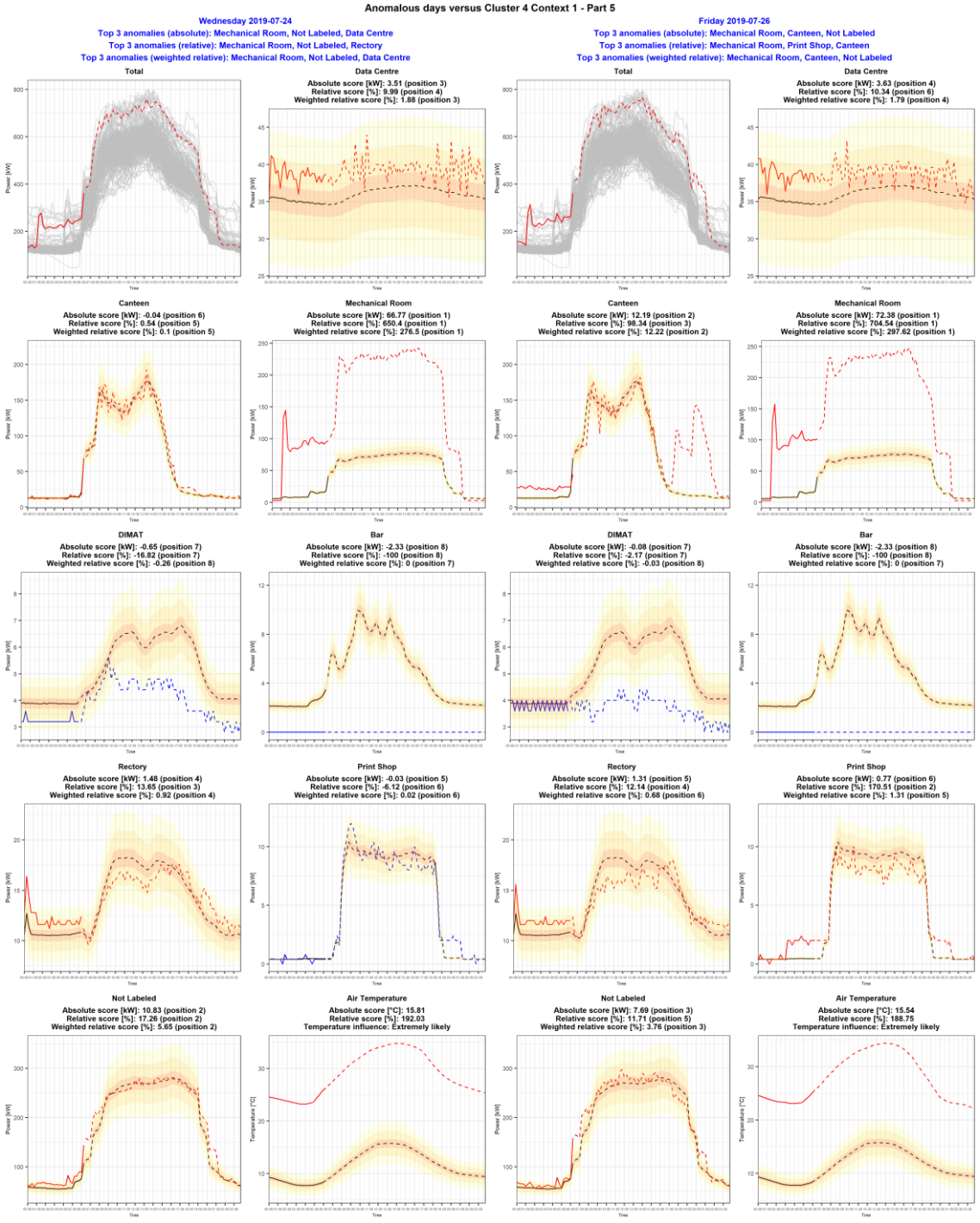

Figure A. 8 - Anomaly diagnosis for cluster number 4 + context number 1, part 5

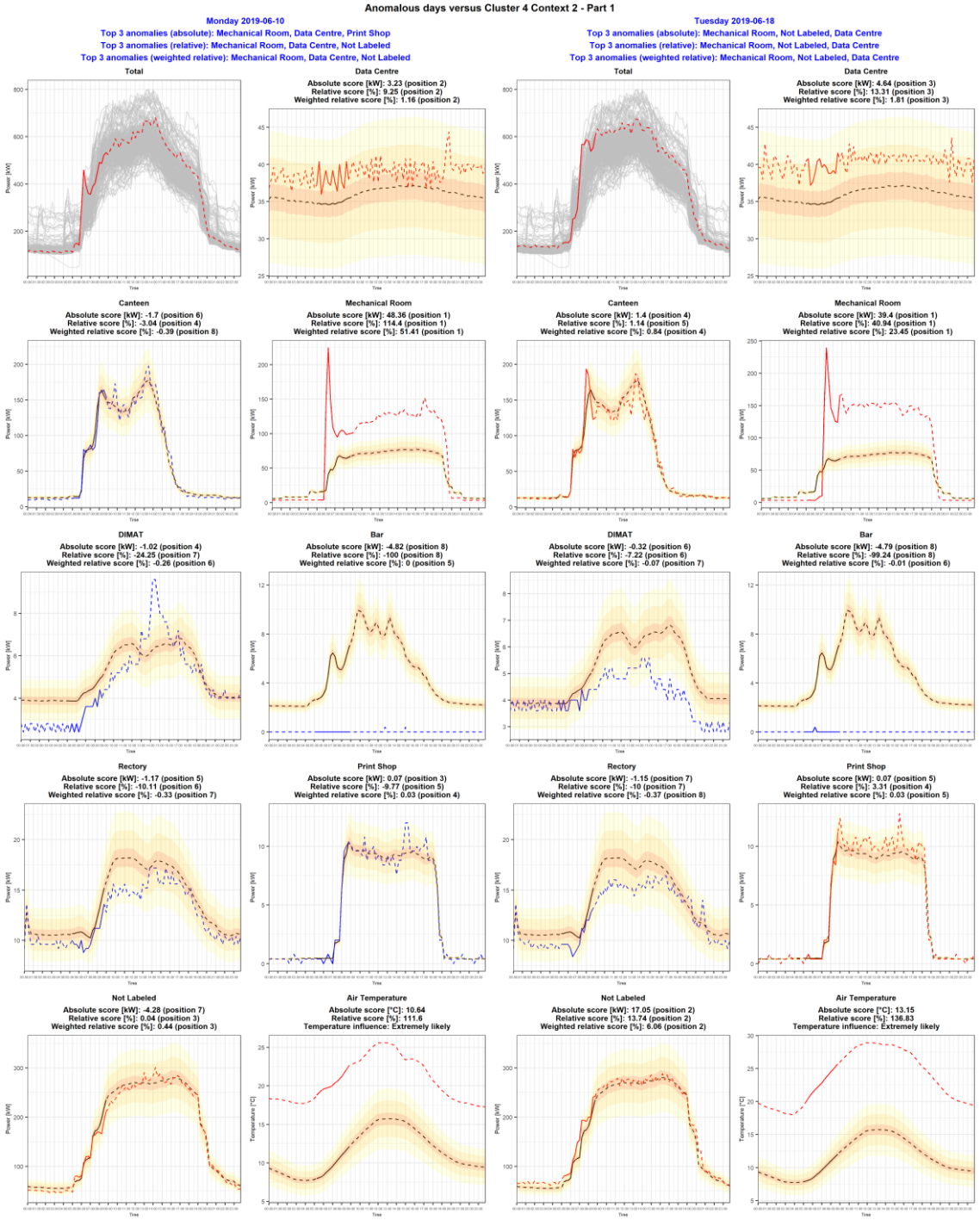

Figure A. 9 - Anomaly diagnosis for cluster number 4 + context number 2, part 1

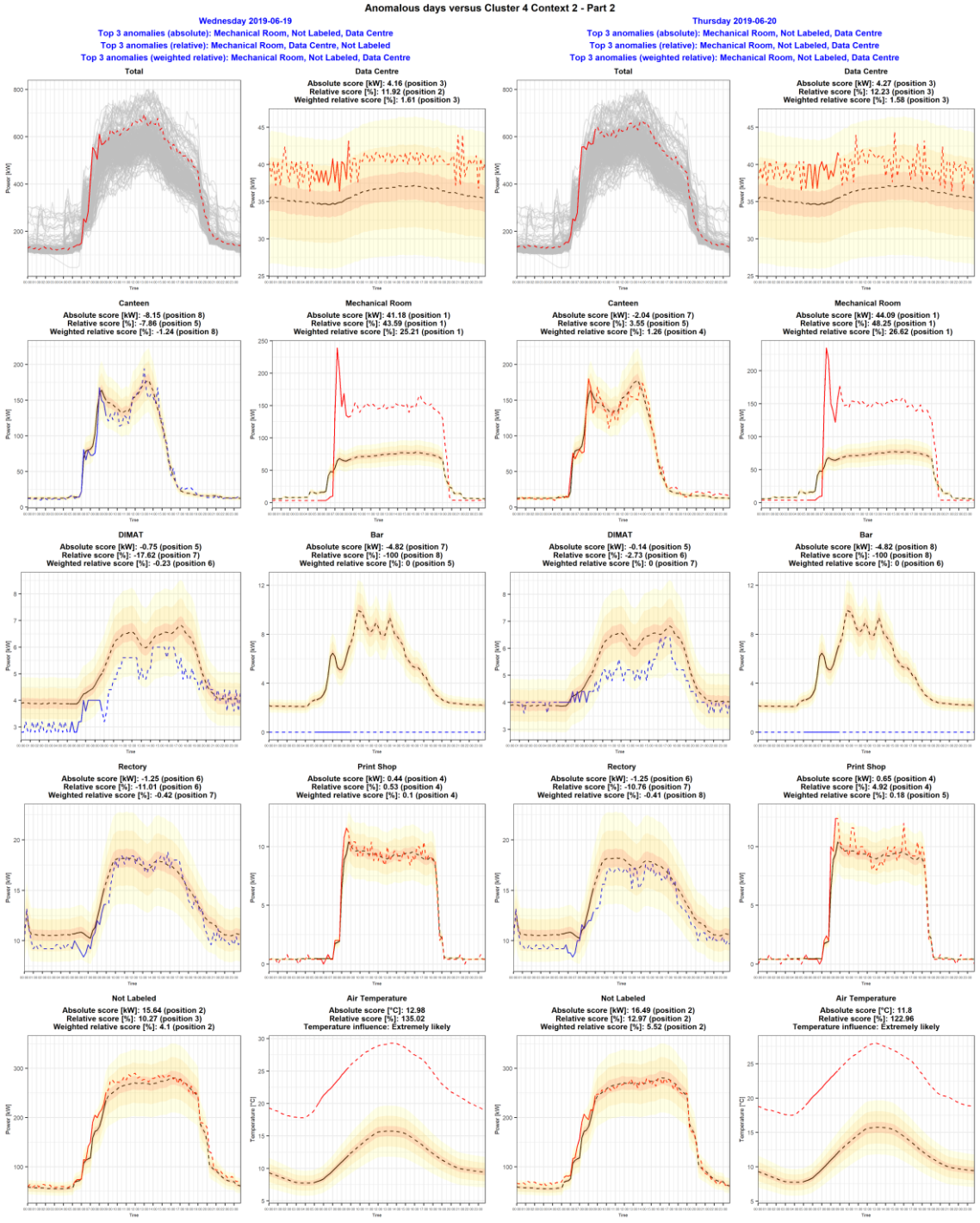

Figure A. 10 - Anomaly diagnosis for cluster number 4 + context number 2, part 2

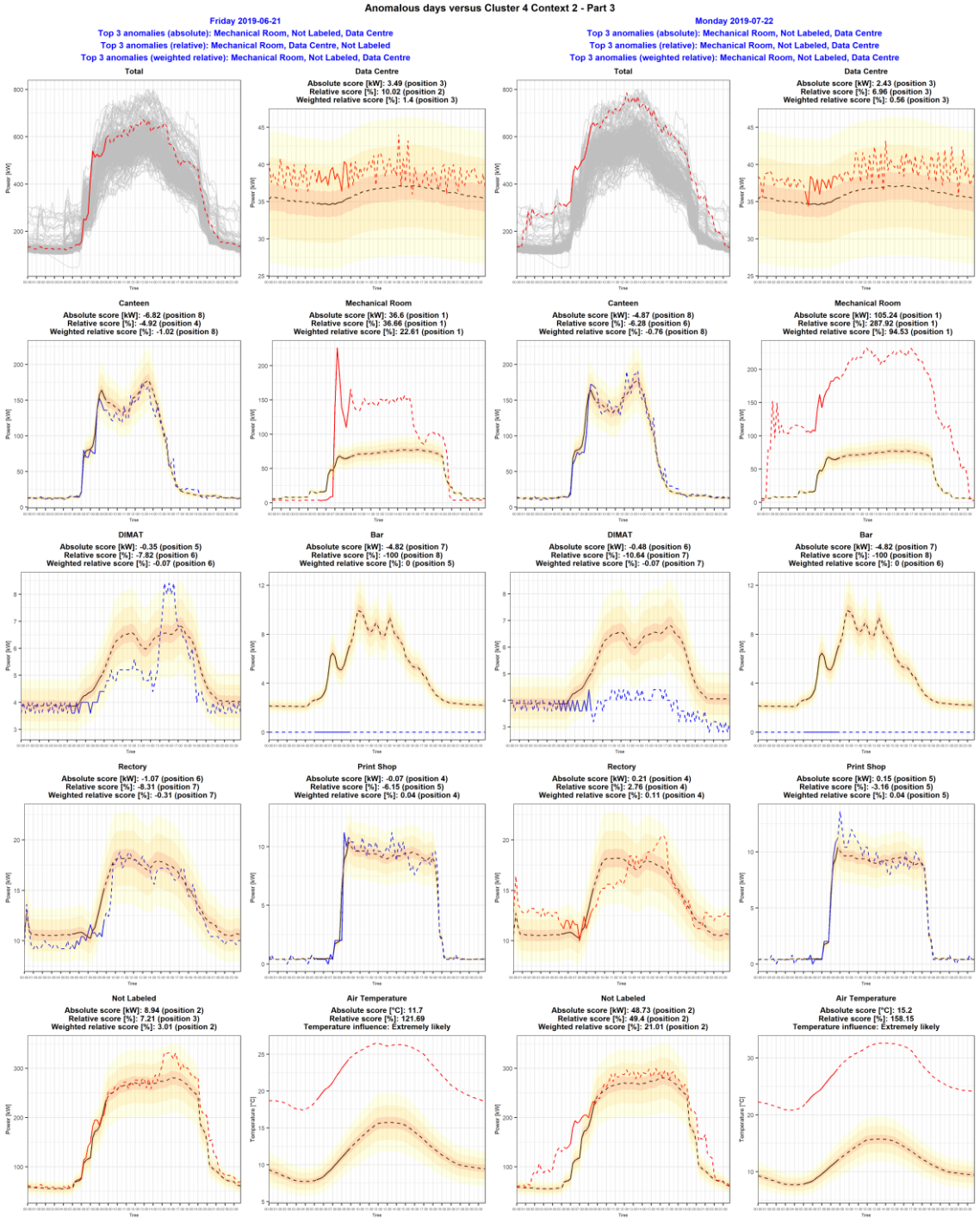

Figure A. 11 - Anomaly diagnosis for cluster number 4 + context number 2, part 3

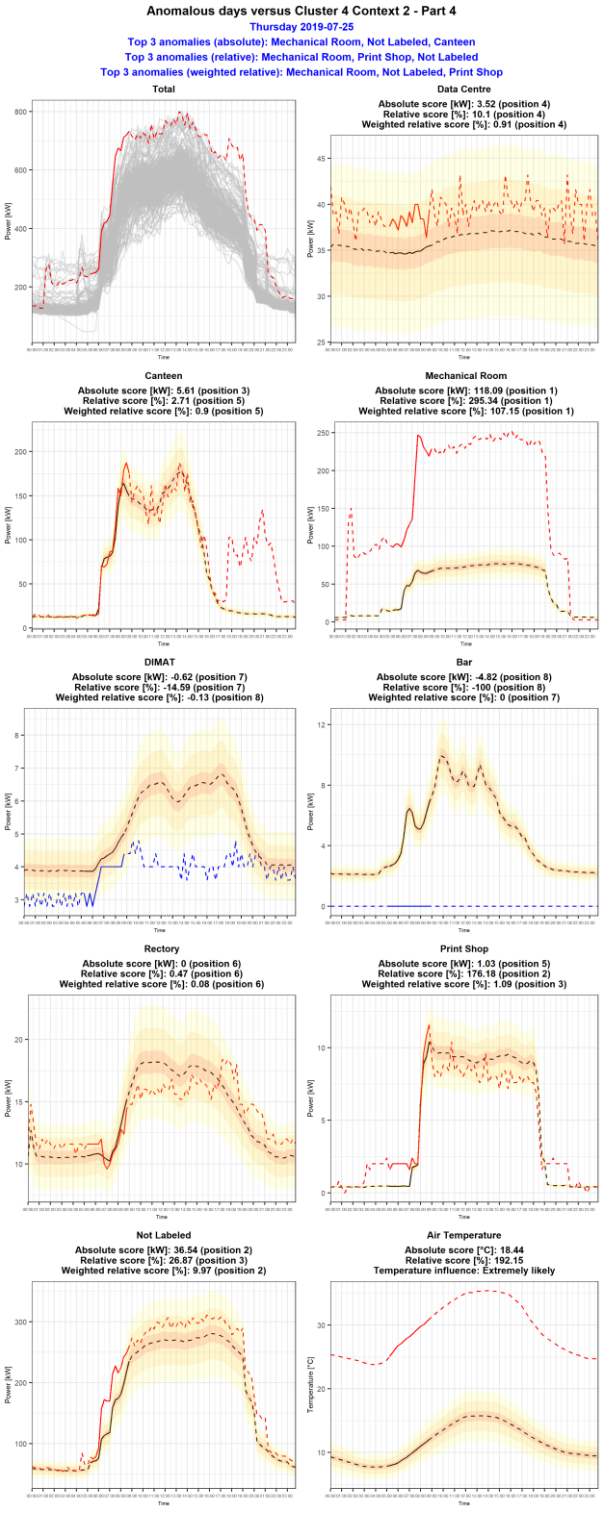

Figure A. 12 - Anomaly diagnosis for cluster number 4 + context number 2, part 4

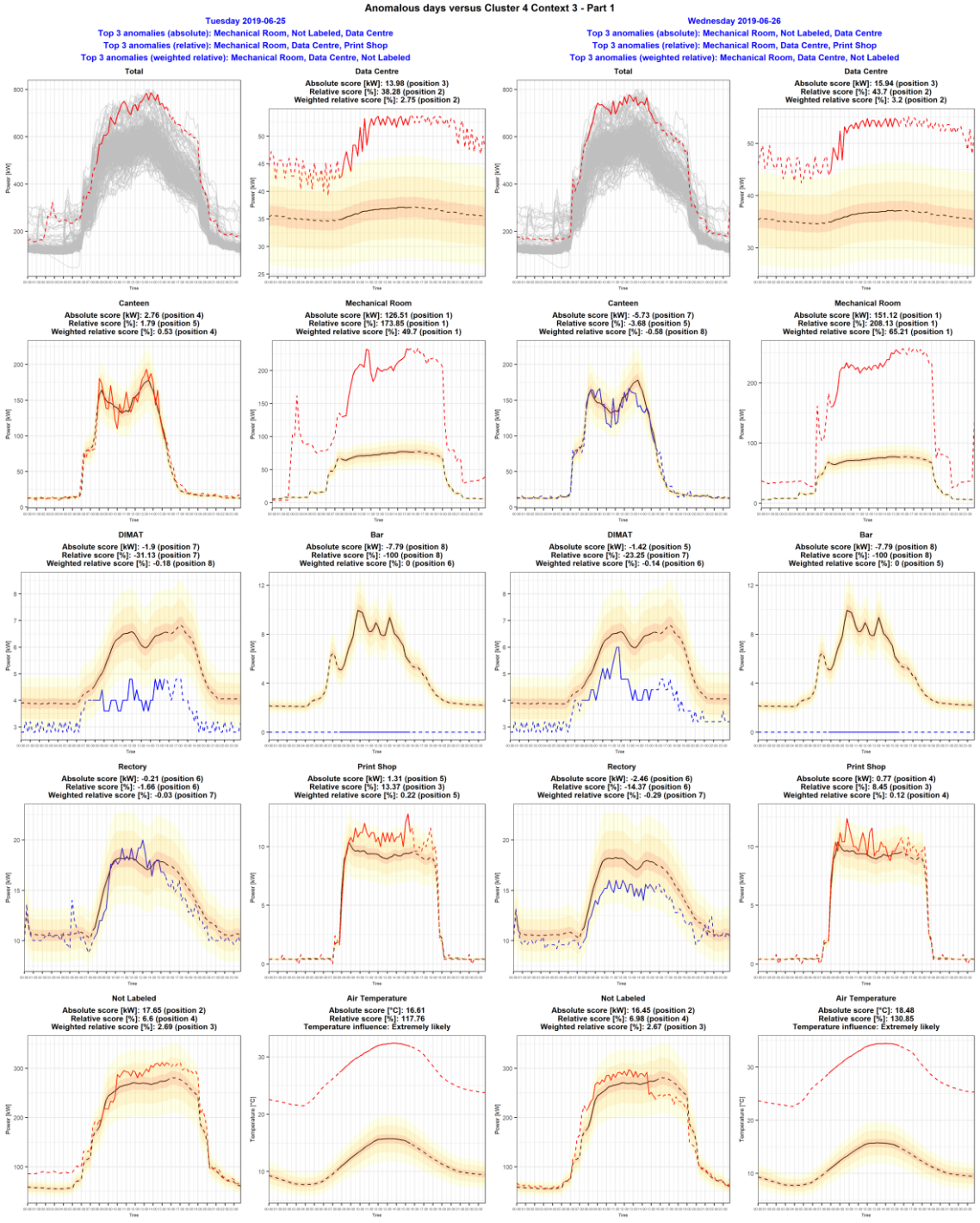

Figure A. 13 - Anomaly diagnosis for cluster number 4 + context number 3, part 1

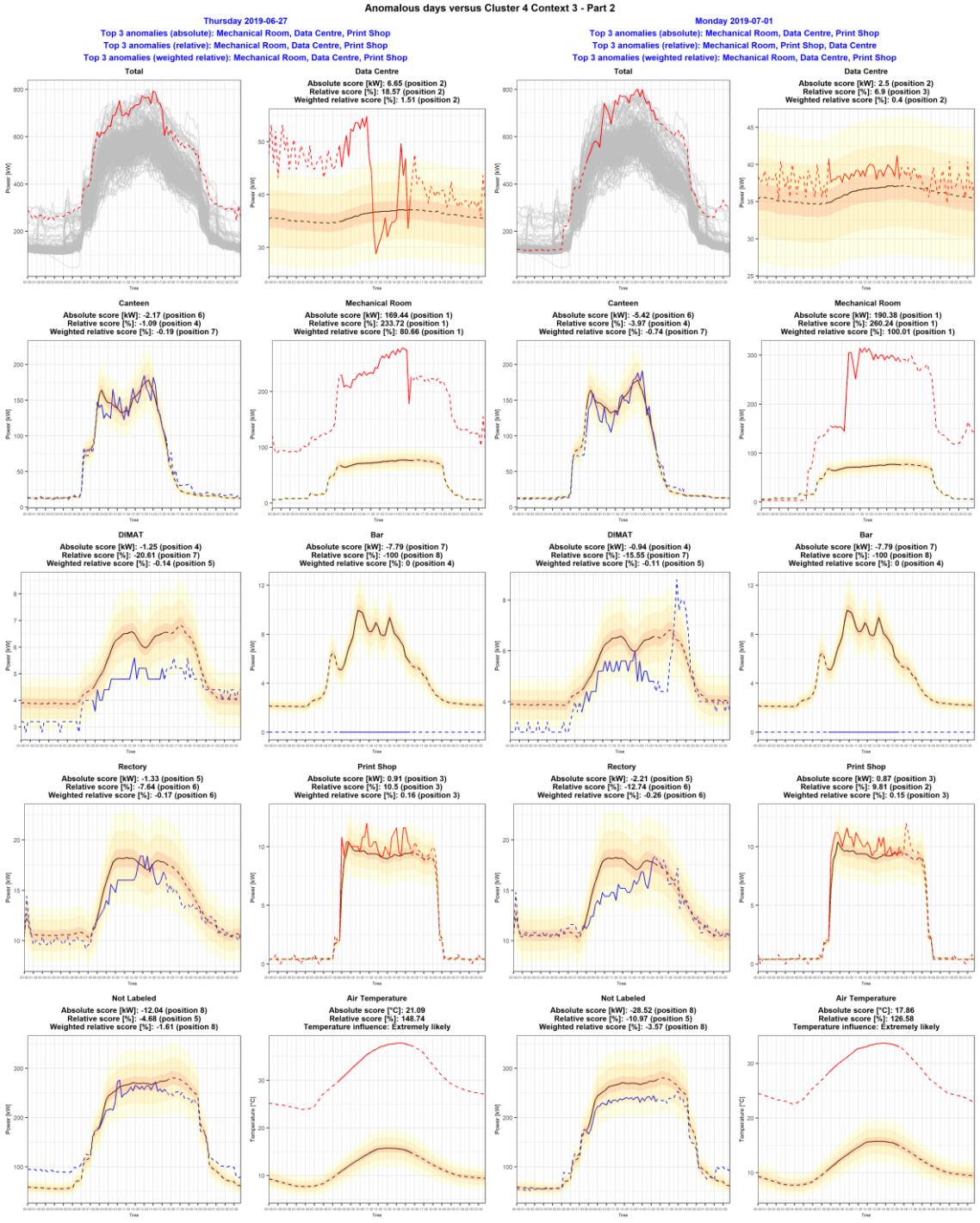

Figure A. 14 - Anomaly diagnosis for cluster number 4 + context number 3, part 2

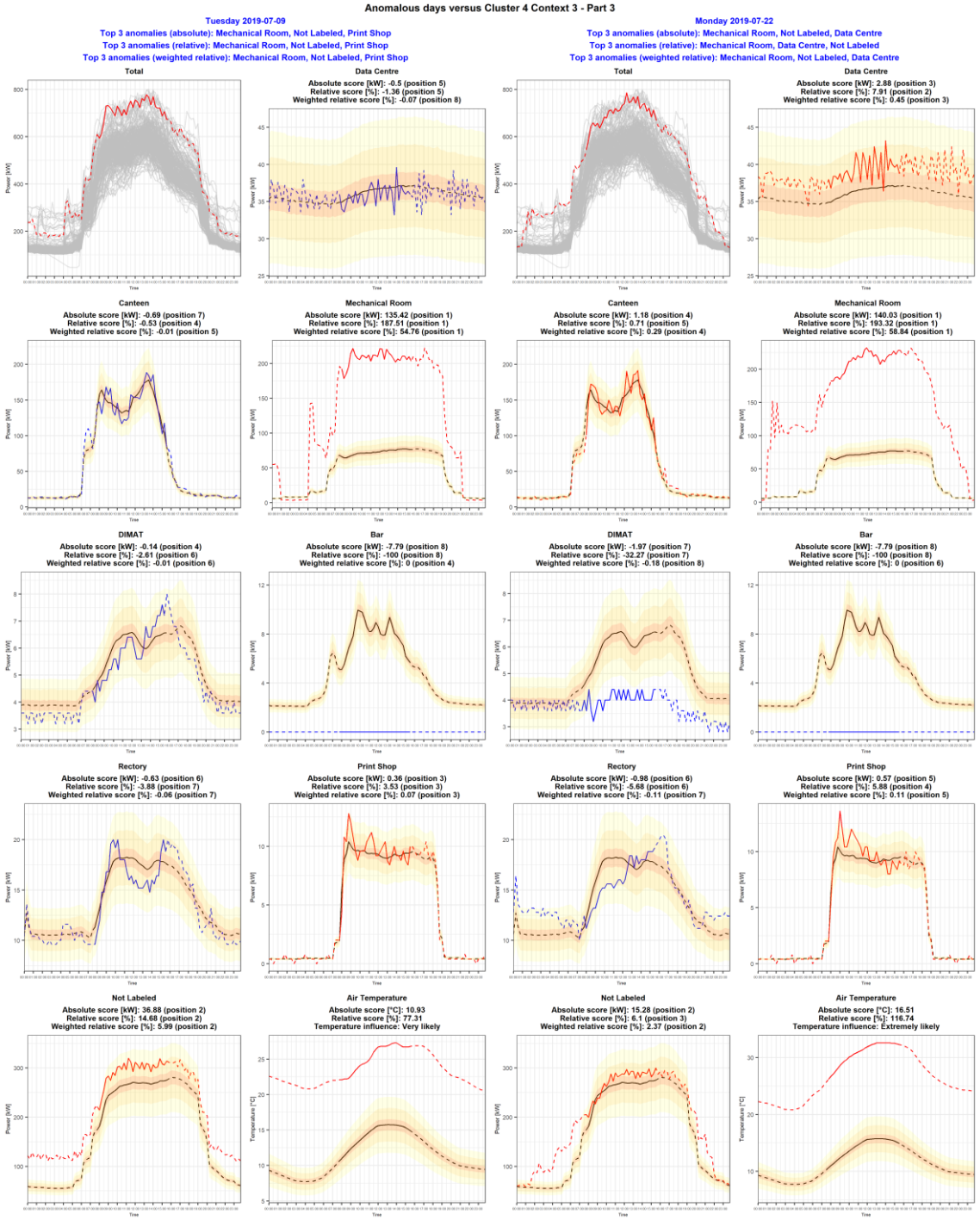

Figure A. 15 - Anomaly diagnosis for cluster number 4 + context number 3, part 3

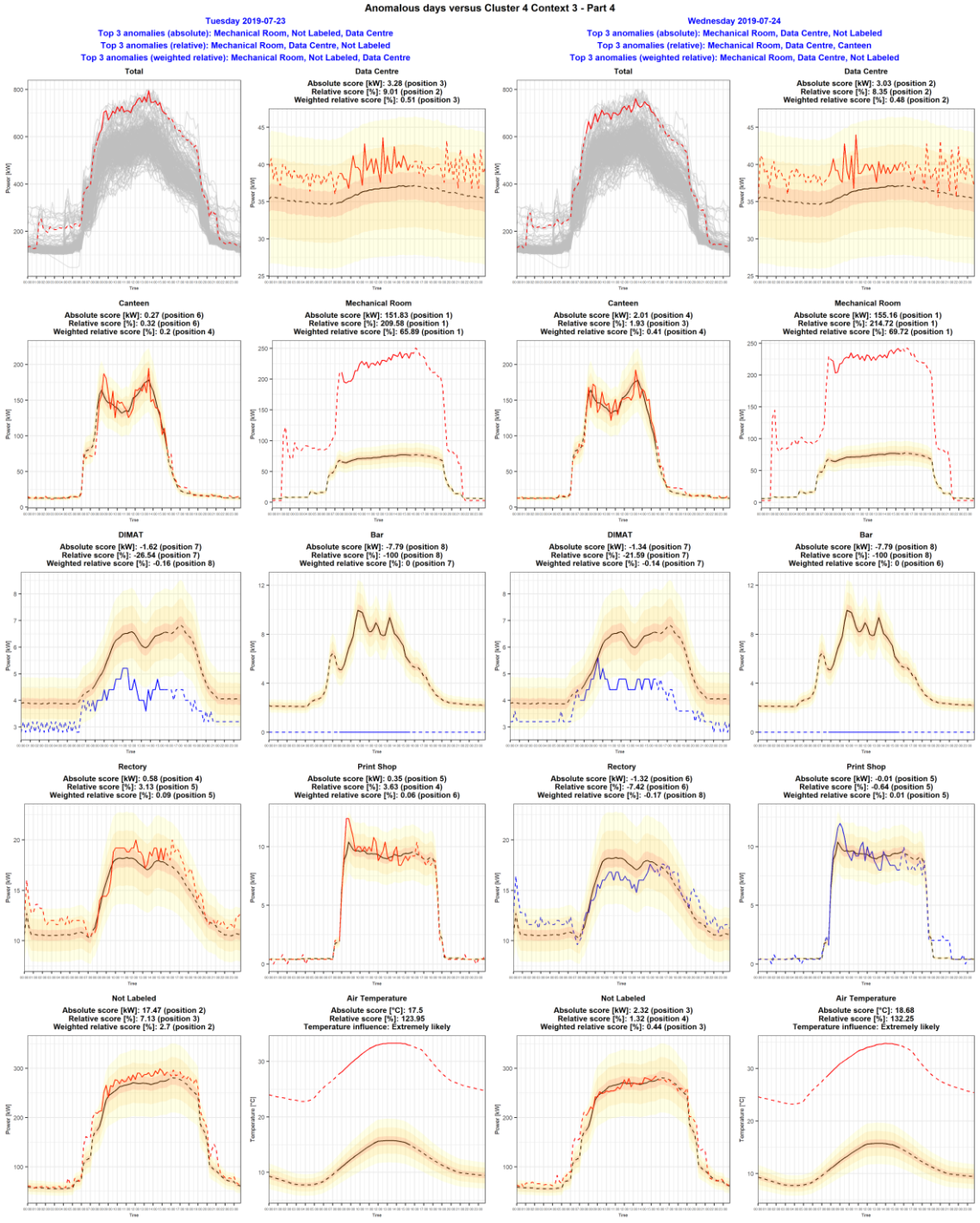

Figure A. 16 - Anomaly diagnosis for cluster number 4 + context number 3, part 4

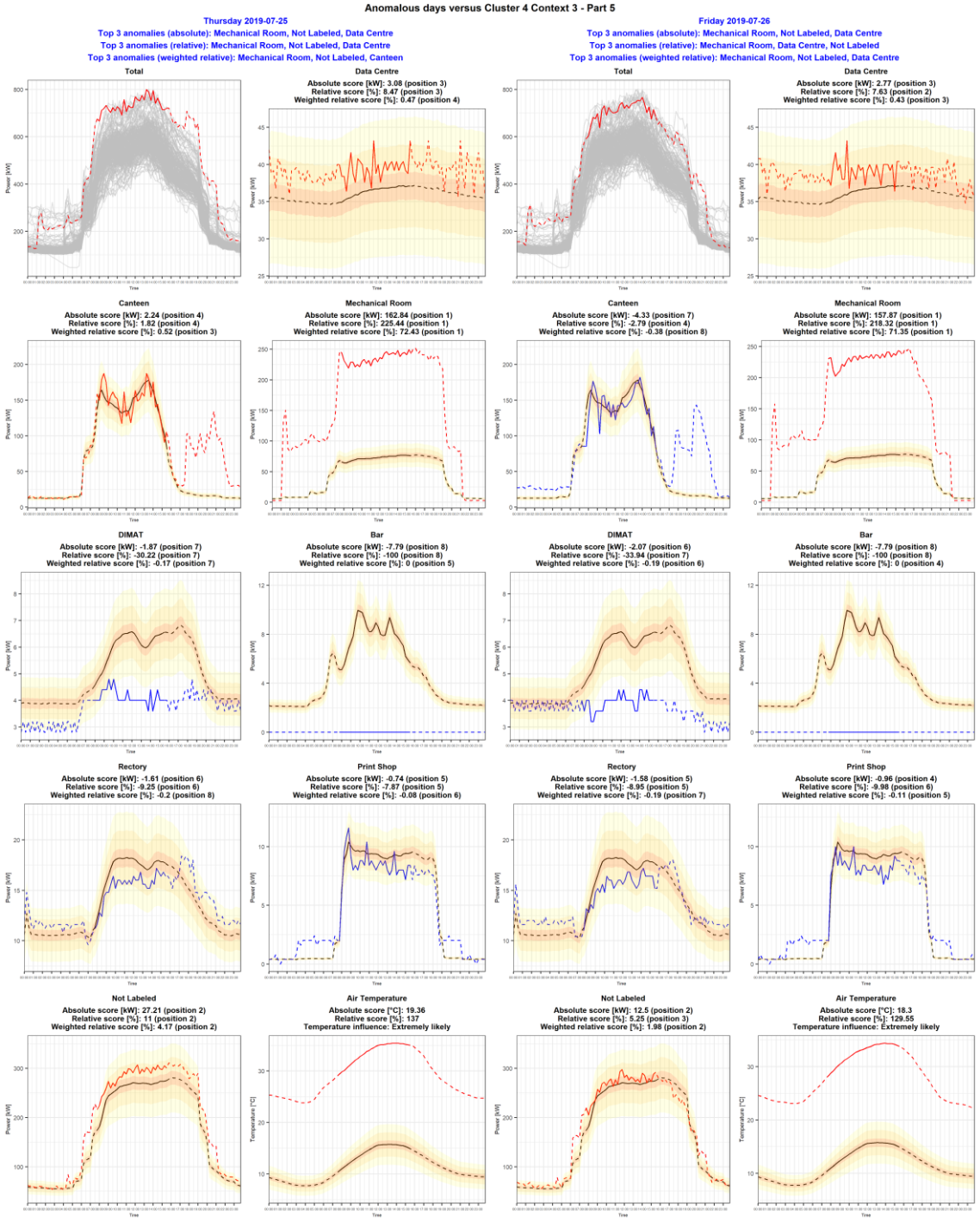

Figure A. 17 - Anomaly diagnosis for cluster number 4 + context number 3, part 5

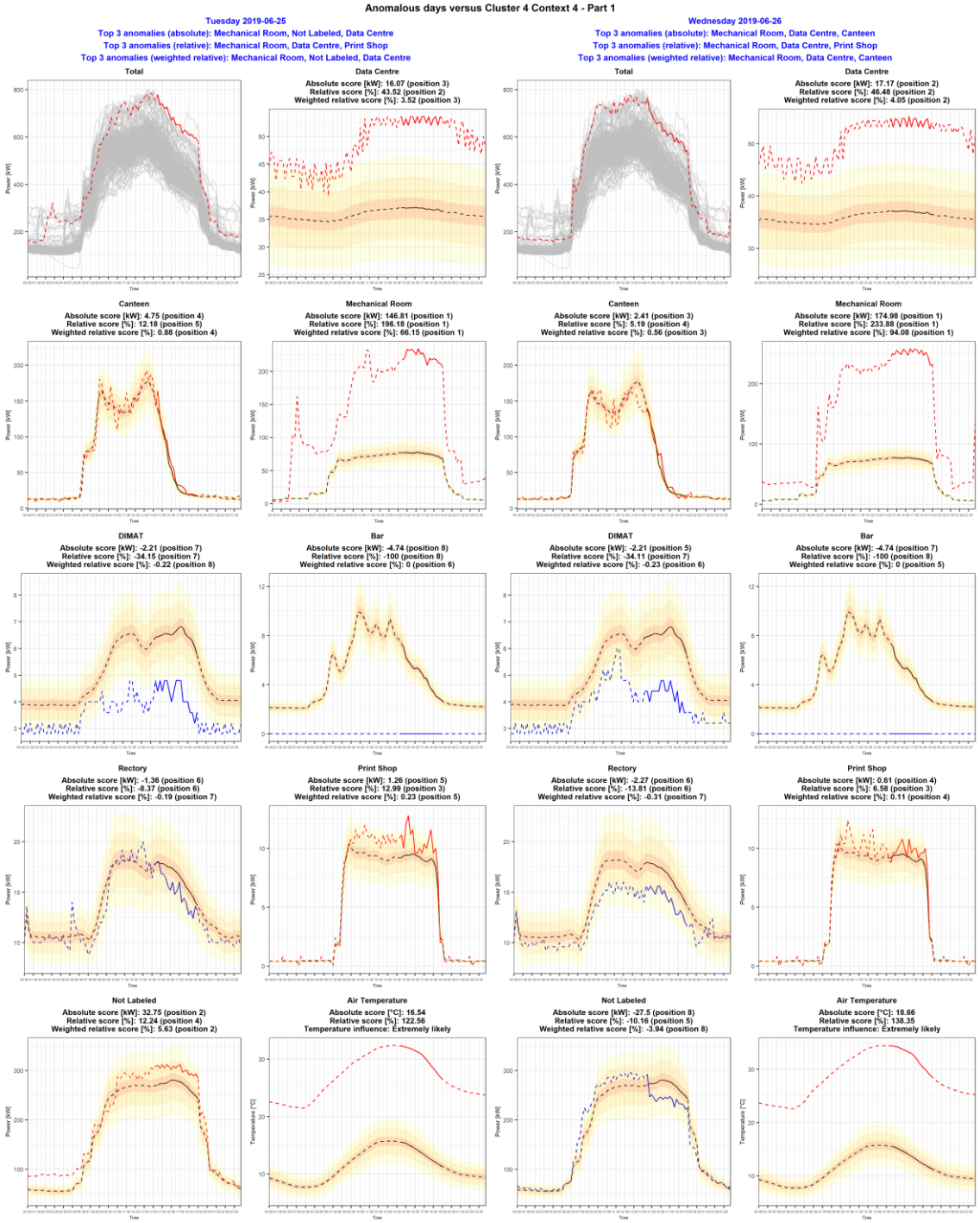

Figure A. 18 - Anomaly diagnosis for cluster number 4 + context number 4, part 1

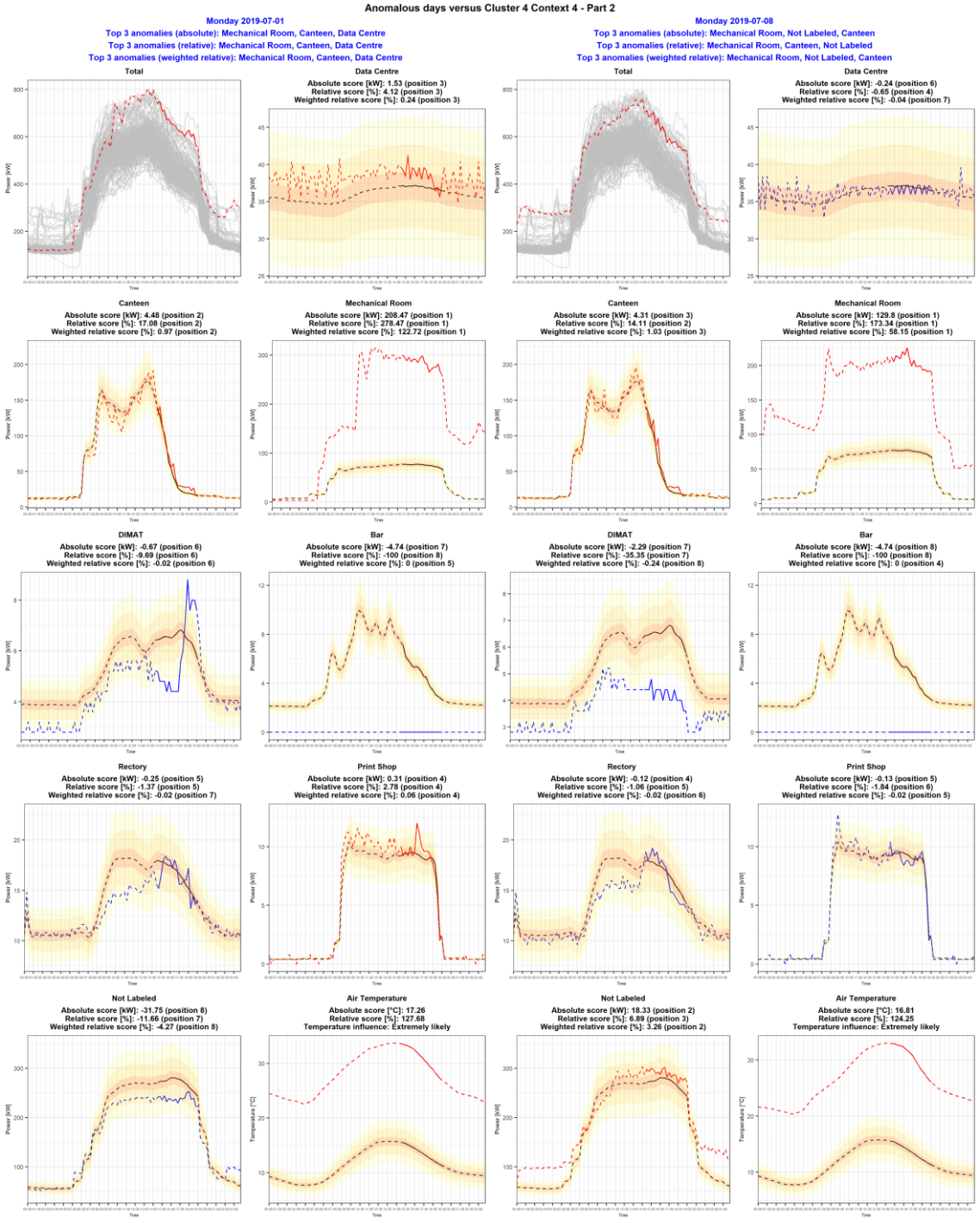

Figure A. 19 - Anomaly diagnosis for cluster number 4 + context number 4, part 2

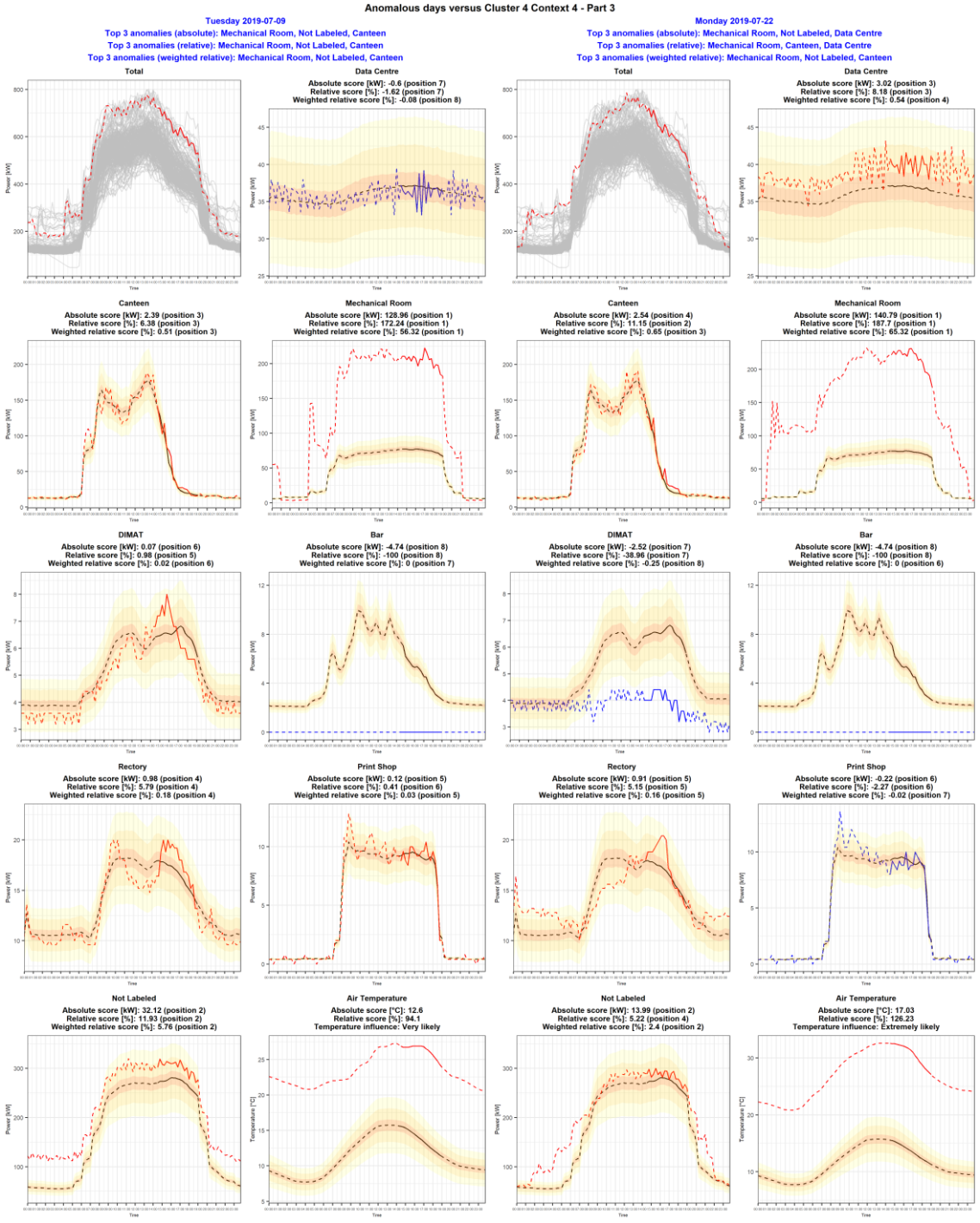

Figure A. 20 - Anomaly diagnosis for cluster number 4 + context number 4, part 3

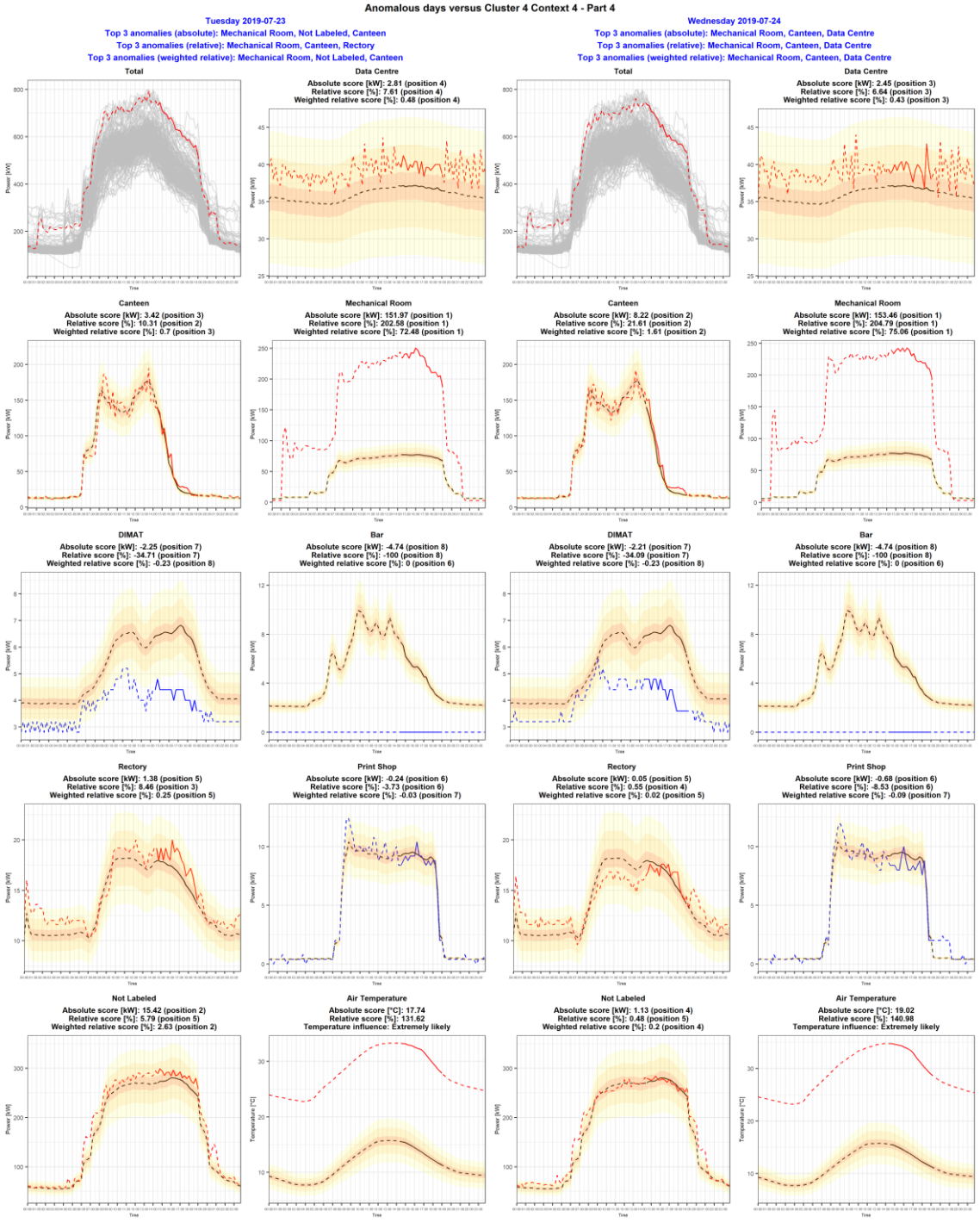

Figure A. 21 - Anomaly diagnosis for cluster number 4 + context number 4, part 4

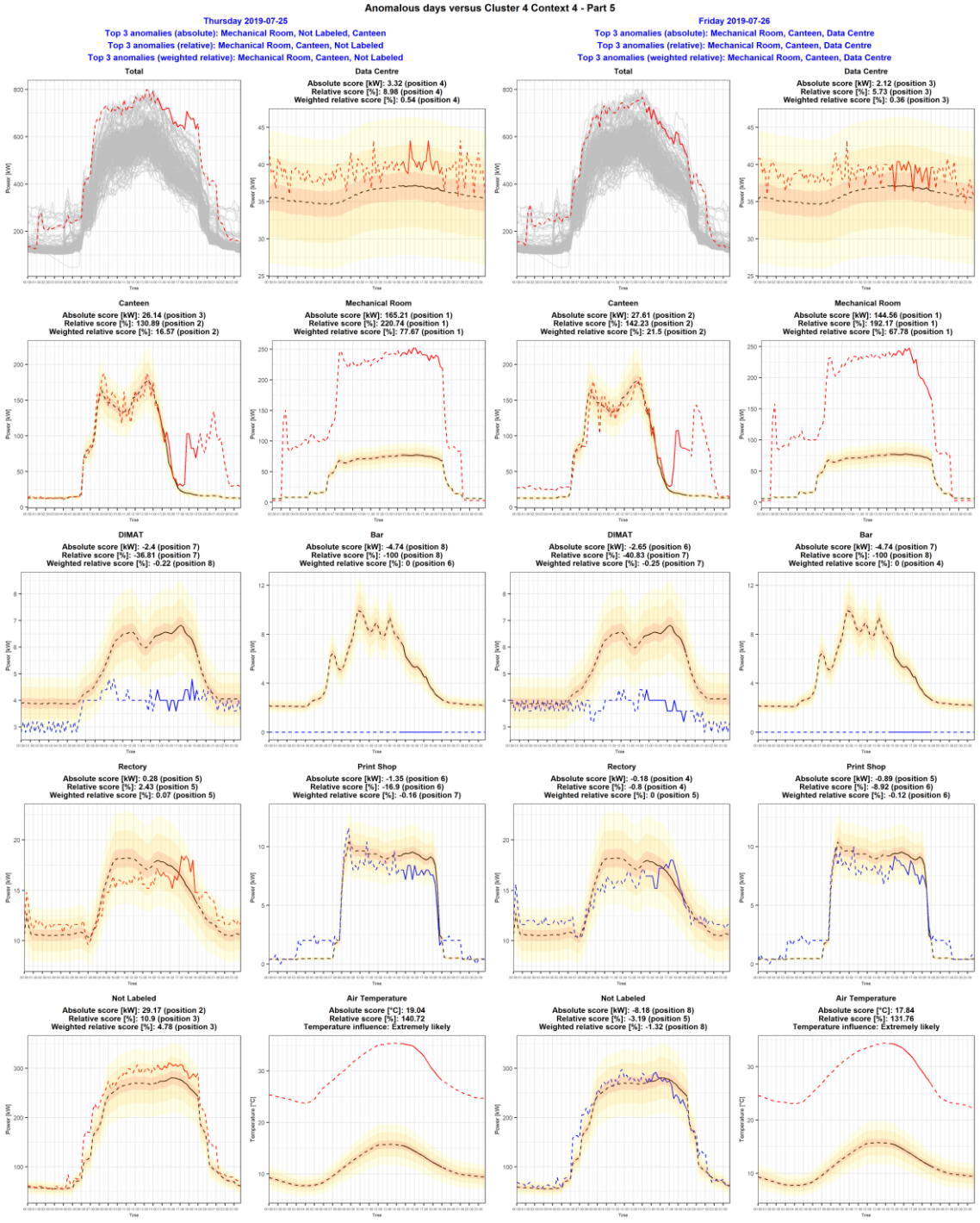

Figure A. 22 - Anomaly diagnosis for cluster number 4 + context number 4, part 5

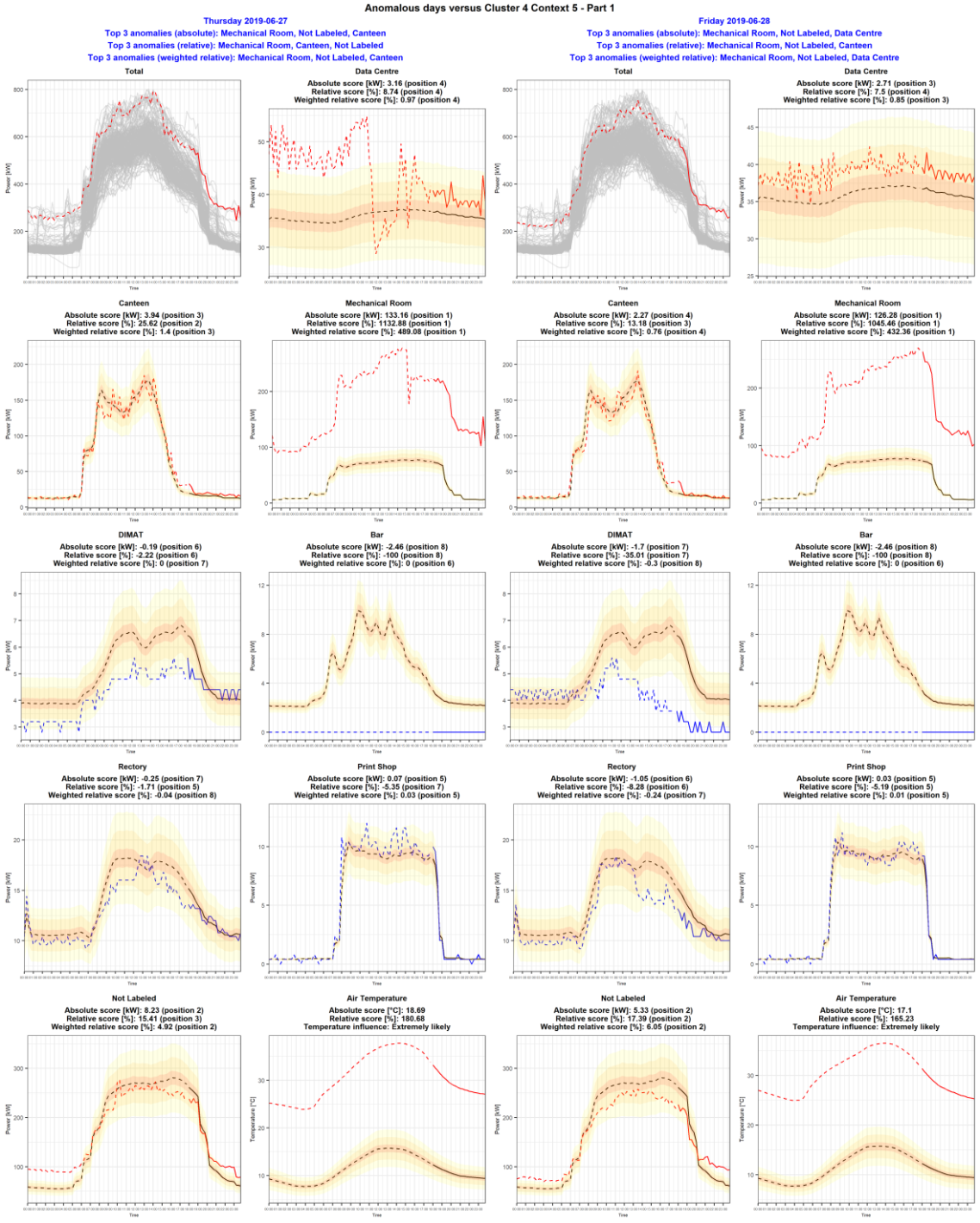

Figure A. 23 - Anomaly diagnosis for cluster number 4 + context number 5, part 1