Politecnico di Torino

Politecnico di Torino

Corso di Laurea in Ingegneria Energetica e Nucleare
Academic Year 2021/2022
March 2022

Master's Degree Thesis

Automated anomaly detection in energy consumption time series of buildings through pattern recognition techniques

Supervisor:
Prof. Alfonso Capozzoli, Phd
Co-Supervisor:
Dr. Marco Savino Piscitelli, Phd
Eng. Roberto Chiosa

Candidate:
Simone Vitale
ID: 267635

Politecnico
di Torino

Politecnico di Torino

ABSTRACT

Commercial buildings are significant consumers of electrical and thermal energy, therefore energy savings, improving energy efficiency, and reducing greenhouse gas emission are the purposes for building owners, operators, and stakeholders. On the other hand, energy analysts have to understand the energy consumption behavior by looking for changes in energy patterns that may imply device failures or anomalous behavior. This master's thesis deals with an energy data-mining approach that performs automated Anomaly Detection, through data analytics techniques called Matrix Profile (MP) and its extension Contextual Matrix Profile (CMP). This work aims to extract from large energy time-series data generated by sub-meters and smart sensors installed in Politecnico di Torino buildings, anomalous energy consumption patterns and to understand the root causes of the detected anomaly. The framework built up combines a hierarchical cluster algorithm, which helps to aggregate power consumption daily patterns, with MP and a final descriptive statistics outliers' analysis.

Contents

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Nowadays energy data analysts are faced with large amount of data due to the entrance in the buildings sector of IoT technologies. IoT stands for Internet of things, and it refers to concept of connecting embedded devices, computers, or smart sensors through wireless or wired network to the internet. These devices can be monitored, interact among themselves and finally they can also exchange data [1]. The Building Automations System (BAS) is a network of sensors, actuators, software and communication protocols, it could be thought as the brain of the building. In reference [2] the authors list the main tasks of BAS, it enables the storing of a large volume of energy data and the associated driving factors, the controlling of building system, the estimating of the energy saving after retrofit actions. In particular, it controls indoor temperature, humidity, ventilation and lighting conditions simplifying management operations, but as reported by reference [3]: ''can't answer questions like: How much energy is consumed at different times of the day? Does the economizer behave appropriately? What is the optimal air handling unit supply air temperature setpoint?'' To answer these questions, we need for Energy Management Information System (EMIS), a tool able to organize, present, visualize, analyze data that come from BAS, makes a Fault Detection and controls, supporting and improving the savings in buildings [3]. The Energy Information System (EIS) as illustrated in *Figure 1-1* is a sub-system of EMIS, its task is essentially monitoring data at meter-level and carrying out automated analysis like predictive energy analysis. The results of these analysis are usually provided in dashboards. On the other hand, when we refer to Fault Detection and Diagnostic System (FDD), we

relate to a software able to detect faults, incorrect occupant behavior and anomalies of building system with rule-based or model-based diagnostic logics.

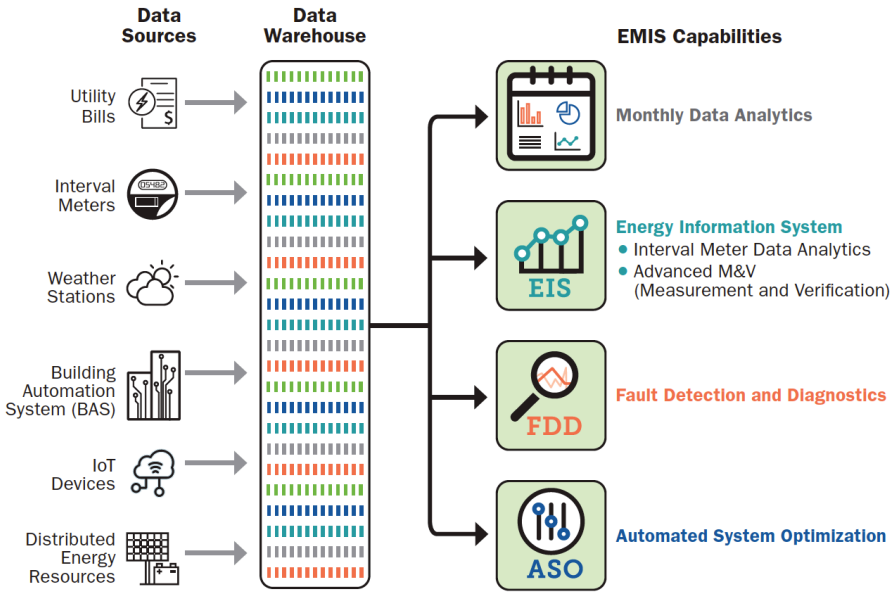


Figure 1-1: Data inputs and key capabilities of EMIS. Reprinted from [3]

The last component Automated System Optimization manages BAS settings continuously in order to improve HVAC system energy consumption while ensuring thermal comfort. With this amount of data, buildings are becoming information-intensive, so, the needs for analyzing large datasets with Data Mining powerful techniques is increasing [4].

1.1 RELATED WORKS

Energy inefficiencies Detection and Anomalies Detection (AD) are Data Mining (DM) application related. This way of discovering hidden knowledge from data is so useful in order to save energy and to improve operational system performance. The scientific paper contains hundreds of methodologies for AD, the most traditional ones exploit physical principle-based methods or statistical analysis which results in a poor performance if the quantity of data examples is too large. Instead, a rising interest to multidisciplinary subject as DM is opening the doors to a promising solution, with brilliant performance results. As reported by [4]: " DM is multi-disciplinary subject, integrating techniques from statistics, machine learning, artificial intelligence, high performance computing etc." DM techniques are mainly of two kinds: supervised and unsupervised. The two categories have strengths and weakness. Supervised techniques are more suitable for modeling complicated relationship but is needed the availability of high-quality training data. By contrast, unsupervised techniques explore structure and correlations among data without the need for training dataset, inputs, outputs variables and without prior information. Among the unsupervised techniques the Authors in [4] mention the Association rule Mining Algorithm. In fact, if an observation meets the antecedent of a frequent association but don't meet the consequence than an anomaly is found. Also, statistical test methods are employed for AD, an example is The Generalized Studentized Deviate Test (GESD). The most of Frameworks in scientific literature apply the GESD method on energy profile computed statistics in order to extract abnormal energy consumption pattern from large Time Series. Capozzoli et al. in Ref. [2] has been introduced a predictive-based methodology for automated detection of

anomalous patterns supported by Symbolic Aggregate Approximation (SAX). This methodology is a multistep one and in addition to SAX transformation, it leverages on classification and regression tree. After the outlier removal stage has been performed, the SAX application helps to reduce dimension of the initial Time Series, without losing key information. The lengths of the non-overlapping windows on time axis are not equal, to better approximate the initial Time Series and the not-equal probability of the symbols encoding each approximated constant segment has also been taken into account (these are Novelties in scientific literature). In the meantime, CART is employed to segment daily period in time windows.

2 METHODS

2.1 MATRIX PROFILE

In time series analysis, most of the time, we are interested in finding anomalies; one method to do this is performing similarity join. Initially, we could compare snippets of the time series against itself by computing the distance between each pair of snippets. Implementing such an algorithm using nested loop is a relatively simple thing but it requires a computational effort even with a short time series. The Matrix Profile technique carries out this task reducing drastically the computation time; it was introduced by Eamon Keogh at University of California at Riverside and Abdullah Mueen at University of New Mexico in 2016 [5]. Essentially, The Matrix Profile is a data structure that annotates time series; it has two main components, the Distance Profile, and the Profile Index. The algorithms used to compute these components involves the use of sliding windows of length m throughout the time series. First, the distances of the windowed sub-sequence against the time series are computed, after that is needed to set an exclusion zone to prevent trivial matches. The exclusion zone helps the algorithms to ignore m/2 indices both before and after the windows index when computing the minimum distance and the nearest-neighbour index. At the end the Distance Profile is updated with minimal values and the Distance Profile with the first nearest-neighbour index.

2.1.1 DEFINITIONS AND NOTATIONS

To figure out what is really Matrix Profile, is helpful to focus on some definitions given by the authors in [6]. The following definitions explains in detail the components of the algorithms.

''Definition 1. A time series T is a sequence of real-valued numbers t_i: T = $t_1, t_2, \dots, t_n$ where n is the length of T"; as mentioned in[6].

''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 $'i'$. Formally, $T_{i,m} = t_i, t_{i+1}, \dots, t_{i+m-1}$, where $1 \leq i \leq n - m + 1$, Figure 2-1 "; as mentioned in[6].

Figure 2-1: Time Series and windowed sub-sequence of length m. Adapted from [5]

''Definition 3. An all-subsequences set 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: $A = T_{1,m}, T_{2,m}, \dots, T_{n-m+1,m}$, where m is a user-defined subsequence length. We use $A[i]$ to denote $T_{i,m}$. Figure 2-2 "; as mentioned in[6].

"Definition 4. A distance profile D is a vector of the Euclidean distances between a given query and each subsequence in an all-subsequences set. Figure 2-3 "; as mentioned in[6].

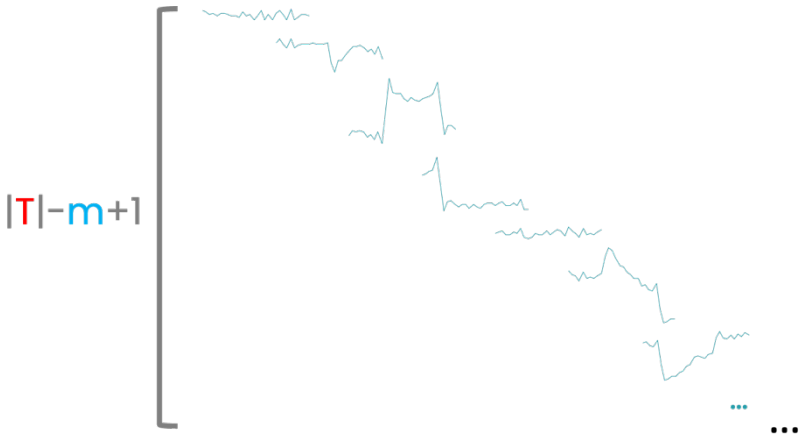


Figure 2-2: all-subsequence ordered set. Adapted from [5]

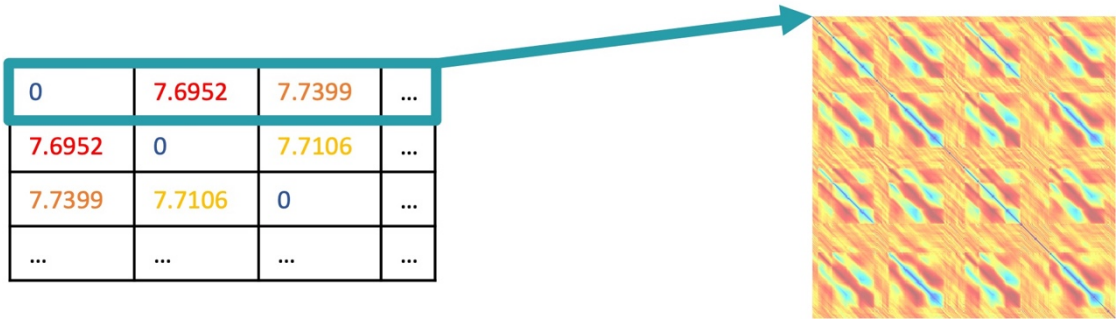


Figure 2-3:Distance Profile on the left and Distance Matrix on the right, where blue is more similar subsequences and red is more dissimilar subsequences. Adapted from [5]

The Authors in [6] take for granted that the distance is measured with the Euclidean distance between the z-normalized subsequences (the subsequences have a mean of zero, and a standard deviation of one). In this sense, Distance Profile annotates the time series T, it records a set of distances. If the query and all-subsequences set belong to the same time series, the distance profile must

be zero at the location of the query, and close to zero just before and just after. Such matches are called trivial matches and are avoided by ignoring an exclusion zone of m/2 before and after the location of the query.

''Definition 5. 1NN-join function given two all-subsequence sets A and B and two subsequences $A[i]$ and $B[i]$, a 1NN-join function $\theta_{1nn}(A[i], B[j])$ is a Boolean function which returns "true" only if $B[j]$ is the nearest neighbor of $A[i]$ in the set B "; as mentioned in[6].

''Definition 6. Similarity join set: given all-subsequence sets A and B, a similarity join set J_{AB} of A and B is a set containing pairs of each subsequence in A with its nearest neighbor in $B : J_{AB} = \{\langle A[i], B[j]\rangle | \ \theta_{1nn}(\ A[i], B[j])\}$. We denote this formally as $J_{AB} = A \bowtie_{\theta\ 1nn} B$, Figure 2-4 "; as mentioned in[6] .

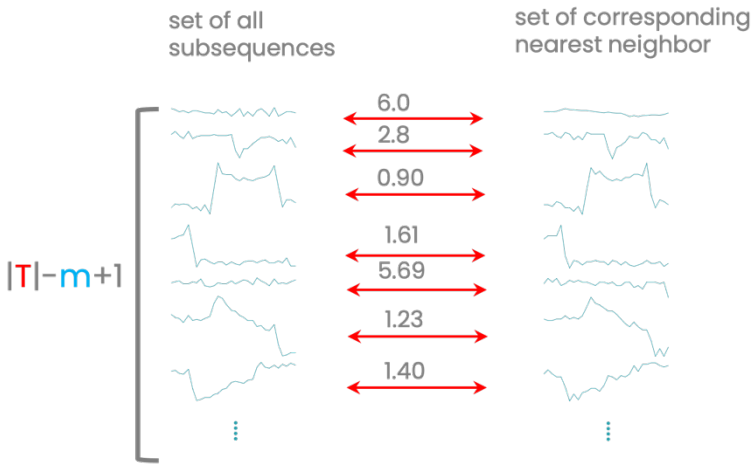


Figure 2-4: Similarity join set. Adapted from [5]

''Definition 7. A matrix profile (or just profile) P_{AB} is a vector of the Euclidean distances between each pair in J_{AB} where $P_{AB}[i]$ contains the distance between $A[i]$ and its nearest neighbor in B''; as mentioned in[6].

''Definition 8. A self-similarity join set J_{AA} is a result of similarity join of the set A with itself. We denote this formally as $J_{AA} = A_{\theta 1nn}A$. We denote the corresponding matrix profile or self-similarity join profile as P_{AA}'',Figure 2-5 ; as mentioned in[6].

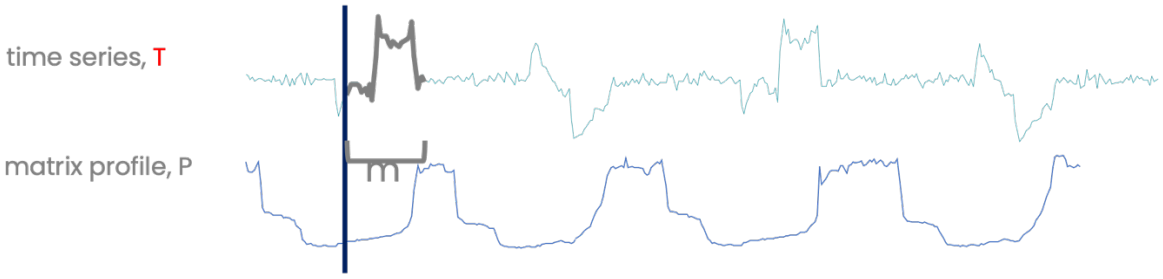


Figure 2-5: Time Series, Time Window and Matrix Profile. Adapted from [5]

The matrix profile value at location i is the distance between T_i and its nearest neighbor, wherever it is. The highest point on the profile corresponds to the time series discord, the lowest points correspond to the locations of the best time series motif pair and the variance can be seen as a measure of the T 's complexity.

''Definition 9. A matrix profile index I_{AB} of a similarity join set J_{AB} is a vector of integers where $I_{AB}[i] = j \ \ if \ \{A[i], B[j]\} \in J_{AB}$'', Figure 2-6; as mentioned in[6].

The nearest neighbor information is contained in the matrix profile index, in this way we can retrieve the nearest neighbor of $A[i]$ by accessing the i th element. It should be considered that the function which computes the similarity join set of two input time series is not symmetric, therefore, $J_{AB} = J_{BA}$, $P_{AB} = P_{BA}$, $I_{AB} = I_{BA}$ [6].

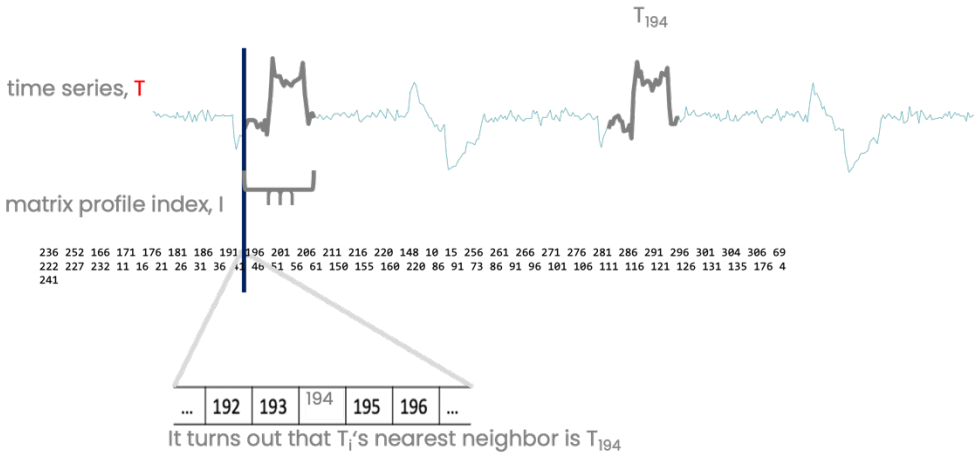


Figure 2-6: Time Series and Matrix Profile Index. Adapted from [5]

2.1.2 GUIDED MOTIF SEARCH

In some cases, there is a need to include in the analysis some domain knowledge. Annotation Vectors are sorted series of values in the range [0, 1] which gives us the relevance of a motif at that index. A 1 in the AV means that any motif starting at that index is important and should be preserved instead a 0 means that the motif can be ignored [7]. As we let it be understood before, the Annotation Vector is a craftly way to ignore insignificant patterns in our dataset. The Annotation Vector introduces the concept of Guided Motif Search [7]; if the data analyst is aware of Motifs in a Time Series, he could award these parts of Time Series adding 1 into Annotation Vector Series, supporting in this way the Motifs Search. But also, Guided Motif Search is the process of penalizing certain portions (undesirable) of z-normalized Euclidean distance Time Series, during Motif Search workflow. If it is known the kind of pattern we are looking for, we can use the Annotation Vector to transform the Matrix Profile, performing successively Motif/Discord discovery. The

new Matrix Profile, Corrected Matrix Profile, which can be used in order to identify desirable motifs or discords, is supplied in the following formula:

$$CMP[i] = MP[i] + (1 - AV[i]) * \max(MP) \quad (2\text{-}1)$$

Where:

- CMP is the Corrected Matrix Profile.
- MP is the original Matrix Profile.
- AV is the Annotation Vector.
- max (MP) is the maximum value of the original Matrix Profile.

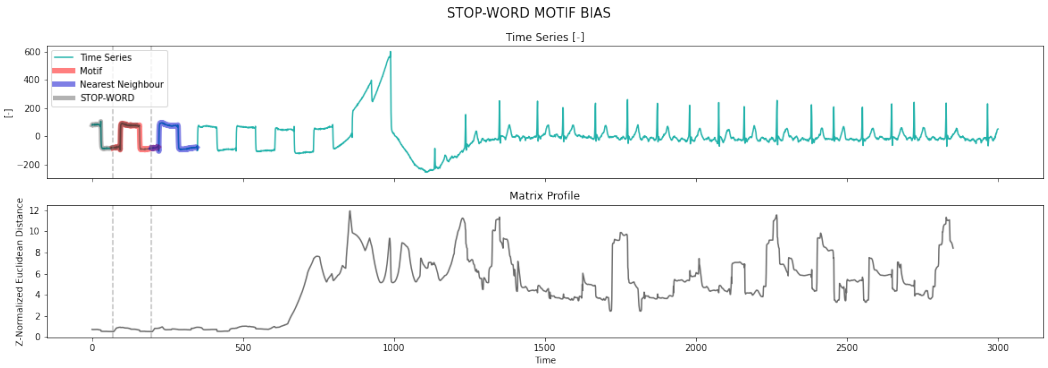


Figure 2-7: Stop-Word Motif Bias. Time Series, Motif, nearest Neighbour (top); Matrix Profile(bottom). Adapted from [7].

Essentially, this formula brings to new Matrix Profile by shifting the undesirable distances towards the maximum of the old Matrix Profile, max (MP) and thereby removing those corresponding subsequences from the pool of potential motifs [7] . In the following sections are reported the main Annotation Vector techniques, developed by the Authors in [7] which have been applied in several research fields. From different domains come from bias function which guides the Motif search. Sometimes, datasets contain frequents patterns which are not useful for

further analysis and overcome more interesting ones. With the term Stop-Word we refer to those words which don't carry meaning or information inside a sentence, but which is plenty of. By analogy with text analytics, the goal is finding a 'bias' function to avoid meaningless or trivial words in this case represented by not-interesting, repeated patterns [7].

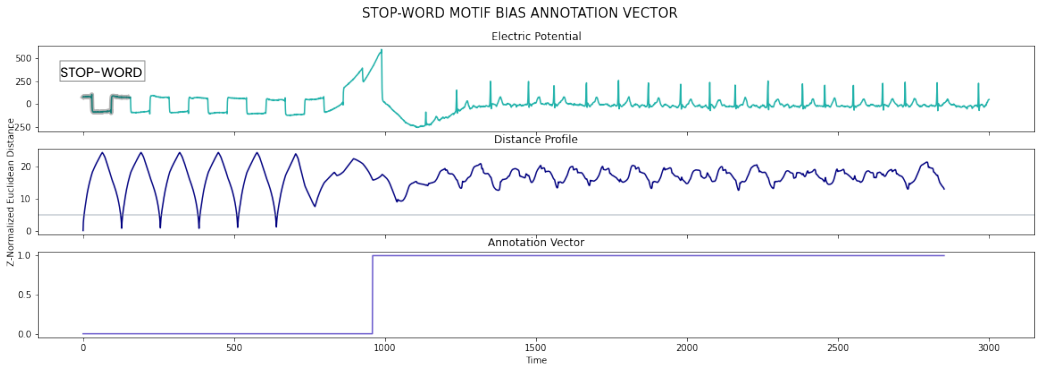


Figure 2-8: Time Series(top); Distance Profile(middle); Annotation Vector(bottom). Adapted from[7]

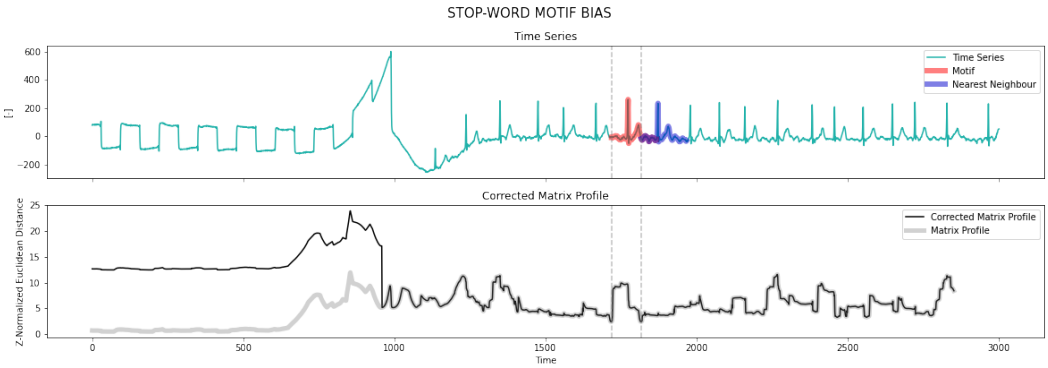


Figure 2-9: Time Series, New Motif, New Nearest Neighbour (top); Corrected Matrix Profile (bottom). Adapted from [7]

To give back an AV, we first calculate the Distance Matrix, a measure of similarity between stop-word and the other subsequences, Figure 2-8 (middle). Then a threshold could be set, in order to establish which subsequence likens to motif

stop-word one. The last step generates the AV vector in this way: all the data between '$3m$' before and '$3m$' after the stop-word motif index meet 1 in AV vector. If in our dataset there are "complex" portions (having many peaks and many valleys, regions with spikes), it is very likely that the Euclidean distances among them is greater than the distances among "simple" ones (in brief, "Euclidean Distance has a bias toward simple profile" [7]). The effect is that it will be difficult discovering "complicated" top-K motifs since they will be overwhelmed by "simple" ones. We can define a complex measure with the following formula (2-2) and generate a Complexity Vector to construct an Annotation Vector that will award motifs in complex regions. The following pseudocode lines deals with the stages from Complex Vector to Simplicity-Bias AV.

Table 2-1: Simplicity-Bias pseudo-code

```
1 annotation vector = complexity vector
2 annotation vector = annotation vector - min (annotation vector)
3 annotation vector = annotation vector / max (annotation vector)
```

$$CE(Q) = \sqrt{\sum_{i=1}^{m-1} (q_i - q_{i+1})^2} \qquad (2\text{-}2)$$

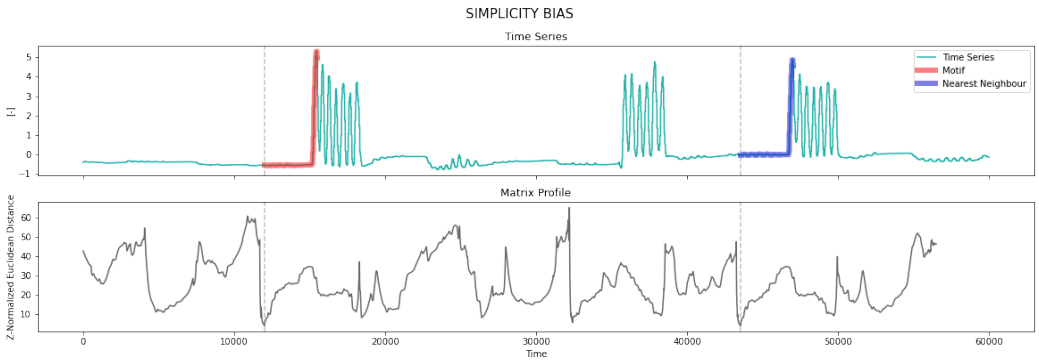


Figure 2-10: Simplicity-Bias; Time Series, Motif and Nearest Neighbour (top); z-normalized (bottom). Adapted from[7]

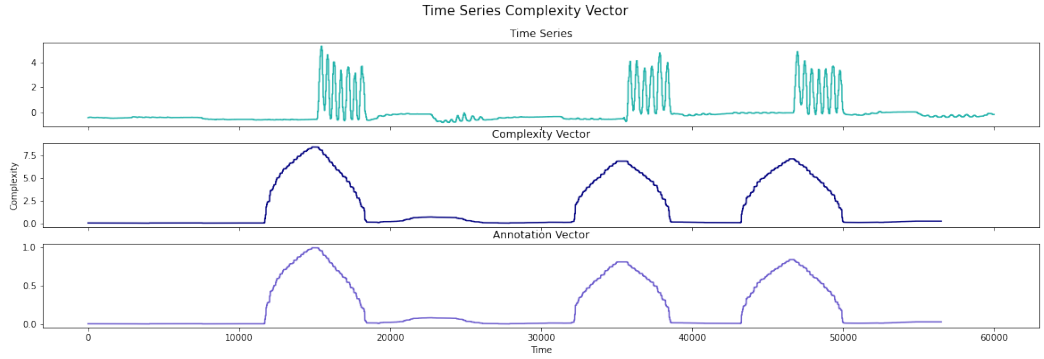


Figure 2-11: Complexity-Vector; Time-Series (top); Complexity Vector (middle); Annotation Vector (bottom). Adapted from [7]

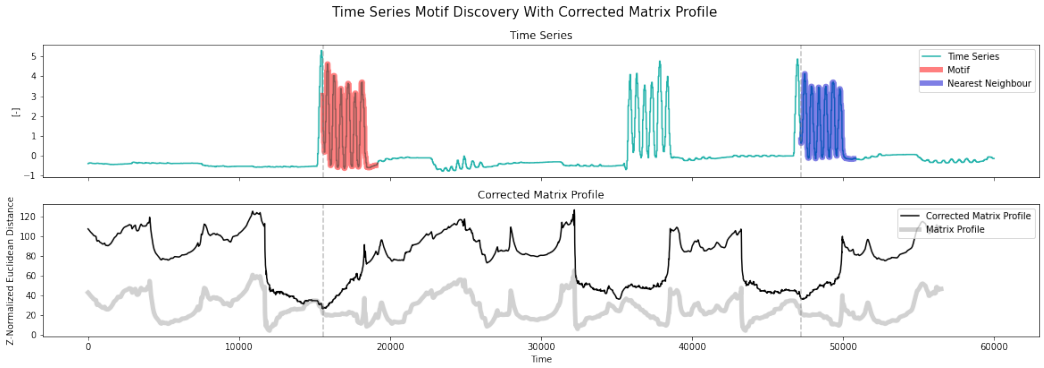


Figure 2-12: Time Series, New Motif, New Nearest Neighbour (top); Corrected Matrix Profile (bottom). Adapted from [7].

Lastly, there are some cases in which we are interested in finding not the best motif, but a Time Series interval which is ‘‘exploitable’’ or ‘‘actionable’’ in some specific way [7]. Several tricks are mentioned in [7] to address these cases, one of them consists in generating AV using additionally Time Series related to the problem (by the name of ‘‘Suppressing Motion Artifact’’). Whereas with the terms ‘‘ Suppressing Hard-Limited Artifacts’’ we refer to those techniques allowing to find Motifs between Time Series upper bound value and Time Series lower bound value, excluding the boundaries [7].

2.1.3 THE Z-NORMALIZATION EFFECTS

As mentioned before, the Matrix Profile is series where each value at a certain index is the distance between z-normalized (zero mean, the variance equal to one) subsequence of time series one, which starts at that index and the best matching z-normalized subsequence of time series two [9]. In this case, the time series one and the time series two are the same, this means that we search for matches in the same time series. The Matrix Profile has been employed to find discords, or rather the subsequence which is the farthest from its nearest match. The z-normalized Euclidean distance firstly commutes each subsequence shape to its normal form (capturing the shape and not the magnitude) and then compares normalized subsequences. The patterns processed with Matrix Profile belongs to several scientific fields, they are signals coming from uncalibrated sensors, natural sources, etc. As reported by the authors in [9], the z-normalization has the disadvantage when dealing with flat subsequences, in fact, the fluctuations in a flat subsequence result in high values in Matrix Profile, against the human intuition of similarity. The effect of flat noisy subsequences impacts the Discord Discovery because the highest values of real discords could be overwhelmed by fluctuations Matrix Profile high values, whereas the Motif Discovery use case is not affected by this effect. The other aspect which limits the application of the z-normalized Euclidean distance is that the magnitudes or amplitudes are not involved when matches are performed. We could say that z-norm Euclidean distance is a shape-comparator, but if our discords are fully influenced by amplitudes, as in the case of energy consumption time series, then the only information about the shape is not sufficient. To overcome these limits, we use not z-normalized Euclidean distance in the following.

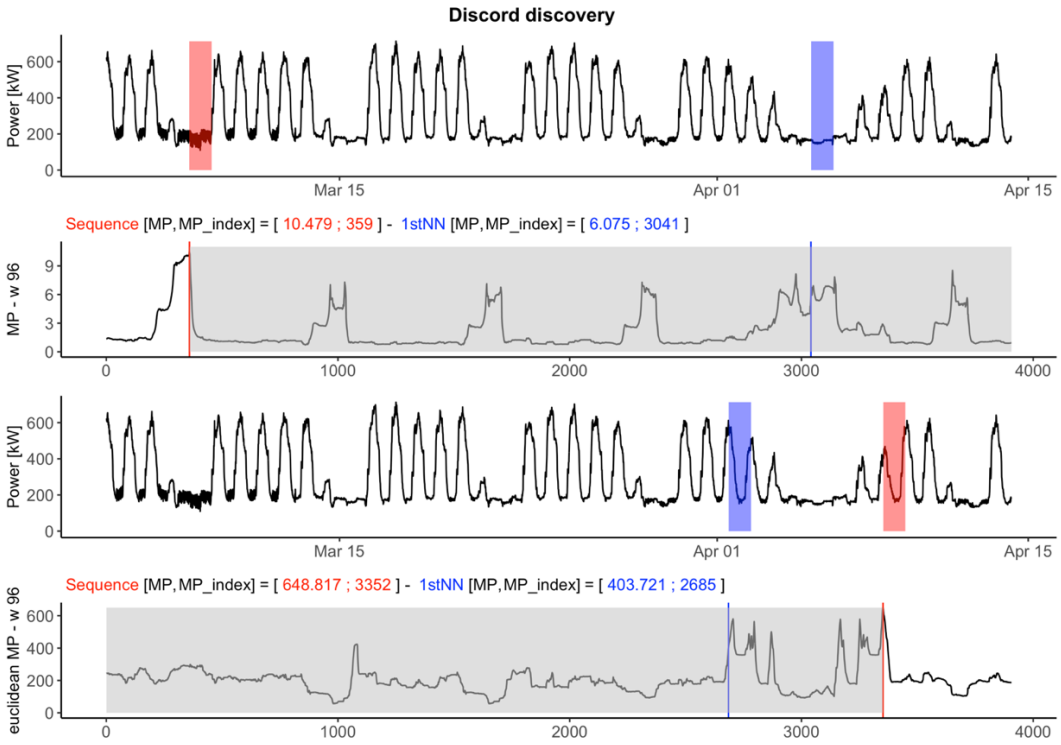


Figure 2-13: Discord discovery; from the top to the bottom, Time Series with top discord in red and 1stNN in blue, z-normalized Euclidean Matrix Profile, Time Serie with top discord in red and 1stNN in blue, not z-normalized Euclidean Matrix Profile.

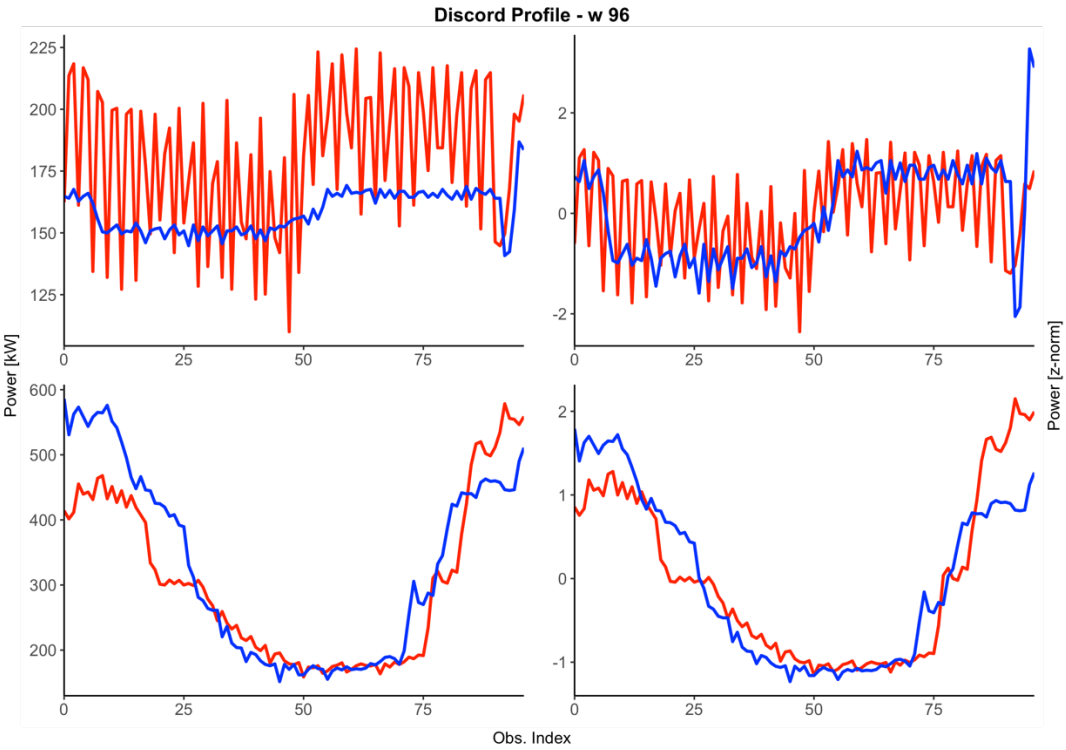


Figure 2-14: Not z-normalized profiles (left), z-normalized profile (right)

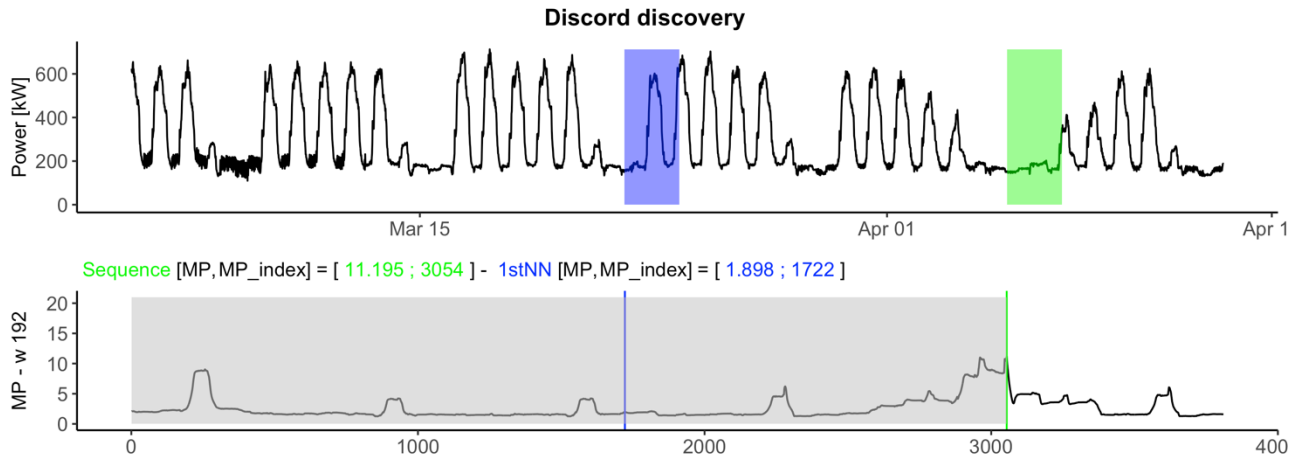


Figure 2-15: Time Series with top Discord and 1stNN in blue (top), z-normalized Euclidean Matrix Profile (bottom).

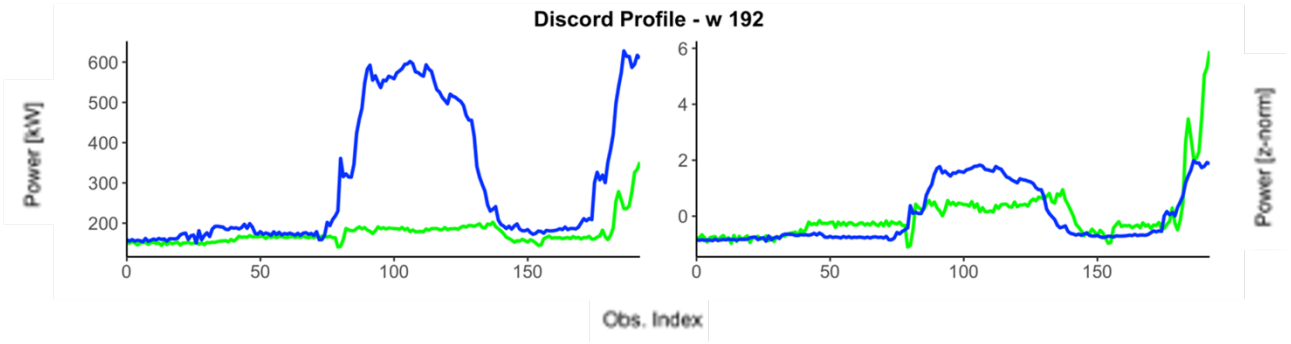


Figure 2-16: top Discord profile and 1stNN in blue (left), top Discord profile under z-normalized Euclidean distance and 1stNN in blue (right).

The Figure 2-13 displays the problem of flatten subsequences, the discord profile index change when the not z-normalized Euclidean distance is used. Meanwhile the Figure 2-14, shows the discord profiles (normalized and not normalized) found by both Matrix Profile techniques; the fluctuations bias the Discord discovery. Last problem regarding the z-normalized distance is shown in Figure 2-16, the discord profile and the 1stNN under z-normalized distance seem to be similar, but actually they are so different, as shown on the left side.

2.2 STATISTICAL MACHINE-LEARNING ALGORITHMS

2.2.1 CART METHOD

CART are machine learning methods that divide data into smaller and smaller groups(non-overlapping) with similar to each other values, using a set of splitting rules, to identify pattern useful for making prediction. As the name implies, Decision Tree methods are based on tree structure. The model, in analogy with flowchart makes ordered logical decisions, each of which come from decision node, that decides according to an attribute value. There are three kind of nodes, root node where the data start to be processed, decision node where data are split into branches that represent decision's choices and last, the leaf node, that contain data after having followed a combination of decisions. From the root node the algorithm selects a feature that is the most predictive of the target class, then data observations are grouped by values of this feature. The algorithm continues the recursive partitioning choosing the best feature each time, until stopping criterion is reached.

Stopping criteria are:

- Nearly all of the observations at the node have the same class.
- The are no remaining distinguishing features within partition.
- The tree has reached a set size limit.

The feature values that split data sample such that partitions includes primarily data of single class, suggest the best split; in addition, a group of data is said to be pure if it contains only a class. The way used to choose the best split involves

entropy index. The entropy index is a measure of purity of a data sample, it is on the range zero one. The value 0 states that the data sample at hand is homogeneous, in contrast, the value 1 underlines a very mixed class values data sample.

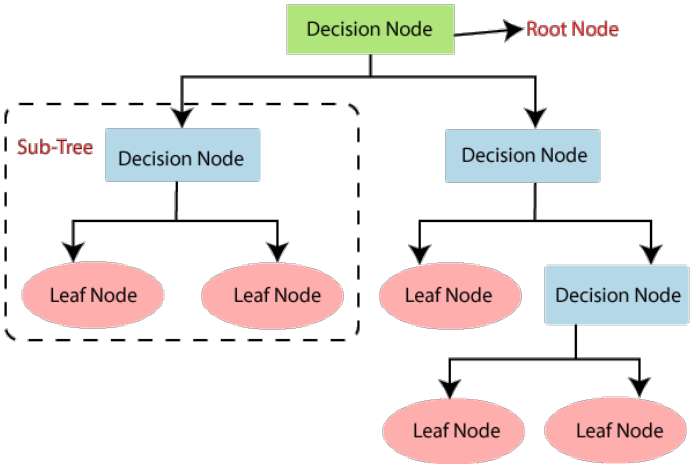


Figure 2-17: Decision Tree structure

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2(p_i)$$

(2-3)

Where:

S = data sample.

c = number of different class levels.

p_i= proportion of falling into class level i.

For example, if we have two class than $p_1 = x$ and $p_2 = 1 - x$, where x is a possible value of proportion, between 0 and 1; accordioning to the previous formula (2-3).

$$S = -x * \log_2 x - (1 - x) * \log_2(1 - x)$$

(2-4)

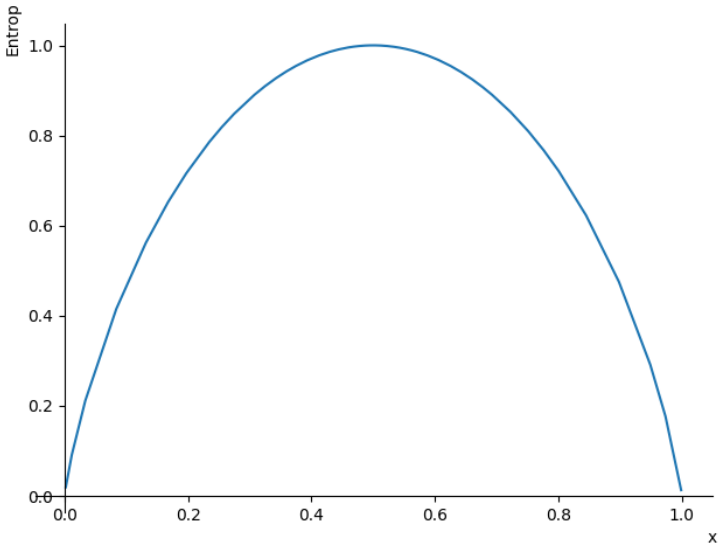


Figure 2-18: Entropy function

The peak of Entropy is reached in $x = 50$ where the data sample contains the same number of objects belonging to the two class. To decide the best feature to split upon, the algorithm calculate the variation of the Entropy between and after a split for all possible features. This kind of calculation is called information gain.

$$InfoGain(F) = Entropy(S_1) - Entropy(S_2) \quad (2\text{-}5)$$

$$Entropy(S_2) = \sum_{i=1}^{n} w_i \, Entropy(P_i) \quad (2\text{-}6)$$

Where:

S is the total entropy resulting from a split.

n is the number of partitions.

w_i is the number of examples falling in partition 'i' (weight).

P_i is the partition 'i'.

The higher the information gain, the better is the feature at creating homogeneous groups after a split. If the information gain is zero, the entropy doesn't reduce, and the feature choice is bad. A decision tree can continue to grow until each example is perfectly classified or there are no more features to split on. The process to reduce the size of the tree is known as 'pruning', It avoid that the tree overfit data, losing the skill to generalize better to unseen data.

2.2.2 AGGLOMERATIVE HIERACHICAL CLUSTERING

Clustering is the process of grouping a set of data obsevations into multiple groups or clusters so that observations within a cluster have high similarity but are very dissimilar to those ones of other clusters. Dissimilarities and similarities are assessed based on certain kind of measures, described in the following. The partitioning is not performed in a supervised manner by humans, but by the clustering algorithm. Hence, clustering can lead to discover previously unknown knowledge and groups within the data. As a data mining algorithm, cluster analysis can be used to gain insight into the distribution of observations into dataset, to observe the characteristics of each cluster, and to focus on a particular set of clusters for further analysis. Alternatively, it may be useful as a reprocessing step for data mining frameworks, which would then operate on the detected clusters and on the related attributes or features. Clustering is known as unsupervised learning because the class label information is not present. In data mining, efforts have focused on finding methods for efficient cluster analysis in large datasets. A hierarchical method creates a hierarchical decomposition of the given set of data observations, it can be of two types, agglomerative or divisive one, based on how the hierarchical decomposition is performed. The

agglomerative approach, also called the bottom-up approach, starts with each observation which constitutes a single separate group. It successively merges the observations until all the groups are merged into one. Hierarchical clustering methods can be distance-based or density-based and continuity-based. This class of methods suffer from the fact that once a merging step is done, it can never be undone. Such techniques cannot correct erroneous decisions; however, methods for improving the quality of hierarchical clustering can be found in scientific literature. Once again, agglomerative methods start with individual observations as clusters, which are iteratively merged to form larger clusters, the process at the next step will operate on the newly generated clusters. These process of merging, if not well carried out, may lead to low-quality clusters. Moreover, the entirely process do not scale well because each decision of merge needs to examine many objects or clusters.

2.2.3 DISSIMILARITY MEASURES:

The following distance measures are written for two vectors or time series snippets x and y, they must be both of the same length, d-vectors. The array elements or variables values may be quantitative (discrete or continuous) or qualitative (ordinal or nominal).

Euclidean distance: it is the usual square distance between the two vectors. It is given by Equation:

$$d(x,y) = \left(\sum_{j=1}^{d} \left(x_j - y_j\right)^2\right)^{\frac{1}{2}} \quad (2\text{-}7)$$

Maximum distance: it is the maximum distance between two components of x and y (supremum norm), as described by Equation:

$$d(x,y) = \sup_{1 \le j \le d} |x_j - y_j|$$

(2-8)

Manhattan distance: is the absolute distance between the two vectors. It is given by Equation:

$$d(x,y) = \sum_{j=1}^{d} |x_j - y_j|$$

(2-9)

Canberra distance: terms with zero numerator and denominator are omitted from the sum and treated as if the values were missing.

$$d(x,y) = \sum_{j=1}^{d} \frac{|x_j - y_j|}{|x_j| + |y_j|}$$

(2-10)

Binary distance: the vectors are regarded as binary bits, non-zero elements are "on", and zero elements are "off". The distance is the proportion of bits in which only one is on amongst those in which at least one is on.

Minkowski distance: is the p-norm, i.e., the p-th root of the sum of the p-th powers of the differences of the components

$$d(x,y) = \left(\sum_{j=1}^{d} \left(x_j - y_j\right)^p\right)^{\frac{1}{p}}$$

(2-11)

2.2.4 WARD'S METHOD

Ward's method says that the distance between two clusters, A and B, is how much the sum of squares will increase when we merge them:

$$\Delta(A,B) = \sum_{i \in A \cup B} \|\overrightarrow{x_i} - \overrightarrow{m_{A \cup B}}\|^2 - \sum_{i \in A} \|\overrightarrow{x_i} - \overrightarrow{m_A}\|^2 - \sum_{i \in B} \|\overrightarrow{x_i} - \overrightarrow{m_B}\|^2 = \frac{n_A n_B}{n_A + n_B} \|\overrightarrow{m_A} - \overrightarrow{m_B}\|^2$$

(2-12)

Where:

m_j is the center of cluster j

n_j is the number of points in it

Δ is the merging cost of combining the clusters A and B.

With hierarchical clustering, the sum of squares starts out at zero (because every point is in its own cluster) and then grows as we merge clusters. Ward's method keeps this growth as small as possible. This is nice if we think that the sum of squares should be small after a merging. Given two pairs of clusters whose centers are equally far apart, Ward's method will prefer to merge the smaller ones, furthermore Ward's method is both greedy and constrained by previous choices as to which clusters to form. This means its sum-of-squares for a given

number k of clusters is usually larger than the minimum for that k, and even larger than what k-means will achieve.

Silhouette Coefficient

Silhouette coefficient has been introduced as a method to explain and to validate within clusters of data consistency. The silhouette coefficient represents the measure of how similar an object is to its own cluster (cohesion) compared to other clusters(separation). A formal definition of the two previous concepts will be required for a later analysis. The cohesion is how closely are the points into a given cluster. Optimal cluster will be characterized from a high value of cohesion. To find Silhouette coefficient we compute, for each point i, the statistics related to the distance from i to all other points in its own cluster, in order to compute the average distance.

$$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(j, i)$$

(2-13)

Where:

i ∈ C_i,

$|C_i|$ is the cardinality of the cluster.

Cluster separation is a statistic that assess how distinct or well-separated a cluster is from other clusters. For Silhouette coefficient is required to compute the distance between a given point i and any other cluster, which i is not a member of. First, we need to perform the average dissimilarities between i and the other

cluster, computed as the average distance between the point and all the members of that cluster.

$$d(i, C_j) = \frac{1}{|C_j|} \sum_{j \in C_j} d(j, i)$$

(2-14)

The equation 2.15 is performed for each cluster $C_j = C_i$ obtained from the clustering algorithm. Once, we highlight a" neighbouring cluster", and the statics b_i used in silhouette is the average distance between that specific cluster. Hence, we get:

$$b_i = \min_{j \neq i} d(i, C_j)$$

(2-15)

With the previous metrics we can define silhouette index, for a given point i as:

$$s_i = \frac{b(i) - a(i)}{\max\{a_i, b_i\}}$$

(2-16)

Which can be also written as:

$$s_i = \begin{cases} 1 - \frac{a_i}{b_i}, & a_i < b_i \\ 0, & a_i = b_i \\ \frac{b_i}{a_i} - 1, & a_i > b_i \end{cases}$$

(2-17)

From this definition is clear that $-1 \leq s_i \leq 1$ if ai < bi if ai = bi if ai > bi. Also, the score is 0 for clusters with size equal to 1. This constraint is added to prevent the number of clusters from increasing significantly. For s_i close to 1 we require $a_i \ll$

b_i. As a_i measures of how dissimilar i is to its own cluster, a small value means it is well matched. At the Meanwhile, a large b_i implies that i is badly matched to its neighbouring cluster. Thus an s_i close to one means that the data is correctly clustered. If s_i is close to -1, then for the same reason, i would be more appropriate if it was clustered in its neighbouring cluster. An s_i near zero means that the sample is on the border of two natural clusters. Silhouette index is calculated separately for each point. In order to provide a median representative metric useful to determine the quality of the clustering we have to combine all the indexes. The average $\bar{s}(i) = s_i$ over all points of a cluster is a measure of how tightly grouped all the points in the cluster are. Thus, the average s_i over all data of the entire dataset is a measure of how appropriately the data have been clustered. It can be used to determine the natural number of a cluster into a dataset, computing the index for each possible k in order to select the maximum.

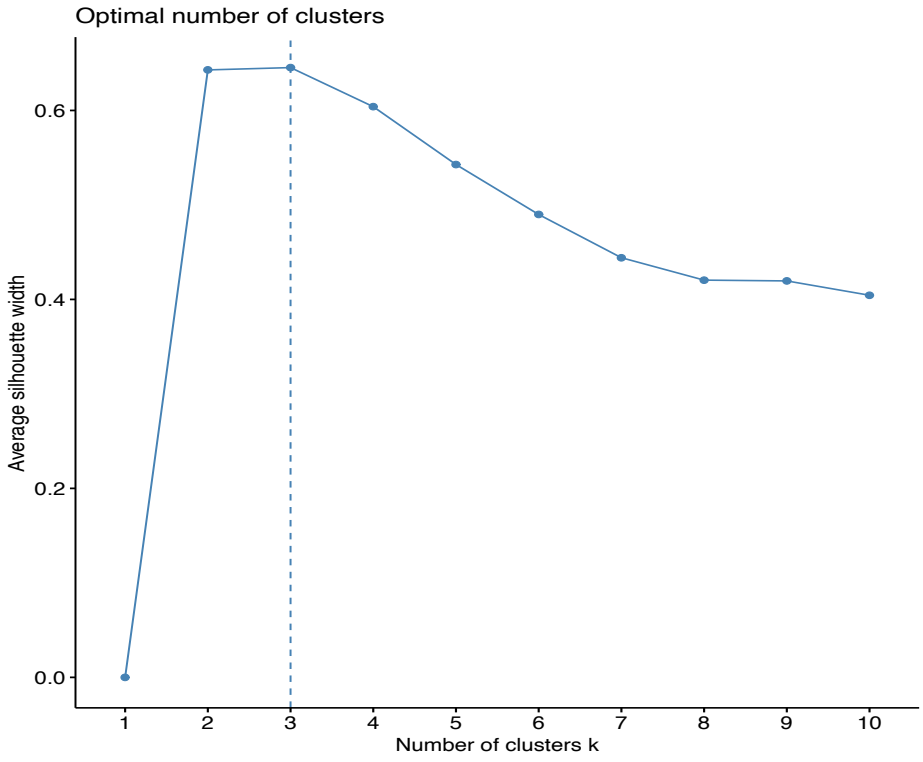


Figure 2-19: Average Silhouette Coefficient and the best number of Cluster K

- $\bar{s}(i) \in (0.70, 1.00]$ → the given partition is extremely reliable
- $\bar{s}(i) \in (0.50,\ 0.70]$ → the given partition is reliable
- $\bar{s}(i) \in (0.25,\ 0.50]$ → the given partition is not so reliable
- $\bar{s}(i) \in (-1.00,\ 0.25]$ → the given partition is not reliable

2.3 CONTEXTUAL MATRIX PROFILE

A new framework is proposed by De Paepe at al. in [10] that focuses on the implicit distance matrix calculation, called the Series Distance Matrix (SDM). This framework takes advantages from distance measures (SDM-generators) and distance processors (SDM-consumers), which can be combined, allowing for more flexibility and easier experimentation. In SDM, the Matrix Profile is one special configuration. Furthermore in [10] Is introduced the Contextual Matrix Profile (CMP) as a new SDM-consumer capable of discovering repeating operating patterns. The strength of CMP is that provides easy visualizations for data analysis and can find anomalies that are not only discords. Series analysis techniques deal with ordered group of observation, rather than independent data points, and unlike non-series, consecutive points in series carry meaning and patterns that will often occur throughout the series. Finding and analysing these patterns can allow better insights in the dataset. Subsequently, they can be used for anomaly detection in contexts where anomalies are not only defined by unique behaviour. Series Distance Matrix (SDM) framework can be considered as the base building block on which other techniques can be built. As mentioned previously it consist of separate components that calculate distances between subsequences of input series (SDM-generators) and components processing these distances (SDM-consumers).

Definition 4. "The z-normalised Euclidean distance $D_{ZE}(A,B)$ between series of equal length $A \in R^m$ and $B \in R^m$ is defined as the Euclidean distance D_E of the z-normalised series $\hat{A}$ and $\hat{B}$".

$$D_{ZE}(A,B) = D_{ZE}(\hat{A},\hat{B}) = \sqrt{(\widehat{a_0} - \widehat{b_0})^2 + \cdots + (\widehat{a_{m-1}} - \widehat{b_{m-1}})^2} \quad (2\text{-}18)$$

Given pairs of series, SDM-generators are responsible for calculating the distances between all pairs of subsequences. Because calculating the full distance matrix requires a high computational effort, it is convenient instead calculate fragments of the distance matrix. These fragments are processed by the SDM-consumers, after which the fragment is discarded, and a new fragment is calculated. This could be used by some consumers, such as the Matrix Profile, to provide approximate intermediate results when processing all data takes a long time, making it well suited for interactive use cases. The CMP, which can easily find repeated patterns in series and inherits the benefits of the Matrix Profile, is deterministic, domain agnostic, exact and is suited for parallelization. As the name implies, the CMP is closely related to the Matrix Profile, and can be best explained making a comparison with it.

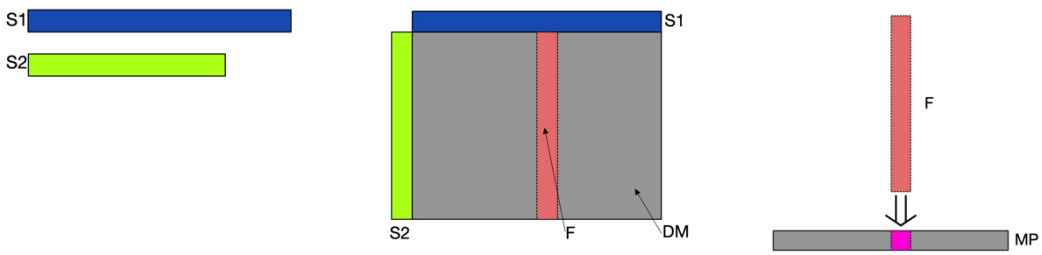


Figure 2-20: Time Series on the left, Distance Matrix in the middle and Matrix Profile on the right. Reprinted from[10]

The Matrix Profile is defined as the column-wise minimum over the entire distance matrix, whereas the CMP is defined as the minimum over rectangular regions of the distance matrix. These rectangles may overlap and may or may not cover the entire distance matrix. Note that the CMP-consumer may be configured in such a way that it calculates the Matrix Profile. In this way, the CMP can be seen as a generalization or extension of the Matrix Profile. The CMP on the other hand looks for the best matching subsequence in ranges over S1 and S2. These ranges allow us to group the data in different ways and can reveal new insightful operating patterns.

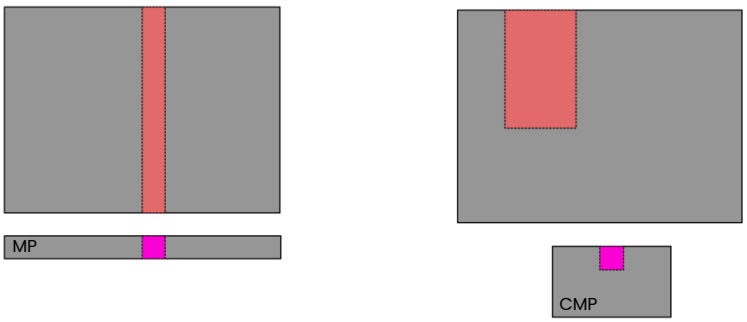


Figure 2-21: Matrix Profile on the left and Contextual Matrix Profile on the right

One benefit of the CMP is that it allows us to discover these patterns in advance when the pattern is unknown in advance. So, assuming we did not know the weekday/weekend similarity beforehand, we could have easily deduced it by visualizing the CMP. The CMP has one other major advantage over a basic distance matrix, it allows for a (time) shift when comparing sequences. The CMP has one other major advantage over a basic distance matrix, it allows for a (time) shift when comparing sequences. One advantage of the CMP over the Matrix Profile for anomaly detection is that the CMP does not depend on the uniqueness

of anomalies (it does not simply find discords), but rather on the expectations of the user regarding normal behaviour. These expectations correspond to the CMP contexts and can be based on the insights retrieved using the CMP for data visualization. As part of the SDM framework, the CMP can be calculated using any distance measure and calculated in parallel with other techniques such as the Matrix Profile

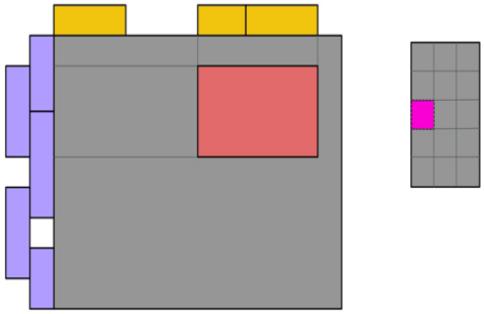


Figure 2-22: Contexts or ranges in yellow and violet, and Distance Matrix (on the left) and Contextual Matrix Profile (on the right).

2.4 OUTLIERS DETECTION TECNIQUES

Once the CMPs for contexts and clusters were obtained, we have several sets of distances that have to be overseen, in order to exclude those ones that are not consistent with the physics of the problem. We could take advantage of the classical statistical techniques to explore in detail which distance is an outlier. This process is well known as Outliers Detection.

Outlier definition of Hawkins [Hawkins 1980]:

“An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism”.

In this case, the outlier detection was performed taking into account the median of each column (representative of a distance of a certain day from the other ones) and then applying some techniques to discover which median (distance and the day to belong to) is an outlier. Higher is the distance higher is the probability of a day(context) to be an outlier. The methods employed to detect outliers are four: boxplot, z-score transformation, elbow-method and the last, Generalized Extreme Studentized Deviate. Below, an introduction of the main features of these statistical methods:

2.4.1 BOXPLOT (BOX AND WHISKERS PLOT)

A box and whisker plot displays the five-number summary of a data set. The five-number summary is the minimum, first quartile, median, third quartile, and maximum. It also shows the spread, the center of a dataset and is useful for indicating whether a distribution is skewed and whether there are potential unusual observations (low probability to happen) in the data set.

2.4.2 Z-SCORE TRASFORMATION

The process of transforming dataset values into z-scores involves the generating of signed numbers, called z-score, such that, the sign of the z-scores (+ or –) identifies whether the values are located above the mean (positive) or below the mean (negative). The numerical value of the z-score corresponds to the number of standard deviations between a value and the mean of the distribution. z = 0 is in the center (at the mean), and the extreme tails correspond to z-scores of approximately –2.00 on the left and +2.00 on the right. If an entire distribution of

values is transformed into z-scores, the resulting distribution of z-scores will always have a mean of zero and a standard deviation of one. The shape of the original distribution doesn't change if transformed in z-score one and the location of any individual score relative to others in both distributions is the same. In conclusion advantage of standardizing distributions is that the values of different distributions can be compared. Those value that falls out the range]-2;2[are possible outliers.

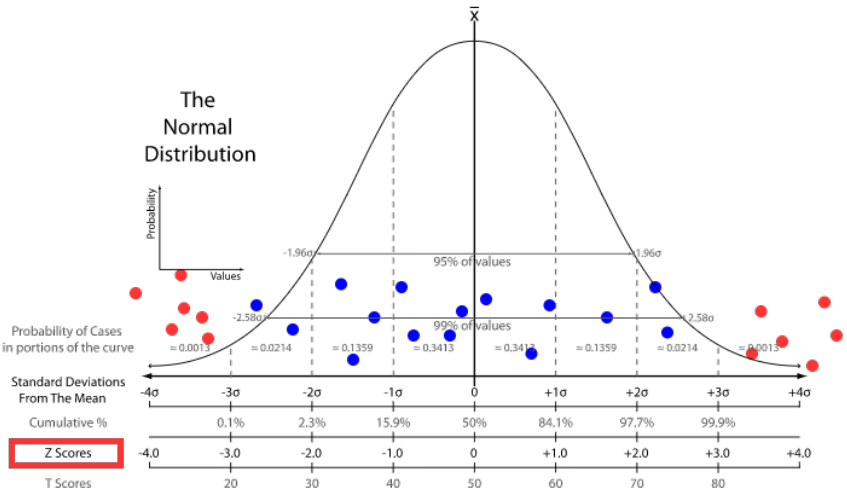


Figure 2-23 : Gaussian distribution with z-score values

$$z - score = \frac{x - \mu}{\sigma} \tag{2-19}$$

2.4.3 ELBOW-METHOD

To detect those distances that are extremally large, another possible method is the elbow method. It sorts the medians of columns and gives them an Anomaly Score. When the rate of change of Anomaly score doesn't vary significantly, the

elbow is found. All distances (context or days) below the threshold are “outliers” or possible anomalies, Figure 2-24.

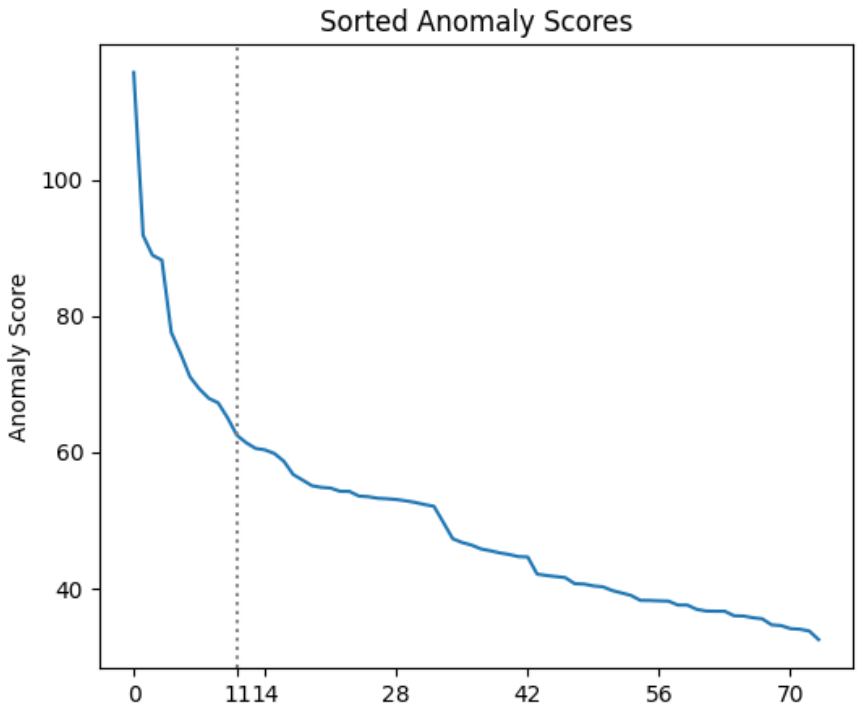


Figure 2-24: The Elbow plot

2.4.4 GENERALIZED EXTREME STUDENTIZED DEVIATE TEST (GESD)

GESD is a simple statistical approach used to detect one or more outliers in a univariate data set that follows an approximately normal distribution. Statistical considerations assume that normal data follow some statistical model and the data not following the model are outliers. The GESD test only requires that an upper bound for the suspected number of outliers be specified. Given the upper bound, r, the generalized ESD test essentially performs r separate tests: a test for one outlier, a test for two outliers, and so on up to r outliers.

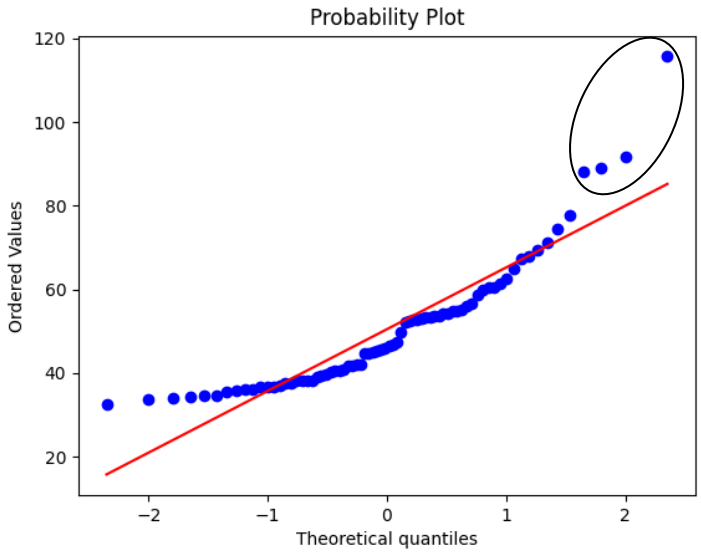


Figure 2-25: Q-Q plot

The generalized ESD test is defined for the hypothesis:

H0: There are no outliers in the data set

Ha: There are up to r outliers in the data set

Our test statistic is given by the formula below:

$$R_i = \frac{\max_i |x_i - \bar{x}|}{\sigma}$$

(2-20)

Here, $\bar{x}$ and σ are sample mean and sample standard deviation, respectively. In GESD we exclude the observation that maximizes |xi $-\bar{x}$ | and then recompute the above statistic with n-1 observations. We repeat this process until r observations have been removed. This results in the r statistics R1, R2 ....., Rr. Corresponding to the r test statistics, compute the following r critical values:

$$\lambda_i = \frac{(n-i)t_{p,n-i-1}}{\sqrt{\left(n-i-1+t^2_{p,n-i-1}\right)(n-i+1)}} \quad i = 1,2\ldots,r$$

(2-21)

where $t_{p,\nu}$ is the 100p percentage point from the t distribution with ν degrees of freedom and

$$p = 1 - \frac{\alpha}{2(n-i+1)}$$

(2-22)

Our Significance level will be denoted by α.

The number of outliers is determined by finding the largest I such that $R_i > \lambda_i$

2.4.5 MEDIAN Z-SCORE

In The Median z-score method median and median absolute deviation are used in place of mean and standard deviation, this implies that the method is less influenced by a single extreme outlier value.

$$MAD = median(|x_i - \tilde{X}|)$$

(2-23)

Where:

$\tilde{X}$ = sample median

$$M_i = \frac{0.6745 * (x_i - \tilde{X})}{MAD}$$

(2-24)

The example should be labeled as outlier if $|M_i| > 3.5$.

2.4.6 ADJUSTED BOXPLOT

Although the classical box-plot method is applicable to both symmetric and skewed data sample, what comes up is a large number of observations dealt with as outliers in case of high skewness in data sample [11]. Vanderviere and Huber have introduced a new trick taking into account the medcouple (MC), a robust skewness measure.

Let $X_n = \{x_1, \dots, x_n\}$ be a data sample from continuous univariate distribution, with $x_1 \leq \cdots \leq x_n$. The MC coefficient could be calculated as the following formula:

$$MC(x_1, \dots, x_n) = med \frac{(x_j - med_k) - (med_k - x_i)}{x_j - x_i} \quad (2\text{-}25)$$

Where:

- $med_k =$ Median of the X_n data set.
- $x_i \leq med_k \leq x_j;\ x_i \neq x_j$.

The fences are computed as follows:

$$\begin{cases} [L, U] = [Q_1 - 1.5 \exp(-3.5\ MC\ )\ IQR,\ Q_3 + 1.5 \exp(4\ MC)\ IQR] & MC \geq 0 \\ [L, U] = [Q_1 - 1.5 \exp(-4\ MC\ )\ IQR,\ Q_3 + 1.5 \exp(3.5\ MC)\ IQR] & MC \leq 0 \end{cases} \quad (2\text{-}26)$$

If the distribution at hand is slightly right skewed, the lower fence moves to the right and more observation in the left side is determined as outlier, if a comparison with classical box-plot method is made; on the other hand, the upper fence identifies less observations as outliers. To summarize the adjusted box-plot method considers boundaries or fences that are free of the effect of skewness.

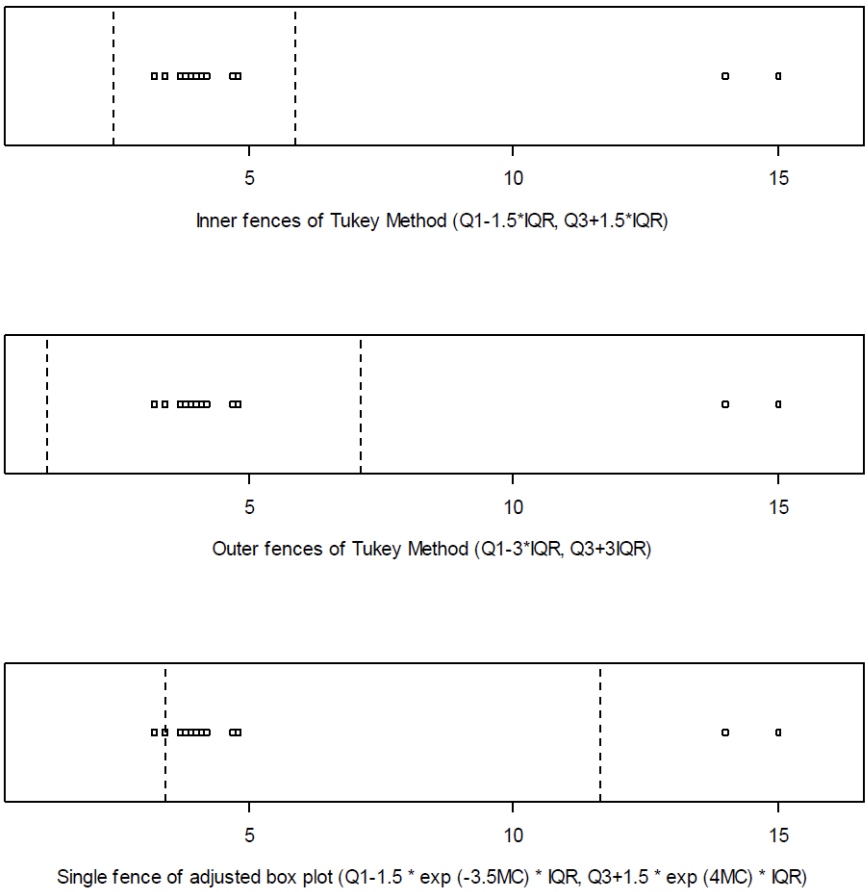


Figure 2-26: Change of the intervals of two different box-plot methods. Reprinted from [11]

3 METHODOLOGY

The methodology built up, deals with an Anomaly Detection framework on total electrical load time series, able to find anomalies patterns at meter level. In this chapter the entire framework process is described, the role of each concatenated algorithm is explained in detail as well as the interactions among algorithms. Data pre-processing is an essential step for data-mining process. The dataset used in this case study comes from a measure equipment installed in a medium voltage substation of Politecnico di Torino. The dataset under discussion has been processed by the BAEDA Lab Team and the physical inconsistencies like negative power rather than nearly-zero power were treated or removed. The further step consists in classifying daily pattern to find pattern distinctive feature.

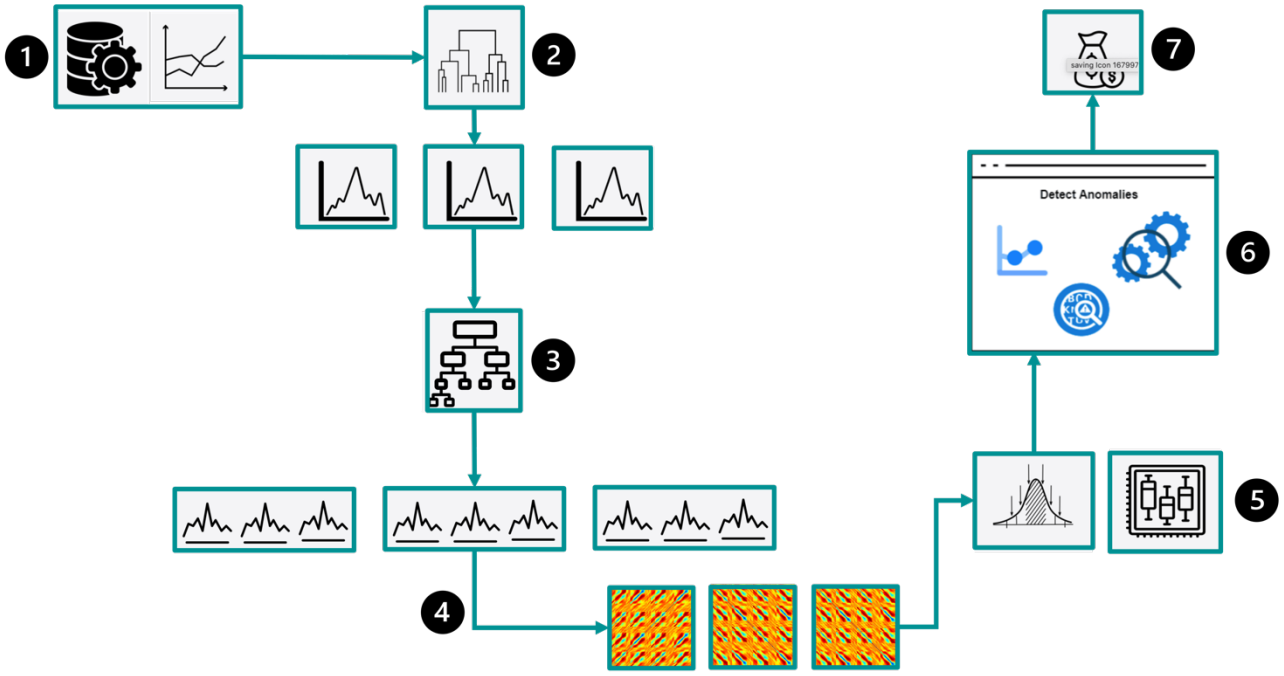


Figure 3-1: framework steps

3.1 CLUSTER SETTING

The cluster analysis belongs to unsupervised class of methods with the classifying task; in this case, the scope of this step analysis is to bring together daily electrical load patterns which are more similar to each other and extract the main features like shape, the weekday type, the season day type and so on. The idea of similarity or dissimilarity is expressed through the concept of distance whereas the method represents the way to groups continuously the daily portions of Time Series (profiles). According to several test made with R libraries, the combination of Euclidean distance with Ward.D method has turned out the best one. This combination gives the best partitions (the most of daily patterns is closer to their own centroid with this combination of distance and methods then with other ones) which could be labeled easily by domain expert figure.

3.2 CONTEXTUAL SUBSEQUENCES

The contextual subsequences are a daily sub-period characterized by a load typical trend and associated with a Context. The task of identify daily subsequences is entrusted to a regression decision tree while the variable used to make decisions inside the leaf nodes is the hour of the day. Contextual subsequences allow to compare time portions with similar load trend, avoiding comparisons with so different profile portions. The CART method needs for hyperparameters tuning process to stop the learning stage and avoiding the overfitting. The hyperparameters act as thresholds that stop the algorithms when they are achieved; they are:

- The minbucket gives the smallest number of observations that are allowed in a terminal node. If a splitting parent node generates child one with less observation than the minbucket, the splitting doesn't occur. The minbucket parameter was set to 120.
- The minsplit is the smallest number of observations in the parent node that could be split further. If a node has less examples than the minsplit than it is labelled as leaf node.
- the maxdepth bounds the tree growth below a certain depth / height. It was set equal to 10

Last hypermeter has been taken into account is the complexity parameter. The complexity parameter cp is employed to measure the Cost-Complexity of our tree; it is useful in pruning the tree and has been set equal to 0, that means no limits to complexity.

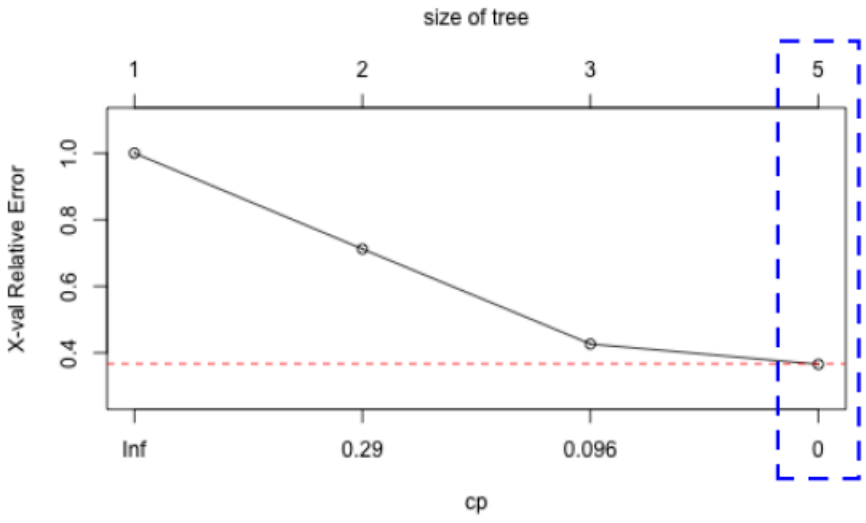


Figure 3-2: cp coefficient

3.3 ANOMALIES BY CONTEXTS AND CLUSTERS

Contextual Matrix Profile stores inside itself distances between time subsequences. Each point displays the Euclidean distance between the two best matching contextual subsequences of two different days. Lower are the distances, better are the match. So, every column and row are representative of a dataset day and to gain further insight, we could isolate those distances that deviate from the majority. This refined approach leads us to discover anomalies respect each context but also respect each cluster. The computation of CMP has been done with the support of customized and flexible Python library: ''Series Distance Matrix''; the creators are the same of the scientific paper [10]. This library includes ''Generators'' and '' Consumers''. With the term generators we refer to a part of library ables to create Distance Matrix according to a set of suitable distances; instead, consumers process DM to generate CMP. The way to establish which distances are far from the rest ones is to apply outlier detection methods, described in the previous section. The next step concerns in querying all the outlier detection techniques to print a list of hierarchical anomalies (sorted by priority) by Contexts and Clusters, according to majority voting rule. Most of statistical outlier detection techniques are parameter free, whereas the adjusted boxplot has required parameter tuning procedure. As regards to GESD method, it has been excluded because some clusters have got few data examples to apply a statistical test. At the end of the framework anomalies have been classified with a colors' legend. Each color is associated with a severity level, which in turn depend on how many outlier detection methods have detected a certain anomaly.

3.4 SAVING ESTIMATION

Given some anomalies that are not 'Thermal Sensitive', the last step tries to put in place a potential saving estimation which give us an order of magnitude of energy saving, based on statical considerations. Thanks to Cluster analysis four categories of day have been found. Each Cluster includes inside it a particular profile called centroid, which is the most representative profile. From physics point of view, the centroid is the '' normal consumption'', it is not affected by any kind of anomalies. Those values that doesn't exceed twice the standard deviation of the centroid values moment by moment, have to be considered as '' normal'', instead those ones that exceed the $+2\sigma$ are overconsumptions. The delta consumption between $+2\sigma$ and overconsumption is the actionable saving, that is, the savings we would have got if there not were the anomaly. We only have on actionable saving when the anomaly profile is a fault.

4 STUDY AND RESULTS

The framework, implemented with the support of software like R and some libraries of python, has been applied to the case study of Politecnico di Torino. The working data come from meter equipment of the medium voltage substation C of the Politecnico di Torino. So, we will have to deal with electrical power consumption data of a building wings and the associated electrical subloads. In the rest of chapter, a description of the dataset is performed. The analysis period involved in this case study is between the first of January 2019 and the 31st of December of the same year whereas the frequency of data recording is quarterly basis. The meter level data represent the total energy consumed by chillers, facilities, and appliances. The sub-loads given in the dataset are: DIMAT department, Bar Ambrogio, Refrigeration unit, Data Center, Print Shop and the last, the Not Allocated Power. The really important datum for the case study analysis is the Total electrical load; in this time series there will be applied the most important data mining technique, which is Contextual Matrix Profile. But before coming there, data are processed with traditional machine learning methods. Since the methodology has already discussed widely in Chapter 4, in the rest of this section will be reported the results. The next paragraph shows the clusters found and how they will be interpreted used by domain expert in the framework.

4.1 CLUSTERS DEFINITION

The clusters definition is useful to reduce dimensionality of the problem. The time series of electrical load is firstly partitioned in daily basis load patterns, and

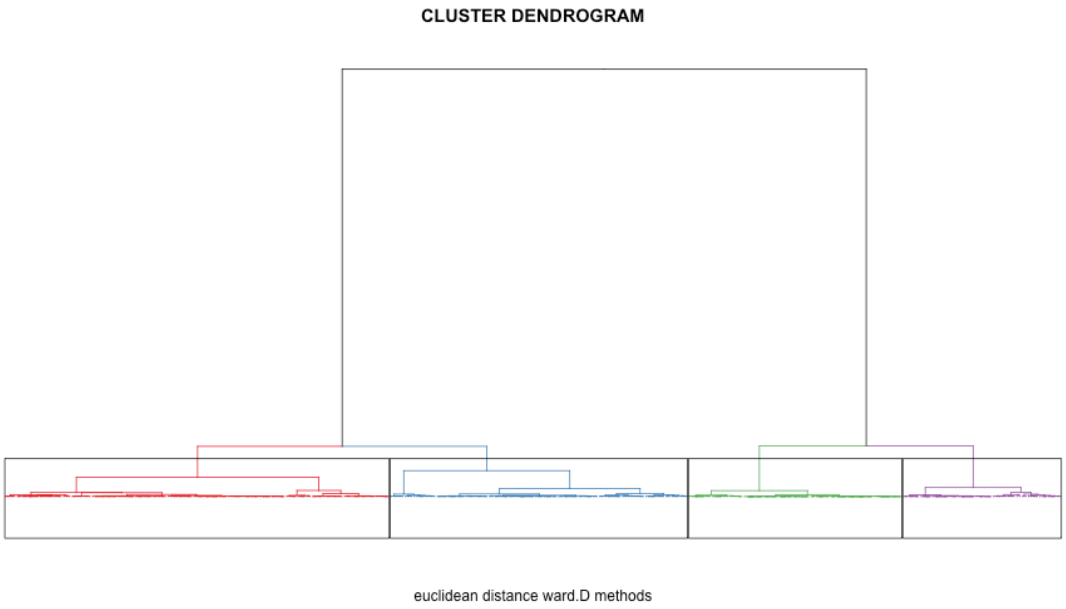


Figure 4-1: Cluster dendrogram using Euclidean distance and Ward.D method

then trough the agglomerative hierarchical cluster technique each pattern (profile) is clustered according to similarity measure. At the end of cluster analysis, each cluster is labeled. The gray profiles are electrical load snippets of the time series, the red ones instead are the centroids. Each centroid is representative of its own cluster and the farther a generic profile is from a centroid, the higher is the probability that it is an anomaly. In detail, the cluster one with flat profiles includes almost all Sundays; the electrical load is just a plateau, a base load which derives from continuously working facilities. The cluster two is the cluster of Saturdays and half-days working. Around the middle of the day the load increases slightly, likely, because of academic activity, then it comes back quickly to the base load. The last two clusters underline the trend of consumptions caused by the occupancy, the use of the appliances, the power for refrigeration, fan and so on. The profiles of the cluster three has a similar shape with respect to those ones of cluster four, but the values keep their selves higher.

The days of June, July and December belong to the third cluster, instead the midseason days have been grouped into cluster four. As concerns the best number of clusters, the Silhouette criterion suggested to cut the dendrogram such that four groups have been generated.

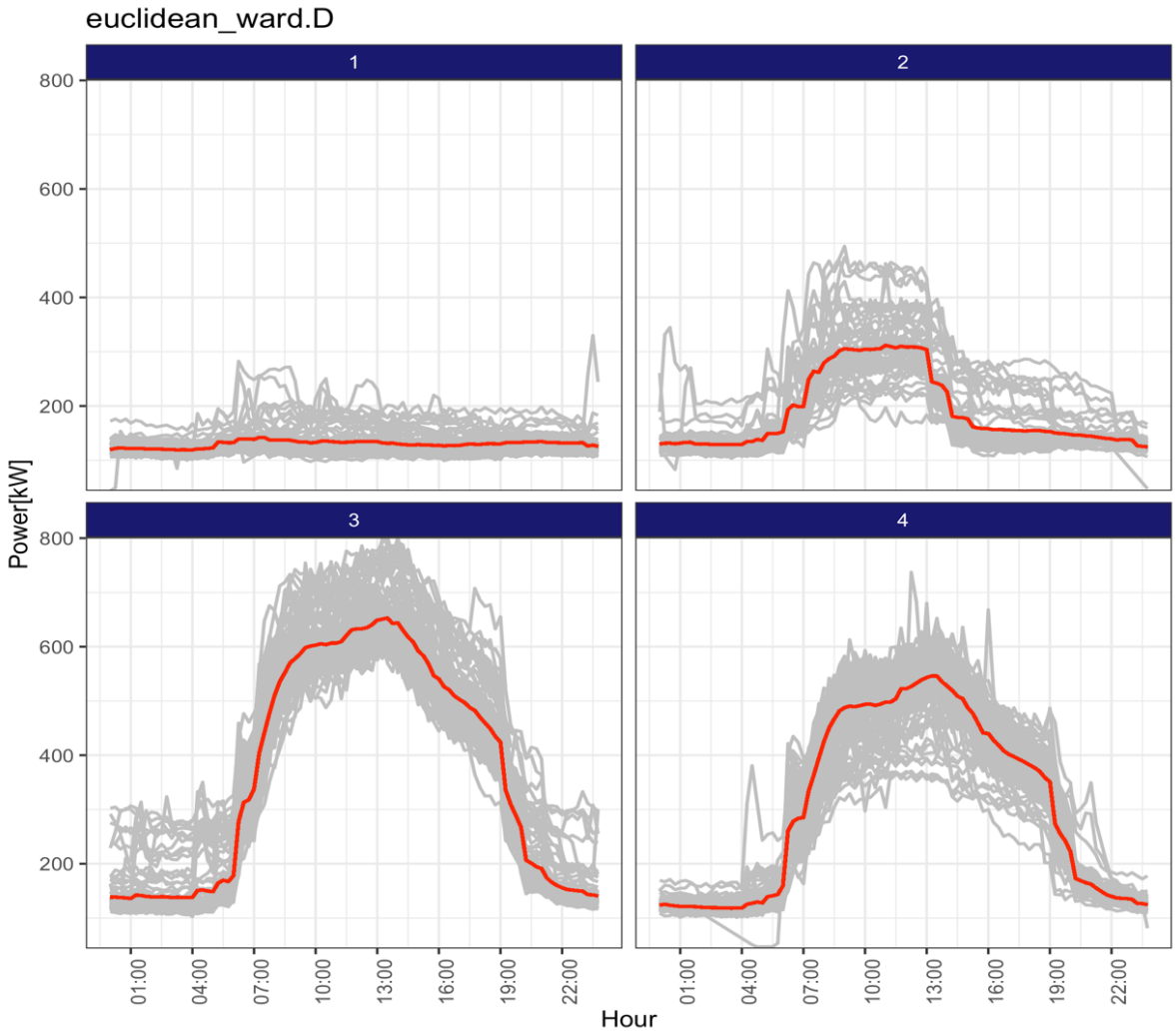


Figure 4-2: Cluster analysis performed on Total Power time series

4.2 DAILY CONTEXTS

This section is entirely dedicated on daily contexts. The context is a powerful tool introduced by Contextual Matrix Profile.

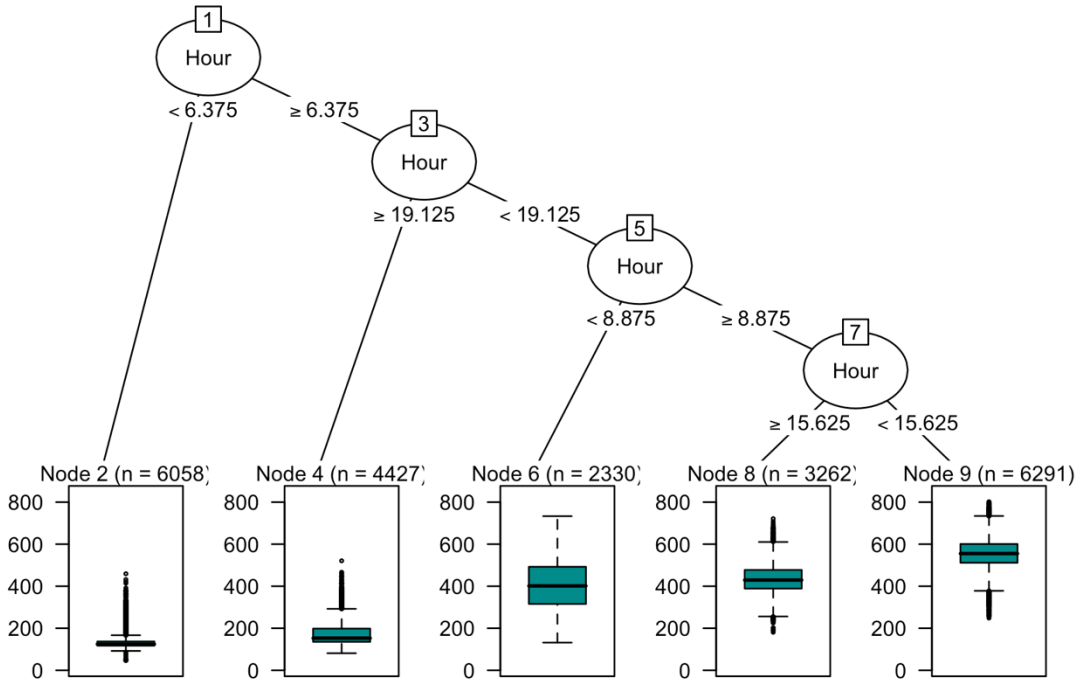


Figure 4-3 : Decision Tree with root, decision, and leaf nodes

The meaning of the context is simple, it is essentially a daily period where anomalous subsequence could start. Detect anomalies is not simple task. First of all, it is mandatory define what is anomalous. In building energy, anomalies are mainly of two kinds, magnitude anomalies and shape anomalies, they may come from components system faults rather than from particular boundary condition like abnormal external temperature or occupancy condition. Another possible event is the lack of datum due to meter failures. The possibility of detecting anomalies in different day periods and day cluster give us the chance to include in our analysis as large number of anomalies as possible, even those that do not come out with traditional methods.

Having said that, the rest of the chapter summarizes the results of CART method and the control parameters in input. The Figure 4-3 is a tree structure with root node, decision nodes and leaf nodes. The target variable in the leaf nodes is the

Total Power consumption, which is continuous variable, so this decision tree is of regression type. Root node and decision nodes have as splitting variable Hour of the day which in turn will be used to define contextual subsequences [2]. To set a good value for time window a lot of tests has been made, the best one, suggests a time windows of length two hours. In the Table 4-1 are listed the main context information: the start, the end, number of observations(examples) which fall into the range, duration, and the daily non-overlapping subsequence period of each context, the same thing of Time Window Context. The Subsequences cover entirely each day, later on they could be classified as normal or anomaly, after the Contextual Matrix Profile will be applied. Subsequence one last 6 hours and 15 minutes and could start between the 00:00 and the 02:00, in this way it covers the first hours of the day, where the power is approximately cost.

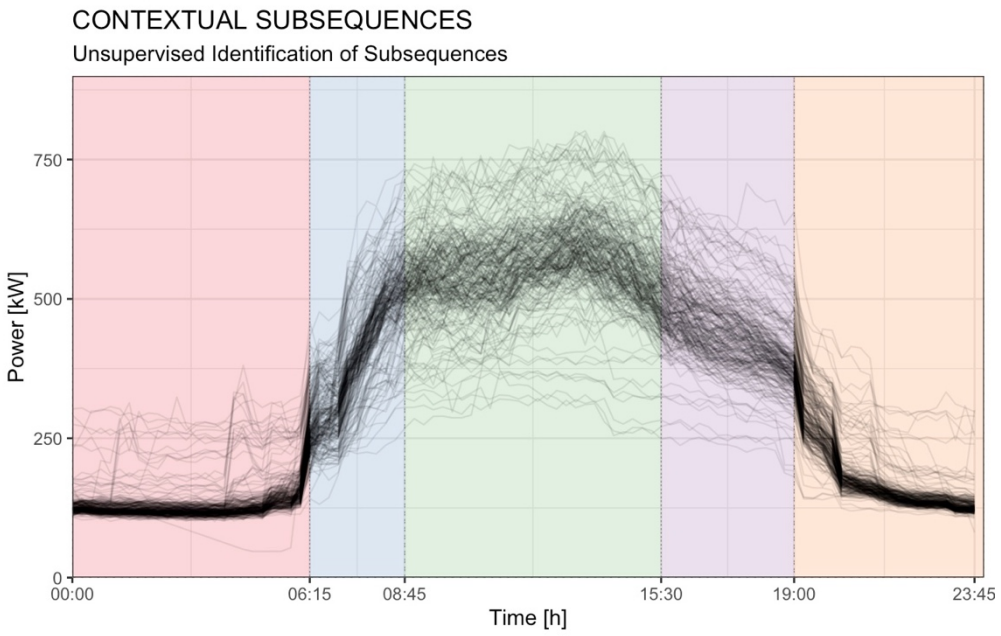


Figure 4-4: daily profiles and contextual subsequences

Table 4-1 : Contexts from CART analysis and relative duration

	from	to	context_subsequence	duration	observations
1	00:00	02:00	Subsequences of 04:15 h that starts between 00:00 and 02:00	2 h	8
2	04:15	06:15	Subsequences of 02:30 h that starts between 04:15 and 06:15	2 h	8
3	06:45	08:45	Subsequences of 06:45 h that starts between 06:45 and 08:45	2 h	8
4	13:30	15:30	Subsequences of 03:30 h that starts between 13:30 and 15:30	2 h	8
5	17:00	19:00	Subsequences of 05:00 h that starts between 17:00 and 19:00	2 h	8

The next subsequence is shorter than the previous one, is focused on the morning power rump-up and lasts less than 3 hours. The middle of the day appears with the maximum electrical load values which have high dispersion with respect data of other subsequences. The Subsequence four opposed to two, concerns the afternoon power rump-down. Lastly the subsequence of evening period, the power of which by analogy with subsequence one (the early hours of the day). The Figure 4-4 reports what has been explained before.

4.3 CONTEXTUAL MATRIX PROFILE OUTCOMES

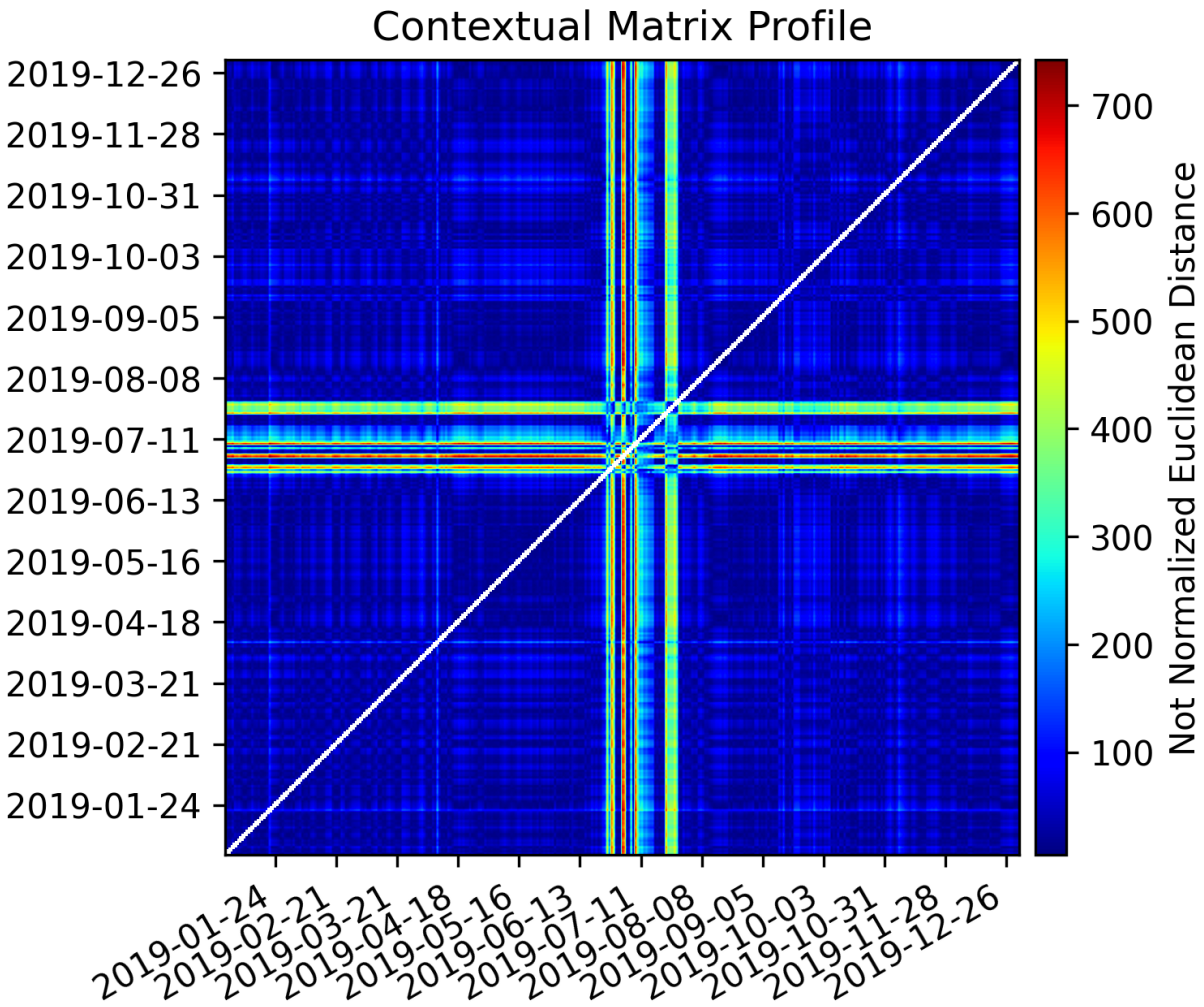


Figure 4-5: Contextual Matrix Profile for Context 1, using not normalized Euclidean Distance

The Figure 4-5 shows Contextual Matrix Profile for Context 1 with subsequence's length of 4 hours and 15 minutes (Contextual Time Window or Contextual Subsequence), starting between the 00:00 and the 02:00 (Context Range). Each Column and each Row accounts for day's Context Range and their intersection represents, as pointed out in the previous chapter, the Euclidean Distance

between the most similar subsequences of 4 hours and 15 minutes length, starting in the Context Range 00:00-02:00; moreover, the subsequences are not normalized when distance measure is applied. The distances in green, higher than the blue ones, are intensive in summer period, this probably means the need for cooling also in the early morning. Therefore, the main information we gain from the Figure 4-5 is about the difference of base load between summer and winter period.

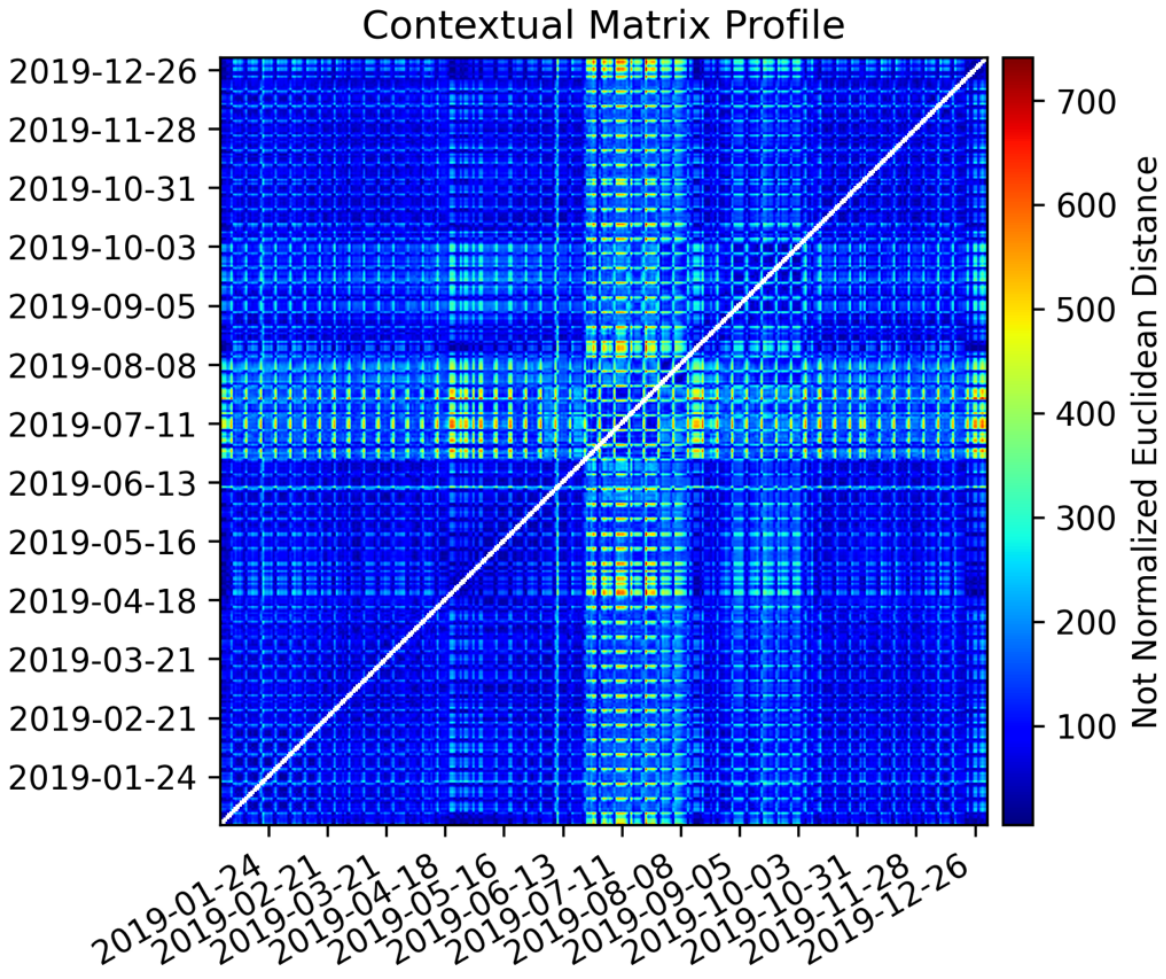


Figure 4-6: Contextual Matrix Profile for Context 2, using not normalized Euclidean Distance

In Contextual Matrix Profile of Context 2, Figure 4-6, at first sight, comes out only the difference between winter and summer months, but with more attention, a slight difference of distances between weekdays and weekend days (recurrent pattern) could be perceived. The reason of this recurrent pattern is explained by dissimilarities in ramp up shape of weekdays compared to that of weekend days one.

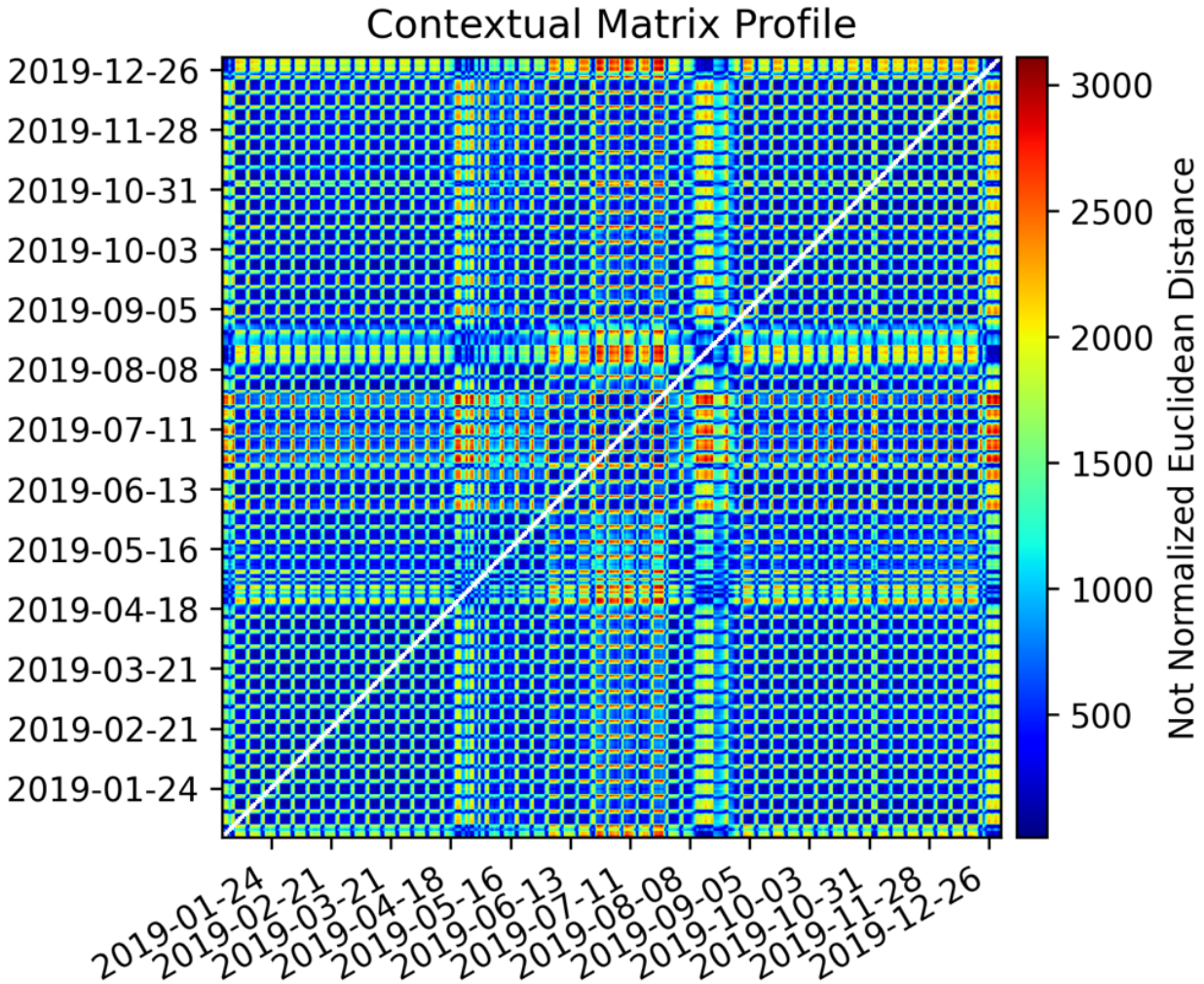


Figure 4-7: Contextual Matrix Profile for Context 3, using not normalized Euclidean Distance

The Matrix Profile for Context 3, Figure 4-7, compares the highest electrical load of the day and what comes up from patterns is still the different behaviour between

summer day and the rest of the year. Also, the midseason days are so far from winter ones. In Figure 4-8, the green squares are localized nearby Easter holidays, Winter holidays and the first two weeks of August. This distances distribution is fully influenced by drivers like external Temperature and occupancy. Lastly, we can clearly see a periodic pattern caused by weekdays versus weekends days.

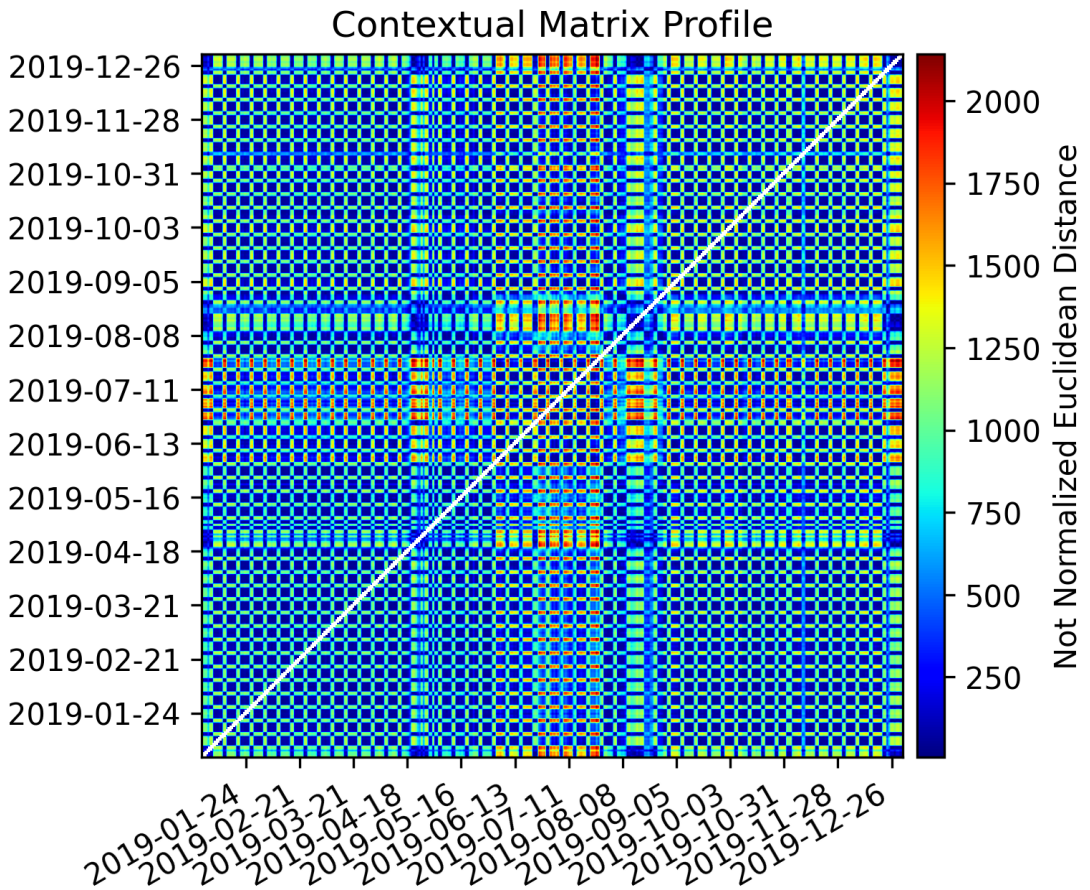


Figure 4-8: Contextual Matrix Profile for Context 4, using not normalized Euclidean Distance

The Contextual Matrix Profile above, Figure 4-8 records Euclidean Distances for the early afternoon period, Context 4. The highest distances are in proximity of holidays and the weekends. The red distances in summer period are instead representative of the variability of summer boundaries conditions which impact the energy consumption profiles and of state of building occupation (all the

activities, this period, are reduced to minimum). The last Context, by analogy with the previous ones, suggests external Temperature as main driver of power profile shape, and again the dissimilarities in base load among winter days, summer days and holidays.

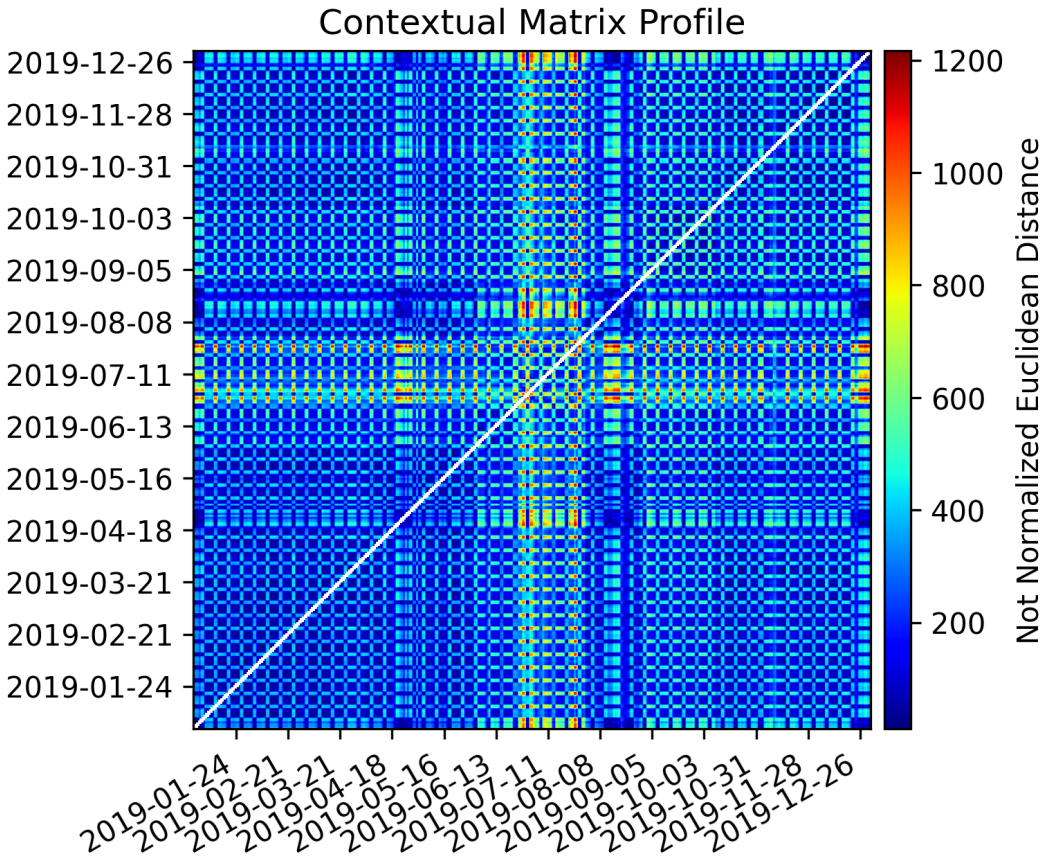


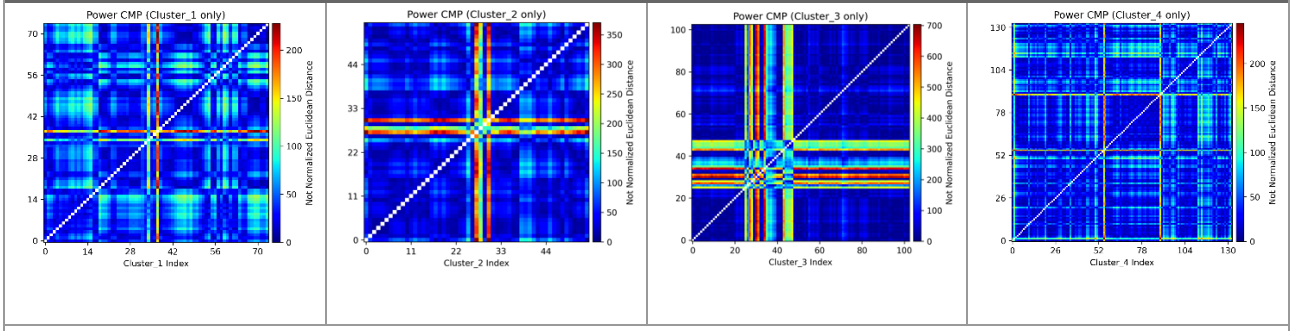
Figure 4-9: Contextual Matrix Profile for Context 5, using not normalized Euclidean Distance

4.3.1 CONTEXTUAL MATRIX PROFILE BY CONTEXTS AND CLUSTERS

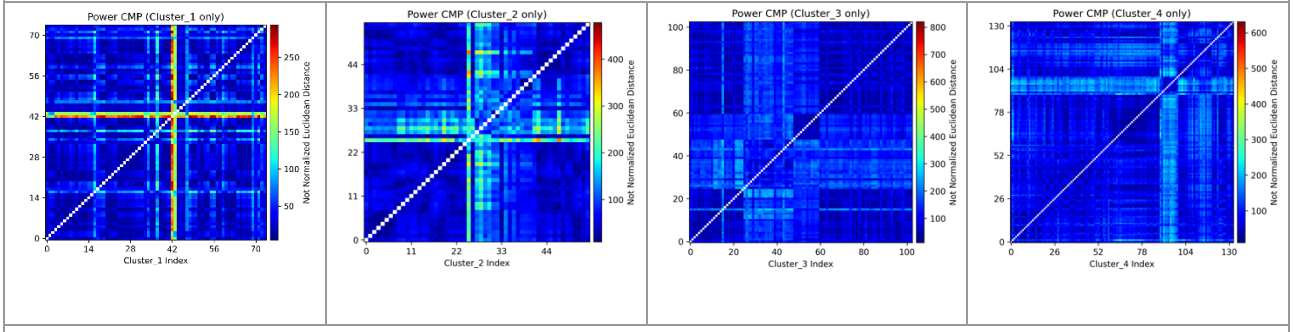
The further in-depth analysis focus on each cluster and aims at identifying unique operating patterns which are synonyms of anomalies. The anomaly, here, belongs to context day period but also to clustered day of the year. This way of carrying out Anomaly Detection give us the chance to exclude from analysis comparisons between different class of days, avoiding bias results. Cluster 1 holds

about seventy days with flat profiles (Sundays or holidays); if we carefully look at the Matrix Profile in Figure 4-10 (the top rows), some columns or rows are marked

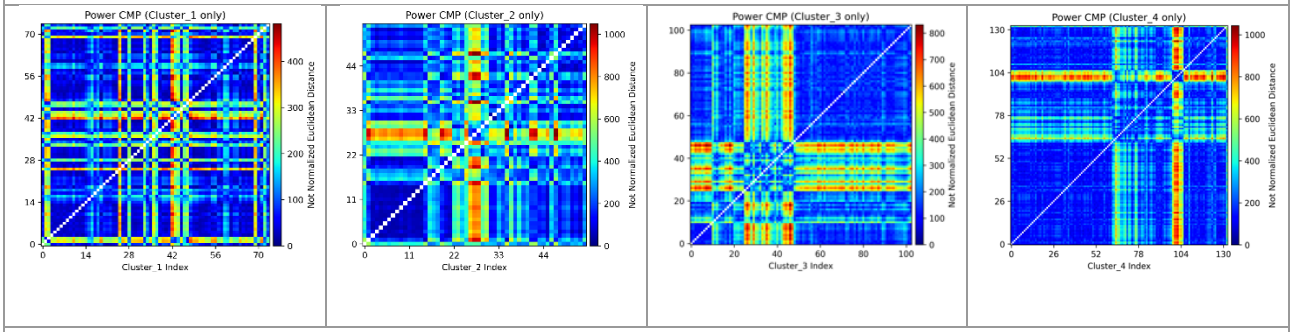
MATRIX PROFILE BY CONTEXT 1 AND CLUSTERS



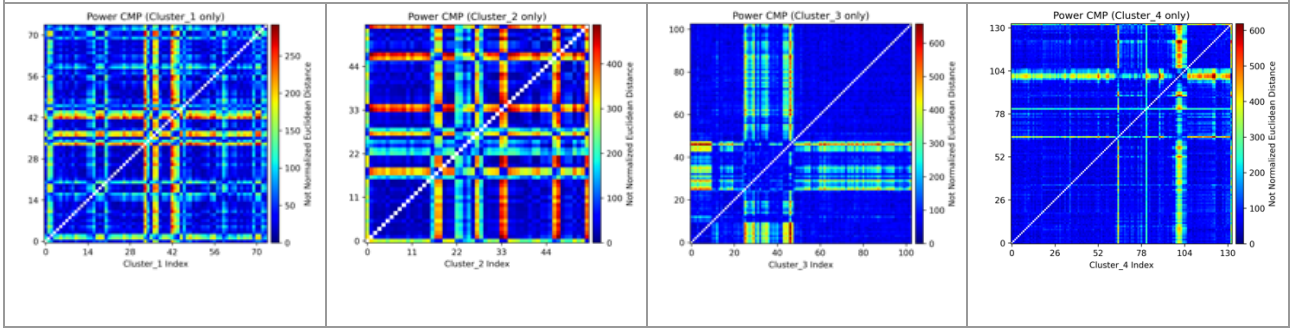
MATRIX PROFILE BY CONTEXT 2 AND CLUSTERS



MATRIX PROFILE BY CONTEXT 3 AND CLUSTERS



MATRIX PROFILE BY CONTEXT 4 AND CLUSTERS



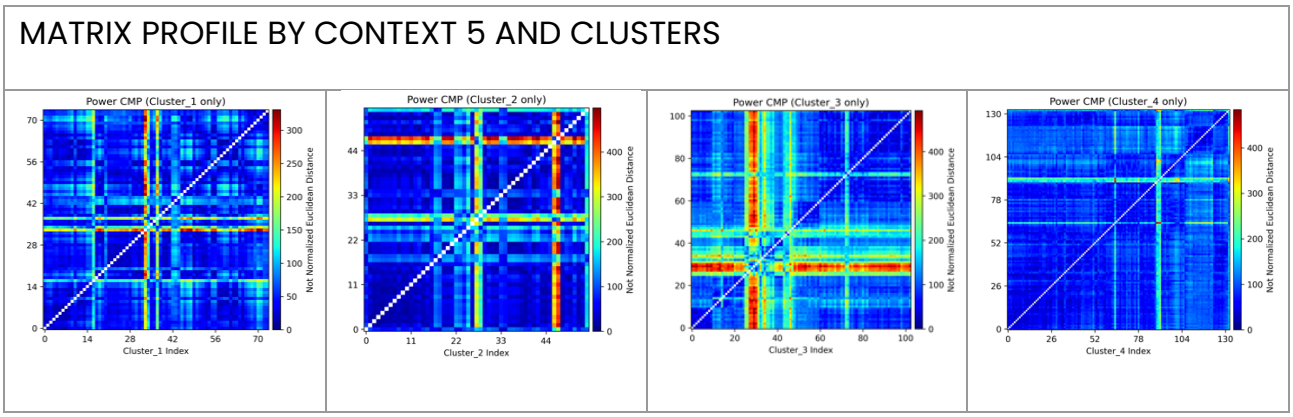


Figure 4-10: Depictions of each Contextual Matrix Profile by Context and Cluster

in red (great distances) and are so far from the other ones; likely, they are anomalies. The anomalies of Context 1 are distributed in the first three cluster, instead the Context 2 presents the most severe anomalies only in Cluster 1, on Sunday. The Context 3 which contains the maximum power loads shows anomalies in almost every cluster, whereas in the Context 4 the red highest anomalies are concentrated in Cluster 2. The last row of the Figure 4-10, the Context 5, that is, the evening consumption period, shows the highest anomalies in Cluster 2. We may conclude: the Contextual Matrix Profile by Cluster and Context shows the dissimilarities between the same contexts of different Cluster days, but it is required another kind of visualization, focused on, operating profile to better capture the nature or the trend of abnormal profile.

4.3.2 ANOMALIES RESULTS

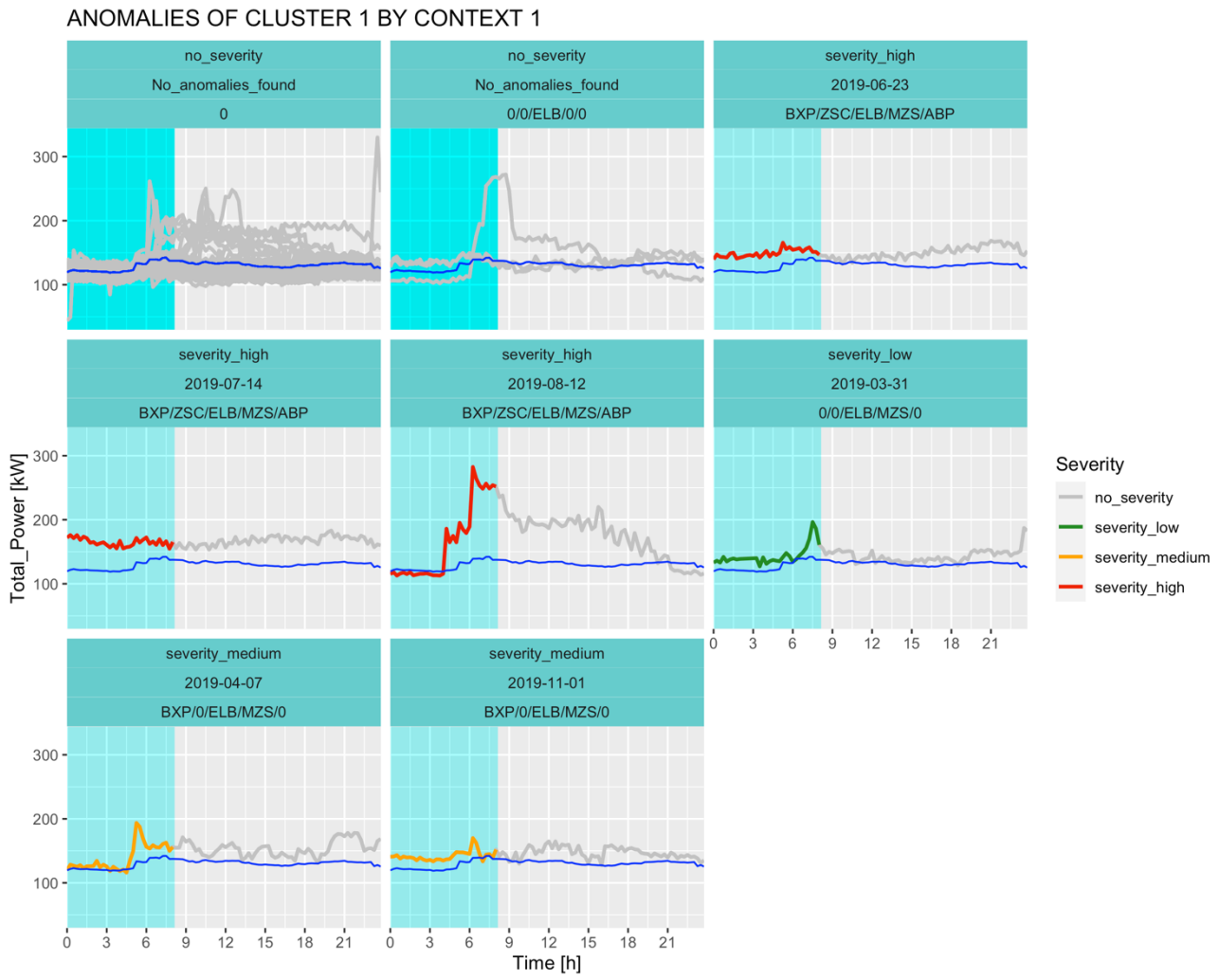


Figure 4-11: Time Series Anomalies by Cluster 1 and Context 1

This paragraph shows anomalies result organized in the following way, each figure holds the anomalies which have been found by the methodology, classified by Clusters, Contexts and Severity. The grey lines in Figure 4-11 are the profiles belonging to Cluster 1 while the blue one is the centroid of cluster. For the moment, we are focusing on Context 1, and what stand out are those subsequences with abnormal values. The red Time Series snippet on 2019/08/12 is abnormal in terms of values and shape and like the red one on 2019/07/14 seems to be driven by external temperature, it is maybe an energy overconsumption. The anomalies labelled with severity medium or low and marked with yellow or green lines are

less severe (not so different from cluster centroid) but have some spikes that makes themselves anomaly.

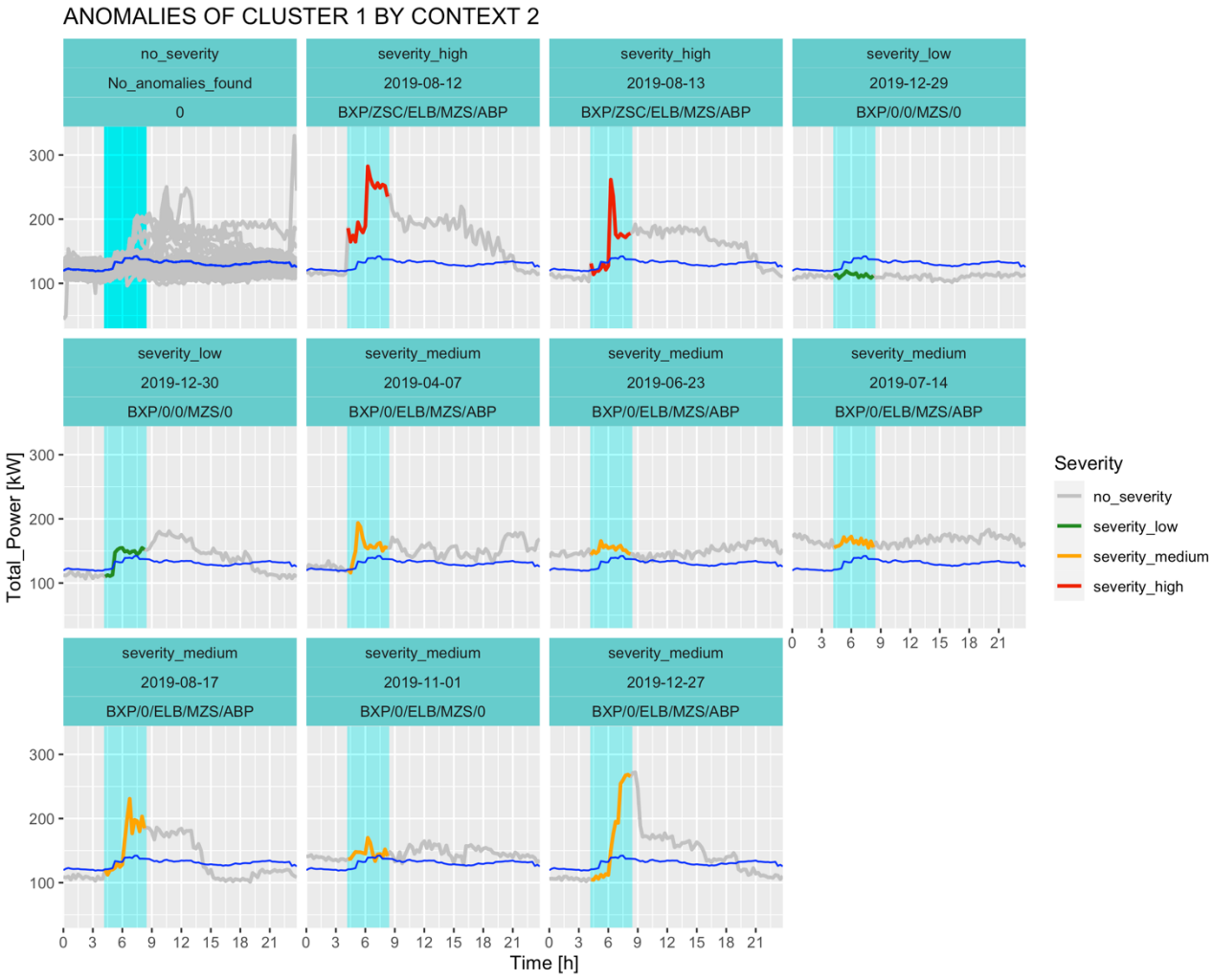


Figure 4-12: Time Series Anomalies by Cluster 1 and Context 2

In Figure 4-12 we could observe abnormal behaviour, especially on 2019/08/13 in which an evident spike reaches 285 kW, so far from the cluster centroid which has a corresponding power value of 150 kW. The same anomaly trend occurs on 2019/12/27 but with medium severity. Another interesting severity occurs on 2019/08/17, the shape of the yellow snippet cannot be associated to abnormal

external driver, instead the sudden jump of power value leads us to think an incorrect data reading.

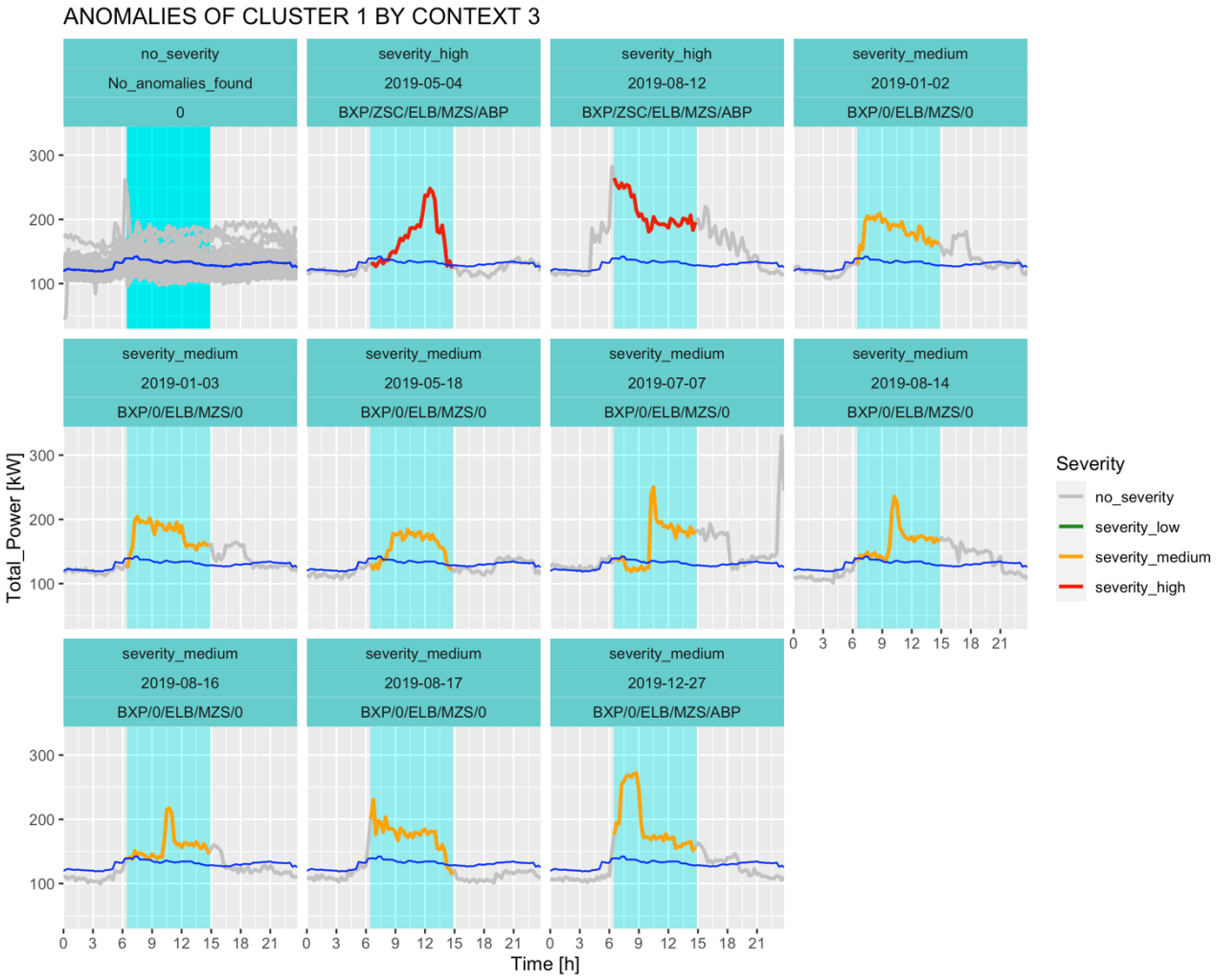


Figure 4-13: Time Series Anomalies by Cluster 1 and Context 3

The anomalies of Cluster 1 and Context 3, Figure 4-13, are related to the shape of Time Series snippets and the spikes could be related to sensor incorrect readings. Instead, the red subsequences are very dissimilar from the centroid so the related energy consumption could be associated with an unusual activity schedule.

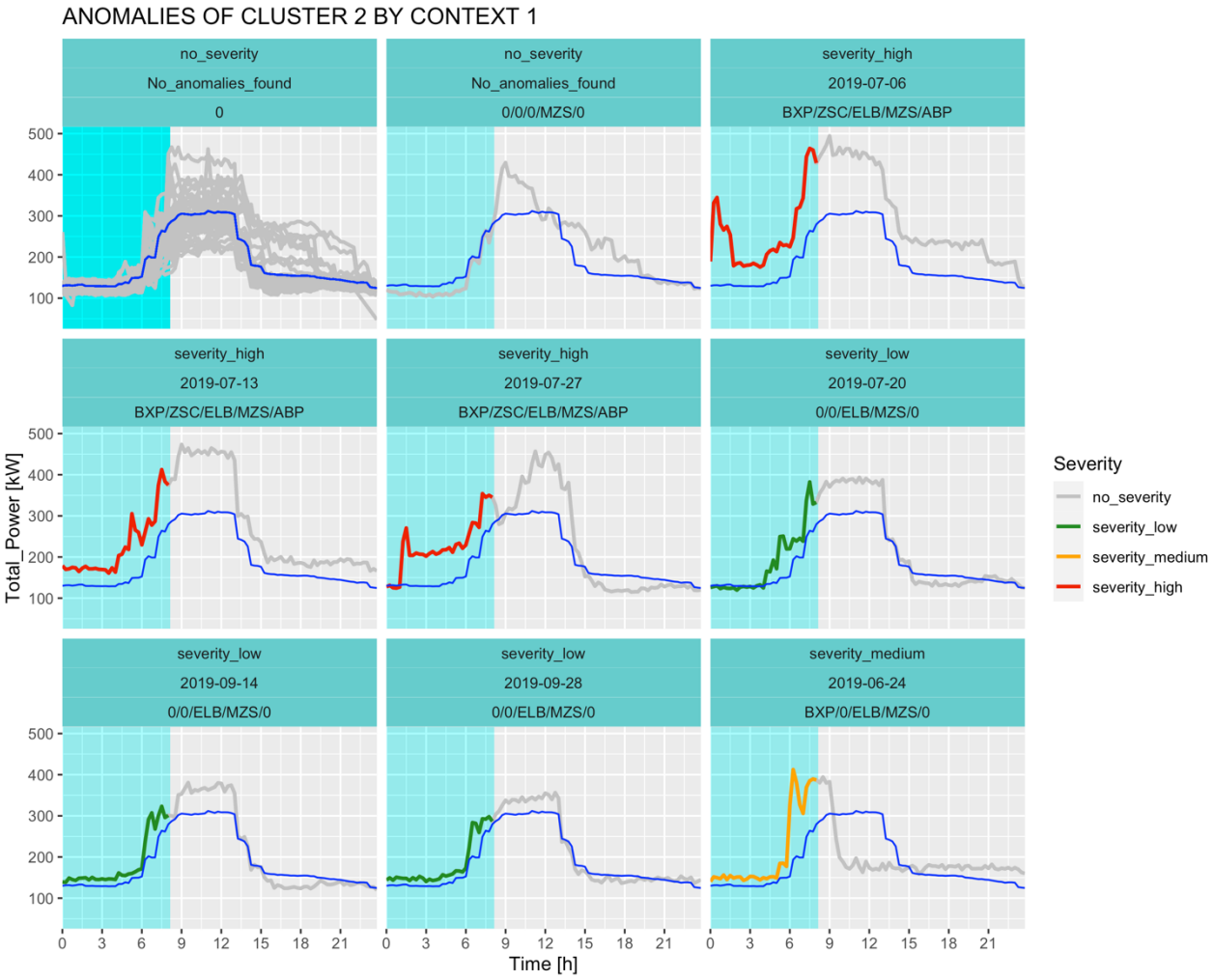


Figure 4-14: Time Series Anomalies by Cluster 2 and Context 1

The Figure 4-14, brings back the anomalies of Cluster 2 and Context 1; the most severe ones reach abnormal values e don't follow the trend of centroid. On 2019/07/06 but also on 2019/07/27 there are some spikes in the early morning, and the power values are extremely high, these are the reasons why they have been labeled as severity high. Whereas the subsequence shape in yellow on 2019/06/24 at a certain point deviate from the centroid and becomes anomalous.

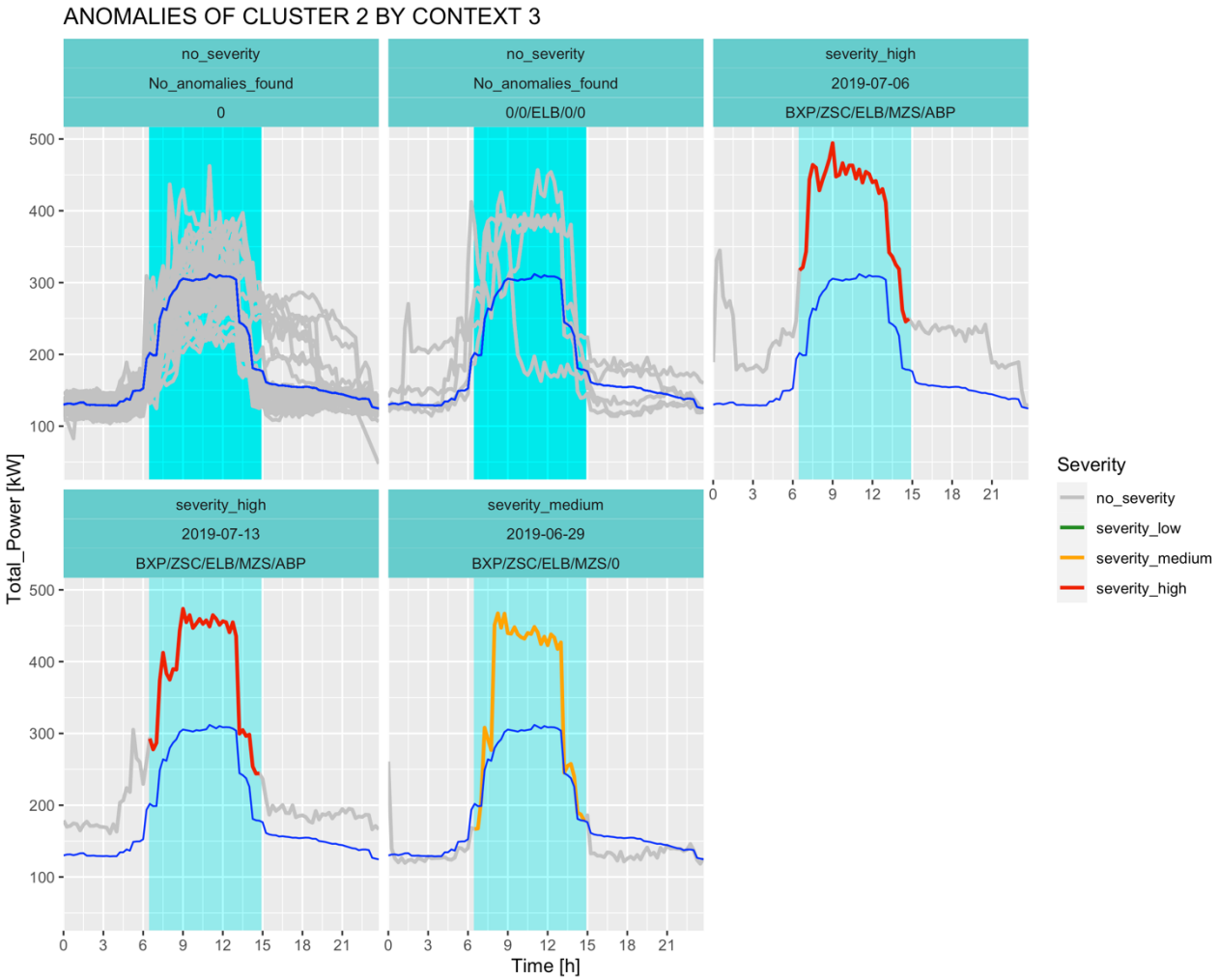


Figure 4-15: Time Series Anomalies by Cluster 2 and Context 3

The previous Figure 4-15, shows clearly what we mean with the term ''magnitude anomaly''. The loads in red and yellow are steadily far from centroid even if the shapes are very similar to the centroid one. Once again, we are not sure about the root case, we could assume that the main driver of this trend is the high external temperature of the middle of the summer day. To answer this doubt, we have to query the sub-load labeled as ''Refrigeration Unit '', if we find high consumption also in ''Refrigeration Unit'', we could validate our starting assumption and state that these profiles are uncommon rather than abnormal.

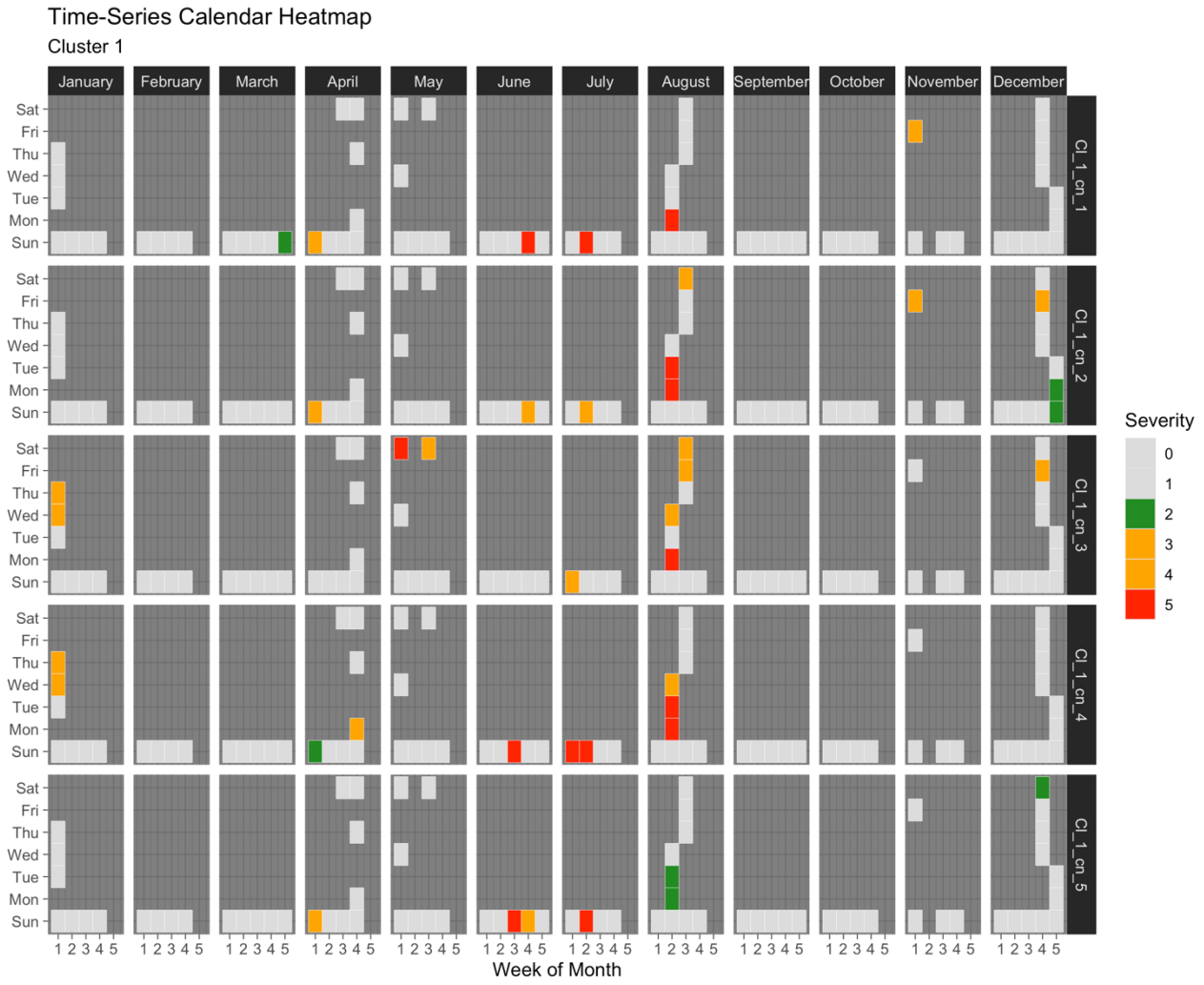


Figure 4-12: The Heat Map Calendar Anomalies by Cluster 1

The figure above, Figure 4-12, is a Calendar Heat Map of the anomalies of cluster one. This kind of visualization is intuitive and immediate to understand. Each subsequence is associated with small rectangle(day). The rows of the heatmap are linked with a context whereas the columns with months of the year. Finally, the left axis reports the weekdays and the bottom one the week of month. All the light grey rectangles are the normal subsequences, most of them are located on Sunday or on holiday.

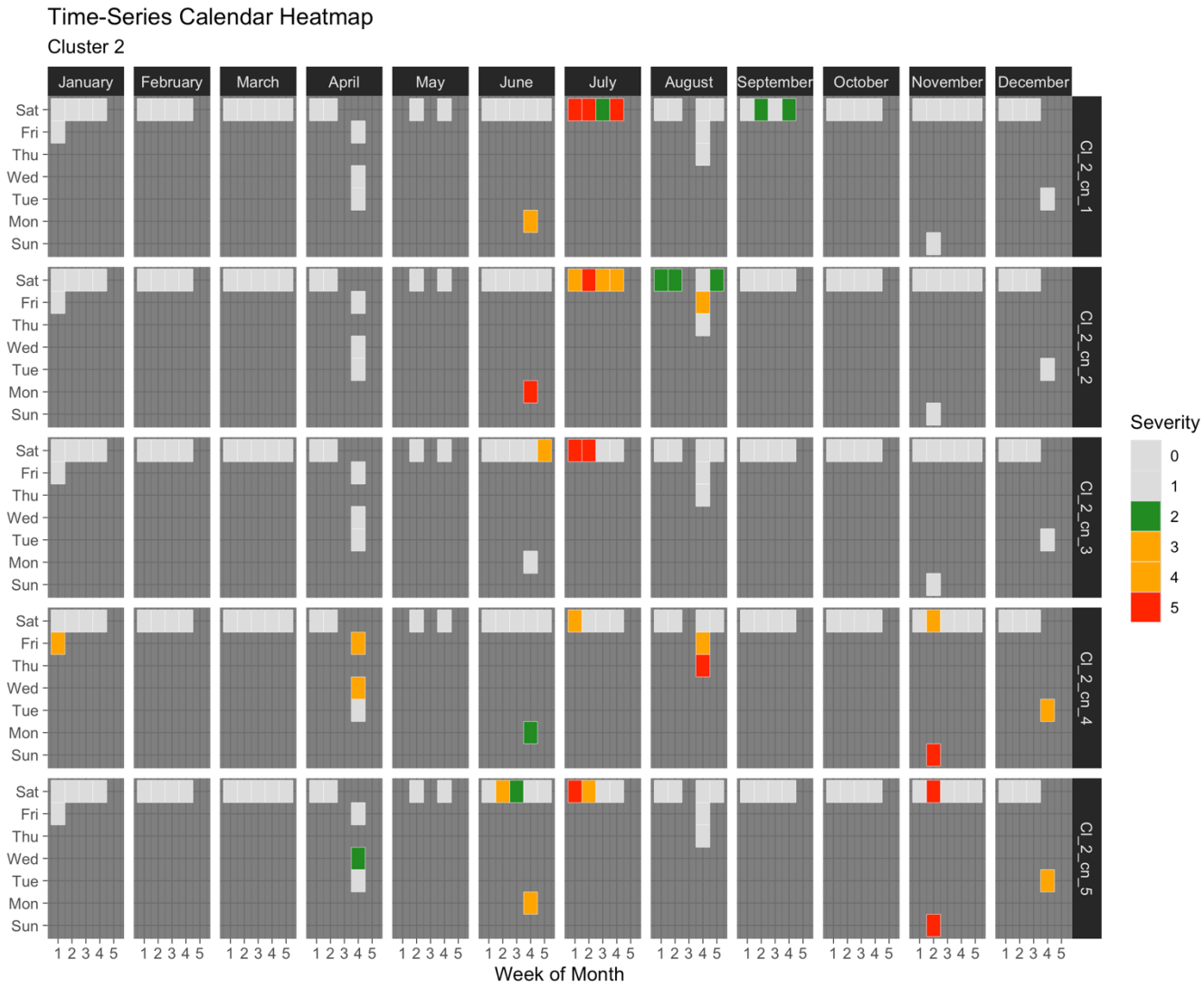


Figure 4-13: The Heat Map Calendar Anomalies by Cluster 2

The second Heat Map calendar displays the anomalies of cluster 2. The light grey rectangles are half a days' work or Saturdays. The main anomalies, marked in red or orange, once again, can be found in Summer and are the so called 'Thermal Sensitive' anomalies. The ones we are interested in are located in the mid-seasons' days in which the energy consumption is not highly influenced by external drivers. The last two figures, Figure 4-14 and Figure 4-15 include the results of Cluster 3 and Cluster 4 respectively. Now that anomalies have been identified, we have to establish which could be associated with a failure condition.

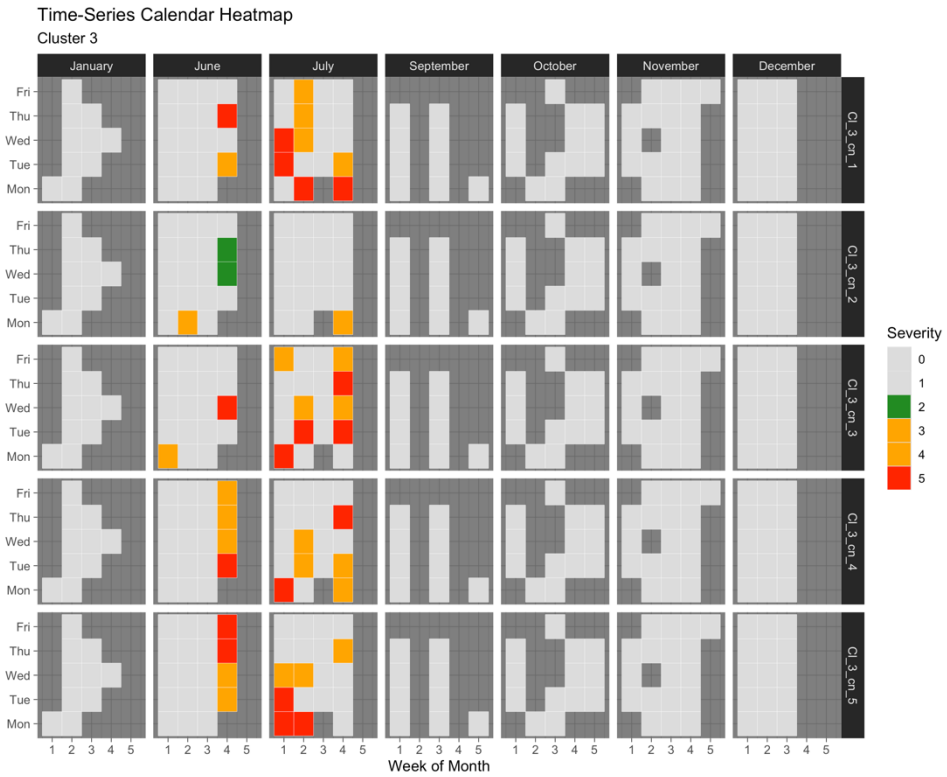


Figure 4-14: The Heat Map Calendar Anomalies by Cluster 3

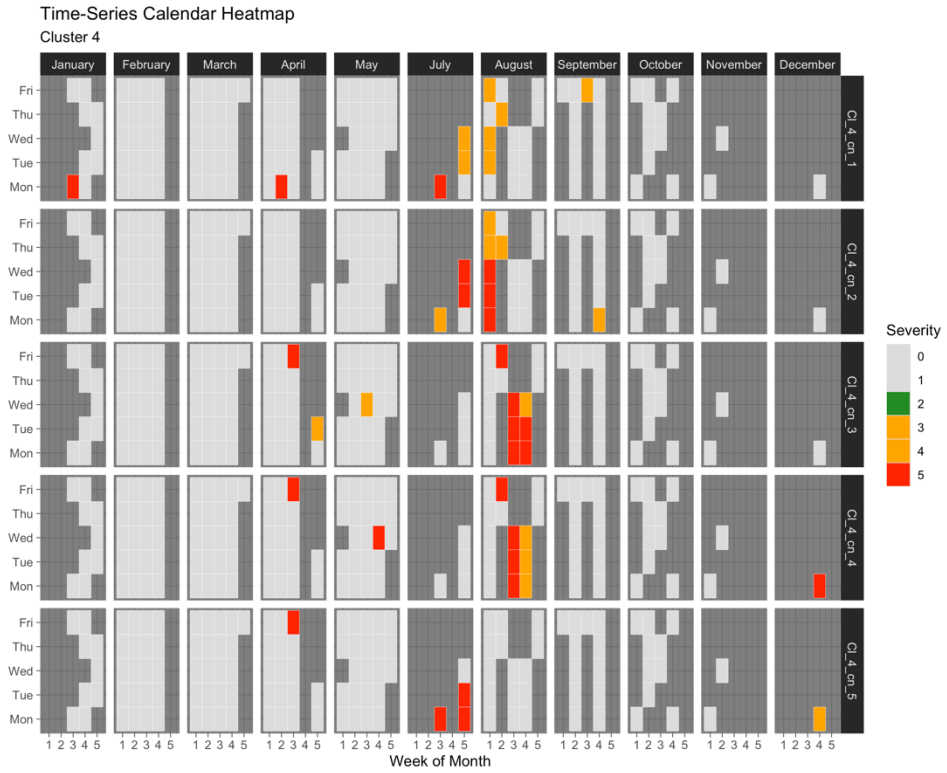


Figure 4-15: The Heat Map Calendar Anomalies by Cluster 4

4.3.3 SAVING POTENTIAL

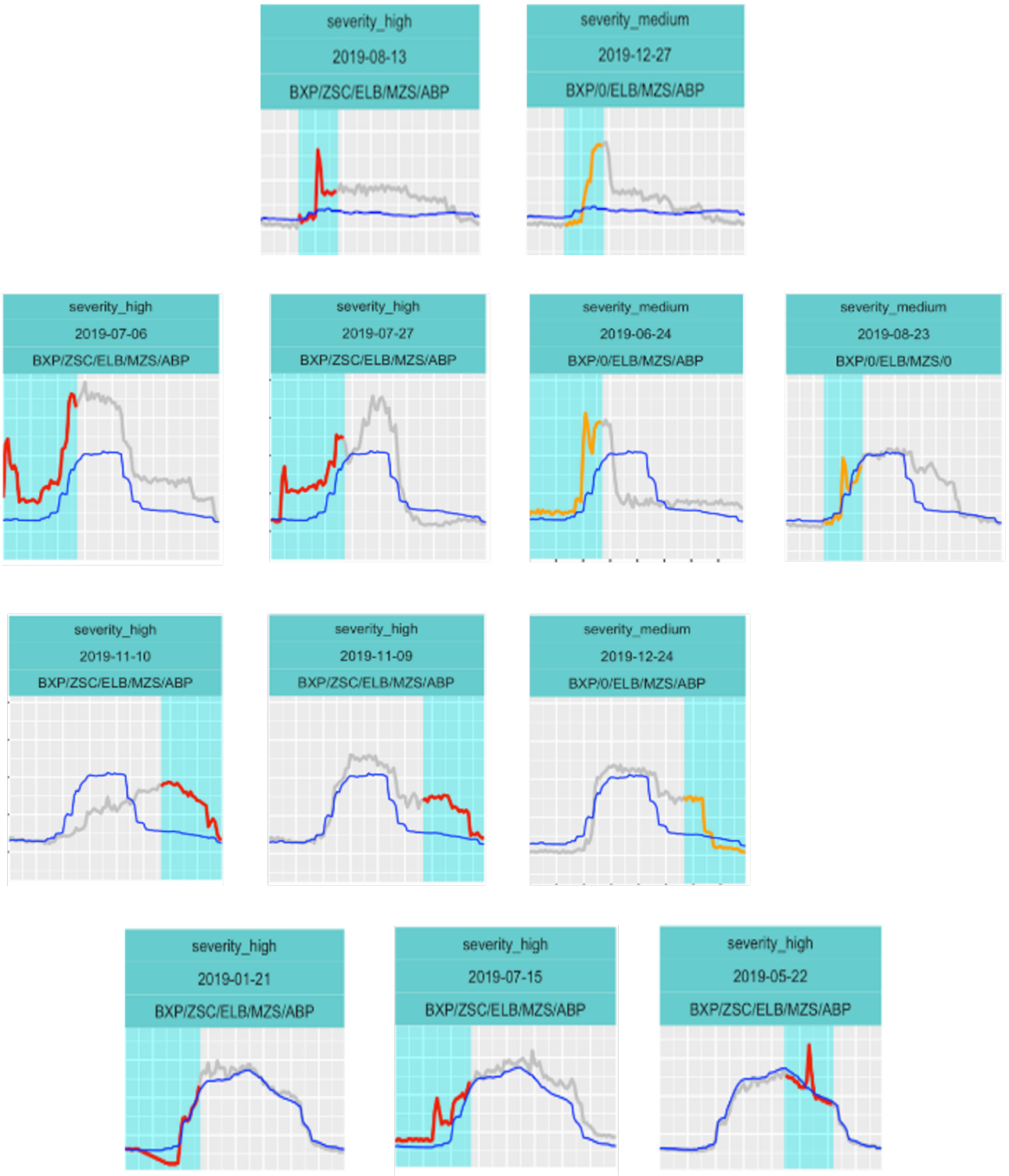


Figure 4-16: subsequences comparable to failure conditions or faults

The figure above, Figure 4-16, shows those anomalies that can be configured as failure conditions. Their typical characteristics are inconsistent punctual values (spikes) as on 2019-05-2022, incorrect readings as on 2019-01-21 and extremely high electrical load as on 2019-11-10.

The Table 4-1 reports the saving estimated as explained in section "Methodology". This type of saving could be actionable if the anomalies were saved in a database and exploited subsequently, comparing them with new data observation. Whereas the figure above contains all the subsequences comparable to failure condition, incorrect readings.

Table 4-2: Saving

Dates	Theoretical Potential Saving [kWh]
12/08/19	366.67
13/08/19	70.62
27/12/19	149.43
06/07/19	559.47
27/07/19	235.95
24/06/19	102.18
23/08/19	24.04
10/11/19	292.59
09/11/19	123.53
24/12/19	31.29
15/07/19	479.53
22/05/19	38.21

5 CONCLUSION AND NEXT STEPS

The work carried out in this master's thesis final project deals with new framework for automated Anomaly Detection at meter level. The starting point of this work is the investigation of Matrix Profile technique. Unfortunately, there are not many scientific papers about Matrix Profile, probably, because the Time Series topic is not so easy to handle. Up to the present day the application fields of Matrix Profile in literature concerns mainly biomedical field and robotics. Although the MP needs to be refined, it is conquering the interest of many Data Analysts and represents the state of art for Time Serie data. The concept of anomaly is a bit complex, there is not only a kind of anomaly, but we expect abnormal profile shape (shape anomaly) and magnitude anomaly, that is, a certain variable could take values so far from the rest ones. The Contextual Matrix, which is more suitable for energy related problem, thanks to its flexibility, is an improvement respect to classical MP. The introduction of contexts allows us to search anomalies in a particular day time, moreover we could thoroughly inspect the type of anomaly and the time location. In this sense CMP can be seen as a step forward respect to traditional MP which treats subsequence as abnormal only if it has maximum distance from its nearest neighbour. The Methodology built up involves, in addition to CMP, other algorithms like CART and hierarchical cluster analysis, resulting globally parameter light. The parameter tuning analysis of CART and cluster algorithm is required in order to fit methodology to the case study, but we have tried to reduce the free parameter in order to keep our methodology as automated as possible. Respect to other anomaly detection framework available in scientific literature, we could leverage on ''The Highly

Desirable Properties of the Matrix Profile'' [12] that are partially inherited by Contextual Matrix Profile.

Here are some of them:

- '' It is exact'': Matrix Profile doesn't provide false positives, when motif discovery, discord discovery, Time Series joins is performed
- '' It is simple and parameter-free'', the only parameter set is time window m (usually known by domain expert and thus it does not require tuning analysis).
- '' It is space efficient'': Matrix Profile construction algorithms take up little space in memory, linear in the time series allowing for big Time Series processing.
- '' It is incrementally maintainable'': we can continuously update our Matrix Profile, so it is possible keeping joins, motifs, discords exactly on streaming data
- Matrix Profile construction is parallelizable
- '' It is free of the curse of dimensionality; it has time complexity that is constant in subsequence length''
- '' It can handle missing data'': Even with missing data, Matrix Profile doesn't provide false negatives.

The work that has been carried out until now could be useful for creating a Dictionary of anomalies, exploitable in a different time period from the training one. This Dictionary could be updated every retraining and could be linked to an EMIS warning system. In this way occupants could be conscious of the occurring anomaly but with the previous information are not aware of the anomalies root

causes. Therefore, the possible future step could be about the use of Contextual Matrix Profile for an in-depth diagnostic analysis with the aid of sub loads massive dataset.

BIBLIOGRAPHY

[1] A. R. Arko, “Anomaly Detection In IoT Using Machine Learning Algorithms,” 2019.

[2] A. Capozzoli, M. S. Piscitelli, S. Brandi, D. Grassi, and G. Chicco, “Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings,” *Energy*, vol. 157, pp. 336–352, Aug. 2018, doi: 10.1016/j.energy.2018.05.127.

[3] H. Kramer *et al.*, “Proving the Business Case for Building Analytics,” 2020, doi: 10.20357/B7G022.

[4] F. Xiao and C. Fan, “Data mining in building automation system for improving building operational performance,” *Energy and Buildings*, vol. 75, pp. 109–118, Jun. 2014, doi: 10.1016/j.enbuild.2014.02.005.

[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] C. C. M. Yeh *et al.*, “Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile,” *Data Mining and Knowledge Discovery*, vol. 32, no. 1, pp. 83–123, Jan. 2018, doi: 10.1007/s10618-017-0519-9.

[7] 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.

[8] 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.

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

[10] 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.

[11] A. Kolbaşi and A. Ünsal, “Science Stays True Here,” Science Signpost Publishing.

[12] 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

ADDITIONAL READINGS

[13] B. Lantz, *Machine learning with R : learn how to use R to apply powerful machine learning methods and gain an insight into real-world applications*.

[14] Z. Yu, B. C. M. Fung, and F. Haghighat, “Extracting knowledge from building-related data - A data mining framework,” *Building Simulation*, vol. 6, no. 2, pp. 207–222, Jun. 2013, doi: 10.1007/s12273-013-0117-8.

[15] Y. Wang, Q. Chen, C. Kang, M. Zhang, K. Wang, and Y. Zhao, “Load Profiling and Its Application to Demand Response: A Review,” 2015.

[16] C. Miller, Z. Nagy, and A. Schlueter, “Automated daily pattern filtering of measured building performance data,” *Automation in Construction*, vol. 49, no. PA, pp. 1–17, 2015, doi: 10.1016/j.autcon.2014.09.004.

[17] M. Molina-Solana, M. Ros, M. D. Ruiz, J. Gómez-Romero, and M. J. Martin-Bautista, “Data science for building energy management: A review,” *Renewable and Sustainable Energy Reviews*, vol. 70. Elsevier Ltd, pp. 598–609, 2017. doi: 10.1016/j.rser.2016.11.132.

[18] C. Fan, F. Xiao, and D. Yan, “Advanced data analytics for building energy modeling and management,” *Building Simulation*, vol. 14, no. 1. Tsinghua University, Feb. 01, 2021. doi: 10.1007/s12273-020-0733-z.

[19] A. Capozzoli, M. S. Piscitelli, and S. Brandi, “Mining typical load profiles in buildings to support energy management in the smart city context,” in *Energy Procedia*, 2017, vol. 134, pp. 865–874. doi: 10.1016/j.egypro.2017.09.545.