Meter Level Electrical Load Anomaly Detection using Contextual Matrix Profile

Red pill or blue pill? Unveiling energy management in buildings with matrix profile

Roberto Chiosaa, Marco Savino Piscitellia, Alfonso Capozzoli a[[1]](#footnote-1)\*, Cheng Fanb

a Department of Energy “Galileo Ferraris”, TEBE Research Group, BAEDA Lab, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy

b affiliation

Abstract

The ……

*Keywords:* Anomaly Detection, Matrix Profile, Energy Information System

1. Introduction

The rapidly growing electrification of buildings energy systems and appliances lead to an increasing electricity demand. On a global scale, direct and indirect CO2 emissions from buildings energy use reached its all-time high in 2019 [1]. The European Commission [2] estimates that buildings sector account for 40% of final energy use and 36% of CO2 emissions. The targets imposed by the European community to reduce greenhouse emissions by at least 55% by 2030 [3] highlight the critical role played of buildings. Considering that [4] estimates that almost 90% of the total energy consumed during the life cycle of a building depends on the building operation, reducing energy consumption, increasing appliances efficiency and prevent energy wastes through an effective energy management is the key to meet climate change goals.

In the last few years, the increasingly widespread use of IoT sensors for building energy monitoring led to an unprecedented acquisition of reliable and accessible real-time data, supporting the transition from the so-called *smart buildings* into more complex energy ecosystems called *cognitive buildings* [5]. Although a great deal of research has been done, the increasing volume of collected building energy data still overwhelms end-users, making it hard to spot energy reduction opportunities, find the root cause of energy anomalies or simply be aware of energy usage in buildings and systems. In the last few years data gathered in the building sector reached the order of zettabyte [6] making buildings not only and energy intensive but information intensive [7]. Building data are heterogeneous and reflects the complex interaction that occurs between occupants, energy systems, the building envelope, and external conditions. Managing those data is not trivial, however if properly managed ingested and analysed, provide the opportunity to gain insight on the building operational behaviour discovering opportunities for savings [8].

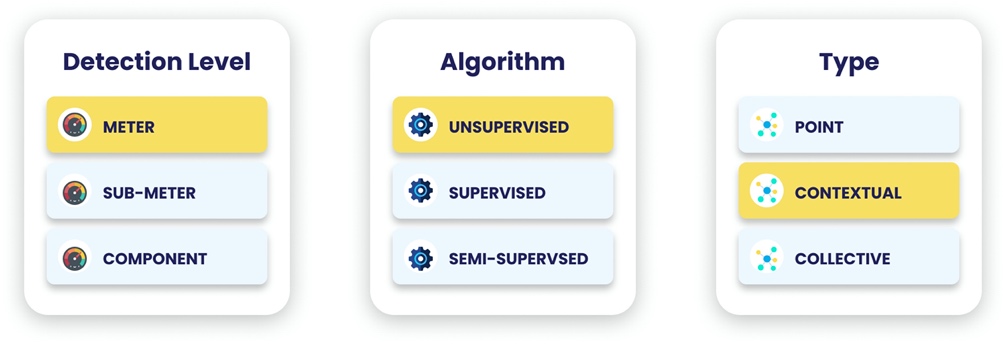
A robust coupling of IoT sensors data, machine learning approaches and energy domain has been proved to be effective in terms of energy savings opportunities to variety of tasks: pattern recognition, energy consumption forecasting, anomaly detection and diagnosis, advanced benchmarking, load profiling, and schedule optimization of building energy systems.

In this paper, we focus anomaly detection of electrical loads in buildings, which is a key application to aid decision makers to reduce wastes and promote sustainable behaviour of end users.

* 1. Anomaly detection: related work

In energy field anomaly detection can be employed in detecting abnormal behaviour of end users, detection of faulty appliance or energy subsystem and spotting technical and non-technical energy loss [9]. Strictly speaking, an anomaly is a region of data with significantly different behaviour from other data and that do not conform to expected values. It can be referred as discord, outlier or exception and its definition is significantly different depending on the field of application. In fact, this classical definition does not take in consideration other form of anomalies that could exist in buildings energy consumptions such as, abnormal occupation patterns, wrong occupants behaviour, incorrect functioning of energy systems, abnormal sub loads consumption and so on [9]. Therefore, when performing anomaly detection in this field is of paramount importance to take into consideration other information sources related to the internal and external environmental conditions, level of detection, occupancy patterns, and join the domain knowledge.

Many categorization have been proposed in literature [10] and some are specific for building environments [6], [9]. The scope of this paper is not to go deep into categorization; thus, we adopted an anomaly classification based on type, level and algorithm as reported in Figure 1.



**Fig. 1.** Classification of anomaly detection method depending on: (a) detection level (b) algorithm (c) anomaly type.

Classification based on type implies a comparison between the observation and the rest of the data. A *point anomaly* is one individual instance or observation that can be considered anomalous when compared to the remaining data. On the other side, a *collective anomaly* is an instance does not represent an anomaly per se, but only if considered within the collection of all the other events instances. Finally, *context anomalies* are anomalies only if considered in a certain context (i.e., boundary conditions) and may not be considered an anomaly if it happens in a different context.

Depending on the detail of electrical load monitored the anomaly detection can be performed at different levels. The *meter level* detection analyses the whole building electrical load, without having any information on the disaggregation of that load among the different sub loads or appliances. *Sub-meter level* detection analyses the disaggregated total electrical load and is usually referred to a specific energy system. Finally, *component level* detection consists in identifying anomalies referring to a given appliance/sensor.

The third is an algorithmic centric classification is based on data-driven anomaly detection techniques. *Supervised* anomaly detection requires to train a machine learning algorithm using labelled dataset (i.e., ground truth) to classify anomalous consumption or not. Although supervised anomaly detection can achieve high-accuracy identification results as demonstrated in academic frameworks, its adoption in real-world is still limited compared to unsupervised methods, due to the absence of power consumption annotated datasets [9]. Examples of supervised algorithms are deep learning, ANN, Regression, Probabilistic models, Traditional classification. On the other side, *unsupervised* anomaly detection consists in detecting rare and unknown anomalous energy patterns without any a priori knowledge. It usually consists in modelling the normal behaviour and then identify patterns that deviates, under the assumption that the number of anomalies is low compared to the observations. Examples of unsupervised algorithms are: … clustering, [11] performs anomaly detection on smart grid though the use of clustering. Finally, there are some semi-supervised algorithms that.

* 1. - Anomaly detection using Matrix Profile

One of the most promising technique for unsupervised anomaly detection in timeseries is Matrix Profile (MP). Introduced by [12] it is a novel algorithm that performs *all-similarity-join-search* among two timeseries, i.e. finding the nearest neighbour for each object of a data collection. Trivial implementations result in excessive computation al time even for modest datasets. Common variants of this problem involve the search of k-nearest neighbour by setting a threshold, which is critical as well as difficult to set [13]. Others perform similarity search by reducing the dimensionality of dataset through PAA ﻿[14], [15] to speed up computation, however, this method causes loss of valuable information.

MP proposes an ultra-fast similarity search under the z-Euclidean distance that does not reduce dimensionality, but calculates the full join, eliminating the need of setting a threshold making the method almost parameter free and exact. The exact and scalable algorithm allows the method to be incrementally maintainable, deterministic in time and so parallelizable on multicore processor to speed up even further the computations.

Given two timeseries and a given subsequence length, the MP algorithm produces two new series: the MP and Matrix Profile Index (MPI). MP is a one-dimensional timeseries that stores the z-normalized Euclidean distance between each subsequence of the first series and the closest matching subsequence (i.e., nearest neighbour) of the second timeseries. MPI is a one-dimensional timeseries that contains the index of where the nearest neighbour is in the second timeseries.

By joining information of MP and MPI many insights could be extracted. Finding the minimum value of the MP is possible to find the best matching subsequence in a series (i.e., motif discovery) on the other side by finding the maximum value of the MP it is possible to find the subsequence with the largest distance to its nearest match, (i.e., discord discovery). In this sense discord discovery may be interpreted as an anomaly detection method that discovers the most unique subsequences in a dataset. Discord discovery using MP as anomaly detection method has been employed with success in different fields.

In medical field [16] proposes an unsupervised real time anomaly detection method based on continuous learning of timeseries shaplets extracted though MP algorithm. Those shaplets are extracted and stored in an anomaly library and then used for anomaly detection in an electro-cardiogram (ECG) timeseries (﻿MIT-BIH database [32]), using a in a sliding window.

An industrial application of anomaly detection is presented in [17] which combines the classical approach of MP with the hamming distance to automatically detect intrusions in the network of a water processing facility.

[18] applies a generalization of MP algorithm called Pan MP find different length anomalies in ﻿automated pedestrian counting system developed in Taipei.

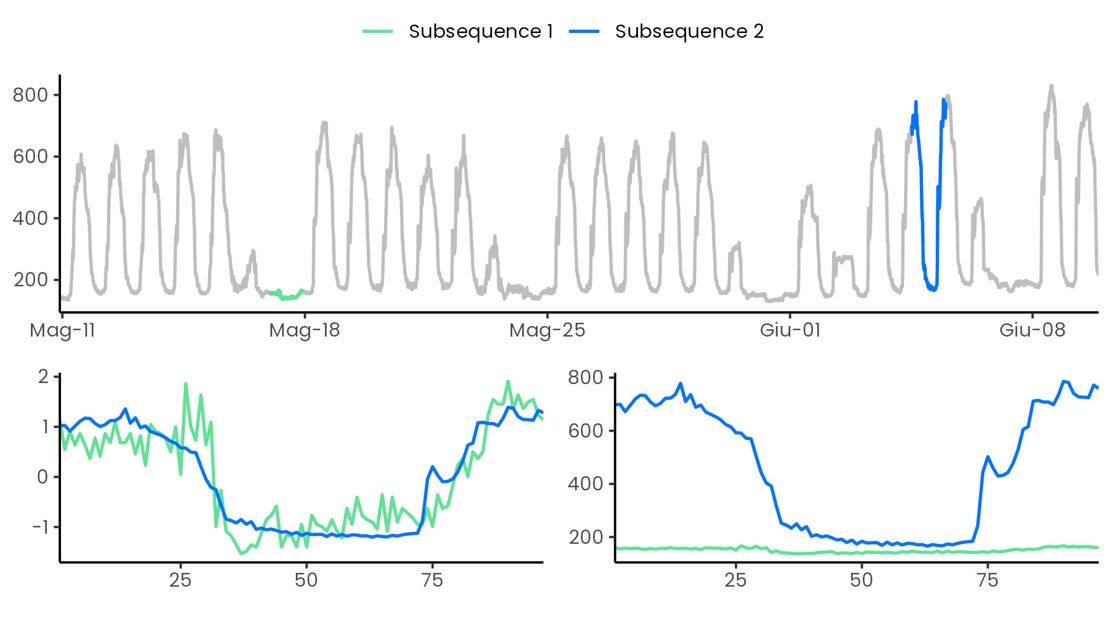
MP is an effective method used to identify anomalies in IT field. [19] introduces ﻿a real time anomaly detection framework based on MP called Real-Time Aggregated Matrix Profile (RAMP), that can identify anomalies in scientific workflows. (building block). [20] Applies a noise elimination technique based on MP on real Yahoo! internet traffic metrics to detect anomalous behaviours; [21] demonstrate how the elimination of noise can help in anomaly detection of noisy date by testing the algorithm on Numenta Benchmark [22].

In the energy field there are few implementations of MP algorithm. [23] Identifies anomalous patterns though a basic application of MP on public building energy traces and then classifies the pattern. [24] demonstrates how MP can be useful in detecting anomalies in different fields in particular in meter swapping and earthquake monitoring. [25] applied an implementation of the classic MP, called Contextual Matrix Profile, in detection of anomalous energy consumption on a ventilation units of three households. [26] applies MP as a part of an automated load profile discord identification (ALDI) based on statistic comparison between normal and anomalous patterns in a large portfolio of buildings.

* 1. Implication MP on energy domain

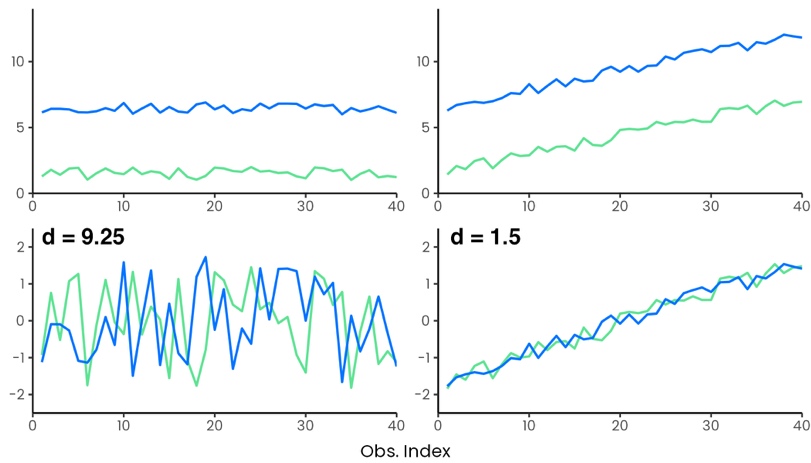
From the previous literature review it emerges that the MP method have been employed successfully in different fields for anomaly detection and the authors have proposed different implementations according to the field of interest. In fact, even if MP is an unsupervised method useful for discord discovery every field have different boundary conditions and different restrictions that cannot be overlooked. In the field of building, the most interesting timeseries are those related to energy consumption. Those timeseries are strictly correlated to many different variables such as occupation, weather conditions, energy systems and so on. A completely unsupervised method may fail to consider the relation with those variables and extract ineffective or trivial results, not useful for anomaly detection.

In buildings, an anomaly is an unexpected behaviour that consists in a strongly different energy consumption. The classic MP, by performing with z-score normalization, searches for each subsequence the nearest neighbour based on shape similarity, however, anomalous shapes not always correspond to anomalous energy consumption, as well as similar shapes in z score not always reflect similar behaviour. With reference to Fig. 2 it is possible to appreciate how three subsequences of electrical load timeseries are considered similar in z score while without normalization the amplitudes are very different reflecting very different energy consumption.



**Fig. 2.** Effect of z-score normalization on two subsequence of electrical load timeseries: (a) electrical load timeseries; (b) comparison between z-score normalized subsequences; (c) not normalized subsequences.

The effect of z-normalization not only does not consider the magnitude of the timeseries but also tends to enhance any fluctuation of the timeseries. By comparing two relatively flat subsequences under z-score normalization the resulting Euclidean distance is higher compared to non-flat subsequences (see Fig.), this results into higher values of MP in flat regions of the timeseries. This issue have been largely analysed in [20] where a smoothing is proposed as possible solution to this issue, beside the trivial solutions of discard flat regions or change the subsequence length.



**Fig. 3.** Effect of z-score normalization of relatively flat subsequences on MP values.

Moreover, the energy consumption pattern changes between weekdays and weekends or holiday, so it would be unfair comparison to compare subsequences pertaining to these groups, the same is to compare subsequences of night hours and daily hours. Introducing domain knowledge to find discords only in sone subgroups of the timeseries became important. [27] introduces the concept of annotation vector used to introduce domain knowledge in the process of motif and discord discovery, which allows to find results that follows users defined constraint and produce better results, closer to expectations of the analyst. This method has been proved to be effective to solve different issues: simplicity bias, actionability bias. However, this method is a posteriori method that does not modify the way MP is calculated, *all-pairs-similarity-search* is always performed and then some regions are excluded form motif/discord search. Sometimes it can be useful to exclude some region or to group subsequences into different groups and then perform the similarity search to discover anomalies by comparing only the interesting regions and excluding others. A solution to this problem have been proposed by [25] where Contextual Matrix Profile (CMP) algorithm permits to define ranges along two timeseries and look for the best matching subsequence among these ranges. This permits different a priori grouping of the timeseries observations so that MP calculation can provide novel and more interesting insights.

* 1. Contribution of the paper

The prompt and accurate discovery of anomalies in building electrical load is the key to reduce energy wastes and enhance energy management in buildings. To this aim the objective of this work is the introduction of an unsupervised anomaly detection procedure based on MP algorithm to detect anomalous electrical load at building level in quasi real time. According to the previous literature review and excursus on implication of MP as anomaly detection method, this paper intends to address the following issues by contributing as follows:

* Employment of automatic unsupervised data driven methods to set MP parameters such as subsequence length, context and groups
* Overcame the issues of z-score normalization and its implication in the energy field with the using the of Euclidean distance between not normalized subsequences when calculating the MP
* Apply a contextual anomaly detection by applying an advanced contextual matrix profile method with the definition of context and groups through an unsupervised fashion
* Propose an almost real time anomaly detection method,
* Robust anomaly score definition based on majority voting
* Define a clever anomaly score that talked into account only positive anomalies (higher power consumption)

The rest of the paper is organized as follows: in Section 2 in section data analytics methods are presented, in Section 3 the methodological framework, in Section 4 results and Section 5 discussions.

1. Description of the Data Analysis Methods
   1. Matrix Profile

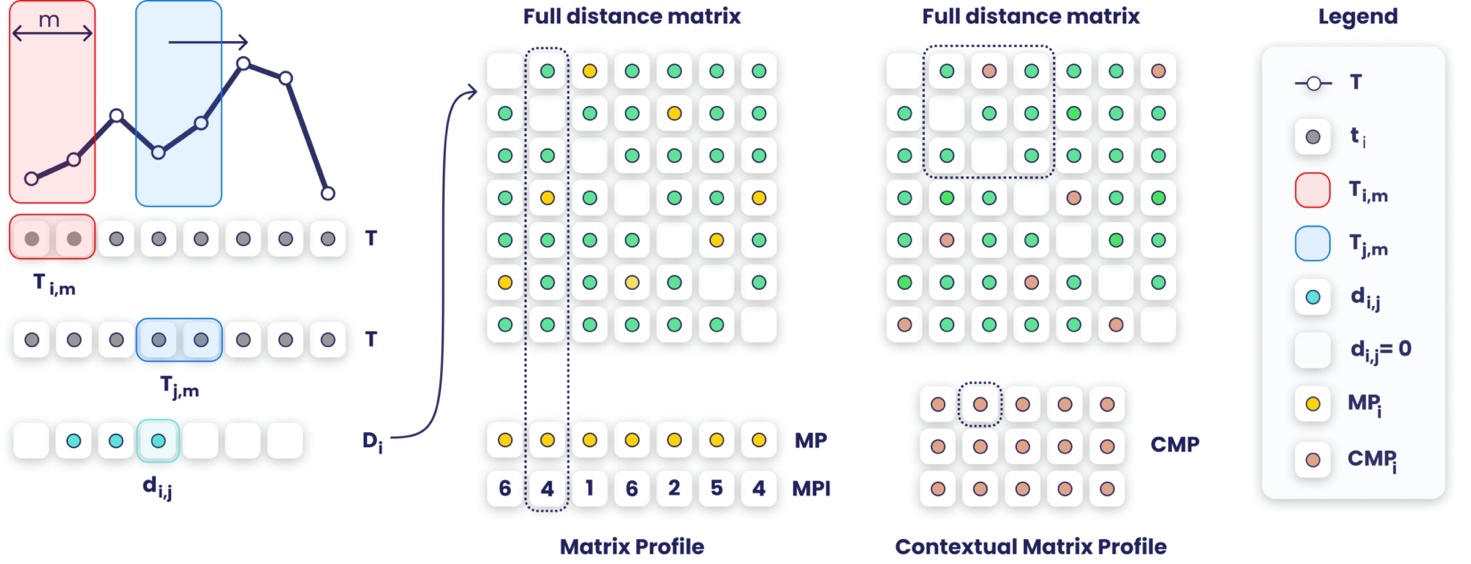
With reference to Figure 4 some fundamental concepts and definitions need to be introduced before explaining the method. First, a *timeseries* is a sequence of real-valued numbers with where is the length of . Since the focus is on local properties of timeseries (i.e., portion of timeseries) a *subsequence* is defined as a continuous subset of values from of length starting from position ; formally defined as with .

An ordered set of all possible subsequences of obtained by sliding a window of length across is called *all-subsequences-set*  of a timeseries and is formally defined as follows: where is a user-defined subsequence length.

By computing the distance between a given query (i.e., subsequence ) and each subsequence in an all-subsequences set it is possible to define is a vector of distances called *distance profile* of a timeseries . Formally, where ) for all where and is the distance metric applied. It is possible to adopt different kind of distances to compute the distance profile [24], [28] but the original method uses the Euclidean distance between the z-normalized subsequences.

If the distance profile is calculated between a query in and the all-subsequences set of (i.e., self-join), by definition the location of the distance profile is zero since the distance is calculated between the query and itself (). Moreover, the distance is close to zero just before and after this location. Those matches are called *trivial matches* and are usually avoided during similarity search by imposing an *exclusion zone* (as function of m, usually set to ) before and after this location.

It is possible to finally define Matrix Profile (MP) as the vector that stores the z-normalized Euclidean distance between each subsequence and its nearest neighbour. Formally, where is the distance profile corresponding to query and timeseries . In other words, it can be generated by extracting the smallest value in each row/column of the full distance matrix. Of course, the construction of the full distance matrix is the most straightforward method but event he less computational efficient, this is the reason why many algorithms has been proposed for the MP calculation, such as mass, such as stamp and stomp that computes the MP in triangular form



**Fig. 4.** step of matrix profile calculation in case of self-join. From left to right is explained the calculation of the element of the distance vector given the query . By calculating the distance vector for the all-subsequences set of , and storing them in a matrix the full distance matrix is composed. MP is the row wise minimum while the comp is the minimum over rectangular regions

With reference to Fig. 4, the MP is the column wise minimum over the entire full distance matrix, meaning that if finds the best matching subsequence (i.e., minimum distance) for any subsequence in . However, as already mentioned in section, comparing regions of timeseries that belongs to different context or operating conditions or different boundary conditions may result into misleading results. Therefore, the Contextual Matrix Profile (CMP) is introduced. CMP is defined as the minimum over rectangular regions of the full distance matrix, allowing to find the best matching subsequence in ranges over and allowing to group data in custom way comparing only portions of with portions of . The CMP calculation is led by the definition of contexts which are a lapse of time in which a subsequence of length may start. For example, given a timeseries of 365 days, with 15-min frequency, by setting a context of from 5:00 to 6:00 and a subsequence length , when computing a row/column of the CMP the distance between the nearest neighbour between five subsequences starting in the given context (i.e., starting at 5:00, 5:15, 5:30, 5:45, 6:00) of a given day with all the subsequences of the context of the other day is calculated. The resulting CMP will have 365 rows/columns﻿ where each point displays the distance between the best matching 2h long subsequence of the two days, lower the distance better the match and vice versa. While context is suitable to create a priori grouping of timeseries, once the MP is calculated it is even possible to further divide the MP into groups that reflect a broader comparison among contexts. In the energy field it may be useful to further group into weekends and weekdays or summer winter to capture weakly or seasonality behaviours otherwise neglected.

* 1. Decision tree

Classification is the task that assigns a class label to unlabelled instances through a classification model which creates a relationship between the predicted or target variable and the predictive variables. Among the numerous classification methods used to describe and explore complex data, the most used are those based on decision trees thanks to their capability to be easily translated in graphical form are commonly used in different fields [29]. Depending on the type of target variable a decision tree can be a classification tree (categorical target variable) or regression tree (numerical target variable) [30].

In this paper is employed the recursive partitioning decision tree called Classification and Regression Tree (CART). Starting from the root (all the available instances) this method proceeds through a binary decision fashion to split the instances in purer subsets (nodes) in a froward stepwise fashion maximizing at each step the purity of each node, yielding local optimum [29] once a stopping condition is satisfied. The purity of each node can be identified through an impurity measure [30]–[32].

* 1. Clustering

Clustering is the process of creating groups (i.e., clusters) based on similarity within some attributes. Clustering algorithms can be categorized into partitional or hierarchical. In the first case, the observations are divided into non-overlapping subsets called clusters. The hierarchical clustering generates non-overlapping clusters, and each cluster can be further divided into subclusters and so on, creating a tree structure.

* 1. Anomaly detection

In this work four statistical model-based outlier detection methods used for outlier identification in univariate timeseries. All those methods accept as input a timeseries and annotates each point of the timeseries with Boolean value: zero if the observation is an outlier, 1 if it is an outlier.

**Inter quartile** defines outliers any of the observations that fall below and above where is the interquartile range () is defined as the difference between the third quartile as the first quartile .

**Z-score standardization** is a model-based outlier detection method which defines an outlier based on the gaussian normal distribution . This method defines outlier any of the observations outside the interval where is a user defined constant in z-score. The normal probability distribution usually defined meaning that the probability to find an observation outlies that range is equal to 2.3%. To apply this method to a not normal distribution z-score standardization is needed.

**Elbow method**: is a graphical method that permits to find the elbow of a curve. By finding the elbow of a univariate vector ordered in descending values it is possible to identify two different regions, the region above the elbow and the one below the elbow, the region above contains the outliers.

**Generalized Extreme Studentized Deviate (GESD):** is an iterative method that progressively evaluates the presence of outliers in a univariate timeseries through a statistical test. The method initialization requires () a presumed number of outliers and confidence interval is set, then for a given the following statistical test is performed:

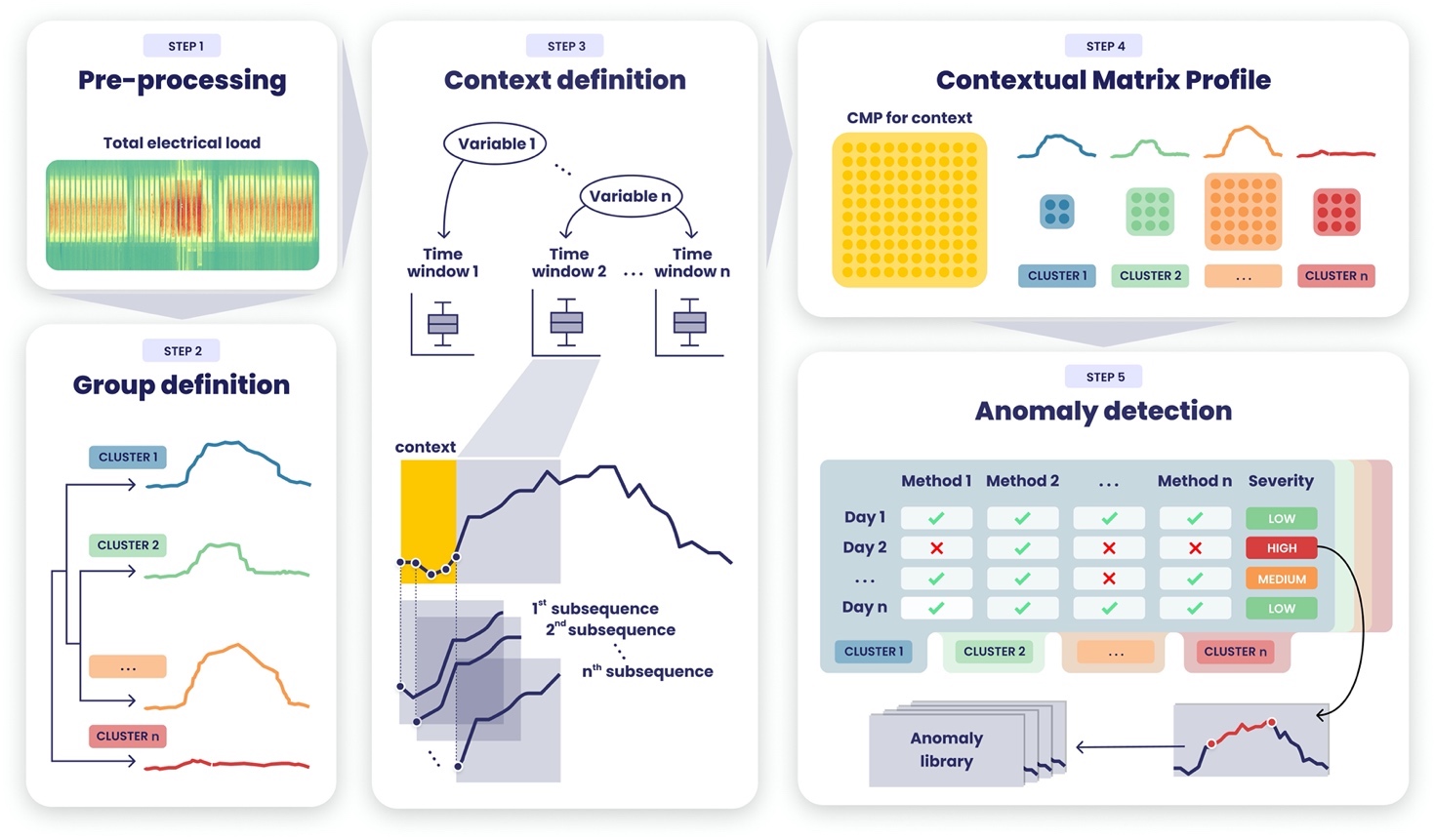
* There are no outliers in the timeseries
* There are up to outliers in the timeseries

The hypotheses test is performed by calculating the statistic and the critical value as follows:

Where and denote sample mean and sample standard deviation of the timeseries, is the timeseries length, is the iteration number,  is the 100p percentage point from the [t distribution](https://www.itl.nist.gov/div898/handbook/eda/section3/eda3664.htm) with ν degrees of freedom and

Methodological Framework

In this section the methodological framework is presented. The method is based on the application of the CMP coupled with unsupervised techniques such as clusters and CART to perform anomaly detection on electrical load timeseries in the most parameter-free and automatic way. The multi-step procedure, reported in Fig. consists in four steps, described in detail in the following paragraphs.



**Fig. 5.** Graphical description of the methodological framework.

**Pre-processing.** The first step consists in data pre-processing is a crucial task for the data analysis workflow. The proposed methodology does not focus on advanced pre-processing techniques since the dataset is assumed to have a good quality with a missing values and inconsistence ratio less than 5% on the overall observations [33]. Thus, the pre-processing is performed through univariate statistical approaches in particular inconsistences removal and missing values imputations through linear interpolation.

**Group definition.** The second step aims to group similar daily load profiles through hierarchical clustering with Euclidean distance. The resulting clusters are considered representative of different operational patterns and are employed in the analysis as a posteriori method to compare results from CMP.

**Context definition.** Within the daily electrical load timeseries different regions are present and different behaviour can be spotted: base load, peak load, ramp up ramp down. The length of the relative time window ﻿ (Mathieu, Price, Kiliccote, & Piette, 2011, [34]) can be defined statistically or inferred from typical schedule and identify sub daily electrical load sequences of particular interest for building energy management. The methodology proposed identifies that sub-daily time windows () through CART. The identification of these region in and unsupervised way has a twofold meaning: (a) automatically identify time windows (b) define the two CMP parameters, subsequence length () and context length () that usually are set a priori based on domain knowledge. The regression tree is constructed using the electrical load as numeric target attribute and the hour of the day as explanatory attribute. This permits to identify, through a cost complexity process, a set of non-overlapping time windows and consequently contexts and subsequence length. Since the interest lies in those subsequences that are mostly contained in the given time window, the subsequence length is always set equal to time window length (). Moreover, the CMP provides the flexibility to investigate similarity of shifted subsequences, thus context is defined as the half of the smallest time window length (). If the smallest time window is two hours long from 6:00 to 8:00 the context is defined as one hour long from 5:00 to 6:00.

**Contextual matrix profile.** CMP is calculated for each context, given that each day have not overlapping contexts the resulting CMP contains one row/column for each day. Given that each row corresponds to a day and each day has been previously classified into clusters, it is possible to further divide the CMP into groups (i.e., cluster).

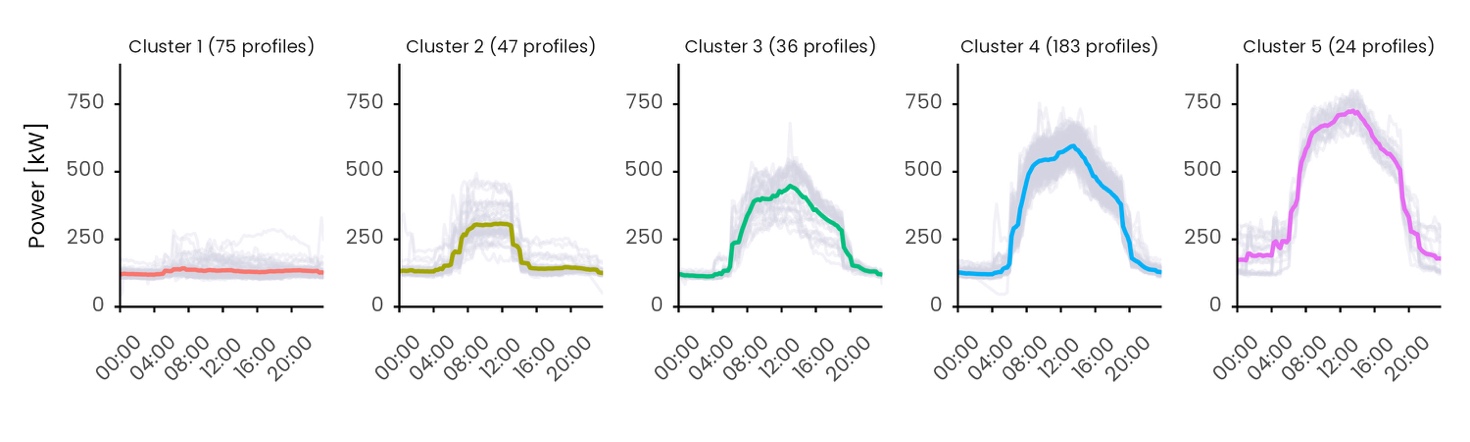
**Anomaly detection.** The anomaly detection is performed for a CMP of a given group of as given context, by applying methods to identify anomalies and the defining the presence and severity of an anomalous through majority voting. Each method is applied for each row/column of the CMP by defining whether a distance is anomalous or not in a Boolean form . Then the severity is then calculated through majority voting, counting by the number of positive detections . Once detected the anomalies are saved into an anomaly library in which context severity and profile are stored.

1. Results

The presented methodology has been tested on the electrical load timeseries of a MV/LV transformer cabin that serves a part of the Italian university campus of Politecnico di Torino (PoliTo). The measurement infrastructure continuously provides the total electrical load with 15 min timestamps. The authors decided to test the presented methodology on a dataset that spans from 1st of January 2019 to 31st of December 2019 even if more recent data are available, mainly because the pandemic COVID completely changed operational patterns and caused a closure of the university from February 2020. The analysis was carried out using the R statistical software [35] for the pre-processing, CART, clustering and visualization and Python [36] for the CMP calculation and anomaly detection methods implementation.

**Pre-processing.** The raw dataset contained 35040 observations with a missing value ratio of less that 0.1%. Inconsistences were removed and missing values imputed through linear interpolation.

**Group definition.** The definition of the groups (i.e., clusters) was performed through a semi-supervised process that combines prior knowledge of the case study with a fully automated, data-driven approach. The results are shown in figure. The a priori knowledge of the typical operational patterns permitted the a priori identification of two clusters: (1) Sundays and public holidays with complete closure of the university (2) saturdays and half-day working weekdays. On the remaining, weekday working days, an unsupervised cluster analysis was performed using the kmeans algorithm. The silhouette index, implemented in the package NbClust [37], was used for the unsupervised search of the optimal number of clusters in a range between two and four clusters. The index identified 3 as the best number of clusters and the daily profiles were arranged in cluster three (3), four (4) and five (5) as shown in figure. .



**Fig. 6.** Daily electrical load profile clusters with the relative centroid.

**Context definition.** The definition of contexts is performed downstream of a sub-daily time window analysis performed with CART, using the R package rpart [38]. To identify meaningful regions of daily load profile with homogeneous electricity consumption values only working days were taken into account (i.e., clusters 4, 5 and 6) excluding days with low standard deviations, i.e., weekends and holidays (clusters 1 and 2). The number of profiles used are 243 for a total number of datapoints of 23328, as shown in figure (b). Total electrical load was used as numerical target variable and time of the day as numerical ordinal predictive variable. The stopping criterion was based on the minimum number of objects in the leaf node such that the minimum length of the time window is 2:30 hours (i.e., minbucket = 60\*2.5/15\*length(unique(df$Date)) ).

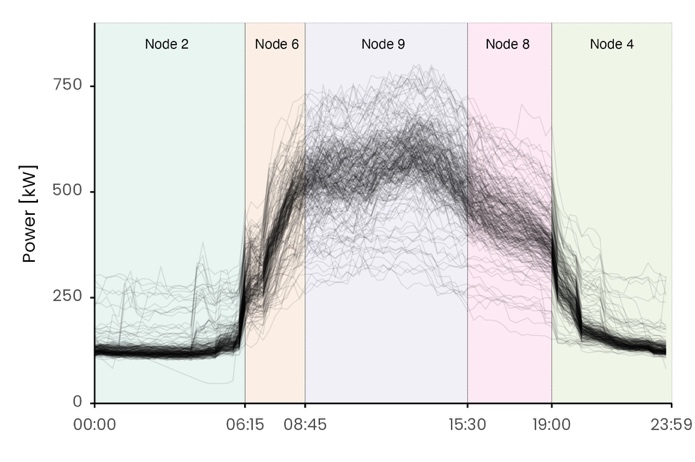
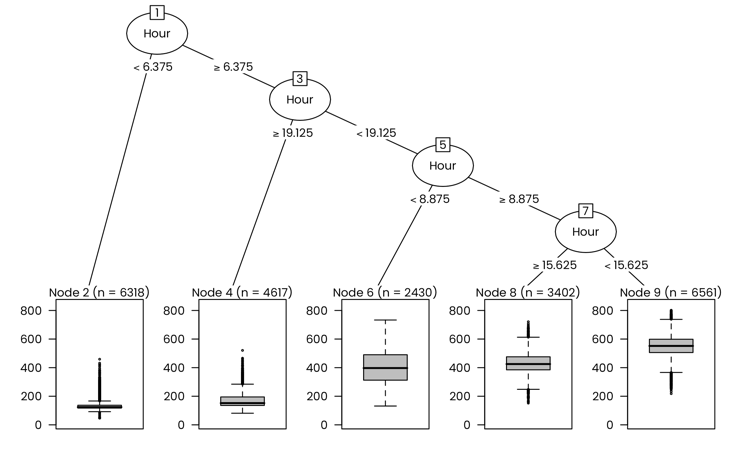


Fig. 1. (a) first picture; (b) second picture.

Figure reports the resulting regression tree, a cost complexity prining process was performed on the regression tree yielding five terminal nodes. Each node refers to a specific time window that can be clearly seen in figure (b) where it can be seen how the time windows effectively isolates sub daily periods of homogeneous load consumption. Time windows with specification on duration number of observations and nomenclature are reorted in table. Time window 1 and 5 refers to …

The resulting time windows are presented in table. To be underlined that the right hand side of the interval is not included in the interval itself. The smallest time window is the second one corresponding to the ramp up of the energy systems with a duration of 2.5 hours. To define a unique context length that can be suitable for all the time windows. One hour

**Table 1.** Summary of resulting time windows and subsequence length.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Time Window | | | | Context | | | |
| ID | Interval | Duration | Observations | ID | Interval | Duration | Observations |
| *tw,1 = m1* | [00:00 - 06:15) | 6 h 15 min | 25 | *mc,1* | [00:00 - 01:00] | 1 h | 4 |
| *tw,2= m2* | [06:15 - 08:45) | 2 h 30 min | 10 | *mc,2* | [05:15 - 06:15] | 1 h | 4 |
| *tw,3= m3* | [08:45 - 15:30) | 6 h 45 min | 27 | *mc,3* | [06:15 - 08:45] | 1 h | 4 |
| *tw,4= m4* | [15:30 – 19:00) | 3 h 30 min | 14 | *mc,4* | [08:45 - 15:30] | 1 h | 4 |
| *tw,5= m5* | [19:00 – 24:00) | 5 h 00 min | 20 | *mc,5* | [15:30 - 19:00] | 1 h | 4 |

**Contextual matrix profile.** The CMP calculation was performed thanks the open source Python library [25]. In figure is presented the CMP calculated for the context. The global CMP can be further divided into groups (right). In the following the authors decided to present the results for context 2.

Immagine che contiene testo, elettronico

Descrizione generata automaticamente

Fig. 1. (a) first picture; (b) second picture.

* 1. Anomaly detection

The anomaly detection module takes as input the contextual matrix profile for a give group and for each row/column (i.e., for each day) computes the median of the Euclidean distances. On this vector are then applied the anomaly detection methods presented in the method sections. The IQR method considers only the positive outliers over 1.5IQR, the z-score only the positive observations over 2 and gest observations considered outliers with a 0.05 tolerance. The elbow method, since it is a pure graphical method is implemented through the python librart knee that

In figure is presented for group 2 (cluster) the results anomaly detection results for the four different methods. In particular 4 anomalies are found in IQR x in zscore etc.

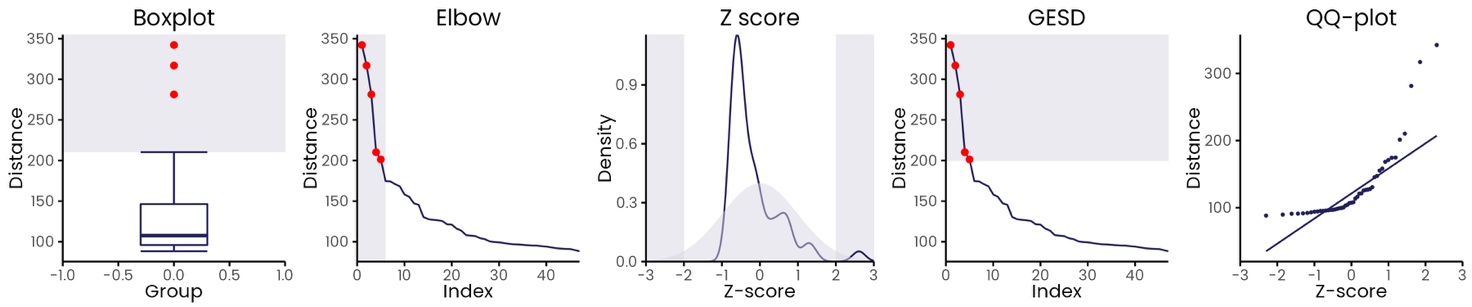


Fig. 1. (a) first picture; (b) second picture.

Each method return a Boolean value 1 (outlier) 0 (not outlier). Then the severity is calculated by summing up the methods. The severity is ranked between 0 and 4. the detected profiles are presented in figure are presented the figure

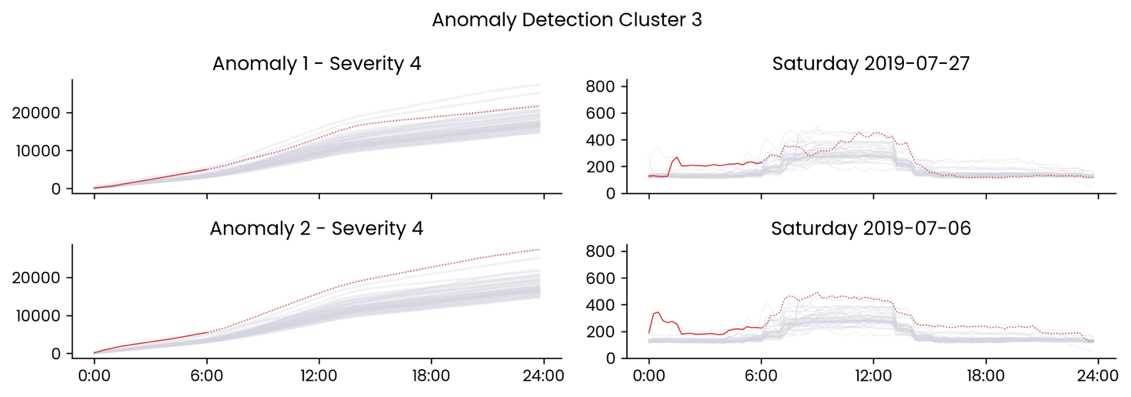


Fig. 1. (a) first picture; (b) second picture.

* 1. Anomaly library

Those anomalies are then

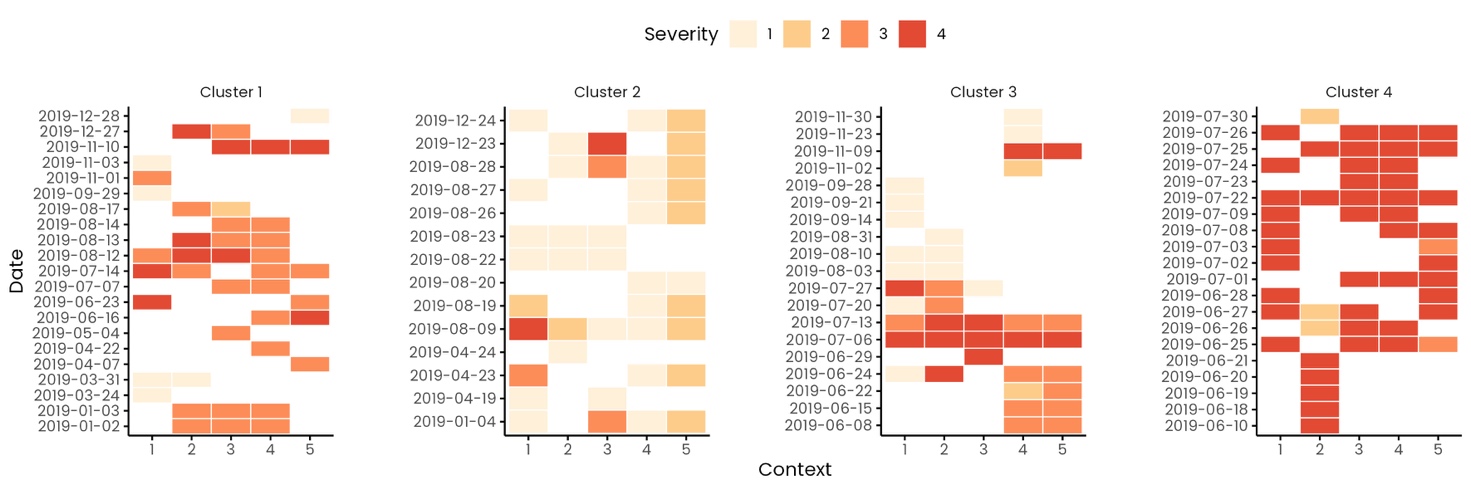


Fig. 1. (a) first picture; (b) second picture.

1. Discussion

Twin freak

﻿For a given subsequence, Matrix Profile computes the Euclidean distance with respect to all other sub- sequences and identifies the minimum distance. Therefore, a repeated anomaly instance would cause false negatives due to the previous anomaly instance being part of all sub- sequence set.

Specifically, frequent/rare subsequences are defined as the ones with the smallest/largest 1-nearest neighbour distance, which are also known as motif/discord. However, discord fails

the ones with the smallest/largest 1-nearest neighbour distance, which are also known as motif/discord. However, discord fails to identify rare subsequences when it occurs more than once in the timeseries, which is widely known as the twin freak problem.

[19] through a semi-supervised model permits to limits the number of subsequences compared, considering for comparison only references with no anomalies.

[39] proposes a method called “Neighbour Profile” based on sampling and density estimation to perform anomaly detection and overcame the issue of twin freak.

1. Conclusion

# Nomenclature

MP Matrix Profile

CMP Contextual Matrix Profile

CART Classification and Regression Tree

References

[1] I. E. A. IEA, “Buildings A source of enormous untapped efficiency potential,” 2020. https://www.iea.org/topics/buildings (accessed Sep. 07, 2021).

[2] Directorate-General for Energy (European Commission), “Clean energy for all Europeans - Publications Office of the EU,” *Publ. Off. EU*, vol. 14, no. 2, p. 3, 2019, doi: 10.2833/9937.

[3] European Parliament, “Directive 2018/2002/EU,” *Off. J. Eur. Union*, vol. 328, no. November, pp. 210–230, 2018, [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018L2002&from=EN.

[4] T. Ramesh, R. Prakash, and K. K. Shukla, “Life cycle energy analysis of buildings: An overview,” *Energy Build.*, vol. 42, no. 10, pp. 1592–1600, 2010, doi: 10.1016/j.enbuild.2010.05.007.

[5] S. Rinaldi *et al.*, “A cognitive-driven building renovation for improving energy effciency: The experience of the elisir project,” *Electron.*, vol. 9, no. 4, 2020, doi: 10.3390/electronics9040666.

[6] L. Erhan *et al.*, “Smart anomaly detection in sensor systems: A multi-perspective review,” *Inf. Fusion*, vol. 67, no. October 2020, pp. 64–79, 2021, doi: 10.1016/j.inffus.2020.10.001.

[7] C. Fan, F. Xiao, and D. Yan, “Advanced data analytics for building energy modeling and management,” *Build. Simul.*, vol. 14, no. 1, pp. 1–2, 2021, doi: 10.1007/s12273-020-0733-z.

[8] 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. pp. 598–609, 2017, doi: 10.1016/j.rser.2016.11.132.

[9] Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira, “Anomaly detection of energy consumption in buildings: A review, current trends and new perspectives,” no. ii, 2020, [Online]. Available: http://arxiv.org/abs/2010.04560.

[10] X. Xu, H. Liu, and M. Yao, “Recent Progress of Anomaly Detection,” *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/2686378.

[11] B. Rossi, S. Chren, B. Buhnova, and T. Pitner, “Anomaly Detection in Smart Grid Data: An Experience Report,” no. 1, pp. 2313–2318, 2016.

[12] C.-C. M. Yeh *et al.*, “Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets,” pp. 1317–1322, 2017, doi: 10.1109/icdm.2016.0179.

[13] C. C. M. Yeh, “Towards a Near Universal Time Series Data Mining Tool: Introducing the Matrix Profile,” University of California, Riverside, 2018.

[14] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, “Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases,” *Knowl. Inf. Syst.*, vol. 3, no. 3, pp. 263–286, 2001, doi: 10.1007/pl00011669.

[15] H. Ren, M. Liu, Z. Li, and W. Pedrycz, “A Piecewise Aggregate pattern representation approach for anomaly detection in time series,” *Knowledge-Based Syst.*, vol. 135, pp. 29–39, 2017, doi: 10.1016/j.knosys.2017.07.021.

[16] M. Alshaer, S. Garcia-Rodriguez, and C. Gouy-Pailler, “Detecting Anomalies from Streaming Time Series using Matrix Profile and Shapelets Learning,” *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 2020-Novem, pp. 376–383, 2020, doi: 10.1109/ICTAI50040.2020.00066.

[17] S. D. D. Anton and H. D. Schotten, “Intrusion Detection in Binary Process Data: Introducing the Hamming-distance to Matrix Profiles,” *Proc. - 21st IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks, WoWMoM 2020*, pp. 347–353, 2020, doi: 10.1109/WoWMoM49955.2020.00065.

[18] 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,” *Proc. - 10th IEEE Int. Conf. Big Knowledge, ICBK 2019*, pp. 175–182, 2019, doi: 10.1109/ICBK.2019.00031.

[19] J. Dinal Herath, C. Bai, G. Yan, P. Yang, and S. Lu, “RAMP: Real-Time Anomaly Detection in Scientific Workflows,” *Proc. - 2019 IEEE Int. Conf. Big Data, Big Data 2019*, pp. 1367–1374, 2019, doi: 10.1109/BigData47090.2019.9005653.

[20] D. De Paepe, D. N. Avendano, and S. Van Hoecke, “Implications of Z-Normalization in the Matrix Profile,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11996 LNCS, no. 768869, pp. 95–118, 2020, doi: 10.1007/978-3-030-40014-9\_5.

[21] D. De Paepe, O. Janssens, and S. Van Hoecke, “Eliminating noise in the matrix profile,” *ICPRAM 2019 - Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, pp. 83–93, 2019, doi: 10.5220/0007314100830093.

[22] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, “Unsupervised real-time anomaly detection for streaming data,” *Neurocomputing*, vol. 262, pp. 134–147, 2017, doi: 10.1016/j.neucom.2017.04.070.

[23] C. Nichiforov, G. Stamatescu, I. Stamatescu, and I. Fagarasan, “Learning Dominant Usage from Anomaly Patterns in Building Energy Traces,” *IEEE Int. Conf. Autom. Sci. Eng.*, vol. 2020-Augus, pp. 548–553, 2020, doi: 10.1109/CASE48305.2020.9216794.

[24] Y. Zhu *et al.*, *The Swiss army knife of time series data mining: ten useful things you can do with the matrix profile and ten lines of code*, vol. 34, no. 4. Springer US, 2020.

[25] D. De Paepe *et al.*, “A generalized matrix profile framework with support for contextual series analysis,” *Eng. Appl. Artif. Intell.*, vol. 90, 2020, doi: 10.1016/j.engappai.2020.103487.

[26] 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 Build.*, vol. 215, p. 109892, 2020, doi: 10.1016/j.enbuild.2020.109892.

[27] H. A. Dau and E. Keogh, “Matrix profile V: A generic technique to incorporate domain knowledge into motif discovery,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. Part F1296, pp. 125–134, 2017, doi: 10.1145/3097983.3097993.

[28] 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*, vol. 34, no. 4. Springer US, 2020.

[29] T. Grubinger, A. Zeileis, and K. P. Pfeiffer, “Evtree: Evolutionary learning of globally optimal classification and regression trees in R,” *J. Stat. Softw.*, vol. 61, no. 1, pp. 1–29, 2014, doi: 10.18637/jss.v061.i01.

[30] P. Tan, *Introduction to data mining*, vol. 12. 2011.

[31] R. Yan, Z. Ma, Y. Zhao, and G. Kokogiannakis, “A decision tree based data-driven diagnostic strategy for air handling units,” *Energy Build.*, vol. 133, pp. 37–45, 2016, doi: 10.1016/j.enbuild.2016.09.039.

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

[33] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,” *Front. Energy Res.*, vol. 9, no. March, pp. 1–17, 2021, doi: 10.3389/fenrg.2021.652801.

[34] J. Zhu, Y. Shen, Z. Song, D. Zhou, Z. Zhang, and A. Kusiak, “Data-driven building load profiling and energy management,” *Sustain. Cities Soc.*, vol. 49, no. March, p. 101587, 2019, doi: 10.1016/j.scs.2019.101587.

[35] R Core Team, “R: A Language and Environment for Statistical Computing.” Vienna, Austria, 2017, [Online]. Available: https://www.r-project.org/.

[36] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[37] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs, “NbClust : An R Package for Determining the,” *J. Stat. Softw.*, vol. 61, no. 6, pp. 1–36, 2014, doi: 10.18637/jss.v061.i06.

[38] T. Therneau, B. Atkinson, and B. Ripley, “Package ‘rpart’: Recursive partitioning for classification, regression and survival trees. R package.” 2019, [Online]. Available: https://cran.r-project.org/package=rpart.

[39] Y. He, X. Chu, and Y. Wang, “Neighbor profile: Bagging nearest neighbors for unsupervised time series mining,” *Proc. - Int. Conf. Data Eng.*, vol. 2020-April, no. April, pp. 373–384, 2020, doi: 10.1109/ICDE48307.2020.00039.

1. \* Corresponding author

   Email address: alfonso.capozzoli@polito.it (Alfonso Capozzoli)

   URL: www.baeda.polito.it (Alfonso Capozzoli) [↑](#footnote-ref-1)