A Generalized Matrix Profile Framework with Support for Contextual Series Analysis

Dieter De Paepe, Sander Vanden Hautte, Bram Steenwinckel, Filip De Turck, Femke Ongenae, Olivier Janssens, Sofie Van Hoecke

IDLab, Ghent University – imec, Ghent, Belgium

Abstract

The Matrix Profile is a state-of-the-art time series analysis technique that can be used for motif discovery, anomaly detection, segmentation and others, in various domains such as healthcare, robotics, and audio. Where recent techniques use the Matrix Profile as a preprocessing or modelling step, we believe there is unexplored potential in generalizing the approach. We derived a framework that focuses on the implicit distance matrix calculation. We present this framework as the Series Distance Matrix (SDM). In this framework, distance measures (SDM-generators) and distance processors (SDM-consumers) can be freely combined, allowing for more flexibility and easier experimentation. In SDM, the Matrix Profile is but one specific configuration. We also introduce the Contextual Matrix Profile (CMP) as a new SDM-consumer capable of discovering repeating patterns. The CMP provides intuitive visualizations for data analysis and can find anomalies that are not discords. We demonstrate this using two real world cases. The CMP is the first of a wide variety of new techniques for series analysis that fits within SDM and can complement the Matrix Profile.

Keywords:
Time Series, Anomaly Detection, Matrix Profile, Distance Matrix, Series Distance Matrix, Contextual Matrix Profile

Email address: `dieter.depaepe@ugent.be` (Dieter De Paepe)

Preprint version, final version available at: https://doi.org/10.1016/j.engappai.2020.103487

1. Introduction

The need for data analysis is increasing as more data is being recorded, stored and made available. One driving factor is the rise of the *Internet of Things* (IoT), where traditional *dumb* devices such as vehicles, household appliances or city infrastructure are enhanced with internet connectivity for monitoring and/or control. In 2018, there were an estimated 7 billion active IoT devices, and this number is expected to double in about 5 years [1]. Many sensors perform periodic monitoring, creating the need for a subdomain of data analysis: series analysis.

Series analysis techniques deal with ordered collections of data points, rather than independent data points. Time series are most common, measuring specific features across time. However, not all series are time series. For example, in [2], skull outlines in images are converted to a series for classification purposes. Unlike non-series, consecutive points in series carry meaning and patterns will often occur throughout the series. Finding and analyzing these patterns can allow better insights in the data.

From a business point of view, series analysis can lead to decreased costs. One such case is maintenance in industry [3]. Today, to prevent the high cost of unexpected machine breakdowns, machine owners perform preventive maintenance periodically. With condition-based maintenance, sensors monitor the health of a machine by recording and analysing time series data to gain insights. This way, machine health is known and owners can better align planned maintenance with the actual need for maintenance, resulting in fewer interventions and decreased maintenance costs and machine downtime. A different business case can be made for trend prediction and anomaly detection [4]. Imagine an online service provider that monitors various metrics related to the usage and load of their services. If the provider is able to gain insight in the usage patterns of the service, he can anticipate certain trends and be made aware of unexpected behavioral patterns of their users. This not only allows the provider to allocate resources more dynamically, but also gives him more time to act on unexpected behavior that might lead to more severe issues.

One state-of-the-art series analysis technique is the Matrix Profile [5], introduced by Yeh et al. in 2016. Given two series $S1$ and $S2$, and a window length m, the Matrix Profile is a new series of length $|S1| - m + 1$ containing the distance between any window of $S1$ and its best matching window in $S2$. By itself, the Matrix Profile can be used to find the *top motifs* (the best matching subsequences in a series) and the *top discords* (the most unique subsequences in a series). Subsequently, it can be used for anomaly detection in contexts where anomalies are defined by unique behavior. Since its inception, many techniques have been published that either extend the Matrix Profile or use it as a building block for new insights [6, 7, 8, 9, 10, 11, 12, 13, 14, 15].

While much progress has been made by going forward with the Matrix Profile, we believe there is also value in taking a step back. One of the implicit steps during the Matrix Profile calculation is the fragmented calculation of the distance matrix of all subsequences of the two input series. In this paper we present the Series Distance Matrix (SDM) framework as the base building block on which specialized techniques can be built, rather than the Matrix Profile itself. To the best of our knowledge, we are the first to present such an overarching framework. Whereas several methods to calculate the distance matrix have been published [5, 6, 16, 13, 14], they have never been suggested as (part of) an overarching framework.

The presented SDM framework separates components that calculate distances between subsequences of input series (*SDM-generators*) and components processing these distances in a meaningful way (*SDM-consumers*). Existing Matrix Profile extensions from literature can be packaged as either SDM-generators or SDM-consumers and plugged into the SDM framework. By separating these components, it becomes easier to combine different techniques freely without additional effort or overhead, resulting in a much broader arsenal of techniques that can be tried on new challenges. Furthermore, distances can be generated once but processed by multiple consumers in combined calculations, resulting in an overall more efficient solution. Lastly, because of this decoupling, components will be smaller, simpler and can be optimized independently from each other.

We also introduce the Contextual Matrix Profile (CMP) and a new SDM-consumer to calculate the CMP. The CMP can be seen as a configurable, 2-dimensional version of the Matrix Profile, that tracks multiple matches across window regions of the series whereas the Matrix Profile tracks one match for each window. Besides data visualization, it can also be used for detecting *anomalies that are not discords.* As a component of SDM, the CMP can be calculated for any distance measure and can be calculated in parallel with other techniques such as the Matrix Profile.

To summarize, our contributions in this paper are as follows: First, we use a new interpretation of the distance matrix to form the generalized SDM framework, which retrofits many published techniques in SDM-generators or SDM-consumers. As second contribution, we introduce the Contextual Matrix Profile as a new SDM-consumer. As final contribution, we created an open source Python implementation of our SDM framework, our CMP-consumer and several Matrix Profile-based consumer and generator implementations based on literature [5, 6, 10, 12, 17, 16, 15]. To the best of our knowledge, this is be the first Python library that provides an implementation combining this many techniques.

The remainder of this paper is structured as follows: Section 2 gives an overview of literature regarding the Matrix Profile. In Section 3, we describe our SDM framework. Section 4 describes our CMP as well as the new SDM-consumer to calculte it. Its value is demonstrated for data visualization and anomaly detection for two real world datasets in Section 5. Finally, we conclude our findings in Section 6.

2. Background and Related Work

In this section, we formalize the definitions used in this paper, summarize the core details of the Matrix Profile and list related literature.

2.1. Definitions

We start by defining the common concepts of *series* and *subsequences*.

Definition 1. A series $\boldsymbol{S} \in \mathbb{R}^n$ is an ordered collection of n real values $(s_0, s_1 \ldots s_{n-1})$.

Definition 2. A subsequence $\boldsymbol{S}_{i,m}$ is the continuous subsequence of $\boldsymbol{S}$ starting at index i of length m: $(s_i, s_{i+1} \ldots s_{i+m-1})$. The subsequence cannot be longer than the original series $(1 \leq m \leq n)$ and has to fall completely within $\boldsymbol{S}$: $(0 \leq i \leq n-m)$.

The distance measure used in the Matrix Profile is the *z-normalised Euclidean distance*. The reason for this is explained in the next subsection.

Definition 3. The z-normalised series $\hat{\boldsymbol{S}}$ is constructed by transforming $\boldsymbol{S}$ so it has a mean $\mu = 0$ and standard deviation $\sigma = 1$: $\hat{\boldsymbol{S}} = \frac{\boldsymbol{S}-\mu_S}{\sigma_S}$.

Definition 4. The z-normalised Euclidean distance $D_{ZE}(\boldsymbol{A}, \boldsymbol{B})$ between 2 series of equal length $A \in \mathbb{R}^m$ and $B \in \mathbb{R}^m$ is defined as the Euclidean distance D_E of the z-normalised series $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$.

$$D_{ZE}(\boldsymbol{A}, \boldsymbol{B}) = D_E(\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}) = \sqrt{(\hat{a}_0 - \hat{b}_0)^2 + \ldots + (\hat{a}_{m-1} - \hat{b}_{m-1})^2}$$

2.2. Matrix Profile

In 2016, Yeh et al. [5] published a novel technique to perform *series subsequence all-pairs-similarity-search* on two series, producing two new series: the Matrix Profile and the Matrix Profile Index. The Matrix Profile is defined as the vector containing the z-normalized Euclidean distances between each subsequence from the first series and its closest matching subsequence from the second time series. The Matrix Profile Index contains the subsequence index in the second series for each match.

Concretely, given two series $\boldsymbol{S1} \in \mathbb{R}^n$ and $\boldsymbol{S2} \in \mathbb{R}^k$ and a subsequence length m, the Matrix Profile $\boldsymbol{M} \in \mathbb{R}^{n-m+1}$ and Matrix Profile Index $\boldsymbol{I} \in \mathbb{R}^{n-m+1}$ are new series such that for each $i \in [0, n-m]$, $\boldsymbol{I}_i$ contains the index of the start of the subsequence of $\boldsymbol{S2}$ of length m that best matches $\boldsymbol{S1}_{i,m}$ and $\boldsymbol{M}_i$ contains the corresponding distance. In the case a *self-join* is performed where $\boldsymbol{S1} = \boldsymbol{S2}$, an additional constraint is added to prevent *trivial matches*, where subsequences match themselves or nearby subsequences.

The default distance measure used is the z-normalized Euclidean distance, which has been shown [18] to provide better results by removing the effect of a changing data offset over time and thus focussing more on shape instead of amplitude. Typical causes of a changing offset are wandering baselines in sensors or natural phenomena (e.g., the gradual change in temperature throughout seasons).

2.3. Related Work

Literature related to the Matrix Profile can be separated into 3 categories: related work focusing on a) the calculation of the Matrix Profile, b) techniques that gain insights from the Matrix Profile or the Matrix Profile Index, and finally, c) ideas from the Matrix Profile for tackling new problems.

a) Calculation of the Matrix Profile

The Matrix Profile was published together with the STAMP algorithm [5], an anytime algorithm to calculate the Matrix Profile (and corresponding Index) of a series of length n in $O(n^2 \log n)$ time. STAMP uses the MASS algorithm [19] to iteratively calculate the distances for each subsequence. Performance was later improved by the STOMP algorithm [6], which uses a dynamic programming technique to reduce the runtime to $O(n^2)$, at the cost of losing the anytime property. Another optimization came with the SCRIMP algorithm [16], which restores the anytime property while retaining the same complexity as STOMP. Finally, ACAMP provides another speed improvement by postponing some operations until the Matrix Profile is completed

[13]. We extended the calculation to reduce the effects of noise when dealing with flat sequences [15, 20], others have made extensions for handling missing data points [21] and support for calculating the multidimensional Matrix Profile [10].

Several recent works have suggested different distance measures to be used in the Matrix Profile. Silva et al. [22] use the Matrix Profile with the (non-normalized) Euclidean distance to perform music recognition and thumbnailing. Akbarinia et al. [13] suggest that using the Euclidean distance, and more general p-norm might be more useful for data analysis in physics, statistics, finances and engineering. Though they present no evaluations, one can expect relevant results for cases where series are not subjected to wandering baselines [18], such as system monitoring. Another distance measure suggested is ψ-DTW [14]. The authors claim that for many application domains, the z-normalized Euclidean distance is too strict while looking for motifs and discords. The ψ-DTW measure performs a non-linear transformation along the (time) axis and can ignore a prefix or suffix of the subsequence being matched. The authors find improved results for domains such as motion tracking (e.g., athlete positioning, motion capture and gesture analysis) and music data mining, though they underline the difficulty of objectively evaluating the relevance of motifs and discords.

b) Gaining insights

Insight in a series can be gained using the Matrix Profile (Index). Motif and discord discovery consist of finding the top matching and worst matching subsequences in a series and can be solved quickly by finding the minima and maxima in the Matrix Profile [5]. Discord discovery can be interpreted as a form of anomaly detection (which has a wide range of applications in machine maintenance, healthcare or system monitoring). In cases where the user knows the type of pattern they are looking for, they can use the Annotation Vector [9] to transform the Matrix Profile before performing motif/discord discovery. Other insights are also possible such as finding gradually changing patterns [11] or finding changes in the underlying behavior being measured [12, 15].

c) Matrix Profile as a building block

The series motifs found by the Matrix Profile have been used for data visualization [7] and classification [8] techniques. Furthermore, a series summarization technique [23] has been published which uses *MPDist*, a distance measure that considers two sequences similar if they share many similar subsequences [24]. The calculation of MPDist involves finding the best match for all subsequences in both series. These could be found by performing a double Matrix Profile calculation, but can also be obtained in a single calculation by processing the subsequence distances in a different way.

As we can see, a wide range of techniques has emerged, most focusing on an aspect closely related to the Matrix Profile.

3. The Series Distance Matrix

Many of the works in Section 2 have started from the idea of the Matrix Profile and created a new algorithm to obtain one specific variation. Looking forward to the future, we can expect the amount of algorithms to rise dramatically as the different distance measures and processing methods are further expanded and combined. Instead, we propose to view these variations as instances of a more generalized framework which we call the *Series Distance Matrix* (SDM).

3.1. SDM: General Concept

We present SDM as a component based framework for deriving insights by processing pairwise distances of the subsequences of pairs of series (this includes self-joins by assuming two equal series). Given pairs of series, *SDM-generators* are responsible for calculating the distances between all pairs of subsequences. Because calculating the full distance matrix is not scalable, we 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. Each consumer is responsible for processing all distance fragments in a way that provides certain insights.

Conceptually, the distance matrix fragments can take any form, however, columns and diagonals have proven to work well for the Matrix Profile. The column based approach is used by the STOMP algorithm [6], it has the advantage of being easier to implement and is more suited for cases where one series is being streamed in an online fashion, since each new data point results in one new column of distance matrix values. The diagonal approach is used by the SCRIMP [16] algorithm. By processing diagonal fragments of the distance matrix, the calculated distances of each fragment are spread over many different pairs of subsequences. This can be utilised 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.

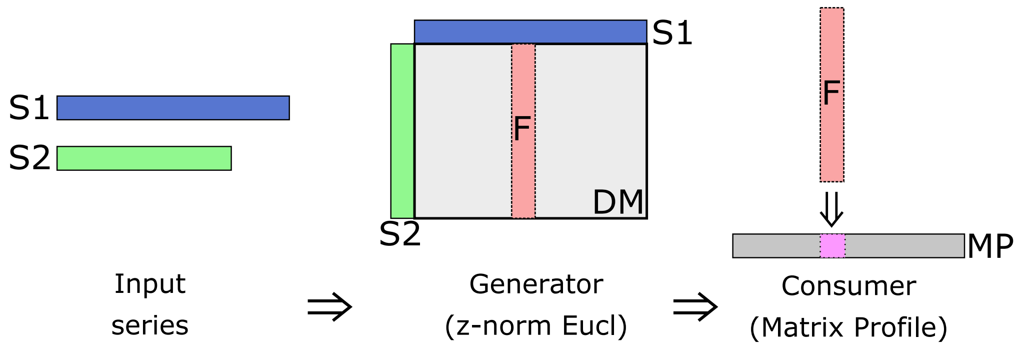

Figure 1: The Matrix Profile calculation fitted into the SDM framework. Starting from two input series (S1, S2), the z-normalized Euclidean distance generator iteratively creates fragments, in this case columns (F), of the distance matrix of all subsequences (DM). Each of these fragments are processed by the Matrix Profile consumer, storing the minimum value for each column in the resulting Matrix Profile (MP).

Figure 1 shows a schematic visualization of the Matrix Profile calculation fitted into the SDM framework.

By separating the distance calculation and processing, we can easily combine generators and consumers to our needs. For example, the techniques described by Akbarinia et al. [13] and Furtado Silva et al. [14] are a combination of the p-norm or ψ-DTW generator with a Matrix Profile consumer. Combinations that have not yet been researched, such as combining a ψ-DTW generator with an MPDist consumer, are - thanks to the SDM framework - just as straightforward. A second benefit is that multiple consumers can be configured for a single generator, instead of having to adjust the algorithms itself, this way reducing calculation overhead. Lastly, by adopting a component based design, each component can be optimized independent of the others. For example, if a faster way is found to calculate the z-normalized Euclidean distance, only one generator has to be updated, instead of every technique using the z-normalized Euclidean distance.

3.2. SDM: Python Implementation

As part of this paper, we released a Python library[1] under the MIT license implementing our SDM framework and CMP consumer. In addition

[1]https://github.com/IDLabResearch/seriesdistancematrix/

to the contributions of this paper, it contains implementations for the noise-corrected z-normalized Euclidean distance ([5, 6, 16, 15]), Euclidean distance, Matrix Profile [5], Multidimensional Matrix Profile [10], Left- and Right-Matrix Profile [11] and VALMOD [17]. It supports batch operations as well as streaming data. At the time of writing, and to the best of our knowledge, this is the first public Python library integrating this many different Matrix Profile related work as consumers and generators in our generic framework.

4. Contextual Matrix Profile

This section covers a new series analysis technique, the CMP, which can easily find repeated patterns in series and shares the benefits of the Matrix Profile: it is deterministic, domain agnostic, exact and is suited for parallelization. The CMP is calculated by the CMP-consumer in the SDM framework. Note that thanks to the SDM framework, we can focus purely on how the calculated distances should be processed, since we can combine the CMP with *any distance measure* that has a corresponding SDM-generator implementation.

As the name implies, the CMP is closely related to the Matrix Profile, and can be best explained in how it differs from it. We make our comparison starting from the distance matrix (the implicit matrix containing the distances of all subsequences from the first input series to all subsequences from the second input series). Where the Matrix Profile is defined as the column-wise minimum over the entire distance matrix, 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. Their configuration is up to the user. A visual comparison of the Matrix Profile and the CMP can be seen in Figure 2. 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 of the Matrix Profile.

Given two input series S_1 and S_2 and subsequence length m, the Matrix Profile looks for the best matching subsequence in S_2 for any subsequence in S_1. The CMP on the other hand looks for the best matching subsequence in ranges over S_1 and S_2. These ranges allow us to group the data in different ways and can reveal new insightful patterns. Specifically, because we aggregate the distances in ranges across both series, the CMP is very good at picking up repeated patterns, even if these patterns are not strictly periodic.

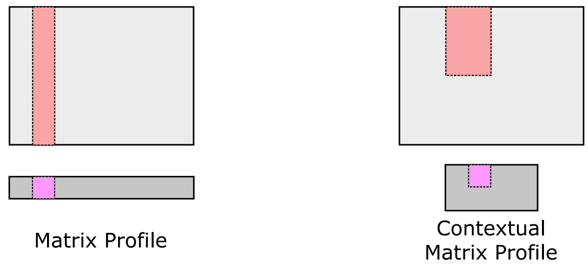

Figure 2: Matrix Profile and CMP differ in how they are created using the distance matrix (light gray). The Matrix Profile (dark gray, left) consists of the column-wise minimum of the values in the distance matrix. The Contextual Matrix Profile (dark gray, right) is created by taking the minimum over rectangular areas. Note that these areas may overlap and may or may not cover the entire distance matrix, depending on the user configuration.

We will show two use cases for the CMP, i.e., data visualization and anomaly detection, but first we discuss more thoroughly how the CMP is calculated.

4.1. Calculating the CMP

Many specialized algorithms could be conceived for specific region configurations. Here, we provide a general purpose algorithm. In this algorithm, the regions of interest are provided by specifying ranges along the dimensions of the distance matrix. This principle is illustrated in Figure 3. One advantage of this approach is that for non-overlapping ranges, the resulting CMP resembles a reduced distance matrix. We will exploit this property in our use cases below.

Our algorithm assumes the distance matrix is provided in a column-wise manner (similar to the STOMP algorithm [6]). A straightforward adaptation for diagonals is also made available in our reference implementation.

The initialization of the CMP-consumer is outlined in Algorithm 1. We take two lists of ranges as input, each defining the contexts for one of the input series. We store the ranges in line 1 and 2. Next, we prepare containers for the CMP and corresponding indices, similar to the Matrix Profile Index. Note that the CMP indices are two-dimensional since we need to track the exact match index for both input series.

The actual calculation of the CMP is listed in Algorithm 2. In line 1, we iterate over all ranges defined over the horizontal dimension of the distance matrix and skip any that do not contain the column being processed in lines 2-4. Next, we iterate over all ranges for the vertical axis. Since all ranges will

Algorithm 1: CMP-consumer Initialization

Input : $R1$, ranges for the vertical axis of the distance matrix. A range is a pair defining a start (inclusive) and end (exclusive) index.
Input : $R2$, ranges for the horizontal axis of the distance matrix.

$v\_ranges \leftarrow R1$;
$h\_ranges \leftarrow R2$;
$cmp \leftarrow |R1| \times |R2|$ matrix, filled with $+\infty$;
$cmp\_index \leftarrow |R1| \times |R2|$ matrix, filled with $(-1, -1)$;

Algorithm 2: CMP-consumer Column Processing

Input : The column index col.
Input : A vector d containing all distances on column col.

```
for j, h_range ← enumerate(h_ranges) do
    if col not in h_range then
        continue
    end
    for i, v_range ← enumerate(v_ranges) do
        dists ← d[v_range];
        min_dist ← min(dists);
        if min_dist < cmp[i, j] then
            cmp[i, j] ← min_dist;
            row ← argmin(dists) + v_range[0];
            cmp_index[i, j] ← (row, col);
        end
    end
end
```

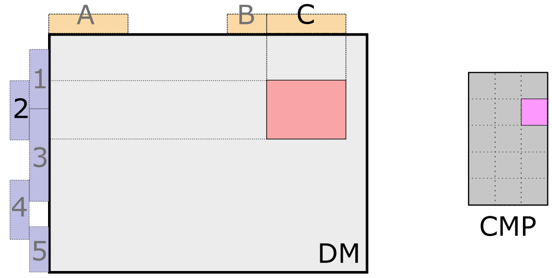

Figure 3: Example of region definitions: a user has specified three horizontal ranges (A, B, C) and five vertical ranges (1...5) on the axes of the distance matrix (DM). Any pair of ranges from both axes corresponds to one region of interest in the distance matrix. The minimum value of the region is calculated and stored in the CMP. Note that the ranges may overlap and may or may not fully cover the distance matrix dimensions.

have some overlap with the distance matrix column, we do not need to filter. In lines 6 and 7, we determine the minimum value of the distance matrix column that is contained in both ranges. We compare this minimum against the best value so far and update the distance and corresponding index if we find a better match (lines 8-12).

Note that when *h_ranges* is very long, a linear scan becomes inefficient. Depending on the intended use, optimizations are obvious: tree maps for general cases, hash based lookup for strictly periodic ranges, or storing the search index for non-overlapping ordered ranges. In this section, we did not attempt to list all possibilities and instead presented the approach best suited for understanding the technique.

Lastly, we briefly discuss the complexity of the CMP. Strictly speaking, the space complexity is constant as it is determined by the configuration of the vertical (V) and horizontal (H) ranges: $O(|H||V|)$. When ranges will be defined in function of the length of the input series (n), $O(n^2)$ is more representative. Note that this last form is overly pessimistic as $|H|$ and $|V|$ will typically be much smaller than n. The time complexity for *processing a single column* is $O(|H| + |V| \times S)$, where S represents the average span of a vertical range. In a typical case where ranges will not overlap, this can be simplified to $O(n)$. As such, a full calculation can be done in $O(n^2)$, the same complexity as the calculation of the Matrix Profile using STOMP.

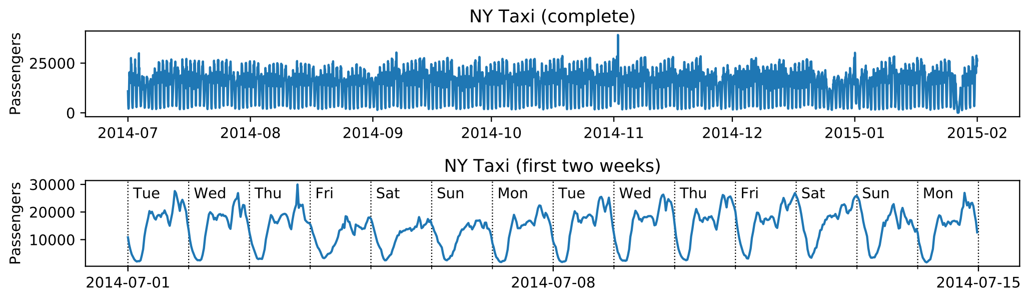

Figure 4: The New York Taxi dataset from the Numenta Anomaly Benchmark. It lists the summed number of taxi passengers in New York at 30 minute intervals. Top: Complete dataset. Bottom: The first two weeks of the dataset, where we see a clear periodic pattern. Note how the pattern for the first Friday, Independence Day, resembles the pattern for a weekend day.

5. CMP for Data Visualization and Anomaly Detection

We will demonstrate the value of the CMP using two different use cases: data visualization and anomaly detection. For both cases, we use the public New York Taxi dataset and a dataset delivered to us by Renson (a ventilation manufacturing company) that we share as part of this publication [25]. Additionally, in our most recent paper [20], we combine the CMP with the noise elimination technique [15] to visualize a UCI activity dataset and show potential for activity segmentation as well. Note that it is not our goal to improve upon the state-of-the-art anomaly detection techniques in this section, but rather to show the potential of the CMP.

All figures in this section were created using Python-based Jupyter notebooks, which we have shared online [25]. Besides providing an easy way to reproduce our results, they offer some additional visualizations we omitted due to size constraints.

5.1. New York Taxi Dataset: Data Visualization

The first dataset is the New York Taxi public dataset from the Numenta Anomaly Benchmark [26]. It lists the total number of taxi passengers in New York city for a period from July 2014 up to February 2015, bucketed per half hour. An overview and excerpt is shown in Figure 4.

We calculated the CMP by self-joining the data using the z-normalized Euclidean distance, using a window length of 44 (22 hours) and a daily context starting at midnight until 02:00 in the morning. Because we are self-joining the data, a constraint prevents any day from matching itself. Simply put, we are asking for the most (shape-wise) similar subsequences between any pair of days, where either subsequence is 22 hours long and can start between midnight and 02:00. These values were based on a quick visual inspection of the data. By choosing a two hour context range and a 22 hour window length, we allow temporal shifts when comparing windows, while always comparing values of the same day. Note that for slightly different values, we obtained similar results. Since the dataset contains 215 days and we define one context per day, the resulting CMP is a 215 by 215 matrix. It is shown in Figure 5. Note that the CMP is symmetrical because of the self-join, higher values in the CMP correspond to more dissimilarity.

When visualized, the CMP can be used to gain insight into the dataset it was built on. For example, the pattern of small squares visible in Figure 5 indicates that there are typically 5 days displaying similar behavior, followed by 2 days of different behavior. These patterns are of course caused by the cycle of weekdays and weekends. Other artefacts standing out are the wide band around New Year, near the end of November (Thanksgiving) and the stripe near the end of January (when a blizzard struck New York), all indicating different behavior in the dataset.

Visualizations like these help data scientists explore new datasets. By inspecting the CMP, they can find patterns and deviations from these patterns that might require further investigation (as we will do in our next use case). Another application is the creation of visual thumbnails for series, helping users to navigate large collections of series. Other thumbnail techniques have been presented using SAX [27] and time series snippets [23] but are unable to provide this degree of insight into the underlying patterns.

Of course, the Matrix Profile can also be visualized to gain insight in a series. We calculated the Matrix Profile using the same parameters as the CMP, it is shown in Figure 6. As mentioned before, the Matrix Profile is a one dimensional vector where high values correspond to more unique subsequences. Looking at the figure, we gain some insights in where the data displays unique behavior, which is further explored in Section 5.2. However, the Matrix Profile is unable to capture the periodic nature of the data since each sequence is compared against all other sequences rather than multiple spans like the CMP does.

As a final demonstration of the possibility to gain insights from visualizing the CMP, we would like to share an unexpected trivia we discovered.

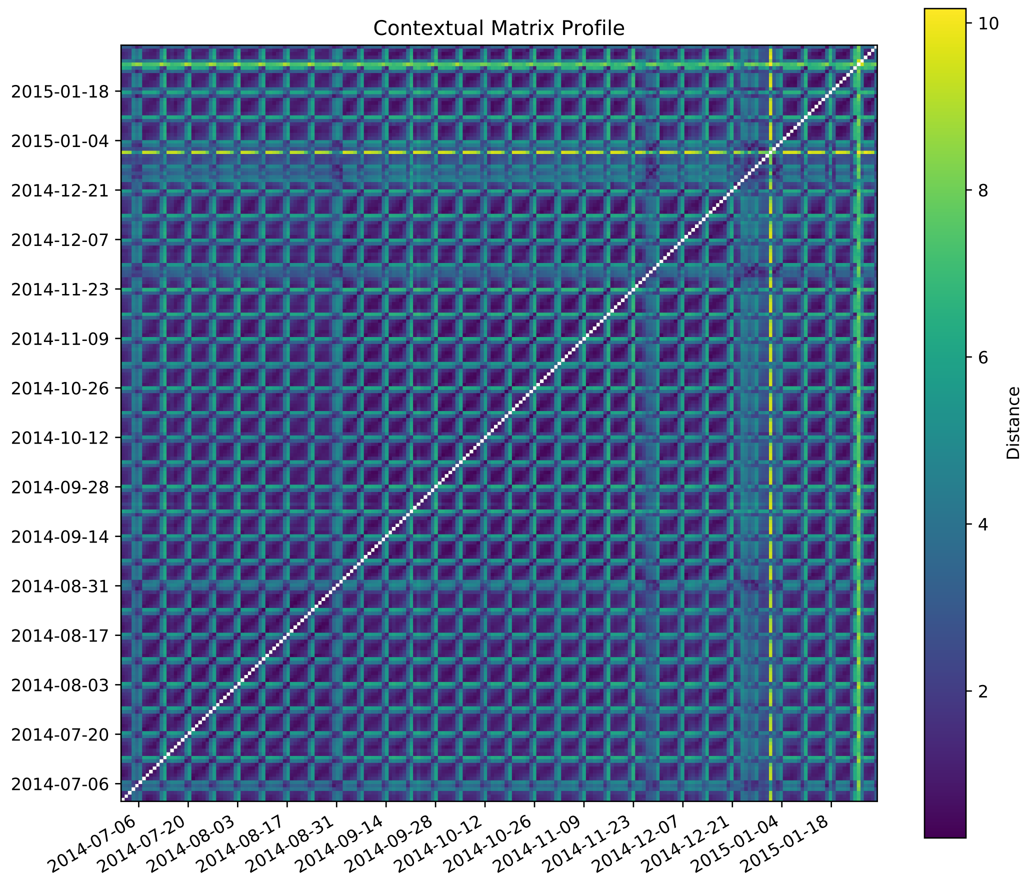

Figure 5: The CMP for the New York Taxi dataset. Each point displays the distance between 2 days, defined as the z-normalized Euclidean distance between the best matching 22 hour long subsequences of both days. Lower distances correspond to a better match. We can clearly see a periodic pattern caused by weekdays versus weekends and the changed behavior around Thanksgiving and between Christmas and New Year. The bright line near the end of January is the effect of a blizzard hitting New York.

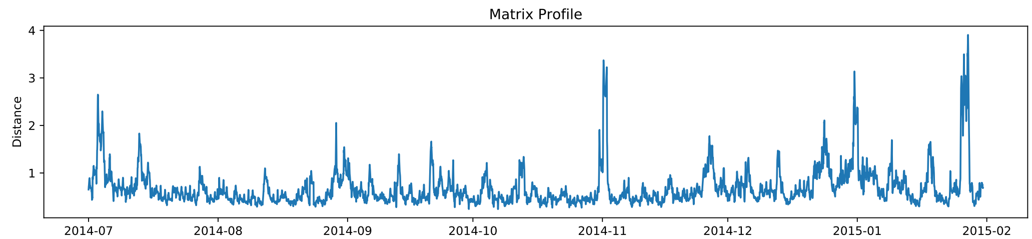

Figure 6: The Matrix Profile for the New York Taxi dataset. Each value represents the distance from the subsequence of the series starting at that index to its nearest match, where higher distances mean more unique subsequences. While we see higher values corresponding to some holidays or other events (discussed in Section 5.2), the periodic nature of the data is not captured in this visualization.

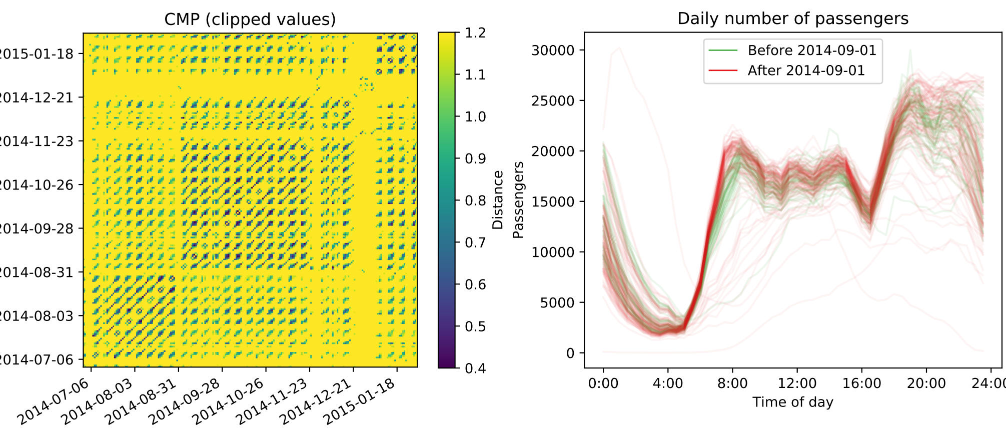

Figure 7: Left: The CMP for the New York Taxi dataset, with values restricted to the range [0.4, 1.2], highlighting the change in distance for days before and after September 1st. Right: The origin of the difference in distances. The number of taxi passengers before and after September 1st differs noticeably around 07:30 in the morning.

Looking carefully, one can see a small difference in the values before and after September 1st (Labor Day). This is more clearly presented in Figure 7 (left). We see the days before Labor Day have a worse match with the days after Labor Day and vice versa, indicating the taxi passenger behavior has changed. Indeed, when looking at the daily graphs (Figure 7 right), we see a noticeable difference in the behavior around 07:30 in the morning: after Labor Day, the number of taxi passengers is higher. The most likely explanation is the start of the school year, which also falls on September 1, enabling parents to leave earlier for work.

5.2. New York Taxi Dataset: Anomaly Detection

As anomalies are defined as *patterns that do not conform to expected behavior* [28], objectively evaluating them is particularly difficult for realistic datasets. What is interpreted as anomalous for one user, might be normal behavior for another [29]. While the New York Taxi dataset contains a ground truth of 5 anomalies (listed in Table 1) that were specified by the dataset provider as "anomalies with known causes"[2], we argue several deviations from expected patterns are present in the data but were not included in the ground truth because of background knowledge not present in the data. As a result, we find the ground truth to be biased towards techniques that find unique behavior, rather than unexpected behavior. Luckily, it is easy to further investigate and validate suspected anomalies, as we will do next.

The visualization of the CMP in Figure 5 already gives a good visual indication about anomalies: on some days the *expected* repetitive pattern is not present. Based on the visual pattern, we divided the contexts into three groups and form smaller CMPs: one containing weekdays and two containing only Saturdays and only Sundays respectively. This is visualised in Figure 8. These reduced CMPs each represent a collection of days that we expect to behave in a similar manner. Since each value in a column (or row) in the CMPs indicates how much a single day (context) deviates from other days (contexts), we can average each column to obtain a single value indicating how much this day deviates from the other days. We define this value as the anomaly score for that day. Note that we average the values in the reduced CMPs, meaning that, e.g. the anomaly score of any Sunday is based on how much it differs from all other Sundays in the dataset, irrespective of the differences with Saturdays or weekdays. After calculating the anomaly score for every day, we ordered all anomaly scores and using the Elbow method, we determined a threshold to obtain 18 anomalous days in total (Figure 8 right). The anomalies are listed in Table 1 and visualized in Figure 9.

We compare the anomalies against those found by the Matrix Profile. The Matrix Profile can be used to find series discords, subsequences that maximally differ from any other subsequence, these discords can be interpreted as anomalies [5]. We calculated the anomalies using the Matrix Profile with a window length of 22 hours (similar as the CMP) and not allowing overlapping anomalies. We obtained 16 anomalies using the Elbow method, which are

[2] https://github.com/numenta/NAB/wiki/FAQ

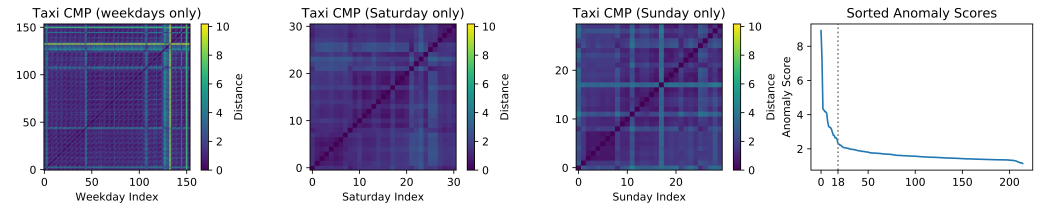

Figure 8: Reduced CMPs from Figure 5, containing only the entries for weekdays (first), Saturdays (second) or Sundays (third) on both axes. Fourth: The anomaly scores (obtained by averaging each column of all reduced CMPs), ordered from high to low. We determined the number of worthy anomalies to be 18.

listed in Table 1 and visualized in Figure 10. Note that the anomalies here have no starting time restriction and can partially cover one or two days.

Of the 25 different anomalies listed in Table 1, only nine are flagged as anomalous by both techniques. For each of these nine, a reasonable explanation could be found, falling into the categories of holiday (Independence Day, Thanksgiving, Martin Luther King Day), holiday predecessor (day before Christmas, New Year's Eve) or large scale event (Climate March, Daylight Savings Time and blizzard). The CMP additionally detected Labor Day, and many weekdays in the Christmas and New Years period, typical days when people take time off from work. Note that since the anomalies by the Matrix Profile can span two days, it would not be fair to consider Christmas and New Year to be found exclusively by the CMP. For one CMP anomaly no clear explanation could be found, though we suspect it is an after effect of the Independence Day celebrations. The Matrix Profile on the other hand exclusively found one weather event, one large scale event (the Millions March against police brutality), Halloween (most likely due to the effect of late-night parties) and four days for which no clear-cut explanation could be found. However, two of the unknown anomalies precede Labor Day, so this could again be an effect caused by people heading out of town for celebrations. Perhaps surprisingly, the Matrix Profile cannot detect Labor Day itself, this is because it closely matches Martin Luther King Day and two weekends in the dataset, meaning it will not be flagged as a series discord.

Rather than looking at individual anomalies, we can also look at the broader picture. By comparing each CMP anomaly against other days of the same type (the second or third column in Figure 9, whichever contains a solid red line), we see that all anomalous days noticeably differ from the majority of the reference days (gray band in the figure). This is less the case for the

Date	Event	Numenta	MP	CMP
Thu 2014-07-03	Evening thunderstorms		5	
Fri 2014-07-04	Independence Day		6	5
Sun 2014-07-06	*Unknown*			15
Sun 2014-07-13	*Unknown*		10	
Fri 2014-08-29	*Unknown*		8	
Sun 2014-08-31	*Unknown*		15	
Mon 2014-09-01	Labor Day			6
Sun 2014-09-21	Climate March		13	17
Fri 2014-10-31	Halloween		9	
Sun 2014-11-02	Daylight Savings Time	x*	3*	9
Thu 2014-11-27	Thanksgiving	x	11*	12
Fri 2014-11-28	Day after Thanksgiving			11
Sat 2014-12-13	Millions March		16	
Wed 2014-12-24	Christmas period		7	3
Thu 2014-12-25	Christmas	x		7
Fri 2014-12-26	Christmas period			10
Mon 2014-12-29	New Year period			14
Tue 2014-12-30	New Year period			18
Wed 2014-12-31	New Year's Eve		4	16
Thu 2015-01-01	New Year	x		1
Fri 2015-01-02	New Year period			13
Fri 2015-01-09	*Unknown*		12	
Mon 2015-01-19	Martin Luther King Day		14*	8
Mon 2015-01-26	Blizzard		2	2
Tue 2015-01-27	Blizzard	x	1	4

Table 1: Anomalies as found by the Matrix Profile (MP) and CMP as well as the ground truth for the dataset (Numenta). The numbers in column CMP and MP correspond to the ordering used in Figure 9 and 10 respectively, where a lower number indicates a higher anomalous behavior.

*: Actually listed on the preceding day, but visual inspection shows the aberrant behavior takes place after midnight.

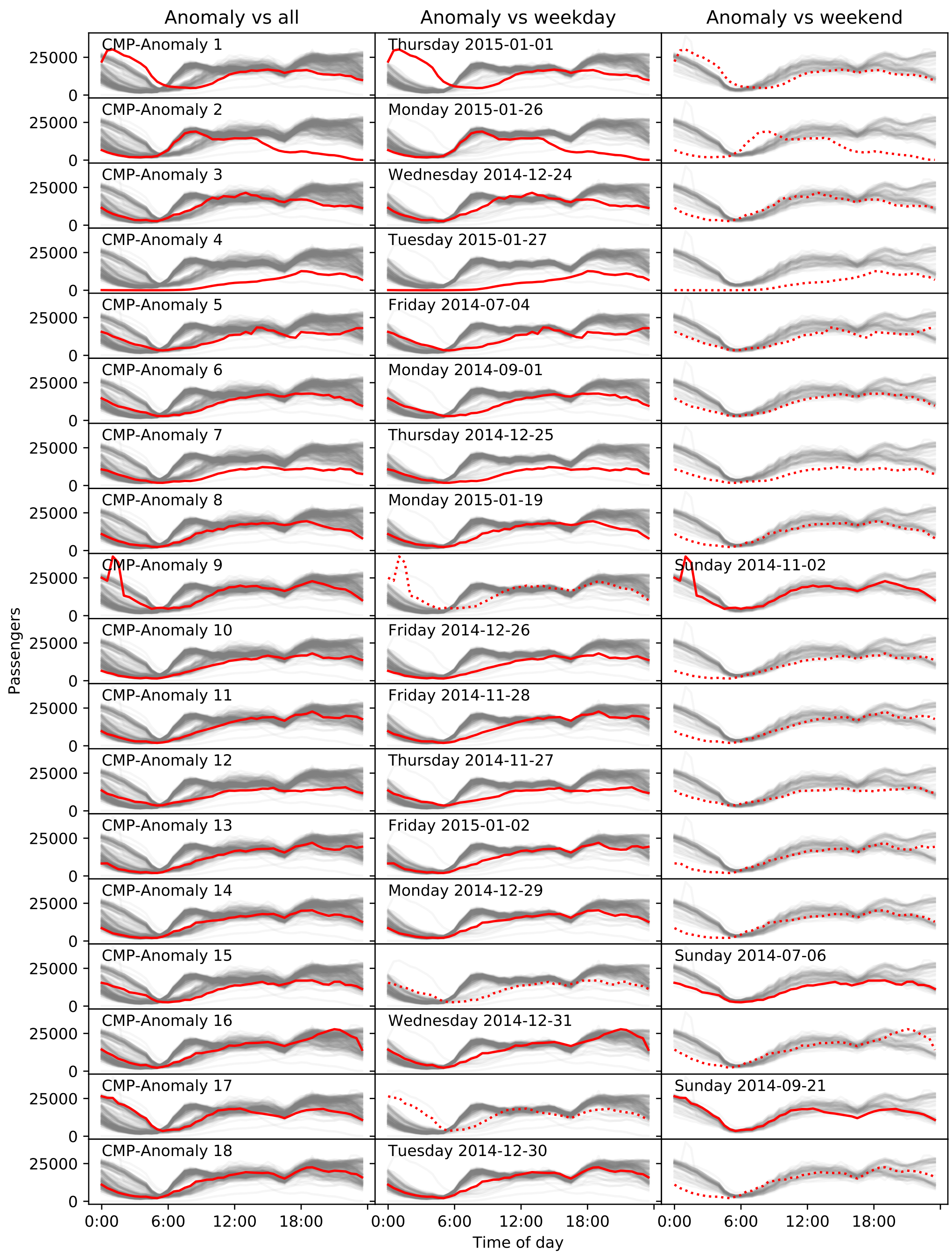

Figure 9: The 18 anomalous days found using the CMP, ordered from most anomalous to least anomalous. Each row shows one anomalous day (red) against all other days in the dataset (gray). A dotted red line is used to visualize the anomaly in the column that does not match its own type (weekday/weekend).

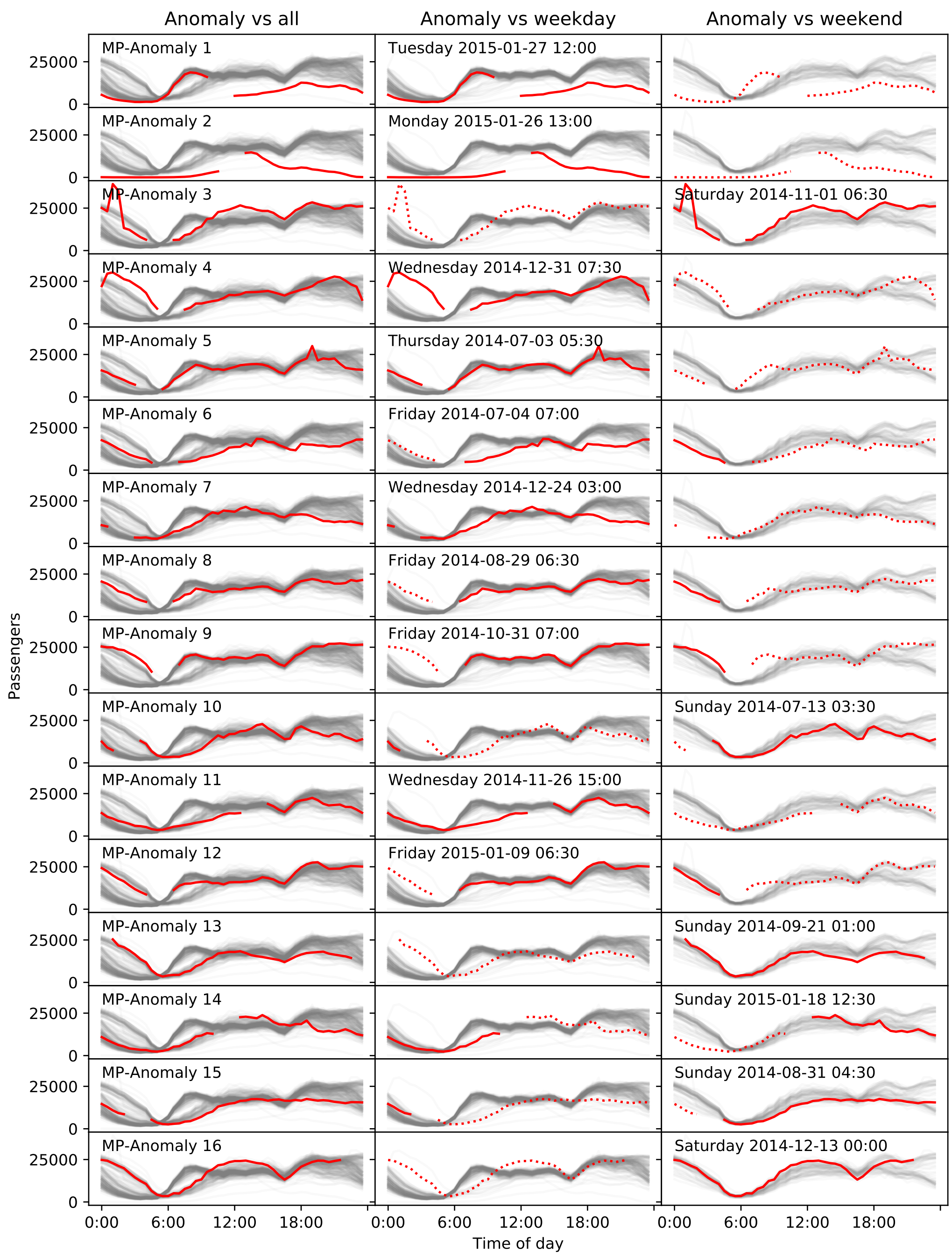

Figure 10: The 16 anomalous sequences found using the Matrix Profile, ordered from most anomalous to least anomalous. Each row shows one anomalous sequence of 22 hours (red) against all other days in the dataset (gray). A dotted red line is used to visualize the anomaly in the column that does not match its own type (weekday/weekend).

anomalies found by the Matrix Profile (Figure 10). Here, about half of the anomalies resemble the reference days, but contain some local variation such as a spike, elongated tail or less pronounced bumps.

The question arises: which of these techniques is best suited for anomaly detection? While we suspect most users will find the results of the CMP to be more insightful for this specific dataset, the general answer remains *"it depends"*. Fundamentally, both techniques are searching for different things. While the Matrix Profile is looking for the most unique patterns (discords) in the series, the CMP based anomaly detection is looking for patterns that differ most from a group of reference contexts. Both approaches will have applications depending on the type of anomalies the user is interested in.

Whereas a simple distance matrix between weekdays and weekends could also have found these anomalies, this assumes knowing the underlying pattern in advance. One benefit of the CMP is that it allows us to discover these patterns in advance when the pattern is *unknown in advance*, which is often the case. 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 (for which the added value is better demonstrated for the next dataset). A similar approach with typical techniques would result in a high complexity, instead we can rely on the computationally efficient implementations of the distance generators of the SDM framework [6, 16].

5.3. Ventilation Dataset: Data Visualization

Our second dataset is a proprietary dataset delivered to us by Renson, a ventilation manufacturing company. It contains measurements of various air quality metrics such as temperature, humidity, carbon dioxide and volative organic compounds, for all rooms within a building that are connected to a ventilation unit, for several anonymized buildings. The users of Renson ventilation products can use this data to observe the functioning of the ventilation system and to estimate the air quality of their home. The metrics are measured at 15 minute intervals and differ per room type. Here, we focus on the CO2 sensor of rooms designated as kitchen. The dataset is shown in Figure 11. Unlike the Taxi dataset, each household has a wide range of distinct daily behaviors and no immediate obvious repeating patterns, it is also not possible to verify any root causes of anomalies. This use case represents

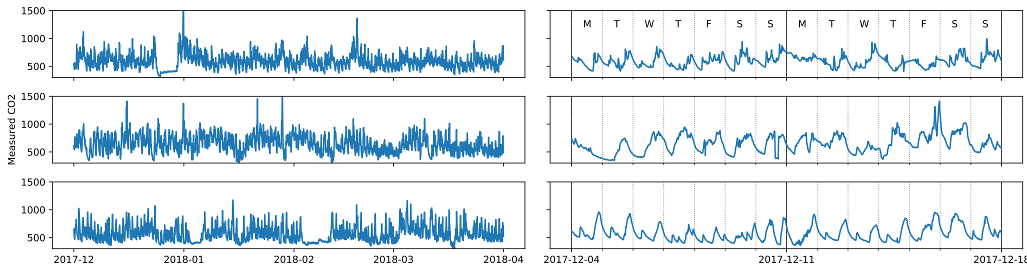

Figure 11: Measured CO2 air content in the kitchen for three ventilation units. Left: The complete datasets. Right: Closeup of two weeks for each corresponding dataset. A day/night pattern is somewhat discernible, but unlike the Taxi dataset, a weekday/weekend pattern is much less obvious.

a typical use case wherein a data scientist has to explore data for which little to nothing is known.

We calculated the CMP using the z-normalized Euclidean distance, using a subsequence length of 3 hours and specifying contexts ranging from 06:00 until (including) 08:00 in the morning. The results are visualized in Figure 12. We see that all three units display very different morning behavior. The first unit displays a pattern that closely resembles the Taxi dataset, with distinct behavior for weekdays, weekends and holidays. It most likely belongs to a family household with regular school and working hours. The second unit shows no clear patterns, though we can see a change near the end of the dataset. The last unit shows a pattern at the start of the dataset, which changes starting January. While we have no explanation for the behavior in these units, the patterns are still interesting to discover and could prove useful for experts. In parallel, we calculated other CMPs for noon and evening, but do not list them in this paper due to size constraints and refer to the accompanying sources for more details [25].

5.4. Ventilation Dataset: Anomaly Detection

After exploring the data, we continue here with the dataset for the first unit. We choose this dataset as it shows most similarity to our expectations of a regular household and should therefore be easier to interpret. Similar to the Taxi dataset, we split the CMP into contexts linked to weekdays and weekends. Since the weekday mornings are very similar, the results are quite similar to those of the Taxi dataset and we refer the reader to the supplementary material for more detailed results. Instead, we will focus on

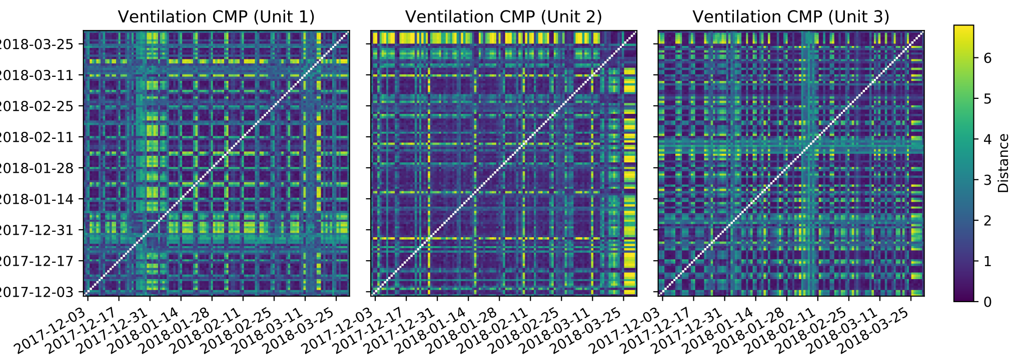

Figure 12: CMP calculated on the morning behavior of three kitchens. The first unit displays a weekday/weekend periodic pattern similar to the Taxi dataset, as well as different behavior around the holiday period. The second unit shows no clear pattern, indicating most mornings have a similar regime. The third unit shows a somewhat periodic pattern that does not match with weekdays/weekends.

the more challenging weekend behavior in this section.

The weekend measurements do not only have a wider range of behavioral patterns, but the start time of these patterns also varies from day to day. Using the CMP calculated on the morning contexts from the previous section, we created a smaller CMP only containing weekend days. Unlike the Taxi dataset, we did not split up Saturdays and Sundays, since there was no distinctive pattern visible for these days in the CMP data visualization. Using the Elbow method, we determined the presence of six anomalies.

Due to the wide variation of the patterns in both values and time, it becomes harder to visualize the anomalies in an intuitive way. One useful approach is a matching table, of which an extract is shown in Figure 13 (the complete figure is available in the source files [25]). Every row of the table corresponds to a single weekend day (one row in the CMP). This day is shown in the first column with the morning context highlighted. The remaining columns show the matches with other weekend days, ordered from best match to worst match. Rather than showing all matches, we simply select the matches on all three quartiles, as well as the best and worst match. Note that each match corresponds to one single value listed in the CMP.

When inspecting the contents of the matching table, we see that the mornings classified as normal have many good matches, only showing minor differences in the third quartile match. The matches for the anomalous mornings already show this level of difference in the first quartile, showing

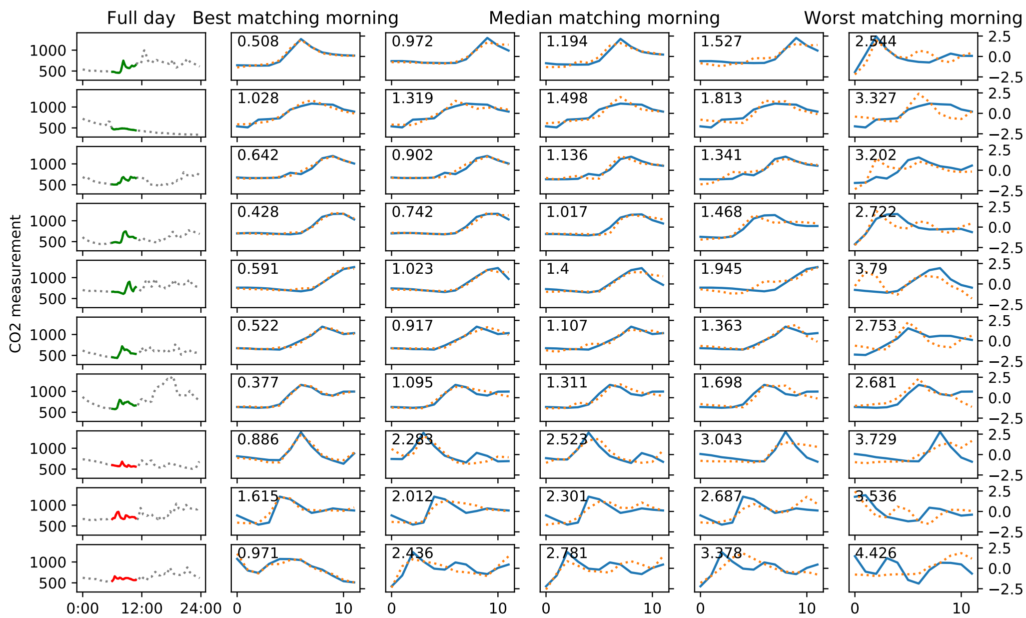

Figure 13: Matching table for subset of weekend days for ventilation unit 1. Each row corresponds to one weekend day, which is displayed in the first column with the morning context (including the window length) highlighted. The first seven rows display days classified as regular (green), the last three show anomalous days (red). The columns show the matching of the morning context (blue) with other morning contexts (dotted orange, one per column). Note that the matching uses subsequences of the context: each blue fragment is a three hour subsequence of the five hour long green/red fragment. For each match, the z-normalized Euclidean distance is displayed in the top left.

that they are in fact uncommon behavior for a weekend morning. This is quantified in the distances listed in Figure 13: the distances of the first quartile match of anomalies are already higher than those of the third quartile of the normal days. Going further into detail, we see that the normal mornings share a common pattern of a plateau followed by a smooth bump and a second, higher plateau. We suspect this pattern is caused by someone waking up, having breakfast in the kitchen and going to an adjacent room. The mornings marked as anomalous show subtly different patterns. The first lacks the second plateau, the second has an earlier start (causing the first plateau to fall outside the context) and also lacks the higher plateau, the third anomaly lacks the distinct high bump at the start. Note that the second normal morning should probably be classified as anomalous. But even though the first spike occurs before the context, the z-normalisation enables a good match between the subtle second bump with the bumps of other days. This again demonstrates the need to finetune the anomaly detection algorithm to the needs of the user.

When looking at the matches in detail, we see how the blue subsequences are not exactly the same for each match. Indeed, the contexts used to produce the CMP allow a time shift: the three hour long subsequence should start between 06:00 and 08:00. As we can see, this flexibility allows us to recognize similar behavioral patterns, despite them not being aligned in time. This flexibility comes at the cost of the user having to define the contexts, often having to rely on expert knowledge of the underlying process. In this case, we relied on our personal experience about kitchen usage patterns to define the contexts.

5.5. Summary

We conclude this section by reiterating our claim that anomaly detection is an inherent subjective topic and difficult to validate. Only when knowing what a user defines as anomalous, can the proper technique be chosen and tried. In this section, we defined normal behavior as behavior that closely matches the majority of the data, and found the CMP to be a suitable technique to detect outliers. We found 18 anomalies for the Taxi dataset, which is more than the five listed as ground truth, and could provide a straightforward explanation for all but one. In the ventilation dataset, we found six anomalies but had no way to validate them independent of the data.

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 *the expectations of the user regarding normal behavior*. 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.

6. Conclusion

In this paper we introduced the Series Distance Matrix framework (SDM), a generalisation of the original approach used to calculate the Matrix Profile. The SDM framework splits the generation and consumption of the all-pair subsequence distances, putting the focus on the distance matrix itself. This allows for easier and more flexible experiments by freely combining components and eliminates the need to re-implement algorithms to combine techniques in an efficient way. The extensions of the Matrix Profile can be fitted in this framework as (part of) a SDM-generator or SDM-consumer. Furthermore, we suspect new techniques will be discovered by further studying the properties of the distance matrix in future work.

We introduced one additional SDM-consumer, namely the Contextual Matrix Profile (CMP). The CMP processes rectangular areas of the distance matrix, compared to the Matrix Profile processing columns. As a result, the CMP is able to compare a range of subsequences against many other ranges, rather than only tracking the best match.

We proved the utility of the CMP for two use cases. When used for data visualization, the CMP was able to reveal repetitive and deviating patterns in the data, making it an ideal first step for data exploration, especially for data containing repetitive patterns. When used for anomaly detection, we defined contexts based on our expectations of the data and were able to find anomalies in the contexts not matching those expectations. Unlike the Matrix Profile, the CMP is able to detect anomalies that are not discords. Both cases were demonstrated on the New York Taxi dataset and a proprietary ventilation metric dataset. In the former, we were able to reasonably explain all patterns and anomalies. In the latter, we showed the visual difference between different ventilation units and relied on the time shift capability of the CMP to discover anomalous mornings.

As part of this publication, we have released a Python implementation of the SDM framework, already comprising implementations for a substantial set of related work. Furthermore, the source code for all use case related processing has been made available online [25].

Acknowledgements

This work was executed within the imec.icon project Dyversify, a research project bringing together academic researchers from imec and industry partners Renson Ventilation NV, Televic Rail & Cumul.io. The Dyversify project was co-financed by imec and received project support from Flanders Innovation & Entrepreneurship (project nr. HBC.2017.0147).

[1] IoT Analytics, State of the iot 2018: Number of iot devices now at 7b market accelerating, 2018. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/.

[2] E. Keogh, L. Wei, X. Xi, S.-h. Lee, M. Vlachos, LB _ Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures, in: Proc. of the 32nd Int. Conf. on Very Large Data Bases, VLDB Endowment, Seoul, Korea, 2006, pp. 882–893.

[3] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, Machinery health prognostics: A systematic review from data acquisition to RUL prediction, Mechanical Systems and Signal Processing 104 (2018) 799–834.

[4] D. Vries, B. van den Akker, E. Vonk, W. de Jong, J. van Summeren, Application of machine learning techniques to predict anomalies in water supply networks, Water Science and Technology: Water Supply 16 (2016) 1528–1535.

[5] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, E. Keogh, Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets, in: 2016 IEEE 16th Int. Conf. on Data Mining (ICDM), IEEE, 2016, pp. 1317–1322.

[6] Y. Zhu, Z. Zimmerman, N. S. Senobari, C.-C. M. Yeh, G. Funning, P. Brisk, E. Keogh, Matrix Profile II : Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins, 2016 IEEE 16th Int. Conf. on Data Mining (ICDM) (2016) 739–748.

[7] C.-C. M. Yeh, H. Van Herle, E. Keogh, Matrix profile III: The matrix profile allows visualization of salient subsequences in massive time series, Proceedings - IEEE Int. Conf. on Data Mining, ICDM (2017) 579–588.

[8] C.-C. M. Yeh, N. Kavantzas, E. Keogh, Matrix profile IV: Using Weakly Labeled Time Series to Predict Outcomes, Proc. of the VLDB Endowment 10 (2017) 1802–1812.

[9] H. A. Dau, E. Keogh, Matrix Profile V: A Generic Technique to Incorporate Domain Knowledge into Motif Discovery, in: Proc. of the 23rd

ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining - KDD '17, ACM Press, New York, New York, USA, 2017, pp. 125–134.

[10] C.-C. M. Yeh, N. Kavantzas, E. Keogh, Matrix Profile VI: Meaningful Multidimensional Motif Discovery, in: 2017 IEEE Int. Conf. on Data Mining (ICDM), IEEE, 2017, pp. 565–574.

[11] Y. Zhu, M. Imamura, D. Nikovski, E. Keogh, Matrix Profile VII: Time Series Chains: A New Primitive for Time Series Data Mining, in: 2017 IEEE Int. Conf. on Data Mining (ICDM), IEEE, 2017, pp. 695–704.

[12] S. Gharghabi, Y. Ding, C.-C. M. Yeh, K. Kamgar, L. Ulanova, E. Keogh, Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels, in: 2017 IEEE Int. Conf. on Data Mining (ICDM), IEEE, 2017, pp. 117–126.

[13] R. Akbarinia, B. Cloez, Efficient Matrix Profile Computation Using Different Distance Functions, 2019.

[14] D. Furtado Silva, G. E. A. P. A. Batista, Elastic Time Series Motifs and Discords, in: 2018 17th IEEE Int. Conf. on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 237–242.

[15] D. De Paepe, O. Janssens, S. Van Hoecke, Eliminating Noise in the Matrix Profile, in: Proceedings of the 8th Int. Conf. on Pattern Recognition Applications and Methods, SCITEPRESS - Science and Technology Publications, 2019, pp. 83–93.

[16] Y. Zhu, C.-C. M. Yeh, Z. Zimmerman, K. Kamgar, E. Keogh, Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds, in: 2018 IEEE Int. Conf. on Data Mining (ICDM), IEEE, 2018, pp. 837–846.

[17] M. Linardi, Y. Zhu, T. Palpanas, E. Keogh, Matrix Profile X, in: Proc. of the 2018 Int. Conf. on Management of Data - SIGMOD '18, ACM Press, 2018, pp. 1053–1066.

[18] E. Keogh, S. Kasetty, On the need for time series data mining benchmarks, Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining - KDD '02 (2002) 102.

[19] A. Mueen, Y. Zhu, M. Yeh, K. Kamgar, K. Viswanathan, C. Gupta, E. Keogh, The fastest similarity search algorithm for time series subsequences under euclidean distance, 2017. http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html.

[20] D. De Paepe, D. Nieves Avendano, S. Van Hoecke, Implications of Z-Normalization in the Matrix Profile, 2019.

[21] Y. Zhu, A. Mueen, E. Keogh, Admissible Time Series Motif Discovery with Missing Data (2018).

[22] D. F. Silva, C.-c. M. Yeh, Y. Zhu, G. E. A. P. A. Batista, E. Keogh, Fast Similarity Matrix Profile for Music Analysis and Exploration, IEEE Transactions on Multimedia 21 (2019) 29–38.

[23] S. Imani, F. Madrid, W. Ding, S. Crouter, E. Keogh, Matrix Profile XIII: Time Series Snippets: A New Primitive for Time Series Data Mining, in: 2018 IEEE Int. Conf. on Big Knowledge (ICBK), IEEE, 2018, pp. 382–389.

[24] S. Gharghabi, S. Imani, A. Bagnall, A. Darvishzadeh, E. Keogh, Matrix Profile XII: MPdist: A Novel Time Series Distance Measure to Allow Data Mining in More Challenging Scenarios, in: 2018 IEEE Int. Conf. on Data Mining (ICDM), IEEE, 2018, pp. 965–970.

[25] Source code for our experiments, 2019. https://sites.google.com/view/generalizing-matrix-profile.

[26] A. Lavin, S. Ahmad, Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark, in: 2015 IEEE 14th Int. Conf. on Machine Learning and Applications, IEEE, 2015, pp. 38–44.

[27] N. Kumar, V. N. Lolla, E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, Time-series Bitmaps: a Practical Visualization Tool for Working with Large Time Series Databases, in: Proc. of the 2005 SIAM Int. Conf. on Data Mining, Society for Industrial and Applied Mathematics, 2005, pp. 531–535.

[28] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A Survey, ACM Computing Surveys 41 (2009) 1–58.

[29] H. Sivaraks, C. A. Ratanamahatana, Robust and accurate anomaly detection in ECG artifacts using time series motif discovery, Computational and Mathematical Methods in Medicine 2015 (2015).