Matrix Profile XXIII: Contrast Profile: A Novel Time Series Primitive that Allows Real World Classification

Ryan Mercer   Sara Alaee   Alireza Abdoli   Shailendra Singh   Amy Murillo   Eamonn Keogh
University of California, Riverside
{rmerc002, salae001, aabdo002, shailendra.singh, amy.murillo} @ucr.edu, eamonn@cs.ucr.edu

Dear Reader, this is an expanded 10-page version of a ICDM 2021 short (6-page) paper

Abstract—**Time series data remains a perennially important datatype considered in data mining. In the last decade there has been an increasing realization that time series data can best understood by reasoning about time series subsequences on the basis of their similarity to other subsequences: the two most familiar such time series concepts being *motifs* and *discords*. Time series motifs refer to two particularly close subsequences, whereas time series discords indicate subsequences that are far from their nearest neighbors. However, we argue that it can sometimes be useful to simultaneously reason about a subsequence's closeness to certain data and its distance to other data. In this work we introduce a novel primitive called the *Contrast Profile* that allows us to efficiently compute such a definition in a principled way. As we will show, the Contrast Profile has many downstream uses, including anomaly detection, data exploration, and preprocessing unstructured data for classification. We demonstrate the utility of the Contrast Profile by showing how it allows end-to-end classification in datasets with tens of billions of datapoints.**

Keywords—Motifs, Multiple Instance, Classification

I. INTRODUCTION

In order to perform various data mining tasks on time series, it can be fruitful to annotate each subsequence with metadata indicating various properties. One such feature is a subsequence's distance to its nearest neighbor within the same dataset. That information can be represented by the Matrix Profile [1]. Small values in the Matrix Profile are called *motifs*, and large values are called *discords*. Both motifs and discords have each been used in hundreds of research efforts . However, we argue that it may be useful to score subsequences with a new piece of meta-data that reflects the property that a subsequence is simultaneously close to its nearest neighbor in certain data but far from its nearest neighbor in other "black-listed" data. We call this property *Contrast*, and the vector that represents it the *Contrast Profile*. While the proposed representation has many uses, for clarity, we will introduce it in the context of subsequence extraction to allow *classification*.

While the time series classification community is very active, most research efforts confine their work to datasets from the UCR archive or similar benchmark datasets [2]. However, for most of these archive datasets, the work of extracting the exemplars from a longer time series has already been done. Here, we argue that extracting the exemplars is actually the most difficult and critical task. In a handful of cases, it may be obvious where the beginning and the end of an exemplar is within a longer time series. But, in many cases, these demarcations may not be clear. Consider Fig. 1.*bottom*, which shows a time series known to have several examples of chicken dustbathing behavior [3]. Even to experts in avian biomechanics, it is not obvious where the dustbathing behavior is. Moreover, even if we knew the answer to that question from an ethological perspective, there is no guarantee that those subsequences would optimize classification accuracy.

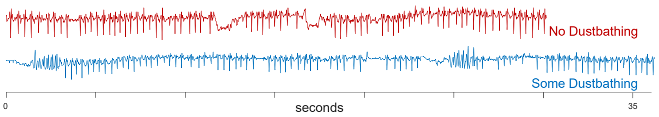


Fig. 1. Two short snippets of behavior from a chicken wearing a backpack accelerometer. The bottom time series is known to contain at least two examples of dustbathing behavior, whereas the top time series is known to be free of this behavior.

This suggests that a technique is needed to annotate each subsequence of the time series with a value that simultaneously represents how close that subsequence is to its nearest neighbor *within* the same time series and how far it is from its nearest neighbor in the time series known to be free of the target behavior. This score would reveal the location of the uniquely conserved behavior, in this case, *dustbathing*.

In Fig. 2, we give a visual intuition of the property of interest: abstracting time series subsequences to points in a high dimensional space. We explicitly consider three data points.

- Point **A** is far from its nearest neighbor in the non-target class, but it is also far from its nearest neighbor within its own target class. It is an anomaly that would score highly on the definition of time series *discord* [4].
- Point **B** in contrast is very close to its nearest neighbor in the target class, but it is also close to its nearest neighbors in non-target class. This point would score highly on the definition of time series *motif* [1].
- Point **C** *is* both very far from its nearest neighbor in the non-target class *and* very close to its nearest neighbor in the target class. This is exactly the property we desire.

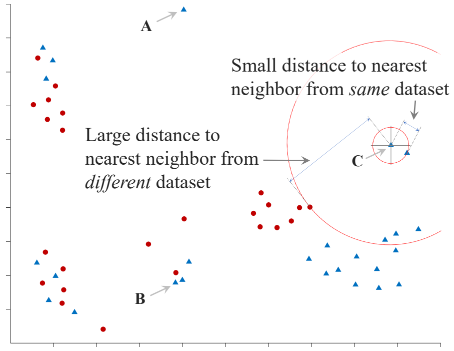

A
Small distance to nearest neighbor from *same* dataset
Large distance to nearest neighbor from *different* dataset
C
B

Fig. 2. A visual intuition of the "contrast" property. Of the three annotated points from the target class, only C is close to a member of its own class, while also being far from its nearest neighbor in the non-target class.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

The rest of this paper is organized as follows. In Section II, we present the necessary definitions and notations. Section III sees a discussion of related work. In Section IV, we present several examples of data mining tasks that can exploit the Contrast Profile before experimentally demonstrating them in Section V. Section VI offers conclusions.

II. DEFINITIONS AND NOTATION

Our data type of interest is *time series*.

Definition 1: A *time series* $\mathbf{T} = t_1, t_2, \ldots, t_n$ is a sequence of real-valued numbers.

Typically, we are not interested in global properties of a time series but rather shapes of small regions called *subsequences*.

Definition 2: A *subsequence* $\mathbf{T}_{i,m}$ is a contiguous subset of values from $\mathbf{T}$ starting at index i with length m.

We can measure the distance between any two time series of equal length using a distance measure. In this work, we use the ubiquitous z-normalized Euclidean distance [1]. One minor modification to the Euclidean distance is that we clip it at $\sqrt{(2*m)}$ because values above this are anti-correlated in the Pearson Correlation space. This is done in order to make the greatest use of the normalized range when working with the Contrast Profile. If we need to measure the distance between a short time series and every subsequence from a long time series, we can produce a *distance profile*.

Definition 3: A *distance profile* $\mathbf{DP}_{i,m}^{(\mathrm{AB})}$ is the vector of distances between each subsequence in reference time series $\mathbf{T}^{(\mathrm{A})}$ and a query subsequence $\mathbf{T}_{j,m}^{(\mathrm{B})}$.

The distance can be computed very efficiently using the MASS algorithm [5]. Fig. 3 illustrates these definitions on a running example of a noisy electrocardiogram (ECG).

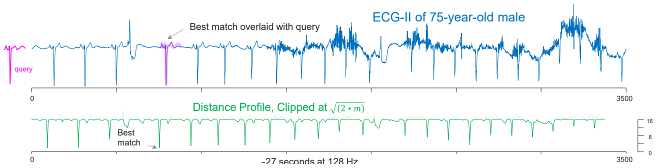


Fig. 3. *top*) A 27-second snippet of an ECG time series. *bottom*) A single heartbeat from earlier in the same dataset was used as a query to produce a distance profile, which has low values when the "sliding" query is similar to a subsequence and is minimized at the best match about five seconds in.

Our proposed ideas leverage the *self-join Matrix Profile* [1].

Definition 4: A *self-join Matrix Profile* $\mathbf{MP}_m^{(\mathrm{AA})}$ of a time series $\mathbf{T}^{(\mathrm{A})}$ is a vector of Euclidean distances between every subsequence $\mathbf{T}_{i,m}^{(\mathrm{A})}$ and its nearest neighbor $\mathbf{T}_{j,m}^{(\mathrm{A})}$. Formally, $\mathbf{MP}_m^{(\mathrm{AA})} = [\boldsymbol{min}(\mathbf{DP}_{1,m}^{(\mathrm{AA})}), \boldsymbol{min}(\mathbf{DP}_{2,m}^{(\mathrm{AA})}), \ldots, \boldsymbol{min}(\mathbf{DP}_{n-m+1,m}^{(\mathrm{AA})})]$

Fig. 4 shows $\mathbf{MP}_{128}^{(\mathrm{AA})}$ for our running example. We can see that the top motifs are a pair of normal heartbeats. Using some out-of-band data (including advice of cardiologist Dr. Greg Mason), we annotated the location of two premature ventricular contractions. While these two beats are similar, they are not as well conserved as normal beats.

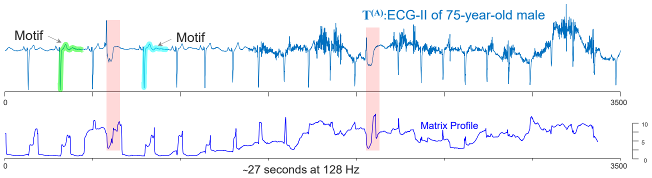


Fig. 4. *top*) The ECG shown in Fig. 3 with its $\mathbf{MP_{128}}$ (*bottom*). The lowest values of $\mathbf{MP_{128}}$ are the Top-1 motif pair, here two normal beats. Also, two PVCs shown highlighted with red bars for future reference.

In addition to subsequence comparisons *within* a time series, it can also be fruitful to make comparisons *between* two time series using the *AB-join Matrix Profile.*

Definition 5: An *AB-join Matrix Profile* $\mathbf{MP}_m^{(\mathrm{AB})}$ between reference time series $\mathbf{T}^{(\mathrm{A})}$ and a query time series $\mathbf{T}^{(\mathrm{B})}$ is a vector of Euclidean distances between each subsequence $\mathbf{T}_{i,m}^{(\mathrm{A})}$ and its nearest neighbor $\mathbf{T}_{j,m}^{(\mathrm{B})}$. Formally,

$$\mathbf{MP}_{128}^{(\mathrm{AB})} = [\boldsymbol{min}(\mathbf{DP}_{1,m}^{(\mathrm{AB})}), \boldsymbol{min}(\mathbf{DP}_{2,m}^{(\mathrm{AB})}), \ldots, \boldsymbol{min}(\mathbf{DP}_{n-m+1,m}^{(\mathrm{AB})})]$$

Note that in general, $\mathbf{MP}_m^{(\mathrm{AB})} \neq \mathbf{MP}_m^{(\mathrm{BA})}$: even with equal lengths, they correspond to different reference time series.

Fig. 5 shows $\mathbf{MP}_{128}^{(\mathrm{AB})}$ for our running example with a region of normal ECG from the same patient.

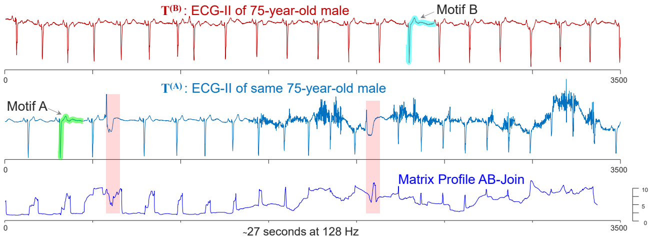


Fig. 5. *top*) Time series $\mathbf{T}^{(\mathrm{B})}$ is a normal ECG time series from the same patient. *center*) Time series $\mathbf{T}^{(\mathrm{A})}$,which contains the behavior of interest, is the original ECG introduced in Fig. 3. *bottom*) The top motif pair, where motif$^{(\mathrm{A})}$ is the unrequited nearest neighbor of motif$^{(\mathrm{B})}$. The red bars foreshadow discovery of two PVCs.

We now exploit an important observation. Note that $\mathbf{MP}_{128}^{(\mathrm{AA})}$ and $\mathbf{MP}_{128}^{(\mathrm{AB})}$ from the last two figures are very similar in most regions. This makes sense. A noisy $\mathbf{T}_{i,m}^{(\mathrm{A})}$ will tend to be just as far from any other $\mathbf{T}_{j,m}^{(\mathrm{A})}$ as it is from any $\mathbf{T}_{k,m}^{(\mathrm{B})}$ (An implication of theorem 1 of [6]). Moreover, a normal heartbeat in $\mathbf{T}^{(\mathrm{A})}$ will tend to have approximately the same low distance to another normal heartbeat, whether that beat happens to come from $\mathbf{T}^{(\mathrm{A})}$ or $\mathbf{T}^{(\mathrm{B})}$. The *only* places showing a significant difference are the locations corresponding to behaviors that are *unique* to $\mathbf{T}^{(\mathrm{A})}$: in this case, the two PVC beats.

We formalize these observations with our proposed representation, the *Contrast Profile*, specializing from the

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

generic $\mathbf{T}^{(A)}$ and $\mathbf{T}^{(B)}$, to consider two time series $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ which have a mild assumption about their contents.

Definition 6: A *Contrast Profile* $\mathbf{CP}_m$ is the difference between Matrix Profiles $\mathbf{MP}_m^{(+-)}$ and $\mathbf{MP}_m^{(++)}$, where $\mathbf{MP}_m^{(+-)}$ joins $\mathbf{T}^{(+)}$ with $\mathbf{T}^{(-)}$, and $\mathbf{MP}_m^{(++)}$ is the self-join of $\mathbf{T}^{(+)}$.

$$\mathbf{CP}_m = (\,\mathbf{MP}_m^{(+-)} - \mathbf{MP}_m^{(++)}\,)/\sqrt{(2*m)}$$

The Contrast Profile is defined for any two time series so long as *m* is shorter than the time series' lengths. However, we proposed to compute the Contrast Profile only when we believe that the two following assumptions are likely to be true:

- $\mathbf{T}^{(+)}$ contains at least two behaviors that are unique to the phenomena of interest.
- $\mathbf{T}^{(-)}$ contains zero behaviors of interest.

Under these assumptions, large values of $\mathbf{CP}_m$ indicate behaviors that appear two or more times in $\mathbf{T}^{(+)}$ while absent from $\mathbf{T}^{(-)}$. Fig. 6 gives a visual intuition of these definitions. Note that $\mathbf{CP}_{128}$ peaks at the locations of the shape that is unique to $\mathbf{T}^{(+)}$ (i.e., the two PVC heartbeats).

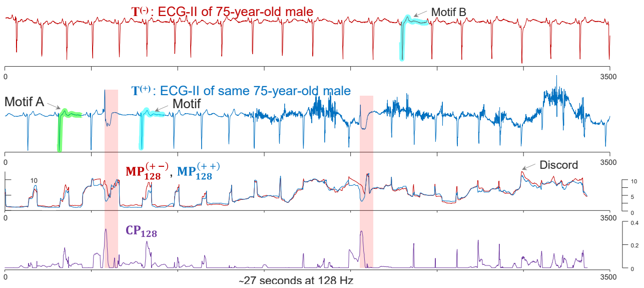


Fig. 6. *top-to-bottom*) Query time series $\mathbf{T}^{(-)}$ contains normal heartbeats. Time series $\mathbf{T}^{(+)}$ contains at least two instances of a behavior of interest. The top discord of the AB-join Matrix Profile (the highest peak), results from a noisy region in $\mathbf{T}^{(+)}$, far from the ground truth labeled with red bars. The top two candidates peak within the ground truth.

The subsequence in $\mathbf{T}^{(+)}$ corresponding to the highest point in the Contrast Profile is called the *Plato*, a backronym of *Pattern likely able to organize,* which is suggestive of a *platonic* ideal for some behavior of interest.

While we use the Matrix Profile as the core function to compute the Contrast Profile, the value optimized is rather simple. The Plato is the subsequence in $\mathbf{T}^{(+)}$ with maximum difference between its nearest neighbor distance in $\mathbf{T}^{(-)}$ and nearest neighbor distance in $\mathbf{T}^{(+)}$. This could be discovered by a classic nested-loop, brute-force algorithm, requiring $O(|\mathbf{T}^{(+)}|\,(|\mathbf{T}^{(-)}| + |\mathbf{T}^{(+)}|)\,m)$. As *m* could be in the thousands, this is clearly intractable. As we will later show, by exploiting the Matrix Profile, we can completely remove the dependence on *m* to produce a highly scalable algorithm.

To summarize, we have shown that at least for our running example, the Contrast Profile can be used to extract discriminating subsequences. This clearly has implications for several downstream algorithms, including classification and novelty/anomaly detection. However, before discussing these, in the next two sections we will consider the Contrast Profile's robustness to noise and the plausibility of the assumptions that warrant its use.

A. *The Contrast Profile's Robustness*

The robustness of the Contrast Profile definition is hinted at in Fig. 6, as it correctly recovers the PVC patterns in spite of sporadic noise caused by motion artifacts. Moreover, we empirically test this robustness in our experimental section. However, it is worth explaining *why* it is so robust. Consider Fig. 7.*top* which shows $\mathbf{CP}_{128}$ computed from two relatively clean ECGs, to hint at the presence of novel patterns (PVCs) in $\mathbf{T}^{(+)}$. Moreover, as we show in [7] (for brevity), while this particular example shows *m* = 128, the PVCs are discovered with any setting in the range *m* = [50:500].

Suppose we concatenate some random data to our time series; how would it affect the Contrast Profile? As Fig. 7.*bottom* shows, the relevant section of $\mathbf{CP}_{128}$ is essentially unchanged.

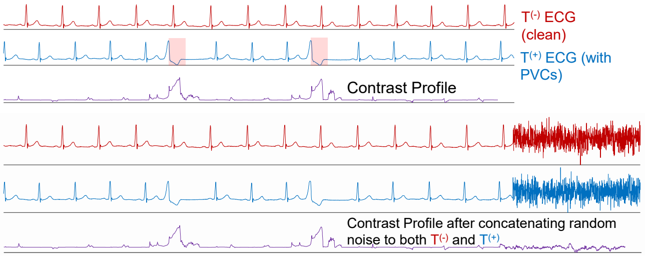


Fig. 7. *top*) Two ECGs and their Contrast Profile. *bottom*) After concatenating random data to the two time series, the section corresponding to the original $\mathbf{CP}$ is virtually unchanged, and the new section has low values, meaning it will not cause spurious matches.

Let us examine this finding for each time series.

- If $\mathbf{T}^{(-)}$ has sections of noise or any irrelevant data, it makes no difference to the $\mathbf{CP}$, because none of the irrelevant data will act as a nearest neighbor to any subsequence from $\mathbf{T}^{(+)}$.
- If $\mathbf{T}^{(+)}$ has sections of noise, it makes no difference to the $\mathbf{CP}$ because the noisy data will be approximately as far from its nearest neighbor in $\mathbf{T}^{(-)}$[1] as it is far from its nearest neighbor in $\mathbf{T}^{(+)}$. Thus, the relevant indices of the $\mathbf{MP}_m^{(+-)} - \mathbf{MP}_m^{(++)}$ calculation will subtract two nearly equal numbers, resulting in a score less than or equal to zero.

Thus, so long as our mild assumptions are true, the Contrast Profile is likely to discover the discriminating behavior.

B. *The Contrast Profile Assumptions*

Recall that our assumptions for the Contrast Profile are that $\mathbf{T}^{(+)}$ contains at least two examples of the desired behavior, but

---

[1] As an implication of theorem 1 of [6]) a noisy subsequence is an approximately equal and *large,* distance to all other subsequences.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

that $\mathbf{T}^{(-)}$ contains zero examples. These are very mild assumptions, nevertheless it is worth considering when such assumptions are warranted.

Our chicken example, shown in Fig. 1 and empirically revisited in Section V.B, is one such documented example where this assumption is justified [3]. We argue that such examples are very common in many domains. For example, in batch processing with *delayed coking* [8], it is possible to produce a bad batch, but not be sure exactly *when* the process began to fail. Thus, it is common to hear a petrochemical engineer report something like *"Everything was running perfectly on Monday, but then we had a couple of bad batches on Tuesday."* Here we can compute the Contrast Profile with $\mathbf{T}^{(-)} \leftarrow$ telemetry(Mon) and $\mathbf{T}^{(+)} \leftarrow$ telemetry(Tues).

Moreover, in some cases we can actually *intervene* in a domain to ensure data that conforms to our assumptions. For example:

- While it was not necessary for our chicken work, we *could* have created a cage that had all the usual food and water supplies, but had a solid floor to prevent the bird from dustbathing. This would have made the $\mathbf{T}^{(-)}$ trivially true.
- For our insect example (cf. Section V.A), suppose we hope to find the signature of "*cell rupture*" feeding as opposed to the more common "*salivary sheath*" feeding [9]. Since it is known what parts of a plant allow such feeding strategies, it would be easy for an entomologist to add/remove the appropriate plant parts into the insectary to build $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ with the Contrast Profile assumptions satisfied.

Finally, our assumptions are satisfied trivially in most examples relating to *human* behavior. For example, if we want to understand what effect (if any) a stretching warm-up routine has on a hurdler's jumps, we can simply ask her to record her workouts with and without the warm-up routine.

C. General Contrast Profile Observations

Note that while the two time series that are input into the Contrast Profile are denoted $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$, there is nothing pejorative about the "negative" time series. It is simply a snippet of data which we know does not have some behavior. That behavior *could* be undesirable, say a seizure, or it could be desirable, say a critical depressurization phase in an industrial process.

The Contrast Profile is bound between zero and one. A value of one corresponding to $\mathbf{T}^{(+)}_{i,m}$ means that $\mathbf{T}^{(+)}_{i,m}$ is a perfect motif in $\mathbf{MP}^{(++)}_{m}$ while also a maximum discord in $\mathbf{MP}^{(+-)}_{m}$ [1]. A value of zero means that $\mathbf{T}^{(+)}_{i,m}$ is conserved at least as much in $\mathbf{MP}^{(+-)}_{m}$ as $\mathbf{MP}^{(++)}_{m}$.

This property is critically different from that of TS-Diff [10], which is optimized solely by maximizing $\mathbf{MP}^{(+-)}_{m}$, a definition that simply tends to point to the noisiest subsequence.

A useful property of the Contrast Profile is that it is length invariant and sampling-rate invariant. For example, we can meaningfully compare scores for length 50 and for length 60, and state which subsequence is better conserved. This provides us with the opportunity to remove the Contrast Profile's only parameter, the subsequence length. We propose the Pan-Contrast Profile (in the spirit of [11]). We can simply compute all Contrast Profiles in some range, and choose the Plato from the one that produces the highest value. To see why we can expect this to work, consider the two extreme cases.

- If m is too small, then we are only comparing tiny fragments of the time series. These are very unlikely to be discriminating.
- If m is too large, then we are comparing the most discriminating subsequence along with extra non-discriminating shapes padded to its prefix or suffix. These non-discriminating sections can only dull the contrast property.

In Fig. 8, we show the Pan-Contrast Profile for the ECG shown in Fig. 7; the example bears out our intuition above. The optimal Plato has a length of 313, which is about the length of the PVC, excluding the QRS peak, which it shares with healthy heartbeats.

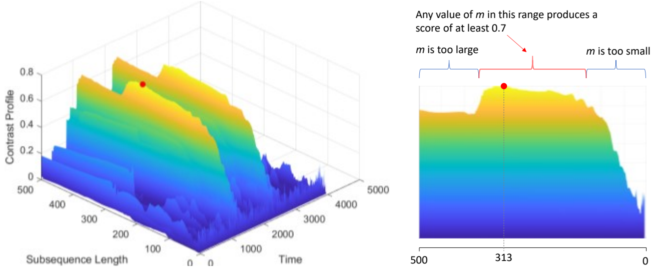


Fig. 8. *left*) The Pan-Contrast Profile for the example shown in Fig. 7. A red dot indicates the largest value. *right*) A side view shows that the Contrast Profile is very robust to its only input parameter. Any subsequence length from 131 to 424 would have produced a score of at least 0.7.

One additional takeaway from this experiment is the relative insensitivity of the Contrast Profile definition to its only parameter. Over a huge range of values (131 to 424) it produces nearly identical values in nearly identical locations.

A computation of a single Contrast Profile requires $O(|\mathbf{T}^{(+)}|^2 + |\mathbf{T}^{(+)}||\mathbf{T}^{(-)}|)$ time. To concretely ground this, the example shown in Fig. 7 takes 0.182 seconds, and the full Pan-Contrast Profile shown in Fig. 8 takes 82 seconds. Note that because the Contrast Profile is based on the Matrix Profile, it inherits many of the Matrix Profile's desirable properties such as time complexity that is completely independent of the subsequence's dimensionality, and the possibility of anytime, online, and GPU-accelerated computation [12].

Thus far, we have only defined the Top-1 Plato. However, it is possible that we may be interested in the Top-K Platos, as we may suspect that the behavior of interest is polymorphic. For example, unlike the simple PVC arrhythmia shown in Fig. 16.*inset*, some arrhythmias such as bidirectional ventricular tachycardia can present themselves with a handful of different shapes even from a single individual. If we are given $\mathbf{T}^{(+)}$ that

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

has at least two examples of each manifestation, we would like to extract them all.

Recall that for time series discords, the Top-K discords correspond to the Top-K peaks in the Matrix Profile. However, that is not the case for the Contrast Profile. To discover the K$^{\text{th}}$ Plato we must ensure that the influence of the K$^{\text{th}}$-1 Plato is first removed from the Contrast Profile. That is trivial to achieve, we simply concatenate the K$^{\text{th}}$-1 Plato to $\mathbf{T}^{(-)}$ and then recompute the Contrast Profile from scratch[2]. All subsequences in $\mathbf{T}^{(+)}$ that were similar to the K$^{\text{th}}$ Plato will then be close to a subsequence in $\mathbf{T}^{(-)}$, and thus their original peaks will vanish.

D. *Online Contrast Profile*

The reader will appreciate that it may be useful to compute the Contrast Profile in an online fashion. While "online" could have several interpretations, we believe the most useful variant will be a fixed $\mathbf{T}^{(-)}$ with an incrementally updated $\mathbf{T}^{(+)}$ in the face of real-time data arrival. As we will show in Section IV.C, this interpretation maps onto a type of anomaly detection.

Assume that we start with a computed $\mathbf{CP}_m$ of length n for $\mathbf{T}^{(+)}$, and some length for $\mathbf{T}^{(-)}$, and we wish to ingest an additional datapoint, the $n+1$ datapoint. This will result in the creation of a new subsequence, NEW, which ends with the $n+1$ datapoint.

What effect will subsequence NEW have on the current $\mathbf{CP}_m$, beyond lengthening it by one?

If NEW is sufficiently dissimilar to any other subsequence in $\mathbf{T}^{(+)}$, then the previous n values of $\mathbf{CP}_m$ will be unchanged regardless of NEW's distance to its nearest neighbor in $\mathbf{T}^{(-)}$.

If NEW is similar to one or more subsequences in $\mathbf{T}^{(+)}$, but also sufficiently close to its nearest neighbor in $\mathbf{T}^{(-)}$, then the previous n values of $\mathbf{CP}_m$ will again be unchanged.

If NEW is similar to one or more subsequences in $\mathbf{T}^{(+)}$, and it is far from any subsequence in $\mathbf{T}^{(-)}$, then we will have to update $\mathbf{CP}_m$ corresponding to those subsequences.

From this, we can see that the previously computed $\mathbf{CP}_m$ values can only increase or stay the same. They can never decrease. Then, adding the $n+1$ value to $\mathbf{CP}_m$ requires computing every index in $\mathbf{DP}^{(+-)}_{NEW,m}$ and $\mathbf{DP}^{(++)}_{NEW,m}$. After outlining the algorithm that maintains the Contrast Profile *Incremental* (ContrastProfile*I*) in TABLE I. , we will explain how this process can be accomplished surprisingly efficiently by exploiting the MASS algorithm [5].

We denote the updated variables with ^{NEW} in the superscript. In line 1, each newly arriving time point $t^{(+)}$ is appended to the expanding time series $\mathbf{T}^{(+)}$. This completes the next subsequence NEW in $\mathbf{T}^{(+),NEW}_m$ in lines 2 and 3. Lines 4 and 5 correspond to updating the contrasting Matrix Profile by first calculating the distance profile $\mathbf{DP}^{(+-)}_{last,m}$ between $\mathbf{T}^{(-)}$ and NEW, then appending the minimum of $\mathbf{DP}^{(+-)}_{last,m}$ to $\mathbf{MP}^{(+-)}_m$ and storing in $\mathbf{MP}^{(+-),\,NEW}_m$.

TABLE I. THE CONTRASTPROFILE*I* ALGORITHM

```
Algorithm: ContrastProfileI($\mathbf{T}^{(-)}$, $\mathbf{T}^{(+)}$, $t^{(+)}$, $\mathbf{MP}^{(+-)}_m$, $\mathbf{MP}^{(++)}_m$, $m$)
Input: negative time series $\mathbf{T}^{(-)}$, positive time series $\mathbf{T}^{(+)}$, a new positive time point $t^{(+)}$ following $\mathbf{T}^{(+)}$, Matrix Profile $\mathbf{MP}^{(+-)}_m$, Matrix Profile $\mathbf{MP}^{(++)}_m$, and subsequence length $m$.
Output: The Contrast Profile $\mathbf{CP}_m$, the incrementally updated Matrix Profiles $\mathbf{MP}^{(+-),NEW}_m$ and $\mathbf{MP}^{(++),NEW}_m$, and the current time series $\mathbf{T}^{(+),NEW}$.
1  $\mathbf{T}^{(+),NEW} = [\mathbf{T}^{(+)}, t^{(+)}]$
2  $last \leftarrow n - m + 1$ // index of last subsequence in $\mathbf{T}^{(+),NEW}$
3  $NEW \leftarrow \mathbf{T}^{(+),NEW}_{last,m}$ // last subsequence in $\mathbf{T}^{(+),NEW}$ of length m
4  $\mathbf{DP}^{(+-)}_{last,m} \leftarrow$ MASS($\mathbf{T}^{(-)}$, $NEW$)) // Begin AB-join update
5  $\mathbf{MP}^{(+-),\,NEW}_m \leftarrow [\mathbf{MP}^{(+-)}_m$, Min($\mathbf{DP}^{(+-)}_{last,m}$)]
6  $\mathbf{DP}^{(++)}_{last,m} \leftarrow$ MASS($\mathbf{T}^{(+)}$, $NEW$) // Begin self-join update
7  $\mathbf{MP}^{(++)\prime}_m \leftarrow$ ElemWiseMin($\mathbf{MP}^{(++)}_m$, $\mathbf{DP}^{(++)}_{last,m}$) // Update prev vals
8  $\mathbf{MP}^{(++),\,NEW}_m \leftarrow [\mathbf{MP}^{(++)\prime}_m$, Min($\mathbf{DP}^{(++)}_{last,m}$)]
9  $\mathbf{CP}_m \leftarrow (\mathbf{MP}^{(+-),\,NEW}_m - \mathbf{MP}^{(++),NEW}_m)$/sqrt(2 * m)
10 return $\mathbf{CP}_m$
```

Because this is a Matrix Profile where the query time series is unchanging, the previously computed values are also unchanged. An extra line of work is done in lines 6 – 8 to update the self-join Matrix Profile because the query time series $\mathbf{T}^{(+)}$ has expanded. The self-join distance profile between $\mathbf{T}^{(+)}$ and NEW is stored in $\mathbf{DP}^{(++)}_{last,m}$. The element-wise minimum between $\mathbf{MP}^{(++)}_m$ and $\mathbf{DP}^{(++)}_{last,m}$ is stored in $\mathbf{MP}^{(++)\prime}_m$, which is then updated to $\mathbf{MP}^{(++),NEW}_m$ after concatenating the minimum value of $\mathbf{DP}^{(++)}_{last,m}$. Finally, in lines 9 and 10, $\mathbf{CP}_m$ is recomputed from the expanded $\mathbf{MP}^{(+-),NEW}_m$ and updated and expanded $\mathbf{MP}^{(++),NEW}_m$.

The time complexity of ContrastProfile*I* is dominated by the MASS function, which performs an O($n$log$n$) FFT operation. The time complexity begins as O($|\mathbf{T}^{(+)}|$ log $|\mathbf{T}^{(+)}|$ + $|\mathbf{T}^{(-)}|$ log $|\mathbf{T}^{(-)}|$), but as the size of $\mathbf{T}^{(+)}$ dominates, the effective time complexity is O($|\mathbf{T}^{(+)}|$log$|\mathbf{T}^{(+)}|$). Each time the function is called, MASS searches a slightly longer time series with n becoming $n+1$. There are no conditional control statements, making the runtime value-invariant to the incoming data.

This time complexity discussion is a little indirect. A more intuitive way to measure the time requirements is by using the Maximum Time Horizon, which answers the question, "*How long can the Contrast Profile be maintained before the maintenance computation is slower than the sampling rate?*"

For example, consider the following two scenarios which refer to an Intel® Core i7-9700 CPU at 3.00GHz with 32 GB of memory (full worked details at [7]).

- If we have a Contrast Profile created with $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ both of length 10,000, and the data is arriving at 10Hz,

---

[2] This is what *logically* must be done, however by caching distance calculations and only recomputing values that could have changed, the time and space overhead for the K$^{\text{th}}$-1 Plato is inconsequential.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

then we can update the Contrast Profile for about 51 hours before the arrival rate is faster than our update time.

- Most automotive GPS loggers update at 1hz. If we have a Contrast Profile created with $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ both of length 10,000, with data arriving at 1Hz, then we can update the Contrast Profile for about 9.5 months before the arrival rate is faster than our update time.

Note that we do not specify the value of m in the above, as the update times are effectively invariant to the subsequence length due to the use of the MASS algorithm.

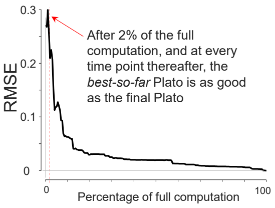
E. Anytime Contrast Profile

The computation time for the Contrast Profile is not particularly onerous relative to the tasks it can be used to solve. Nevertheless, it is natural to ask where the current limits of computation are in terms of the size of the datasets we can consider. Here we can take advantage of the fact that the Contrast Profile is largely comprised of calls to the *Matrix Profile*. In recent years Matrix Profile has become a widely studied algorithm, with multiple high-performance computational paradigms now available, including distributed computation and GPU implementations. Here we will show that we can also exploit the *anytime* computation property from the Matrix Profile [13].

Clearly it would be pointless to first compute $\mathbf{MP}_m^{(+-)}$, and only *then* compute $\mathbf{MP}_m^{(++)}$ in anytime fashion, as we would have to wait for about half the overall time to get the first approximation. Fortunately, the SCRIMP++ algorithm for the Matrix Profile has perfect *interruptibility* and *preemptability* [13]. This means we can spend p% of our time computing $\mathbf{MP}_m^{(+-)}$, then suspend it and then switch to spending p% of our time computing $\mathbf{MP}_m^{(++)}$, then toggle back to $\mathbf{MP}_m^{(+-)}$ etc.

The only question is what value 'p' should have. If there is a large cost in suspending and then resuming an algorithm, this can be a difficult question to answer. But for the SCRIMP++ algorithm, this cost is inconsequential (less than 0.0001% of the overall cost). Because of this, we can make p arbitrarily small. For simplicity we chose $p = 0.5\%$. We can measure the utility of the anytime property in two ways, the rate at which the best-so-far Contrast Profile converges on the final Contrast Profile (as measured by the RMSE), and by asking at what point the best-so-far Plato is as good as the final Plato. Here we define "as good as" to mean able to obtain within 1% of the latter's classification accuracy on holdout data. In Fig. 9 we answer both questions for the ECG example considered in Section V.C.

Note that while the convergence plots for a Matrix Profile are strictly monotonic, this is not necessarily the case for the Contrast Profile. Nevertheless, as this figure shows, both numerically and semantically, the Contrast Profile converges very quickly.



After 2% of the full computation, and at every time point thereafter, the *best-so-far* Plato is as good as the final Plato

Fig. 9. Anytime convergence plot for the ECG data shown in Fig. 16.

III. Related Work

The work closest in spirit to the Contrast Profile is in time series *contrast-set mining* by Lin and Keogh [10]. Their work proposes to solve the same problem of identifying subsequences that maximize the differences between two time series. This work predates the Matrix Profile but in retrospect can be seen as returning the subsequence pointed to by the location that maximizes $\mathbf{MP}_m^{(+-)}$. The problem with this definition is that it tends to simply return the most complex and/or noisiest subsequence. For example, for the data shown in Fig. 7.*bottom,* it will return a subsequence of the pure noise at the right end of the $\mathbf{T}^{(+)}$ signal (see [2]).

If we generalize beyond time series, multiple instance learning, pioneered by Dietrich et al. [14], is another closely related concept. In order to identify a unique feature of a desired class, two labeled "bags" are provided: The class-positive bag must contain at least one sample with the desired property, while the class-negative bag must contain zero samples with the desired property. There have been some attempts to generalize this framework to time series by researchers noting that you cannot (meaningfully) convert the time series problem to a classic multiple instance framework by use of "sliding windows". However, the proposed methods require significant feature engineering and a change of representation. In comparison, the Contrast Profile works directly on the *raw* data.

Moreover, the Contrast Profile is tasked with finding discriminating exemplars, but is completely agnostic as to what classification algorithm will then be used. In contrast, [15] is tied to a particular Auto-Regressive Hidden Markov Model classification paradigm. Finally, the HMM requires the learning of six explicit parameters, whereas the Contrast Profile has one parameter, or zero parameters for the Pan-Contrast Profile.

The general literature of time series classification is vast; however, it is mostly orthogonal to this work. As we noted above, virtually all research efforts only consider time series objects after they have been extracted from a longer time series. However, they are silent as to *how* they can be extracted. In most cases, the community has bypassed this issue by only evaluating on the UCR archive, where many of the datasets were processed with human annotations and access to out-of-band information and domain knowledge.

IV. Algorithms That Exploit The Contrast Profile

We believe that the Contrast Profile may be a useful primitive within dozens of higher-level algorithms. In this section we give some concrete examples.

A. End-to-End Time Series Classification

As we noted in the introduction, *given* discriminative subsequences (i.e., in the UCR format [2]) that characterize a behavior, time series classification is generally a simple task. We argue that *finding* such discriminative subsequences can be extremely difficult. Clearly the Contrast Profile has the potential to mitigate this difficulty. For concreteness, we outline a basic approach:

- Identify two snippets of time series that conform to the Contrast Profile assumptions (See Section II.B).

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

- Run the Pan-Contrast Profile to discover the Plato.
- Use this Plato with a threshold t to discover similar instances, label them as the class that $\mathbf{T}^{(+)}$ represents.

Note that while the Euclidean distance is the natural distance measure to use, other measures such as DTW are possible [8]. We need to set a threshold; here we must resort to heuristics. For example we can use 3 × the distance for the Plato to its nearest neighbor (recall that we are assuming that the Plato's nearest neighbor is *also* an example of the desired behavior). Finally, the above assumes that there is a single template for the desired behavior. If we think it may be polymorphic, we can use the technique discussed in Section II.C to find the Top-K Platos instead. This is a very simple technique for end-to-end classification, but as we will show on diverse real-word problems, *extremely* effective.

B. Exploratory Data Mining

We believe that the most frequent use of the Contrast Profile may be in exploratory data mining. That is, searching datasets with the goal of finding interesting hypotheses to further evaluate. As this is a difficult task to evaluate objectively, we will confine ourselves to a single example here. In Fig. 9.*top* we show a pair of four-year long time series that record pedestrian traffic in two locations in Melbourne, Australia. A city manager may wish to know, *"What happens only in Bourke Street but not elsewhere?"*

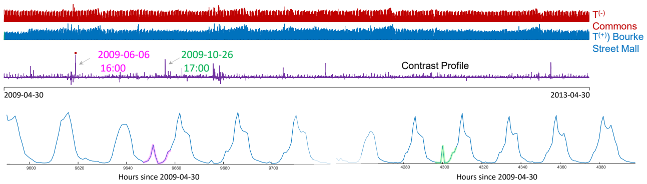


Fig. 10. *top*) Two four-year-long time series and their $\mathbf{CP}_{12}$. *bottom*) The two contrasting subsequences shown in context.

This is a difficult question to answer; the data have daily, weekly and seasonal variation in addition to spikes and dips caused by both cultural events (protests, flash mobs, strikes) and severe weather events. As shown in Fig. 9.*bottom* there is a subtle distinction between the two time series. Only the Bourke Street data has small spikes at what is otherwise the quietest part of the day. This might be a coincidence, but we have an additional 4.75 years of data, and we find that this distinction holds up with a handful of similar events over the longer time period. What is the cause of this pattern? Using Reddit we crowdsourced that question, and one volunteer noted that the area is dominated by a branch of Myers, Australia's largest department store chain. She went on to note *"That was around the time of the new store opening up and I worked there at the time. We had ..* (EOD fire drills) *evacuations mid 2009-mid 2010. ..the Bourke doors being the only way to access Myer some days"*. In brief, the volunteer believes this burst is the evacuation of Myers during mandatory fire drills, with staff pouring out into the street. Whatever the reason, this example shows the potential utility of the Contrast Profile to find regularities that would otherwise be difficult to notice.

C. Anomaly Detection: Solving the Twin Freak Problem

While there are dozens of approaches for the task of anomaly detection in time series, it has long been known that shape-based methods (i.e. *discords* and their variants) can be competitive [4]. Here we propose a novel interpretation of anomaly detection, the Contrast Profile Anomaly Detector (CPAD). We propose creating a $\mathbf{T}^{(-)}$, which contains all the "allowable" behavior of a system, and then expanding $\mathbf{T}^{(+)}$ online as new data arrives (see TABLE I.). The $\mathbf{T}^{(-)}$ may be a natural time series or created from the concatenation of several time series. For example, if we wanted to build an anomaly detector to monitor pedestrian traffic at a train station, we could create a concatenation of one weekday, one weekend day, one bank holiday, one rainy day etc.

This idea is somewhat similar to "golden batch" monitoring, where engineers find or create an ideal representation of a manufacturing process, and continuously compare an ongoing process to it [8]. However, our formulation generalizes in several ways. We do not need to obtain or understand the process in detail, we simply need to *obtain* data that we think covers the space of allowable behavior.

The CPAD has a very unusual property that makes it something of a hybrid between an anomaly detector and a regime-change detector. In particular, the first occurrence of a new pattern will not register a high score. It is only when the *second* occurrence is seen that the Contrast Profile value spikes to signal an anomaly.

We believe that there are problems for which this is the ideal behavior. Recall the $\mathbf{T}^{(+)}$ shown in Fig. 6. It shows that ECGs are often contaminated with noise, which are typical sensor artifacts. We typically do not want an anomaly detector to alert for these (in any case, they are trivial to monitor with simpler methods). However, if we are monitoring a patient recovering in an ICU and we see novel beats, they may have sudden onset paroxysmal atrial fibrillation.

As with the previous task, we will confine our consideration to anecdotal examples, saving space for more forceful and quantitative evaluation of classification in the next section.

We consider data from a house that has individual appliances metered. This particular house happens to have two refrigerators. Refrigerator power demand has a very approximately square wave appearance, as the cycle between *on*/*off* compressor cycles. However, the timing of these cycles drift as the room temperature drifts and in response to someone opening the refrigerator door. In addition, at the very beginning of the *on* cycle, there is an increased demand as the compressor struggles to build rotational inertia. The latter variability is somewhat unique to each device depending on the rotational mass. This allows us to perform a natural experiment. We use a diverse selection of snippets from the refrigerator time series as $\mathbf{T}^{(-)}$. We then use the online Contrast Profile algorithm outlined in TABLE I. to monitor newly arriving data from the same stream. After 4.5 hours, we switched the leads to monitor the *other* refrigerator in the same house. This "refrigerator swap" models the situation where a system unexpectedly changes to allow instances from a new and unanticipated class. This is exactly what happens for certain cardiological diseases, or in

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

batch processing if one component (say a valve or pump) begins to fail and then produces a different batch profile.

As Fig. 11 shows, the difference caused by the refrigerator switch is too subtle to be seen, at least at this scale. Nevertheless, the CPAD algorithm clearly captures this event.

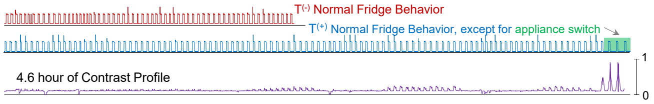


Fig. 11. An example of CPAD. *top*) A time series that represents normal behavior of a fridge. *center*) A snippet from the same refrigerator, with the final moments swapped out for a different fridge from the same house. *bottom*) The Contrast Profile strongly spikes to indicate conserved novel behavior.

For this dataset we could update the online Contrast Profile 3,660 times faster than real time.

It is interesting to note that the classic discord definition has had one criticism levied at it for over a decade, in dozens of papers. In [16] the authors noted "*discords miss similar anomalies*" , likewise [17] notes the discords fail "*because our dataset includes several anomalies that are similar to one another*". In other words, if there are two or more occurrences of undesirable behaviors, and they happen to have the same shape in each occurrence, then by definition it is a motif rather than a discord. Note that our framing of anomaly detection completely solves that problem, because as shown in this example, it can *only* find anomalies that occur at least twice. Thus, the union of the discords discovered using the Matrix Profile [1] and the anomalies discovered with the proposed Contrast Profile can be used to create a hybrid-definition that includes all anomalies, independent of how often they occur.

V. EXPERIMENTAL EVALUATION

To ensure that our experiments are reproducible, we have built a website [7] which contains all data/code for the results, in addition to many experiments that are omitted here for brevity. We have created a detailed document that with concrete details of our experiments, that we believe will allow anyone to reproduce all our experiments with less than one hour of effort. All experiments were conducted on an Intel® Core i7-9700 CPU at 3.00GHz with 32 GB of main memory.

While we hinted at several downstream algorithms that can exploit the Contrast Profile, we will mostly confine our attention to end-to-end classification. We remind the reader that the Contrast Profile is a data mining tool whose output can be repurposed as a nearest neighbor classifier, which allows for objective measures of utility.

As noted in the related work section, to the best of our knowledge, there is no other algorithm that performs prototype extraction under our assumptions. Where appropriate we consider the baseline of the default rate representing a *random* selection of K candidate behaviors.

A. Insect Behavior Classification

Sapsucking insects (insects in the orders Hemiptera and Homoptera) are insects that feed by sucking nutrients from plants. This behavior is typically not destructive by itself but can spread diseases from plant to plant. Worldwide, across all crops/insects, this results in billions of dollars in crop losses each year. The primary tool used to study these insects is the electrical penetration graph (EPG), which as shown in Fig. 12, produces a complex and noisy time series that reflects the insect's behavior [9].

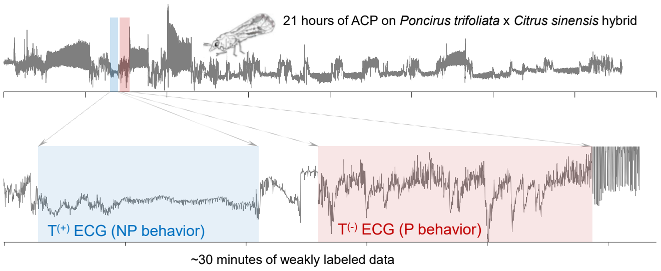


Fig. 12. *top*) 21 hours of Asian citrus psyllid (ACP) feeding behavior on citrus. *bottom*) A zoom-in of a small fraction of the data.

We managed to obtain 21 hours of such data that was annotated by a combination of algorithms and humans (exploiting out-of-band information). Using the two regions shown in Fig. 12.*bottom*, that conform to our algorithm's mild assumptions, we ran the Contrast Profile to produce the Plato shown in Fig. 13.

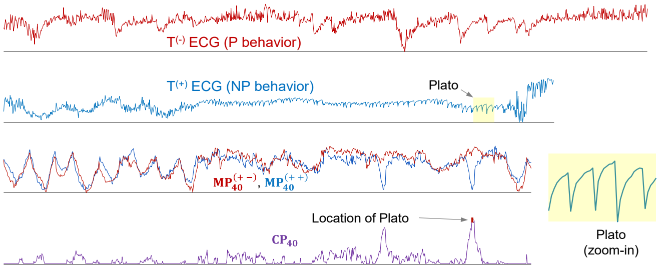


Fig. 13. *top-to-bottom*) The weakly-labeled instances shown in Fig. 12 have their $\mathbf{MP}_{40}^{(+-)}$and $\mathbf{MP}_{40}^{(++)}$computed to produce the $\mathbf{CP}_{40}$, which strongly peaks to indicate the location of the Plato.

Using this template to find the Top-100 instances in the full dataset (excluding training data), the Plato had an error-rate of 7%, whereas the Top-1 motif in $\mathbf{T}^{(+)}$ had an error-rate of 32%, not much better than the default error rate of 36.9%

B. Chicken Behavior Classification

Here we revisit the chicken behavior example considered in Fig. 1. First, we should explain why the data is weakly- labeled. The accelerometer worn by the bird *was* approximately synchronized with a video camera trained on the coop. However, technical limitations meant that the synchronization had an error of up to ± 3 seconds. By comparison, the dustbathing behavior we were tasked with quantifying is known to last about 0.5 to 3 seconds. Thus, a domain expert was able to locate 30-second regions *with* and *without* the behavior, but not provide annotations at a *finer* temporal resolution. In Fig. 14 we use the two time series shown in Fig. 1 to compute $\mathbf{CP}_{120}$ in an attempt to find a Plato that can act as a "signature" for dustbathing.

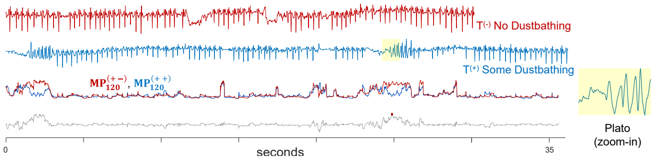


Fig. 14. *top-to-bottom*) The weakly-labeled instances shown in Fig. 1 have their **$MP_{120}^{(+-)}$** and **$MP_{120}^{(++)}$** computed to produce the **CP_{120}**, which strongly peaks to indicate the location of the Plato.

We used this Plato to search a 12,679,054,727 datapoint archive of chicken behavior for the one thousand best matches. The returned matches are shown in Fig. 15.

Domain experts examined the results and confirmed that all the returned subsequences are true positives.

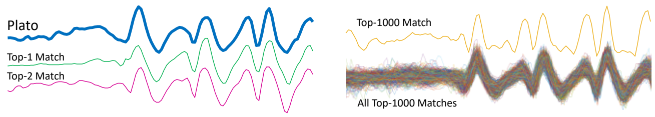


Fig. 15. The Plato used for dustbathing classification (*top.left*). Selected matches returned by a nearest neighbor search using the Plato discovered in Fig. 14. The Top-1000 matches (*bottom.right*).

The discovery of the Plato took 0.3 seconds. Surprisingly, the exact Top-1000 search in the 12.7 billion datapoints of disk-resident data (corresponding to four years of behavior) took only 55 minutes using the MASS algorithm.

C. Electrocardiogram Classification

We consider a 23.5 hour (10,828,800 datapoints at 128Hz), ECG dataset from a 46-year-old male (MIT 14046). The dataset has a beat-by-beat annotation created by a combination of specialized algorithms and human expert inspection.

We examine the first five minutes of the data to find a 30-second region free of anomalies **$T^{(-)}$** and a 30-second region with *at least* two (actually, *six*) anomalies **$T^{(+)}$**, which we later learned are PVCs. Running the Pan-Contrast Profile for lengths 32 to 128 (a ¼ second to a full second) produced the 91-datapoint Plato shown in Fig. 16.

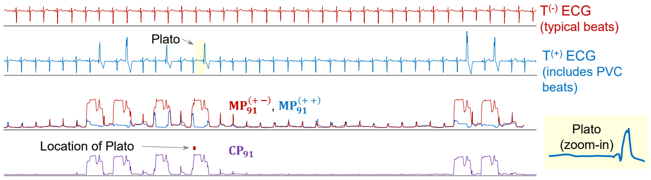


Fig. 16. top-to-bottom) Two weakly-labeled snippets from first 5 minutes used to compute their **$MP_{91}^{(+-)}$** and **$MP_{91}^{(++)}$**, which then produce the **CP_{91}**, with high amplitude peak to indicate the location of the Plato.

By (naively) extrapolating the density of PVCs we see in our tiny training snippets, we expect to see 3,934 additional PVCs in the remaining 23.42 hours of data. Retrieving the Top-3,934 nearest neighbors we achieve 0.9992 precision. By contrast the default rate is only 0.1645.

Here our recall was only 0.4031, because we underestimated the number of PVCs. If we "cheat" and retrieve the ground truth number of PVCs (there are 9,753), we obtain 0.9047 precision and 0.9047 recall. As impressive as this is, these results appear to be pessimistic. An audit of our "false negatives" by cardiologist Greg Mason suggests that they are mostly mislabeled in the original data (discussion moved to [7]).

A recent paper surveyed thirteen approaches on datasets, including the one above (see Table 9 of [11]). While these works are not all directly comparable with each other or this work, the accuracy we report would place us high in this list. More important however is the speed and simplicity of our method. Consider:

- The median number of parameters that need to be tuned by the thirteen methods in [11] is seven (most are based on CNNs or LSTMs); in contrast, we have no parameters to tune.
- For a much smaller dataset, [11] notes that it took 4.5 hours to train the model and then took 0.05 seconds to classify each beat. In contrast, we needed 16.3 seconds to learn our Plato model and 0.00000294 seconds to classify each beat.
- The other approaches worked with *extracted* beats and, thus, required domain dependent code to first do the extraction. In contrast, we worked with the *raw* data without explicitly extracting beats or using any cardiological knowledge.

Our ability to classify each heartbeat in just 0.00000294 seconds may strike the reader as being implausibly fast, but it is possible using Mueen's MASS algorithm [5]. MASS takes just 0.3390 seconds to process a time series of length 10,828,800 containing 115,278 beats. This is a quarter of million times faster than real time. Of course, being so much faster than real time is of limited utility for monitoring an individual patient but is very useful for mining large data archives.

Finally, lest the reader think we chose an easy ECG, in [7] we repeated this experiment with other ECGs with similar results.

D. Model Comparison

We will now compare the Contrast Profile as a classifier to two other leading time series models: shapelets and LSTM. We acknowledge that there is not a 1:1 correspondence in purpose of the models and we will explain the possible discrepancies.

1) Comparison to Shapelets

Time series shapelets have very different assumptions to the proposed Contrast Profile. However, the Contrast Profile can duplicate at least *some* elements of time series shapelets (the opposite is not true). To see this, consider the following:

We can produce a positive time series **$T^{(+)}$** by concatenating all exemplars of one class, and **$T^{(-)}$** by concatenating all exemplars of the other class(es). Given two such assembled time series, we can simply hand them over to the Contrast Profile. The **Trace** dataset, which does not have polymorphic classes, is from the UCR Archive [2] that illustrates our purpose of extracting a single Plato. Using

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

uniform m of $(\frac{1}{8} * sampleLength)$ for all four classes results in an error rate of 0.17, but the misclassifications come solely from the fourth class, whose main feature is a rising edge, which also appears in other classes. Increasing m for class 4 to $(\frac{3}{4} * sampleLength)$ lowers the error rate to 0.09, Platos shown in Fig. 17. For comparison, the error rate for INNED and the default error rate are 0.24 and 0.71 respectively.

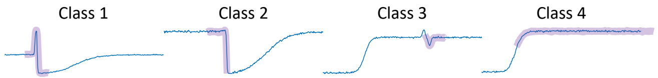


Fig. 17. Samples of each of the four classes in the **Trace** dataset overlaid with Plato.

The Plato's error rate is competitive, given that shapelets have access to many strongly *labeled* extracted samples.

2) Comparison to LSTM

For the sake of completeness, we also include a comparison to an off-the-shelf LSTM approach [18]. This experiment comes with a disclaimer. The Contrast Profile is *not* a classification algorithm. It simply extracts patterns that can be used for nearest neighbor classification, one of the simplest classification algorithms.

We test on the chicken behavior classification from Section V.B. As a reminder, the $\mathbf{T}^{(-)}$ (Class-0) training data contains behaviors to be ignored while the $\mathbf{T}^{(+)}$ (Class-1) training data contains at least two instances of the desired behavior, as well as other behaviors likely exhibited in $\mathbf{T}^{(-)}$. The LSTM's XTrain variable is constructed by extracting each subsequence from $\mathbf{T}^{(-)}$ and $\mathbf{T}^{(+)}$, then assigning a 0 or 1 accordingly in YTrain. Two parameter changes were made to the original model: [inputSize = 120, numClasses = 2]. The 51-minute training completed with 100% training accuracy. The test accuracy on a set of 2000 samples balanced between dustbathing and non-dustbathing was 60.20% compared to 98.55% for the Plato. We have no doubt that the results for the LSTM could be improved by some preprocessing. However, our point here is simply that in this real-world challenging problem, we can quickly achieve results that are better than the community's current "go-to" solution, and we can do this with significantly less human effort and parameter tuning.

VI. CONCLUSIONS

We have introduced the Contrast Profile, a novel data structure that allows a user or algorithm to reason about the differences between two time series. We reiterate that the Contrast Profile is *not* a classification algorithm, but it can help any downstream time series classification algorithm by finding discriminative prototypes. Beyond allowing end-to-end time series classification with only the weakest possible assumptions/annotations of the data, we have shown that the Contrast Profile has several other uses in data mining, including anomaly detection and data exploration. We have shared all code and data with the community [7], to allow it to confirm and exploit our findings.

REFERENCES

[1] C. M. Yeh *et al.*, "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec. 2016, pp. 1317–1322. doi: 10.1109/ICDM.2016.0179.

[2] "Welcome to the UCR Time Series Classification/Clustering Page." https://www.cs.ucr.edu/~eamonn/time_series_data_2018/ (accessed Jan. 17, 2021).

[3] A. C. Murillo, A. Abdoli, R. A. Blatchford, E. J. Keogh, and A. C. Gerry, "Parasitic mites alter chicken behaviour and negatively impact animal welfare," *Scientific Reports*, vol. 10, no. 1, Art. no. 1, May 2020, doi: 10.1038/s41598-020-65021-0.

[4] T. Nakamura, M. Imamura, R. Mercer, and E. Keogh, "MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives," p. 11.

[5] M. Abdullah, "The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance." https://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html (accessed Jan. 18, 2021).

[6] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is 'Nearest Neighbor' Meaningful?," in *Database Theory — ICDT'99*, Berlin, Heidelberg, 1999, pp. 217–235. doi: 10.1007/3-540-49257-7_15.

[7] "Contrast-Profile." https://sites.google.com/view/contrastprofile (accessed Jan. 05, 2021).

[8] "Online Amnestic DTW to allow Real-Time Golden Batch Monitoring | Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining." https://dl.acm.org/doi/abs/10.1145/3292500.3330650?casa_token=UbpbhUA-AVAAAAAA:q_989yn-jqUl_K5tJXlXvGclelo-cjs07zQAM9MDufctnKDu_IjL0ahMfQ4SPGuQp5-UAcU40k65Gg (accessed Dec. 29, 2020).

[9] D. S. Willett, J. George, N. S. Willett, L. L. Stelinski, and S. L. Lapointe, "Machine Learning for Characterization of Insect Vector Feeding," *PLOS Computational Biology*, vol. 12, no. 11, p. e1005158, Nov. 2016, doi: 10.1371/journal.pcbi.1005158.

[10] J. Lin and E. Keogh, "Group SAX: Extending the Notion of Contrast Sets to Time Series and Multimedia Data," in *Knowledge Discovery in Databases: PKDD 2006*, Berlin, Heidelberg, 2006, pp. 284–296. doi: 10.1007/11871637_29.

[11] O. Yildirim, U. B. Baloglu, R.-S. Tan, E. J. Ciaccio, and U. R. Acharya, "A new approach for arrhythmia classification using deep coded features and LSTM networks," *Computer Methods and Programs in Biomedicine*, vol. 176, pp. 121–133, Jul. 2019, doi: 10.1016/j.cmpb.2019.05.004.

[12] Y. Zhu *et al.*, "Exploiting a novel algorithm and GPUs to break the ten quadrillion pairwise comparisons barrier for time series motifs and joins," *Knowl Inf Syst*, vol. 54, no. 1, pp. 203–236, Jan. 2018, doi: 10.1007/s10115-017-1138-x.

[13] S. Zilberstein and S. Russell, "Approximate Reasoning Using Anytime Algorithms," in *Imprecise and Approximate Computation*, S. Natarajan, Ed. Boston, MA: Springer US, 1995, pp. 43–62. doi: 10.1007/978-0-585-26870-5_4.

[14] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1, pp. 31–71, Jan. 1997, doi: 10.1016/S0004-3702(96)00034-3.

[15] X. Guan, R. Raich, and W.-K. Wong, "Efficient Multi-Instance Learning for Activity Recognition from Time Series Data Using an Auto-Regressive Hidden Markov Model," p. 10.

[16] Y. Bu, L. Chen, A. W.-C. Fu, and D. Liu, "Efficient anomaly monitoring over moving object trajectory streams," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 159–168.

[17] W. M. Kouadri, M. Ouziri, S. Benbernou, K. Echihabi, T. Palpanas, and I. B. Amor, "Quality of sentiment analysis tools: the reasons of inconsistency," *Proceedings of the VLDB Endowment*, vol. 14, no. 4, pp. 668–681, 2020.

[18] "Sequence Classification Using Deep Learning." https://www.mathworks.com/help/deeplearning/ug/classify-sequence-data-using-lstm-networks.html (accessed Jan. 21, 2021).

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE