Linear-trend normalization for multivariate subsequence similarity search

Thibaut Germain
Université Paris Saclay
Université Paris Cité, ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli
F-91190, Gif-sur-Yvette, France.
thibaut.germain@ens-paris-saclay.fr

Charles Truong
Université Paris Saclay
Université Paris Cité, ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli
F-91190, Gif-sur-Yvette, France.
charles.truong@ens-paris-saclay.fr

Laurent Oudre
Université Paris Saclay
Université Paris Cité, ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli
F-91190, Gif-sur-Yvette, France.
laurent.oudre@ens-paris-saclay.fr

***Abstract*—Finding repeating or anomalous subsequences in long time series is a crucial task in numerous data analysis pipelines. Most of those methods share a common step where they compute the pairwise similarity between all subsequences of a time series or between a fixed subsequence and a time series. However, the presence of a trend in a time series may cause changes in the shape of subsequences, making the similarity measure less reliable. This article introduces a new normalization scheme called LT-normalization (for Linear Trend) to prevent this phenomenon. It generalizes the well-known Z-normalization by removing the linear trend and scaling the subsequences to unit variance. Like the Z-normalization, we show that the LT-normalization has a computationally efficient recursive formulation. Thanks to this recursion property, the LT-normalized matrix profile can be computed with the same quadratic complexity as the classical Z-normalized matrix profile. Our procedure can naturally cope with multivariate signals. Empirical results on synthetic and real datasets show that the LT-normalized matrix profile has competitive performances for the best motif pair, similarity search, and motif set discovery problems.**

***Index Terms*—time series, similarity measure, normalization, similarity search**

I. Introduction

Over the past two decades, motif discovery [24] and anomaly detection [20] on large time series have gained attention in the research community. Algorithms that solve those tasks take a long time series as input and find repeating or abnormal subsequences. Applications range from medicine [26], to seismology [28], and industry [27]. Most of those methods share a common step: computing the pairwise similarity between all subsequences of a time series or between a fixed subsequence and a time series. This task is often called "subsequence similarity search". For instance, the matrix profile and its variants [10], [15], [29] efficiently perform subsequence similarity search to find the best motif pair in a temporal signal or anomalous subsequences, among other time series primitives. For such algorithms, choosing a similarity measure between subsequences is an important step as it defines the kind of time series primitives that can be detected.

In several applications, the distance must be robust to some deformations of the sequences. For instance, in signals exhibiting a smooth trend, the same pattern can appear several times with different vertical offsets and scales. Depending on the task, such distortions should not affect the ability of an algorithm to detect patterns or anomalies. While complex shape deformations must be dealt with an appropriate distance measure (e.g., DTW), a common approach is to preprocess subsequences before computing sliding Euclidean distances for simple deformations and large time series. More precisely, subsequences are usually normalized before being compared. The normalization step has been shown to influence significantly the performance of data mining algorithms for time series [16].

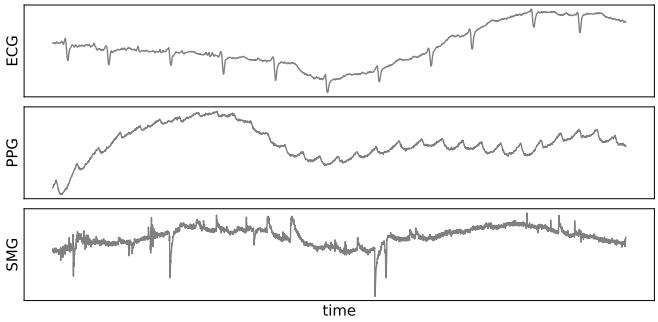

Fig. 1. **Top:** Electrocardiogram (ECG), **Middle:** Photoplethysmogram (PPG), **Bottom:** Seismogram (SMG).

Many normalization procedures exist in the literature, like Z-normalization, MinMax normalization, and UnitLength normalization. (see [16] for a review). In motif discovery, Z-normalization is the most popular for two reasons. First, it makes subsequent analysis robust to two frequent distortions, offset and scale. Second, it can be computed efficiently, which is crucial for algorithms that compute similarities between all subsequences. Indeed, the Z-normalized Euclidean distance between two subsequences can be computed in constant time by intelligently keeping track of specific quantities and computing similarities in the time order. Thanks to this algorithmic trick, the matrix profile algorithm has time complexity $\mathcal{O}(l^2)$, where l is the time series' length, which is significantly faster than the naive version, which has time complexity $\mathcal{O}(wl^2)$, where w is the subsequences' length. This trick is

essential for similarity search as signals can have hundreds of thousands or millions of samples [10], [15], resulting in approximately the same number of subsequences. It contrasts with other task settings, like classification or clustering, where such acceleration tricks are impossible to compare whole time series. For this reason, most normalization procedures used in classification or clustering cannot be applied out-of-the-box for subsequence mining.

Despite its popularity, Z-normalization is not robust to some deformations that affect real-world data, and that can be considered meaningless in some applications. In particular, deformations caused by trends are common in real-world time series, and removing their effect is necessary when comparing subsequences for some applications. Whenever the trend is smooth enough, its effect on a time series can be locally approximated as a linear trend, as illustrated in Figure 1. The top two figures show an electrocardiogram (ECG) and a photoplethysmogram (PPG), where the repeated patterns correspond to heartbeats and the trend is due to the movements of the subjects. The last figure shows a seismograph (SMG) with successive earthquakes altered by seismic noise. In all cases, the slowly evolving trend changes the orientation of the repeated patterns, making them difficult to detect with the Z-normalized distance. Trend suppression algorithms can help to diminish these phenomena. However, current state-of-the-art algorithms do not achieve perfect results, are time-consuming, or require fine-tuning [7], [22]. Also, the longer the subsequence, the more pronounced this phenomenon is.

Contributions. In this paper, we introduce a novel normalization procedure, denoted LT-normalization (LT refers to Linear Trend). Combined with the Euclidean distance, it yields the LT-normalized Euclidean distance, invariant to linear trends, vertical offsets, and amplitude shifts. This normalization can be integrated without any computational overhead into state-of-the-art algorithms for motif discovery and anomaly detection in long time series thanks to careful implementation. For instance, similarly to the Z-normalization, the matrix profile with LT-normalized Euclidean distance has complexity $\mathcal{O}(l^2)$. Our normalization scheme is easily extendable to the multivariate case. This improvement is thus cost-free in terms of complexity. We show that adding this extra step can improve the performance of the matrix profile in some use cases with simulated and real-world data.

Structure of the paper. The first section defines the matrix profile and provides an overview of related research. Then we present the LT-normalization and some of its properties; in particular, we describe its fast computation. The last section contains the experimental evaluation that demonstrates the effectiveness of the proposed approach for the motif pair, similarity search, and motif set problems.

II. Background

This section presents some fundamental definitions and an overview of fast normalization algorithms for subsequence similarity search. For ease of comprehension, all definitions are given in the univariate case. They can be straightforwardly extended to the multivariate setting.

A. Definitions

We will follow the formalism introduced in papers [10], [29], [31] for consistency with previous work.

Definition 1 (Time series)**.** *A time series is an ordered sequence $S = [s_1, \dots, s_l]$ of length l of real-valued coefficients ($s_i \in \mathbb{R}$).*

Definition 2 (Subsequence)**.** *The subsequence of a time series $S \in \mathbb{R}^l$ of length w starting at index $i \in [1, \dots, l - w + 1]$ is the sequence : $S_i^w = [s_i, \dots, s_{i+w-1}]$*

Definition 3 (Overlapping subsequences)**.** *Two subsequences (S_i^w, S_j^w) of a time series $S \in \mathbb{R}^l$ overlap if $|i - j| < w$.*

In the following, we fix a time series $S \in \mathbb{R}^l$, a window length $w > 0$, and a distance function $d : \mathbb{R}^w \times \mathbb{R}^w \mapsto \mathbb{R}_+$. To ease notation w is not mentioned $S_i^w = S_i$

Definition 4 (Distance Profile)**.** *The distance profile between S_i and S is the real valued sequence $D_i = [d(S_i, S_j)]_{j=1,\dots,l-w+1}$*

Definition 5 (Matrix Profile)**.** *The matrix profile P of S is the sequence of distance to the nearest non-overlapping subsequence. Formally, $P = [\overline{\min}(D_i)]_{i=1,\dots,l-w+1}$ where $\overline{\min}(D_i)$ is the distance to the nearest non-overlapping subsequence of S_i.*

Definition 6 (Index Profile)**.** *The index profile of S is the integer valued sequence $IDX = [\arg\overline{\min}(D_i)]_{i=1,\dots,l-w+1}$*

Thus, the pair (P, IDX) provides the location and the distance to the nearest non-overlapping neighbour of each subsequence according to the distance d.

Definition 7 (Z-normalized Euclidean distance)**.** *The Z-normalized distance between $x \in \mathbb{R}^w$ and $y \in \mathbb{R}^w$ is :*

$$d_Z(x, y) = \left\| \frac{x - \mu_x \mathbf{1}}{\sigma_x} - \frac{y - \mu_y \mathbf{1}}{\sigma_y} \right\|$$

where $\|.\|$ is the norm associated to the Euclidean inner product $\langle ., . \rangle$, $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^w$ the unit vector, $\mu_x = w^{-1} \sum_{i=1}^{w} x_i$ the empirical mean of x and $\sigma_x^2 = w^{-1}\langle x, x \rangle - \mu_x^2$ the empirical variance of x.

B. Fast computation for Z-normalization

Here we only focus on the Z-normalization scheme. More procedures can be found in [16], although they are given in the classification/clustering context. It is not obvious if any can be extended for the task of subsequence similarity search, meaning that there is an efficient implementation as the one we describe now.

While applicable to any distance metric, the matrix profile is frequently utilized in conjunction with Z-normalized distance [25]. This preference stems from the computational complexity of the brute-force method, which computes the matrix

profile in $\mathcal{O}(Cl^2)$ time, with C representing the computational time for distance calculation between two subsequences. To address this challenge, current algorithms leverage the insight that efficient computation of the matrix profile is achievable for certain distances and normalization techniques (at the expanse of storing additional data for each subsequence). The Z-normalized distance is one of them, as stated in the following proposition.

Proposition 1. *The Z-normalized distance between S_i and S_j can be written as:*

$$d_Z(S_i, S_j) = \sqrt{2\Big(w - \frac{I_{i,j} - w\mu_i\mu_j}{\sigma_i\sigma_j}\Big)}$$

where $I_{i,j} = \langle S_i, S_j \rangle = \sum_{k=0}^{w-1} s_{i+k}s_{j+k}$ is the inner product between S_i and S_j.

Proof. See [28]. □

The algorithms STAMP [29], STOMP [31] and SCRIMP++ [30] follow this idea and they compute efficiently the matrix profile for the Z-normalized distance. According to proposition 1, the Z-normalized distance is computed with the mean and the variance of both subsequences and their inner product. Means and variances of all subsequences are computed in advance in $\mathcal{O}(l)$ [17]; they represent the additional data. The algorithms' time complexity resides in their ability to compute the inner products. For the three algorithms mentioned above, the resolution scheme consists of computing the distance profile for each subsequence successively. STAMP [29] computes the inner products between a subsequence and all the others with the Fast Fourier Transform (FFT). Its time complexity is $\mathcal{O}(l^2 \log(l))$. In addition to means and variances, STOMP [31] stores the inner products of the first subsequence with all the others. Then the distance profiles are computed successively thanks to the recursive property on the inner product. Its time complexity is in $\mathcal{O}(l^2)$. Both STAMP and STOMP are offline algorithms, SCRIMP++ is an anytime algorithm in $\mathcal{O}(l^2)$ that benefits from approaches of both offline methods.

These three algorithms are the foundational work around the matrix profile and several variations have been proposed depending on the context. For instance, VALMOD [10] computes the matrix profile for a range of window lengths. mSTAMP [28] computes the matrix profile of multidimensional time series.

C. Fast computation for other normalizations

While various algorithms have been proposed for the Z-normalized distance, only some are concerned with other distances. An algorithm is interested in a variation of the elastic distance DTW [21]. It focuses on finding patterns whose length may vary over time. Lastly, in [5], they study the limitation of the Z-normalized distance in noisy time series and propose a variation of the Z-normalized distance corrected by the noise variance. They present better empirical results for the task of anomaly detection. Nevertheless, no distance has been proposed for time series with trend.

III. LT-NORMALIZATION

This section first recalls some elementary transformations of time series, introduces the LT normalization, its theoretical properties, the algorithmic procedure for fast computation in a subsequence similarity search setting and the extension to the multivariate case.

A. Transformations

In some contexts, it is desirable to compare subsequences independently of certain transformations. In the following, we formalize four common transformations applicable to subsequences.

Definition 8 (Amplitude shift)**.** *A sequence $x \in \mathbb{R}^w$ whose amplitude is shifted by $\lambda > 0$ is the sequence $\lambda x = (\lambda x_1, \ldots, \lambda x_w)$.*

Definition 9 (Offset shift)**.** *A sequence $x \in \mathbb{R}^w$ whose offset is shifted by $b \in \mathbb{R}$ is the sequence $x + b\mathbf{1} = (x_1 + b, \ldots, x_w + b)$.*

Definition 10 (Linear shift)**.** *A sequence $x \in \mathbb{R}^w$ that is linearly shifted by $at + b\mathbf{1}$ is the sequence $x + (at + b\mathbf{1}) = (x_1 + b, \ldots, x_w + a(w-1) + b)$ where $a \in \mathbb{R}$, $t = (0, \ldots, w-1)$ and $b \in \mathbb{R}$.*

Definition 11 (Additive white Gaussian noise)**.** *A noisy version of the sequence $x \in \mathbb{R}^w$ is the sequence $x + \epsilon = (x_1 + \epsilon_1, \ldots, x_w + \epsilon_w)$ where ϵ_i are $i.i.d$ and $\epsilon_1 \sim \mathcal{N}(0, \sigma^2)$ with $\sigma > 0$.*

Note that the amplitude, offset and linear shifts can occur simultaneously in real-world settings.

B. Definition of the LT-normalization

In this subsection, we introduce the LT-normalization (LT stands for Linear Trend), which is robust to linear, offset, and amplitude shifts. We first only consider the univariate case: the extension to the multivariate case will be described in Section III-E.

Definition 12 (LT-normalized distance)**.** *The LT-normalized distance between sequences $x \in \mathbb{R}^w$ and $y \in \mathbb{R}^w$ is:*

$$d_{LT}(x, y) = \left\| \frac{x - (\alpha_x t + \beta_x \mathbf{1})}{\|x - (\alpha_x t + \beta_x \mathbf{1})\|} - \frac{y - (\alpha_y t + \beta_y \mathbf{1})}{\|y - (\alpha_y t + \beta_y \mathbf{1})\|} \right\|$$

where $t = (0, \ldots, w-1)$ and (α_x, β_x) are solutions of the linear regression problem:

$$\underset{(a,b)\in\mathbb{R}^2}{\arg\min} \|x - (at + b\mathbf{1})\|^2 \tag{1}$$

Remark 1. *The linear regression problem (1) has an explicit solution:*

$$\begin{cases} \alpha_x = \mathrm{cov}(x, t)/\sigma_t^2 \\ \beta_x = \mu_x - \alpha_x \mu_t \end{cases} \tag{2}$$

where $\mathrm{cov}(x, t) = \frac{1}{w}\langle x, t \rangle - \mu_x\mu_t$, $\mu_t = \frac{w-1}{2}$ and $\sigma_t^2 = \frac{w^2-1}{12}$

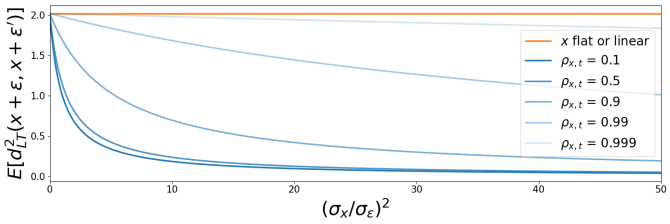

Fig. 2. Expected LT-normalized distance value between two noisy version of a sequence x as a function of the ration between the sequence variance and the noise variance for different value of the Pearson correlation between x and t.

C. Properties of the LT-normalization

According to Definition 12, the LT-normalization removes the linear trends of both sequences and sets the norm of the detrended sequences to the unit norm. Thanks to these operations, the LT-normalized distance is invariant to linear, offset, and amplitude shifts as shown by the following proposition.

Proposition 2. *The LT-normalized distance is invariant to linear, offset, and amplitude shifts. For any* $(x, y) \in \mathbb{R}^w \times \mathbb{R}^w$, $\lambda_x, \lambda_y > 0$, $(a_x, a_y) \in \mathbb{R} \times \mathbb{R}$ *and* $(\beta_x, \beta_y) \in \mathbb{R} \times \mathbb{R}$*:*

$$d_{LT}(\lambda_x(x + a_x t + \beta_x \mathbf{1}), \lambda_y(y + a_y t + \beta_y \mathbf{1})) = d_{LT}(x, y)$$

Proof. Application of Definition 12 to the shifted sequences. □

In the following proposition, we study the influence of Gaussian noise on the LT-normalization.

Proposition 3. *Suppose a sequence* $x \in \mathbb{R}^w$ *and two Gaussian vectors* $\epsilon, \epsilon' \sim \mathcal{N}(0, \sigma_\epsilon^2 I_w)$ *which represent noise. If* σ_x, μ_x, α_x *and* σ_ϵ *are known, then :*

$$\mathbb{E}[d_{LT}^2(x + \epsilon, x + \epsilon')] = \frac{2w}{w-1}\left(1 + \left(\frac{\sigma_x}{\sigma_\epsilon}\right)^2 (1 - \rho_{x,t}^2)\right)^{-1}$$

where $\rho_{x,t} = \mathrm{cov}(x,t)/(\sigma_x \sigma_t)$ *is the Pearson correlation between* x *and* t*.*

Proof. We assume that we work with an unbiased estimator of the variance, thus, $\alpha_x \sigma_t = \rho_{x,t} \sigma_x$. Noticing that $\beta_x = \mu_x - \alpha_x \mu_t$, we have: $\eta_x^2 = \|x - (\alpha_x t + \beta_x \mathbf{1})\|^2 = w(\sigma_x^2 - \alpha_x^2 \sigma_t^2)$, and thus:

$$\eta_{x+\epsilon}^2 = \eta_{x+\epsilon'}^2 = \frac{w-1}{w}\left(\sigma_x^2 + \sigma_\epsilon^2 - \rho_{x,t}^2 \sigma_x^2\right)$$

As well, $(2\sigma_\epsilon^2)(\sum_{i=1}^w (\epsilon_i - \epsilon_i')^2) \sim \chi^2(w)$. Then, by linearity of the mean and independence between x and ϵ the proposition 3 is verified. □

The LT-normalized distance between two noisy versions of a sequence depends on the signal-to-noise ratio $\frac{\sigma_x}{\sigma_\epsilon}$. However, it also depends on the Pearson correlation between x and t. From Proposition 3, the LT-normalized distance is maximal, thus does not detect matches when sequences are close to being flat ($\sigma_x^2/\sigma_\epsilon^2 \to 0$) or linear ($\rho_{x,t}^2 \to 1$). Figure 2, illustrates this behaviour; it expresses the expected LT-normalized distance as a function of the ratio of variances for different values of the Pearson correlation between x and t. The LT-normalized distance has the expected behaviours. In the case of a noisy time series with a trend, the LT-normalized distance between two almost flat or linear subsequences is high. Consequently, when the trend is assumed to be locally linear, the LT-normalized distance between two subsequences of pure trend is high. On the other hand, the distance between two noisy versions of a sequence that is not flat or linear is low regardless of any linear, offset, and amplitude shifts. Therefore, the distance between two motif occurrences remains low even in the presence of a trend. Where the Z-normalized distance cannot distinguish motifs from trends, the LT-normalized distance can differentiate them.

Algorithm 1 ComputeInnerProduct

Require: Q a query sequence, S a time series
  $w \leftarrow Lenght(Q)$, $l \leftarrow Length(S)$
  $S_a \leftarrow append\ S\ with\ w\ zeros$
  $Q_r \leftarrow reverse\ Q$
  $Q_a \leftarrow append\ Q\ with\ l\ zeros$
  $Q_{af} \leftarrow FFT(Q_a)$, $S_{af} \leftarrow FFT(S_a)$
  $I \leftarrow iFFT(ElementwiseProduct(Q_{af}, S_{af}))$
  return $I[w : l]$

Algorithm 2 ComputeCoefficients

Require: S a time series, w a window length
  $\mu, \sigma \leftarrow ComputeMeanStd(S, w)$ ▷ See [17]
  $T \leftarrow [1, \ldots, w]$
  $IT \leftarrow ComputeInnerProduct(T, S)$
  $\alpha \leftarrow ComputeSlople(IT, \mu)$ ▷ See eq: (2)
  $\eta \leftarrow ComputeNorm(\alpha, \mu, \sigma)$ ▷ See eq: (4)
  return μ, α, η

D. Fast computation of the LT-normalization with application to subsequence similarity search

Proposition 4 (LT-normalized matrix profile time complexity)**.** *The matrix profile combined with the LT-normalized distance can be computed in* $\mathcal{O}(l^2)$ *for a time series* $S \in \mathbb{R}^l$ *and any subsequence length* $w > 0$*.*

Proof. Thanks to an adaptation of STOMP algorithm [28], the matrix-profile combined with the LT-normalized distance can be computed in $\mathcal{O}(l^2)$ for a time series $S \in \mathbb{R}^l$ and any window length $w > 0$. Indeed, the STOMP algorithm computes the matrix profile with the Z-normalized distance in $\mathcal{O}(l^2)$. Its efficiency comes from a recursive formulation of the Z-normalized distance which allows the computation of any subsequence distance profile in $\mathcal{O}(l)$. The recursive formulation relies on additional data that are computed in a preprocessing step in $\mathcal{O}(l \log(l))$.

In what follows, we show that the LT-normalized distance also has a recursive formulation, which relies on additional data computable in $\mathcal{O}(l \log(l))$ during a preprocessing step. Following the framework of the STOMP algorithm, the matrix profile computation is in $\mathcal{O}(l^2)$.

Algorithm 3 ComputeMatrixProfile

Require: S a time series, w a window length
  $l \leftarrow Lenght(S)$
  μ, α, $\eta \leftarrow ComputeCoefficients(S, w)$
  $I \leftarrow ComputeInnerProduct(S[1:w], S)$, $I_{init} \leftarrow I$
  P, $IDX \leftarrow InitializeMatrixProfile(I, \mu, \alpha, \eta, w)$
  for $i = 2$ **to** $l - w + 1$ **do**
    for $j = l - w$ **downto** 1 **do**
      $I[j+1] = I[j] + S[i+w] \cdot S[j+w] - S[j] \cdot S[i]$
    end for
    $I[1] \leftarrow I_{init}[i]$
    $D \leftarrow ComputeDistanceProfile(I, \mu, \alpha, \eta, w)$ ▷ See eq: (3)
    $P[i]$, $IDX[i] \leftarrow FindNonOverlappingMinimum(D, i)$
  end for
  return P, IDX

Lemma 1. *The LT-normalized distance between the subsequences S_i and S_j can be expressed as:*

$$d_{LT}(S_i, S_j) = \sqrt{2\Big(1 - \frac{I_{i,j} - w(\mu_i\mu_j + \alpha_i\alpha_j\sigma_t^2)}{\eta_i\eta_j}\Big)} \quad (3)$$

where $I_{i,j} = \sum_{k=0}^{w-1} s_{i+k}s_{j+k}$ is the inner product between S_i and S_j, μ_i and σ_i are the mean and the variance of S_i, α_i is the trend estimator of S_i and:

$$\eta_i = \|S_i - (\alpha_i t + \beta_i \mathbf{1})\| = \sqrt{w(\sigma_i^2 - \alpha_i^2\sigma_t^2)} \quad (4)$$

is the norm of S_i without linear trend.

Proof. It suffices to show that for $x \in \mathbb{R}^w$ and $y \in \mathbb{R}^w$:

$$d_{LT}(x, y) = \sqrt{2\Big(1 - \frac{\langle x, y\rangle - w(\mu_x\mu_y + \alpha_x\alpha_y\sigma_t^2)}{\eta_x\eta_y}\Big)}$$

where $\eta_x = \|x - (\alpha_x t + \beta_x \mathbf{1})\|$.

Noticing that $\beta_x = \mu_x - \alpha_x\mu_t$, we have: $\eta_x^2 = w(\sigma_x^2 - \alpha_x^2\sigma_t^2)$, and: $\langle x - (\alpha_x t + \beta_x\mathbf{1}), y - (\alpha_y t + \beta_y\mathbf{1})\rangle = \langle x, y\rangle - w(\mu_x\mu_y + \alpha_x\alpha_y\sigma_t^2)$. Then, with: $\|x/\|x\| - y/\|y\|\|^2 = 2\,(1 - \langle x, y\rangle/\|x\|\|y\|)$, Lemma 1 is verified. □

As for the matrix profile combined with the Z-normalized distance, the recursion occurs on the inner product. The following proposition exhibits the recursion.

Lemma 2 (Inner product recursion)**.** *Knowing the inner product $I_{i,j}$ between the subsequence S_i and S_j, the inner product between S_{i+1} and S_{j+1} can be computed in $\mathcal{O}(1)$ with the recursion:*

$$I_{i+1,j+1} = I_{i,j} + s_{i+w}s_{j+w} - s_i s_j$$

According to Equation 3, we store the additional data: $(\mu, \sigma, \alpha, \eta) = (\mu_i, \sigma_i, \alpha_i, \eta_i)_{i=1,\dots,l-w+1}$. μ and σ are computable in $\mathcal{O}(l)$ with the procedure presented in [17]. From Equation 2, it is known that for any i, $a_i = w^{-1}\langle S_i, t\rangle - \mu_i\mu_t$. It suffices to compute the inner product between t and all subsequences to get α. Thanks to Algorithm 1, which was first introduced in [29], these inner products can be computed in $\mathcal{O}(l\log(l))$. Finally, from Equation 4, η can be computed in $\mathcal{O}(l)$ with σ and α. All additional data can be computed in $\mathcal{O}(2l\log(l))$ and the Algorithm 2 summarizes their computation. Thanks to the additional data $(\mu, \sigma, \alpha, \eta)$ and the inner product recursive property (Lemma 2), each distance profile can be computed in $\mathcal{O}(l)$. □

We summarize the adaptation of STOMP algorithm for the LT-normalized distance in Algorithm 3.

Remark 2. *The anytime algorithm, SCRIMP++ [30], and parallel computation with GPUs [28] could also be adapted to the LT-normalized distance. The only difference with original algorithms [28]–[30] resides in the computation of the additional data α and η.*

E. Extension to multivariate time series

The extension of LT-normalization to the multivariate setting is straightforward and given in the following definition.

Definition 13. *The multivariate LT-normalized distance between $x \in \mathbb{R}^{d\times w}$ and $y \in \mathbb{R}^{d\times w}$ is*

$$d_{MLT}(x, y) = \sqrt{\frac{1}{d}\sum_{k=1}^{d} d_{LT}^2(x^{(k)}, y^{(k)})}$$

where $x^{(k)}$ is the k^{th} dimension of the signal.

IV. Experimental Evaluation

We evaluated the performance of the LT-normalized distance on three data mining tasks for time series:

- **Motif Pair Discovery:** Identifying the two most similar non-overlapping subsequences in a time series.
- **Similarity search:** Identifying all non-overlapping subsequences in a time series that are similar to a query subsequence.
- **Motif Set Discovery:** Identifying sets of subsequences encompassing every occurrence of distinct repeated patterns in a time series.

For reproducibility, the source code and all datasets are available on our webpage [1].

In what follows, we present the datasets and the experimental results for each task. In the last section, we evaluate the scalability with the time series and the subsequence length of the STOMP algorithm with the LT-normalized distance.

A. Datasets

We conducted our experimental evaluation on several labeled datasets constructed from real and synthetic time series. The datasets are described in more detail in the following paragraphs.

1) Real-world data:

(R-1) **mitdb-1, (2 dimensions)** [6], [13]: The MIT-BIH Arrhythmia Database contains 48 half-hour recordings of two-channel ambulatory electrocardiograms (ECGs) sampled at $360Hz$. Cardiologists annotated the heartbeats according to 19 categories[1]. We divided all recordings into time series of 1 minute. We selected time series of healthy patients (id: 100, 101, 103, 117, 122, according to [18]) that contain only normal heartbeats and randomly selected 100 time series.

(R-2) **mitdb-2, (2 dimensions)**: We randomly selected 100 1-minute ECGs from MIT-BIH. The number of repeated patterns varied between 1 and 4.

(R-3) **ptt-ppg, (7 dimensions)** [12]: Pule-Transit-Time PPG dataset contains recordings from 22 healthy subjects performing three physical activities: sit, walk, and run. Each time series includes multiple sensors: photoplethysmogram (PPG), inertial, pressure and ECG. All recordings were sampled at $500Hz$, the heartbeats were annotated by cardiologists from ECGs. We kept the *run* activity and the photoplethysmogram channels as well as the ECG channel. For all subjects, we divided the recordings into 40-seconds time series, and we randomly selected 100 time series. This results in a labelled dataset of 100 time series with a single repeated pattern.

(R-4) **arm-coda, (27 dimensions)** [4] is a dataset of 240 multivariate time series collected using 34 Cartesian Optoelectronic Dynamic Anthropometers (CODA) placed on the upper limbs of 16 healthy subjects, each of whom performed 15 predefined movements such as raising their arms or combing their hair. Each sensor records its position in 3D space. To construct the dataset, we kept sensors describing the upper body (28,17,10,21,16,8,0,5,11) and 12 of the predefined movements ((0,5,6,),(4,7,8),(9,11,12),(10,13,14)). We selected the first two occurrences of all movements. Then, by tuple of 3 movements, the occurrences of the movements were randomly placed along the time axis for each subject, sensor. The distance between two consecutive occurrences is sampled uniformly over $[50, 450]$. A Gaussian noise with a signal-to-noise ratio of 0.01 was added to all time series. This resulted in a dataset of 64 time series.

2) Synthetic data: We have generated one dataset per data mining task with the following scenarios:

(S-1) **m-pair, (1 dimension)**: There is 1 pattern of length 100 and with 1 dimension that repeats twice.

(S-2) **s-search, (5 dimensions)**: There is 1 pattern of length 100 and with 5 dimensions that repeats 50 times.

(S-3) **m-set, (5 dimensions)**: There are 5 patterns of length 100 and with 5 dimensions. For each pattern, the number of occurrences is sampled uniformly between 2 and 10.

All time series are generated using the same protocol: occurrences of the N repeated patterns are randomly placed on top of a random walk, and Gaussian noise is added to the resulting time series. In all scenarios, the amplitude of the Gaussian noise is set to 0.1, and given a fundamental frequency of $4Hz$, a pattern is generated as the sum of the sine function of the hundred first harmonics, with the phases and the amplitudes are uniformly sampled over $[-\pi, \pi]$ and $[-1, 1]$. For the motif pair dataset, we generated 200 time series for each random walk variance step between 0 and 0.5 by steps of 0.01, and the interval between the occurrences is uniformly sampled over $[100, 900]$. For the similarity search and the motif set datasets, we generated 100 time series such that the amplitude of the random walk is set to 0.2, and the interval between two consecutive occurrences is uniformly sampled over $[10, 90]$.

[1]https://archive.physionet.org/physiobank/annotations.shtml

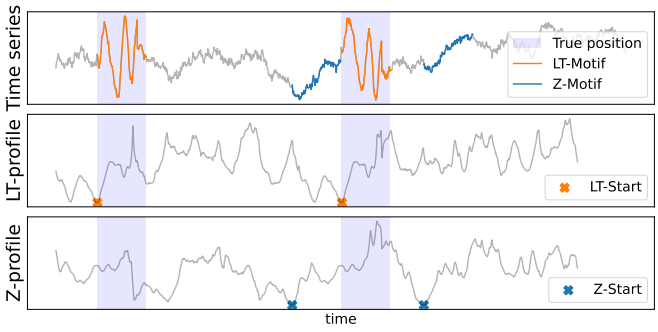

Fig. 3. **Top:** Synthetic time series with a trend and one motif that occurs twice. True motif locations are highlighted in light purple. The predicted best motif pair is colored in orange for the LT-normalized distance and blue for the Z-normalized distance. **Middle:** Matrix profile with the LT-normalized distance. The starting location of the predicted best motif pair is in orange. **Bottom:** Matrix profile with the Z-normalized distance. The starting location of the predicted best motif pair is in blue.

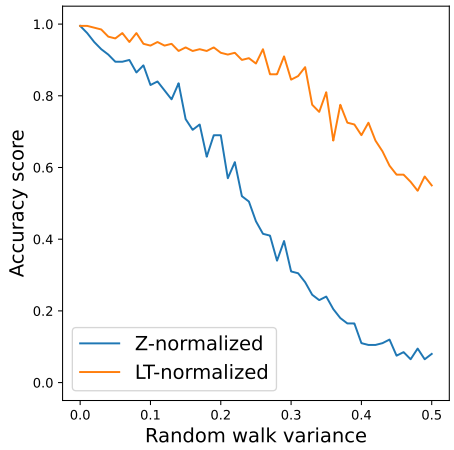

Fig. 4. Accuracy scores for the LT-normalized and Z-normalized distances as a function of the random walk variance.

B. Best Motif Pair

In this experiment, we investigated the influence of the trend on the performance of LT-normalized and Z-normalized distances for solving the Best Motif Pair problem. This problem [10] consists of finding the pair of non-overlapping subsequences whose distance is minimal compared to all other non-overlapping subsequence pairs. The matrix profile provides an exact solution to this problem, which

corresponds to the indices associated with its minimum $(\arg\min(P), IDX_{\arg\min(P)})$. We used this resolution scheme to compare the performance of the distances. Figure 3 illustrates the best motif pair problem and its resolution with the matrix profile. The top figure shows a time series of the m-pair dataset (S-1). The next figures show the LT-normalized and Z-normalized matrix profiles with the predicted best motif pair locations. The true motif pair was recovered with the LT-normalized distance, while the Z-normalized distance identified a pair of nearly linear subsequences of the trend.

To evaluate the influence of the trend on the best motif pair prediction, we considered the m-pair dataset (S-1), where time series have been generated for different values of the random walk variance. This parameter controls the trend's regularity: it decreases as the variance increases. To measure the performance, we compute an event-based accuracy score. A best motif pair prediction is counted as a true positive if, for each subsequence, the predicted location overlaps the real location by at least 50%. Figure 4 shows the accuracy scores of both distances as a function of the variance of the random walk.

When the variance of the random walk is zero, there is no trend; by construction, the signal-to-noise ratio is, on average, equal to 22 dB. In this case, it is expected that the best motif pairs are most likely well predicted with both distances; indeed, both empirical scores are equal to one. However, the empirical results show that as soon as the variance of the random walk increases, the Z-normalized accuracy score decreases. On the other hand, Proposition 3 suggests that the best motif pair remains detectable with the LT-normalized distance when it can be assumed that the trend can be locally approximated with a linear sequence. The empirical results are congruent with this observation: the accuracy score remains consistently high for a low random walk variance (between 0 and 0.2), and then the score decreases as the regularity of the trend decreases.

Thanks to its linear shift invariance, the detection of the best motif pairs with the LT-normalized distance is more robust to the deformations induced by the trend.

C. Similarity search

In this experiment, we evaluated the performance of LT-normalized and Z-normalized distances in solving the similarity search problem.

The similarity search problem [9] consists in identifying all occurrences of a query sequence in a time series. A classical approach [14] first computes the Z-normalized distance profile between a query sequence and a time series. From the distance profile, the starting locations of the occurrences are identified with local minima below a given threshold. This approach can be extended to the LT-normalized distance, and we used this resolution scheme to evaluate the performance of the two distances.

We performed our experiment on datasets where the time series have one pattern that repeats multiple times: s-search (S-2), mitdb-1 (R-1), and ptt-ppg (R-3). Figure 5 illustrates the resolution of the similarity search problem on an ECG (A) and a PPG (B). In both cases, the top right plot shows the query sequence corresponding to the repeated pattern's first occurrence. The top right plot shows the raw signal, and the plots below show respectively the LT-normalized and Z-normalized distance profiles. For both time series, the distance profiles are minimal at the starting locations of occurrences of the query sequences. However, the Z-normalized distance profile is sensitive to the trend, and the distance remains high for some occurrences. On the contrary, the trend less affects the LT-normalised distance profile, and the distance remains consistently low at the starting location of occurrences. The LT-normalized distance is better suited for the similarity search on these two time series.

TABLE I
F1-SCORES ON THE MOTIF SET DISCOVERY TASK. EUCLIDEAN (EUC), Z-NORMALIZED (Z), LT-NORMALIZED (LT), TREND REMOVAL & Z-NORMALIZED (STL+Z).

distance dataset	Euc	STL+Z	Z	LT
s-search (S-2)	0.20	0.86	**0.87**	0.86
m-set (S-3)	0.25	0.62	0.62	**0.62**
mitdb1 (R-1)	0.42	0.54	0.50	**0.58**
mitdb2 (R-2)	0.16	0.44	0.43	**0.45**
ptt-ppg (R-3)	0.54	**0.58**	0.53	0.57
arm-coda (R-4)	0.25	**0.26**	0.25	0.25

We computed ROC curves for each distance and dataset according to the procedure described in [16]. We counted a predicted occurrence as valid if it overlapped with a real occurrence by at least 75%. The results are shown in Figure 6. On average, the LT-normalized distance outperformed the Z-normalized distance as it had a higher AUC score across all datasets. It is also worth noticing that the ROC curves of the LT-normalized distance are consistently above those of the Z-normalized distance. Indeed, the LT-normalized distance is a generalization of the Z-normalized distance to a broader class of deformations. As a result, the LT-normalized distance profiles are more robust to the deformation induced by the trend. Therefore, the number of true occurrences detected with the LT-normalized distance is at least as good as that of the Z-normalized distance.

D. Motif set discovery

In this experiment, we evaluated the performance of LT-normalized and Z-normalized distances in solving the motif set discovery problem.

The motif set discovery problem [11] consists in identifying all occurrences of each repeated pattern present in a time series. A heuristic based on the matrix profile has been proposed to solve this problem [2], [31]. This algorithm can be extended to the LT-normalized distance, and we used it to evaluate the performance of both distances. We also added two baselines, a matrix profile with the Euclidean distance and a second matrix profile with the Z-normalized distance where time-series are preprocessed using a trend removal algorithm: A Seasonal-Trend Decomposition Procedure Based on LOESS

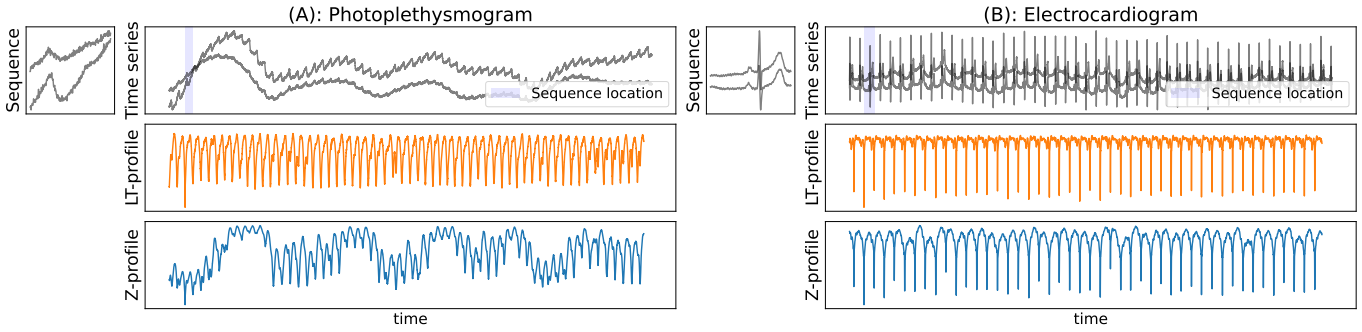

Fig. 5. Similarity search on: (A) photoplethysmogram, and (B) electrocardiogram. In both cases, top left: the query subsequence, top right: the time series with the query subsequence location in blue, middle: LT-normalized distance profile, bottom: Z-normalized distance profile. Due to the trend, some occurrences of the query subsequences are missed with the Z-normalized distance profile while they are all identifiable with the LT-normalized distance profile.

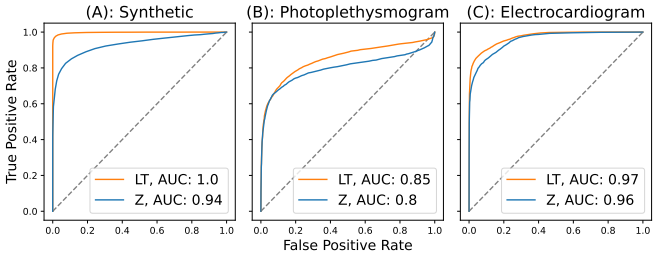

Fig. 6. ROC curves of the similarity search problem for LT-normalized (orange) and Z-normalized (blue) distances on the datasets: (A) s-search, (B) ptt-ppg, and (C) mitdb-1. The LT-normalized distance performs better than the Z-normalized distance.

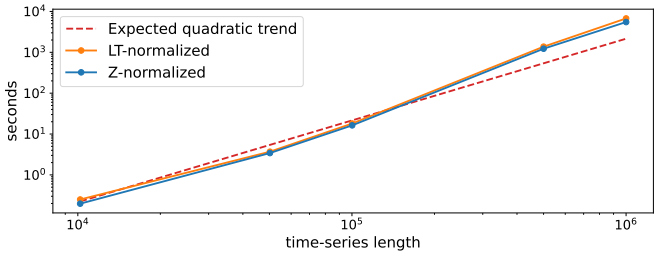

Fig. 7. Scalability of the matrix profile with the time series length for LT-normalized (blue) and Z-normalized (orange) distances.

(STL) [3]. In terms of settings, the algorithm requires the number of sets to discover, which we assumed to be known, a subsequence similarity ratio, which we set to 3, and a subsequence length, which we set to be the average motif length for each dataset. The STL algorithm period is also set to the average motif length.

We ran the experiment on all datasets except the m-pair, as its time series contains a single motif that repeats twice. We evaluated the performance with the event-based f1-score [23]. We counted a pair of predicted/real occurrences as valid for the precision (resp. recall) metric if the length of their intersection is greater than 50% of the length of the predicted (resp. real) occurrence. We used the Hungarian matching algorithm [8], [19] to match the predicted motif sets and occurrences with the real ones.

Experimental results are shown in Table I. Compared to the Z-normalized distance, the motif set algorithm performs better or equally using the LT-normalized distance except on the s-search dataset. Often, LT and STL+Z perform similarly, meaning that removing a linear trend is useful, and both methods succeed at doing this. However, we will see in the scalability experiment that STL+Z is more computationally burdensome than LT.

E. Scalability

In this experiment, we evaluated the scalability of the matrix profile with respect to the time series length for the LT-normalized and Z-normalized distances. We considered the STOMP algorithm [31] to compute the matrix profile with both distances. We generated 50 time series based on the m-set scenario (S-3) with lengths of 10K, 50K, 100K, 500K, and 1M. We measured the computation time for a subsequence length of 100. The average computation time is shown in Figure 7. Even though the LT-normalized distance generalizes the Z-normalized distance and performs better on several tasks, the matrix profile's computation time is equivalent for both distances and evolves according to its quadratic complexity. STL+Z (not shown on the plot) takes around 1 minute to process 100K samples, compared to a dozen seconds for LT and Z.

V. Conclusion

We have introduced the LT-normalization, a generalization of the Z-normalisation that is robust to linear, offset, and amplitude shifts. Combined with this normalization, the matrix profile can be computed in quadratic time with only a slight modification of the state-of-the-art algorithms: STOMP or SCRIMP++. Empirical results show competitive results on several data sets for the best motif pair, the similarity search and the motif set discovery problems.

VI. Acknowledgments

This work was supported by grants from Région Ile-de-France (DIM MathInnov). Charles Truong is funded by the PhLAMES chair of ENS Paris-Saclay.

References

[1] https://github.com/thibaut-germain/lt-normalized.

[2] Andrew Van Benschoten, Austin Ouyang, Francisco Bischoff, and Tyler Marrs. Mpa: a novel cross-language api for time series analysis. *Journal of Open Source Software*, 5(49):2179, 2020.

[3] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *J. Off. Stat*, 6(1):3–73, 1990.

[4] Sylvain W Combettes, Paul Boniol, Antoine Mazarguil, Danping Wang, Diego Vaquero-Ramos, Marion Chauveau, Laurent Oudre, Nicolas Vayatis, Pierre-Paul Vidal, Alexandra Roren, and Marie-Martine Lefèvre-Colau. Arm-coda: A dataset of upper-limb human movement during routine examination. https://www.ipol.im/pub/pre/494/.

[5] Dieter De Paepe, Diego Nieves Avendano, and Sofie Van Hoecke. Implications of z-normalization in the matrix profile. In *International Conference on Pattern Recognition Applications and Methods*, pages 95–118. Springer, 2019.

[6] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[7] Michael Hippke, Trevor J David, Gijs D Mulders, and René Heller. Wōtan: Comprehensive time-series detrending in python. *The Astronomical Journal*, 158(4):143, 2019.

[8] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[9] Rake& Agrawal King-lp Lin and HSSK Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceeding of the 21th International Conference on Very Large Data Bases*, pages 490–501. Citeseer, 1995.

[10] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. Matrix profile x: Valmod-scalable discovery of variable-length motifs in data series. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1053–1066, 2018.

[11] JLEKS Lonardi and Pranav Patel. Finding motifs in time series. In *Proc. of the 2nd Workshop on Temporal Data Mining*, pages 53–68, 2002.

[12] Philip Mehrgardt, Matloob Khushi, Simon Poon, and Anusha Withana. Pulse transit time ppg dataset. *PhysioNet*, 10:e215–e220, 2022.

[13] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3):45–50, 2001.

[14] Abdullah Mueen, Sheng Zhing, Yan Zhu, Michael Yeh, Kaveh Kamgar, Krishnamurthy Viswanathan, Chetan Gupta, and Eamonn Keogh. The fastest similarity search algorithm for time series subsequences under euclidean distance, August 2022. http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html.

[15] Takaaki Nakamura, Makoto Imamura, Ryan Mercer, and Eamonn Keogh. Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives. In *2020 IEEE international conference on data mining (ICDM)*, pages 1190–1195. IEEE, 2020.

[16] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron Elmore, and Michael J Franklin. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(11):2774–2787, 2022.

[17] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):1–31, 2013.

[18] Lucie Saclova, Andrea Nemcova, Radovan Smisek, Lukas Smital, Martin Vitek, and Marina Ronzhina. Reliable p wave detection in pathological ecg signals. *Scientific Reports*, 12(1):6589, 2022.

[19] Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11225–11234, 2021.

[20] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.

[21] Diego Furtado Silva and Gustavo EAPA Batista. Elastic time series motifs and discords. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 237–242. IEEE, 2018.

[22] Amit Singhal, Pushpendra Singh, Binish Fatimah, and Ram Bilas Pachori. An efficient removal of power-line interference and baseline wander from ecg signals by employing fourier decomposition technique. *Biomedical Signal Processing and Control*, 57:101741, 2020.

[23] Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. Precision and recall for time series. *Advances in neural information processing systems*, 31, 2018.

[24] Sahar Torkamani and Volker Lohweg. Survey on time series motif discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(2):e1199, 2017.

[25] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26:275–309, 2013.

[26] Rutuja Wankhedkar and Sanjay Kumar Jain. Motif discovery and anomaly detection in an ecg using matrix profile. In *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2019, Volume 1*, pages 88–95. Springer, 2021.

[27] Dazhi Yang and Dennis van der Meer. Post-processing in solar forecasting: Ten overarching thinking tools. *Renewable and Sustainable Energy Reviews*, 140:110735, 2021.

[28] Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamonn Keogh. Matrix profile vi: Meaningful multidimensional motif discovery. In *2017 IEEE international conference on data mining (ICDM)*, pages 565–574. IEEE, 2017.

[29] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 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)*, pages 1317–1322. Ieee, 2016.

[30] Yan Zhu, Chin-Chia Michael Yeh, Zachary Zimmerman, Kaveh Kamgar, and Eamonn Keogh. Matrix profile xi: Scrimp++: time series motif discovery at interactive speeds. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 837–846. IEEE, 2018.

[31] Yan Zhu, Zachary Zimmerman, Nader Shakibay Senobari, Chin-Chia Michael Yeh, Gareth Funning, Abdullah Mueen, Philip Brisk, and Eamonn Keogh. Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 739–748. IEEE, 2016.