mSIMPAD: Efficient and Robust Mining of Successive Similar Patterns of Multiple Lengths in Time Series

CHUN-TUNG LI and JIANNONG CAO, The Hong Kong Polytechnic University
XUE LIU, McGill University
MILOS STOJMENOVIC, Singidunum University

A successive similar pattern (SSP) is a series of similar sequences that occur consecutively at non-regular intervals in time series. Mining SSPs could provide valuable information without *a priori* knowledge, which is crucial in many applications ranging from health monitoring to activity recognition. However, most existing work is computationally expensive, focuses only on periodic patterns occurring in regular time intervals, and is unable to recognize patterns containing multiple periods. Here we investigate a more general problem of finding similar patterns occurring successively, in which the similarity between patterns is measured by the *z*-normalized Euclidean distance. We propose a linear time, robust method, called *Multiple-length Successive sIMilar PAtterns Detector* (mSIMPAD), that mines SSPs of multiple lengths, making no assumptions regarding periodicity. We apply our method on the detection of repetitive movement using a wearable inertial measurement unit. The experiments were conducted on three public datasets, two of which contain simple walking and idle data, whereas the third is more complex and contains multiple activities. mSIMPAD achieved F-score improvements of 3.2% and 6.5%, respectively, over the simple and complex datasets compared to the state-of-the-art walking detector. In addition, mSIMPAD is scalable and applicable to real-time applications since it operates in linear time complexity.

CCS Concepts: • **Mathematics of computing** → **Time series analysis**; • **Information systems** → *Nearest-neighbor search*; • **Human-centered computing** → *Ubiquitous and mobile computing*;

Additional Key Words and Phrases: Successive similar pattern, repetitive movement detection, periodicity detection, matrix profile

ACM Reference format:
Chun-Tung Li, Jiannong Cao, Xue Liu, and Milos Stojmenovic. 2020. mSIMPAD: Efficient and Robust Mining of Successive Similar Patterns of Multiple Lengths in Time Series. *ACM Trans. Comput. Healthcare* 1, 4, Article 23 (September 2020), 19 pages.
https://doi.org/10.1145/3396250

The presented work was supported by the National Key R&D Program of China with Project No. 2018YFB1004801. This work was partially supported by the Research Grants Council of Hong Kong under RGC No. C5026-18G and the Serbian Ministry of Science and Education, via TR32054, "Digital Signal Processing, Synthesis of an Information Security System."
Authors' addresses: C.-T. Li and J. Cao, The Hong Kong Polytechnic University, 11 Yuk Choi Road, Kowloon, Hong Kong, China; emails: chun-tung.li@connect.polyu.hk, csjcao@comp.polyu.edu.hk; X. Liu, McGill University, 845 Sherbrooke Street West, Montreal, Quebec, Canada; email: xueliu@cs.mcgill.ca; M. Stojmenovic, Singidunum University, Danijelova 32, Belgrade, Serbia; email: mstojmenovic@singidunum.ac.rs.

2637-8051/2020/09-ART23 $15.00
https://doi.org/10.1145/3396250

1 INTRODUCTION

Successive similar patterns (SSPs), or series of similar subsequences that occur successively, are prevalent in the physical world in areas such as seasonal weather, biosignals, and human behavior. Mining SSPs in time series means recognizing the appearance of successive recurring patterns, which is a challenging problem of great influence [34]. For example, repetitive physical motions characterize many interesting types of exercise including walking [4, 22], running [10], and free weight training [9, 28], which can be detected as the repeating patterns occurring in wearable sensor data. SSP detection is the enabling technology for exercise tracking using wearable devices that evaluate the activeness of an individual and can provide guidelines for daily activities to promote physical health. Despite exercise tracking, SSP detection can be applied to analyze heartbeat signals from electrocardiography (ECG) by searching unusual patterns within periodical signals for abnormal heartbeat detection [6]. Factory assembly work can be analyzed by estimating the lead time using SSPs detected from wearable sensor data [16]. In this work, we focus on the use case of repetitive movement detection—a problem that aims to identify the repeating physical motion of human activities—using wearable inertial measurement unit (IMU) data. Automatic repetitive movement detection is warranted, as it is the fundamental building block for human activity recognition. Specifically, representative patterns can be extracted efficiently from long-time series of each of the detected segments to facilitate human behavior studies and wearable healthcare applications [7].

Although numerous methods for repeating patterns finding have been proposed based on periodicity detection, most of them can only handle a single fixed period and fail to detect periodic patterns when disturbances appear or the patterns are misaligned [29, 35]. There are also patterns with multiple periods that may not be present all the time and their recurrence may be shifted. For instance, an athlete lifting weights may perform multiple successive repetitions, where the interval between each repetition may vary due to muscle fatigue. Different moves can produce different periods, and the shift of each repetition can constitute asynchronous periodic patterns. Existing methods typically require extensive domain knowledge to determine and learn many parameters [21, 28] and make assumptions to the target patterns such as the fixed periodicity [9, 22]. Therefore, it is desirable to have an SSP detection method that is parameter light and robust to unknown patterns with variations. Such a general approach can reduce the effort devoted to scenario-based repetitive movement detection, as it requires barely any domain knowledge.

SSP detection is related to, yet different from, periodic pattern mining and periodicity detection. In periodic pattern mining, the focus is on finding the pattern in symbol sequences that is fully or partially matched with other occurrences of the pattern [30]. Although Yang et al. [29] introduced an efficient method for asynchronous periodic patterns mining, it is not a straightforward process to convert real-valued time series to symbol sequences when prior knowledge is missing [13]. However, periodicity detection focuses on estimating the period of the recurring patterns [18]. The key difference of SSP is the relaxation of the periodicity assumption. It does not postulate a regular interval among the successive patterns, which is more flexible in covering the general set of repeating patterns in reality.

Mining multiple-length SSPs is not trivial for the following reasons. Unlike periodic patterns, the concept of SSP is ambiguous and difficult to define without prior knowledge of the target pattern. A high computational cost is associated with this problem due to the relaxation of the periodicity assumption. Searching for variable interval patterns between each repetition is intractable even for a small number of repetitions. Additionally, a time series may contain multiple lengths of repeating patterns, which makes it difficult to determine which length a pattern belongs to.

Here, we outline a novel, efficient, and robust matrix profile [33]-based algorithm that finds SSPs with multiple lengths in multi-dimensional real-valued time series. We first introduce a definition of SSP based on the concept of the Range-Constrained Matrix Profile (RCMP) and proposed the Range-Constrained Multi-Dimensional Scalable Time Series Ordered Matrix Profile (RC-mSTOMP) to compute the RCMP efficiently. We then present the

Successive sIMilar PAtterns Detector (SIMPAD) on the basis of the RCMP, which requires two inputs: the target pattern length l and the maximum displacement of pattern m. SIMPAD has barely any periodicity constraints, and the result can be computed and updated in an online fashion efficiently. We extended the proposed method and introduced the *Multiple -length Successive sIMilar PAtterns Detector* (mSIMPAD) for finding SSPs with multiple lengths within a time series. It provides an estimation of the pattern length and potentially assists in applications such as representation learning for pattern recognition.

Given their ubiquity and availability in smartphones and wearable devices [5, 12, 24], IMUs are the dominantly used source of data for physical activity assessment. It is a well-established area for evaluating the performance of the proposed method in real-world applications. Experiments were conducted on three public datasets, and we achieved promising results compared to the state-of-the-art (SOTA) repeating pattern–based walking detectors. It shows that the performance of the proposed method is insensitive to the input parameters. We also examined the robustness of our method on low-quality sensor data to evaluate its suitability to the emergence of battery-free, low sampling frequency, power consumption–optimized wearable devices. Additionally, we examined the empirical computational cost and demonstrated the linear relationships to the length and number of dimensions of the input time series. The code used in this study is freely available to all researchers and can be found at https://github.com/chuntungli/mSIMPAD. We summarize our contributions as follows:

- We formally define an SSP, which makes no assumption regarding the periodicity of the target pattern. On this basis, we introduced the RCMP, a modification of the matrix profile that is more efficient and superior in the case of SSP detection.
- We propose SIMPAD, a general SSP detection method based on the RCMP that is robust, efficient, and parameter light, which can facilitate various healthcare applications including exercise tracking and heartbeat monitoring. This method is then extended to capture SSPs with multiple lengths, which we call mSIMPAD.
- We evaluate the performance of SIMPAD and mSIMPAD on three public datasets both empirically and by examining their computational costs. The experiments demonstrate the superior performance of the proposed methods over the SOTA repeating pattern–based walking detectors.
- We provide guidelines for parameter settings by investigating the effect of different values. We also examine the influence of low-quality input data, and the result affirms that the proposed methods have practical value in handling battery-free wearable devices.

The rest of the article is organized as follows. Section 2 summarizes the recent findings on repetitive activity detection and motif discovery. Section 3 introduces the preliminaries of the proposed algorithm. In Section 4, we introduce the RCMP and mSIMPAD in detail. Section 5 presents the experimental evaluation on three public datasets. Section 6 discusses the potential problems and applications of mSIMPAD. Finally, Section 7 covers the conclusion and future direction of this work.

2 RELATED WORK

Our work is closely related to the detection of periodicity in time series, which is an active field of research in the data mining community that has been studied extensively. Autocorrelation function (ACF)-based methods and fast Fourier transform (FFT)-based methods are the two major approaches to date for periodicity detection in time series [8, 18, 26, 27, 35]. ACF computes the correlation of a sequence to a previous sequence candidate with varying lags, and the period is determined by the lag that maximizes the ACF. FFT converts a sequence from the time domain to the frequency domain and determines the period as the frequency that has the maximum power. Generally, the two methods have the same $O(n \log n)$ computational cost, and the major drawback of these methods is that they assume the pattern has the same periodicity. It fails when the periodicity varies over time, and the result is sensitive to the frequency that is being estimated.

In human activity recognition, ACF- and FFT-based methods have been widely used to detect activities that are composed of repetitive movements. Rai et al. [22] proposed a normalized-autocorrelation-based approach to identify the repetitive pattern of walking from IMU data. Brajdic and Harle [4] conducted a comprehensive evaluation of walk detection comparing the supervised and unsupervised methods including ACF- and FFT-based approaches. They show that all of the studied methods achieve comparable results. Physical exercises also consist of a set of repetitions of the same movement, which make them a detection candidate here. Guo et al. [9] extracted the magnitudes of the IMU data and computed the ACF to identify and count each repetition per set using data collected from wearable mobile devices. These approaches are especially robust when the period is known *a priori*. However, the period is usually not known and may vary over time in many real-world applications.

Xie et al. [28] decomposed a movement (*complex-activities*) into a series of small-range movements (*meta-activities*) and used the sequence of meta-activities to recognize a complex-activity. They collect angular information during physical exercise and apply dynamic time warping (DTW) to identify meta-activities to overcome this issue. The work of Maekawa et al. [16] is similar to this work, where it also identifies similar patterns in the repetition to evaluate assembly work in a factory. It identifies the motif within the IMU time series of each repetition and uses the interval of motif to estimate the lead time of the operation process.

Motif discovery has been extensively studied, but a breakthrough was made recently by Yeh et al. [33], who proposed an efficient algorithm, namely the STAMP to compute the matrix profile. Several extensions have been made in the following years for multi-dimensional time series, as well as toward the improvement of the efficiency of the algorithm [31, 32]. Mirmomeni et al. [17] proposed to leverage the matrix profile for mining SSP by examining the number of nearest neighbor arch crossings at each sample of the time series; however, this method will fail when a similar pattern appears in a faraway region of the series. Gharghabi et al. [7] introduced a temporal constraint to exclude arches from undesired regions in a different context for time series segmentation, but these methods inherit the same computational costs of the matrix profile that requires $O(n^2)$ time. To overcome these issues, we propose a general SSP detection based on the matrix profile with a time constraint during computation that can remove information from undesired regions while being efficient enough for real-time applications.

3 PRELIMINARY

3.1 Successive Similar Patterns Mining

In this article, we investigate SSPs mining from sensory data. Periodicity detection has been studied extensively, where different fields define it in different ways. We unite these definitions by starting with the definitions of the useful notations. A time series T is a sequence of real valued numbers, and a subsequence $T_{i,l}$ of T is a continuous subset of the values from T of length l starting from position i. Formally, $T_{i,l} = [t_i, \ldots, t_{i+l-1}]$. The distance between two subsequences $dist(T_{i,l}, T_{j,l})$ is measured by the z-normalized Euclidean distance:

$$dist(T_{i,l}, T_{j,l}) = \sqrt{\sum_{p=1}^{l} \left(\frac{t_{i+p-1} - \mu_{i,l}}{\sigma_{i,l}} - \frac{t_{j+p-1} - \mu_{j,l}}{\sigma_{j,l}} \right)^2}. \tag{1}$$

It is the root squared difference of the z-normalized values of two subsequences, where $\mu_{i,l}$ is the mean of $T_{i,l}$ and $\sigma_{i,l}$ is the standard deviation. This can be simplified as follows:

$$dist(T_{i,l}, T_{j,l}) = \sqrt{2l \left(1 - \frac{QT_{i,j} - l\mu_i\mu_j}{l\sigma_i\sigma_j} \right)}, \tag{2}$$

where $QT_{i,j}$ is the dot product of the two subsequences. A *successive similar pattern* is a subsequence $T_{i,l}$ of T where a *similar* subsequence $T_{j,l}$ appears within a *nearby range*. We then define similar as a small distance

between the two subsequences, and a nearby range refers to a small displacement between two subsequences. Formally, a subsequence $T_{i,l}$ is an SSP iff $\exists T_{j,l} : dist(T_{i,l}, T_{j,l}) < \alpha$, for $i \neq j$ and $|i - j| < m$, where $\alpha \in \mathbb{R}$ is some threshold in which $\alpha \geq 0$ and $m \in \mathbb{Z}$ is a user-defined window length. Then we define the problem as follows.

Problem 1 (Successive Similar Patterns Mining). *Given a multi-dimensional data series T, target subsequence length l, and the searching range m. We want to identify the subsequences that contain an SSP in T.*

3.2 Matrix Profile

Before we introduce the proposed method, a brief introduction to the matrix profile is provided as a background. This is a method recently proposed by Yeh et al. [33] for all-pair-similarity-search across a time series. The matrix profile is defined as a vector $MP = [mp_1, \ldots, mp_{n-l+1}]$ that stores the minimum distance of the subsequence to its nearest neighbor for every subsequence in T. The pair of subsequences that has the minimum distance, namely the motif pair, can be easily identified from the valley of the *MP*. The original matrix profile was developed for uni-dimensional time series, and it has been recently extended to process multi-dimensional time series. We suggest that interested readers refer to the work of Yeh et al. [32].

Efficient algorithms have been proposed to compute the matrix profile, including STOMP [31] and STAMP [33]. The former iterates the time series in sequential order, making it more efficient, whereas the latter is an anytime algorithm that iterates the time series in random order to produce an approximated result at any iteration. Theoretically, the computational cost of STAMP is $O(n^2 \log n)$, which was later superseded by SCRIMP++ [36], and both STOMP and SCRIMP++ are $O(n^2)$, where n is the length of the time series T.

We are now ready to introduce our proposed methods given the preceding definitions. In the next section, we will introduce SIMPAD to efficiently solve the SSP detection problem. Then, we will present mSIMPAD, which is an extension for multiple-length SSP detection.

4 METHODOLOGY

The detection method has two parts. First, we introduce SIMPAD to identify the segments that potentially contain SSPs, namely the set of "valleys" from the RCMP. Second, we choose a combination of valleys that maximize the likelihood of the segments being repetitive using a maximum weighted independent set (MWIS) algorithm. For simplicity, *repeating patterns* and *SSPs* are used interchangeably in the rest of the article.

4.1 Range-Constrained Matrix Profile

The original matrix profile calculates the distances between every subsequence to the rest of the time series and only preserves the distances and their corresponding indices for its nearest neighbor. However, such an approach allows the nearest neighbor to be located anywhere in the time series, which might not be of our interest, as the SSP should appear in a period that is considered "short." Figure 1 shows a case where an abnormal heartbeat due to ventricular contractions can hardly be identified by the regular matrix profile because of the coincident matching but is fairly notable in the RCMP. If similar ventricular contractions appeared multiple times over the entire ECG recording, it may identify the contractions as SSP with the regular matrix profile by accident. Therefore, we introduce the RCMP, where the nearest neighbor is calculated only within a given range. For ease of presentation, we refer to *MP* as the vector of RCMP in the rest of the article.

The idea of the RCMP is to find the nearest neighbor only within the searching range (length of the search window) m. Instead of calculating the entire distance profile, we calculate a range-constrained distance profile $DP_{i,m}$ for every subsequence $T_{i,l}$ in T. $DP_{i,m}$ is an intermediate vector to store the distances of subsequence $T_{i,l}$ to other subsequences from $T_{i-m,l}$ to $T_{i+m,l}$. Then the minimum value in $DP_{i,m}$ is selected to update the *MP*. We modified the mSTOMP algorithm in the work of Yeh et al. [32] introducing RC-mSTOMP, which produces the

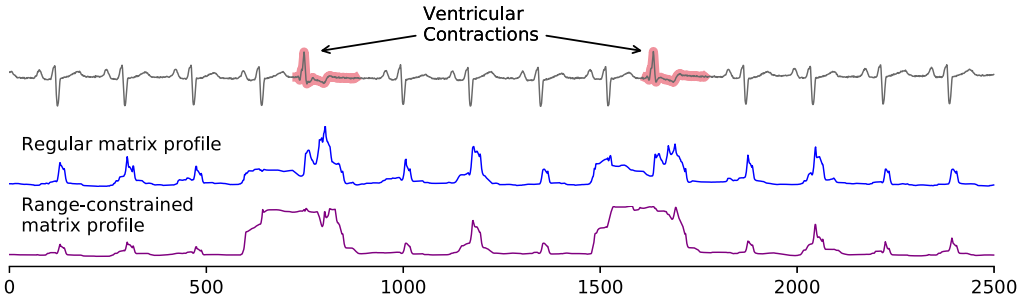


Fig. 1. Top: A snippet of ECG data in the Beth Israel Deaconess Medical Centre (BIDMC) PPG and Respiration dataset [2]. Middle: The regular matrix profile computed from the ECG signal. Bottom: The range-constrained matrix profile that clearly indicates the two ventricular contractions.

MP in $O(nm)$ time. This modification not only ensures a local similarity search but also significantly improves the efficiency of computing the RCMP. The details of this algorithm can be found in Algorithm 1.

ALGORITHM 1: Range-Constrained Multi-Dimensional Matrix Profile (RC-mSTOMP)

Input: d-dimension time series $\boldsymbol{T}$, int l, int m
Output: MP

double[] MP; int[] IP;
double[] $QT \leftarrow$ slidingDotProduct($\boldsymbol{T}_{1,l}, \boldsymbol{T}_{1,m}$);
int $s \leftarrow$ sum($\boldsymbol{T}_{1,l}$); int $ss \leftarrow$ squaredSum($\boldsymbol{T}_{1,l}$);
int $sr \leftarrow 2m+1$; int $dv \leftarrow \boldsymbol{t}_1$; int $nv \leftarrow \boldsymbol{t}_{m+1}$;
for $i \leftarrow 1$ *to* $|\boldsymbol{T}|$ **do**
  if $i > 1$ **then**
    $QT \leftarrow QT - dv \times \boldsymbol{T}_{i-m,sr} + nv \times \boldsymbol{T}_{i-m+l,sr}$;
    $s \leftarrow s - dv + nv$;
    $ss \leftarrow ss - dv^2 + nv^2$;
  $DP_{i,m} \leftarrow$ calcDistProfile($QT, \boldsymbol{T}_{i,l}, \boldsymbol{T}_{i-m,sr}, s, ss$);
  $DP_{i,m} \leftarrow$ columnWiseAscendingSort($DP_{i,m}$)
  $DP'_{i,m} \leftarrow$ double[d, sr] = {0,...,0}
  for $k \leftarrow 1$ *to* d **do**
    $DP'_{i,m} \leftarrow DP'_{i,m} + DP_{i,m}[k,:]$
    $DP''_{i,m} \leftarrow DP'_{i,m} \div k$
    $MP[k,:] \leftarrow$ elementWiseMin($MP[k,:], DP''_{i,m}$)
  end
  $dv \leftarrow \boldsymbol{t}_{i-m}$; $nv \leftarrow \boldsymbol{t}_{i+m+1}$;
end

Let $T_{i,l}$ and $T_{j,l}$ be the motif pair, which are two subsequences that have the lowest mutual distance between each other. One may imagine that a subsequence repeated once is a motif pair that forms two valleys in the MP at the location of the pair. However, the distance between $T_{i+1,l}$ and $T_{j+1,l}$ should also be small since most of the distances between the subsequences overlapped with the motif pair. Therefore, instead of having two (or the number of repetitions) separated valleys, the MP covering SSP should be a flat valley.

With this observation, we can identify SSPs by finding the valleys in the MP. The details of SIMPAD can be found in Algorithm 2. We first compute the MP providing the time series T and the target subsequence length l as

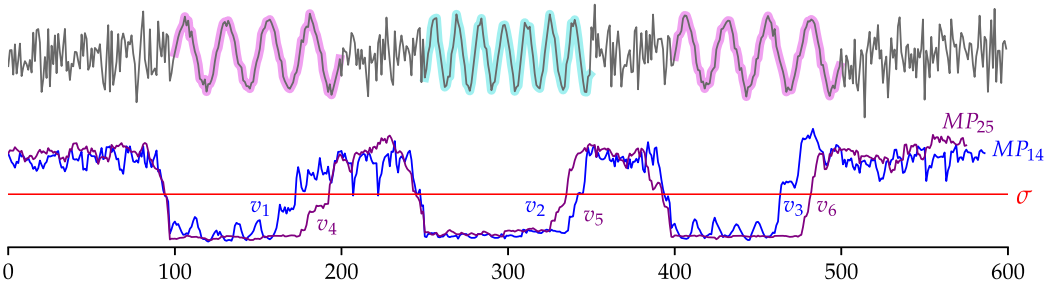


Fig. 2. An artificial signal contains three regions of sine waves with two intervals: from 100 to 200 and 400 to 500, the interval is 25, and from 250 to 350, the interval is 14. The bottom shows the MP with length 25 in purple and 14 in blue.

ALGORITHM 2: Successive sIMilar PAtterns Detector (SIMPAD)

Input: d-dimensional time series T, int l
Output: RP

```
MP ← RC-mSTOMP(T, l);
double α ← otsu_thresh(MP);
RP ← Boolean[|MP'|] = [0, ..., 0];
counter ← 0;
for i ← t to |MP| do
    if MP[i] ≤ α then
        if counter ≥ l then
            RP[i − l : i] ← 1;
        else
            counter ← counter + 1;
        end
    else
        counter ← 0;
    end
end
```

input. Then the key of this algorithm is to decide a suitable threshold to distinguish repeating and non-repeating components. This is a difficult task, as the distance of the MP correlates to the d and l. We assume that the input series T is a composition of repetitive subsequences and non-repetitive subsequences so that the MP is either at a distance of SSP or non-SSP segments. Then we apply the method of Otsu [19], which is a popular binarization method in the field of image processing to determine the threshold α. Those subsequences with distance below α are *valid* as the valleys shown in Figure 2. Finally, to avoid false positives caused by chance, we only accept subsequences that are valid for at least l consecutive timesteps. The result is stored in a Boolean vector RP of length $l - m + 1$, which indicates if the corresponding subsequence $T_{i,l}$ in T contains a repeated pattern or not.

Note that we could replace the RCMP by the regular matrix profile and perform the same detection pipeline for SSP identification. However, it will generate an MP that includes the nearest neighbor from anywhere of the entire time series and potentially degrades the detection performance. To overcome this issue, we might adopt the windowing approach by letting the window size equal the search range m while computing the regular matrix profile for each window. We then obtain the full MP by concatenating the regular matrix profiles of each window to perform SSP detection. This can ensure the range constraint, but the windows are assumed to be independent, which results in a loss of information coherency of the pattern as a whole. Instead, the RCMP incorporates the

range constraint in the computation, which preserves information coherency and reduces the computational cost. We include the detection results of SIMPAD and mSIMPAD with a regular matrix profile and those with a sliding-window-based regular matrix profile in Section 5.3.2.

4.2 Multiple-Length Successive Similar Patterns Detection

In the previous section, we introduced SIMPAD, which can be applied for fixed-length SSP detection. It assumes that all of the repeated patterns have the same length, so it may fail when patterns with different lengths appear within one time series. To tackle this problem, we present mSIMPAD to capture repeated patterns of different lengths automatically. The basic intuition is that for each potential pattern length, we compute the RCMP accordingly and identify the valleys as the candidates of repeated patterns. We then choose a set of valleys that best fits the patterns.

Let $\hat{l}$ be the *true length* of a repeated pattern that equals the interval length. The quality of a fit is then defined by the distance between $\hat{l}$ to the detected subsequence length l. In reality, $\hat{l}$ is usually not known and may vary over time. We assume that *a valley with larger sum of depth is a better fit to the repeated pattern.* The rationale behind this is that a larger value in the valley implies a clearer differentiation between repeated pattern and non-repeated patterns. Figure 2 shows an example where the area of valleys is larger when l is closer to $\hat{l}$. Finding a better fit to a repeated pattern is similar to searching for a valley with a larger sum.

To better illustrate the problem, we further introduce the notation of *MP*s and valleys with different target lengths. MP_l is the *MP* of T with subsequence length l. A valley $V_l \in \mathbb{R}^{|V_l|}$ is a sequence of differences between the subsequence $D \in \mathbb{R}^{|V_l|}$ of MP_l with some real value threshold α such that all values of V_l are less than α. Formally, $V_l = [\alpha - d_i | d_i < \alpha, \forall i \in [1, 2, \ldots, |V_l|]]$, where $d_i \in D$.

Identifying patterns with multiple periodicities in T maps to finding the best fit of valleys from multiple-length *MP*s. Given that we have computed all of the *MP*s for different l, there are two key issues when choosing the set of valleys that best fit the patterns. The first issue is that the scale of the distance depends on l. From Equation (1), we notice that longer subsequences tend to have larger distances, and therefore the difference between α and d_i could be larger. That being said, the distance incurs a strong bias to pairs of subsequences with larger l. To mitigate this effect, we obtain the length normalized distance by factorizing the *MP* by $sqrt(1/l)$, which is known to be better than simply factorized by l [14]. With the length normalized distance, we can compare the similarity of subsequence pairs with different length and the corresponding valleys.

The second issue is that at any point in T, there should be at most one valley chosen as the best fit of a given pattern. Although one valley may overlap with other valleys with different l, choosing one valley will reject the others. Using Figure 2 as an example, choosing v_1 will reject v_4 and vice versa. This could become a lot more complicated when $|L|$ is large, and one valley may be overlapped with multiple other valleys either from one length or different lengths. The longest valley is always chosen if we simply find valleys that yield the largest sum. To overcome this issue, we introduce the assumption that the sum of all selected valleys is maximized when all of the best fits of repeated patterns are found. Instead of choosing those longest valleys, the overall objective is to find a set of valleys that maximize the total sum. It allows the method to choose several shorter valleys over one long valley if the total sum is larger. Formally, the subproblem is defined as follows.

Problem 2 (Multiple Length Successive Similar Patterns Mining). *Let $V = \{V_1, V_2, \ldots, V_{|V|}\}$ be the set of valleys found in MP at all candidates l, and let $idx(V_l) = [x_1, x_2, \ldots, x_{|V_l|}]$ be the function mapping the valley to the corresponding index in T. The objective is to find the subset $V_{opt} \subseteq V$ such that the total sum of valley $\sum_{V_i \in V_{opt}} \sum_{j=1}^{|V_i|} v_j$ is maximized where $idx(V_i) \bigcap idx(V_j) = \emptyset, \forall V_i, V_j \in V_{opt}$, and $i \neq j$.*

This problem is related to the MWIS problem [20], which is an NP hard problem to find a subset of weighted vertices in a graph such that there exists no edge between any pair of the selected vertices while the sum of the weights is maximized. We can generate the graph $G = \{V', E\}$ for vertices $V' = \{v'_1, v'_2, .., v'_{|V|}\}$ as the sum

of valleys V as $v'_i = \sum_{j=1}^{|V_i|} v_j$. Then we can generate the set of edges E if two valleys are overlapped, formally $E = \{(v'_i, v'_j) | idx(V_i) \bigcap idx(V_j) \neq \emptyset\}$. The objective is then defined as

$$\begin{aligned} &\max_{V_{opt} \subseteq V'} \sum_{v_i \in V_{opt}} v_i \\ &s.t. \forall u, v \in V_{opt} : (u, v) \notin E. \end{aligned} \tag{3}$$

Notice that the graph generated is sparse with numerous components, since the valleys come from different parts of the time series and are separated by regions that have no repeated patterns. To find the solution more efficiently, we can divide the problem into multiple subproblems by the components in G while having the same optimal solution. This can drastically reduce the search space to speed up the computation. Then we apply the branch and bound approach in the work of Pardalos and Desai [20] for each of the subgraphs and finally combine the result to obtain the optimal solution.

The details of mSIMPAD can be found in Algorithm 3. First, we compute the MP for each potential pattern length and searching range and determine α using Otsu's method to identify the valleys from lines 2 to 6. Note that it is possible to let L be $[2, 3, \ldots, n/3]$ and M be $2 \times L$ if domain knowledge is missing. We generate graph G and separate it into multiple subgraphs in line 7. Lines 8 through 10 outline the loop for each subgraph to find the MWIS that is the best fit of valleys, and it is stored in V_{opt}. Finally, the repeated patterns are annotated as the indexes of selected valleys correspond to T.

ALGORITHM 3: Multiple-length Successive sIMilar PAtterns Detector (mSIMPAD)

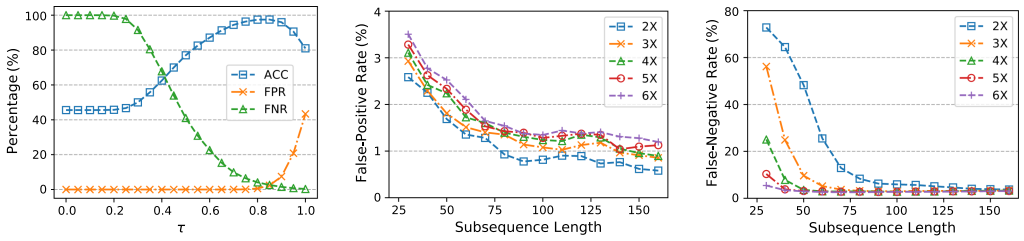
Input: d-dimension Time Series $\boldsymbol{T}$, int[] L, int[] M
Output: RP
1 $RP = [False, \ldots, False]$;
`// Find valleys from` MP `at different` l
2 **for** $l, m \leftarrow L, M$ **do**
3 $\quad MP_l \leftarrow$ RC-mSTOMP(T, l, m);
4 $\quad \alpha \leftarrow$ otsu(MP_l);
5 $\quad V[l] \leftarrow$ findValleys(MP_l, α);
6 **end**
`// Find best match from` V
7 $G \leftarrow$ generateGraphs(V);
8 **for** $G' \leftarrow G$ **do**
9 $\quad V_{opt} \leftarrow$ MWIS(G');
10 **end**
11 $RP[idx(V_{opt})] \leftarrow True$;
12 return RP;

5 EXPERIMENTAL EVALUATION

The experiment aims to answer the following questions: (1) How do different thresholds α, subsequence lengths l, and search ranges m affect the detection accuracy? (2) Does the proposed algorithm achieve a comparable result to the SOTA walking detectors? (3) Can it detect different forms of repetitive movement? (4) How does it scale to different sizes of input? We first introduce the metric for performance evaluation.

5.1 Performance Metric

For a time series T of length n, we identify if the subsequence $T_{i,l}$ that contains repetitive movement, where $i \in [1, \ldots, n - l + 1]$ and l is the window size of the search. This process produces an $n - l + 1$ detection result



(a) (b) (c)

Fig. 3. (a) Effect on accuracy (ACC), false-positive rate (FPR), and false-negative rate (FNR) of different τ. (b, c) Effect on FPR and FNR of different l and m.

denoted as $RP = [rp_1, rp_2, \ldots, rp_{n-l+1}]$:

$$rp_i = \begin{cases} 1, & \text{if } t_{i,l} \text{ contains SSP} \\ 0, & \text{otherwise.} \end{cases}$$

The ground truth of each trace contains a repetitive segment indicated by t_{start} and t_{end}, and we derive the truth label of each subsequence $T_{i,l}$ as 1 if $t_{start} \geq i \geq t_{end}$ and 0 otherwise. Then we define *accuracy* (ACC), *false -positive rate* (FPR), *false -negative rate* (FNR), and error rate (ERR) as follows:

$$\begin{cases} \text{ACC} = \frac{TP+TN}{|RP|} \\ \text{FPR} = \frac{FP}{FP+TN} \\ \text{FNR} = \frac{FN}{FN+TP} \\ \text{ERR} = \frac{FP+FN}{|RP|} \end{cases}$$

		Prediction 1	Prediction 0
Label	1	*TP*	*FN*
	0	*FP*	*TN*

5.2 Parameter Estimation

First, we study the effect of the parameters of SIMPAD on a dataset [4] collected from 27 participants using a conventional smartphone with an embedded accelerometer sampled at 100 Hz. The participants were told to walk at different speeds with different placement of the smartphone—for instance, carry by hand, in a pocket, in a backpack, or in a handbag. Then the ground truth was obtained by manually labeling each of the traces from the camera recording, which has the indicated start and end times when participants were walking. The previous study in the work of Brajdic and Harle [4] shows that both supervised and unsupervised approaches can accurately detect walking segments in which the median error rate is less than 2%.

The parameters were estimated by evaluating the performance on walk detection using the ground truth provided. We start by examining the effect of the threshold α. It is assumed that no repeating patterns appear in the first e subsequences, so the first e distances were used to obtain the baseline distance d_{base} for non-repetitive subsequences as $d_{base} = \sum_{i=0}^{e} mp_i$. We let $e = 100$, as the earliest walking session begins at 837. We let $0 \leq \tau \leq 1$ be a user-defined ratio to obtain threshold α manually by multiplying the baseline distance such that $\alpha = \tau \times d_{base}$. A range of values $[0, 0.05, 0.1, \ldots, 0.95, 1.0]$ for τ were examined over all traces, where we fix the other parameters for $l = 100$ (≈1 second), which is about to cover a *stride* (two steps), as the average steps per second is about 2 [22]. Let the searching range m be three times the subsequence length ($3l$). We excluded the first e subsequences in the evaluation, as we are using these distances to determine α.

Figure 3(a) shows the performance of different values of τ, where we found that lower τ results in higher FNR, and higher τ results in higher FPR. It suggests that the distance of the RCMP can effectively differentiate

between repetitive and non-repetitive subsequences. The value of τ was fixed as 0.85 for the following evaluation on different subsequence lengths l and searching ranges m.

Ideally, the subsequence length should be exactly the same as the length of the repeating pattern such that it matches the next cycle. However, repetitive movements in reality vary in every repetition, which makes the value difficult to determine. Similarly, the searching range should cover somewhere around $i + l$, as the next cycle should begin right after the current cycle. Unfortunately, the lag between each cycle varies and the shape might be deformed. Therefore, a larger search range provides a better chance to find a more similar nearby cycle, thus lowering the distances.

We demonstrated the relations between different subsequence lengths and searching ranges in Figure 3(b) and 3(c), in which the performance is calculated from the mean of the detection result over all traces. This shows that the FPR is insensitive to l, although a relatively higher FPR occurred when l is smaller. This is due to the smaller l providing fewer points to compare in a subsequence, so it is more likely to mismatch the non-repetitive subsequences randomly. On the contrary, the FNR seems to be very sensitive to l, as it increases drastically when l is small. The reason is that for a repetitive movement with a normal cycle of length $\hat{l}$, 100 in this case, if $l \ll \hat{l}$ (i.e., $l < \hat{l}/2$), then the subsequence contains only a small portion of the complete cycle. While the searching range is small, it cannot cover any portion of the next cycle with the given window. Therefore, the FNR is significantly higher for $l \leq 50$ and $m \leq 3 \times l$ given that the average length of a walking cycle is about 100. Fortunately, the larger m can tolerate the negative effect of a small l better, as it provides a better chance for the small portion of a cycle to be matched with the next cycle given a large window. But it comes with a minor drawback that the FPR is slightly increased.

As discussed earlier, the dataset in the work of Brajdic and Harle [4] is relatively simple and contains only idle and walking data. The authors achieved a median of total error of less than 2% with the best parameters. Unfortunately, we were unable to reproduce the result with the reported parameters, so we compare the performance mentioned in the work of Brajdic and Harle [4]. SIMPAD achieved a comparable result, as the median of total error rate is 1.78% using the parameters ($l = 100, m = 6 \times l, \tau = 0.85, e = 100$).

5.3 Repetitive Movement Detection

We first evaluate if the proposed methods achieve comparable results to the SOTA walking detectors on the HAPT [1] dataset. Then we study if mSIMPAD is generic enough for the general repetitive movement detection by evaluating the performance on PAMAP2 [23] that contains various activities.

5.3.1 Evaluation of Robustness. HAPT is a dataset collected from 30 volunteers using an accelerometer and gyroscope on a smartphone at a 50-Hz sampling rate. It contains different forms of activities, including walking, climbing up or down stairs, sitting, standing, lying, and transitions between activities. We use the precision = $TP/(TP + FP)$, recall = $TP/(TP + FN)$, and F-score = $2 * (p \times r)/(p + r)$ for the evaluation.

We compare the proposed methods to the widely used walk detection algorithms, namely normalized autocorrelation–based step counting (NASC) [22] and short-term Fourier transform (STFT) [3] correspond to the ACF-based method and the FFT-based method. These algorithms were the best-performing algorithms as noted in the work of Brajdic and Harle [4]. NASC excluded segments of the time series where the standard deviation was below a threshold σ_{thresh} over a window std_{win}, then it performs normalized autocorrelation over a window of 2 seconds with a range of time span τ_{min} to τ_{max} for those remaining subsequences. Those subsequences are then asserted to be walking if the maximum of the normalized autocorrelation exceeded another threshold R_{thresh}. STFT is a Fourier transform–based method that calculates the frequency domain energy of the vertical acceleration signal with consecutive windows of size dft_{win}, and it affirms walking if the total energy of the interested frequencies exceeds threshold dft_{thresh}.

The raw linear accelerations obtained from accelerometer were used for SIMPAD, mSIMPAD, and NASC. Since STFT takes the vertical velocity as input, we apply the Madgwick algorithm [15] to estimate the orientation of

Table 1. List of the Parameter Values Used in Each Algorithm for the HAPT Dataset

Algorithm	Parameter Value
NASC	$std_{win} = 40, \sigma_{thresh} = 0.24, R_{thresh} = 0.4, \tau_{min} = 40, \tau_{max} = 100$
STFT	$dft_{win} = 60, dft_{thresh} = 0.25, freq_{min} = 0.01Hz, freq_{max} = 7Hz$
SIMPAD	$l = 50, m = 5 \times l$
mSIMPAD	$L = [40, 50, 60], M = 5 \times l \in L$

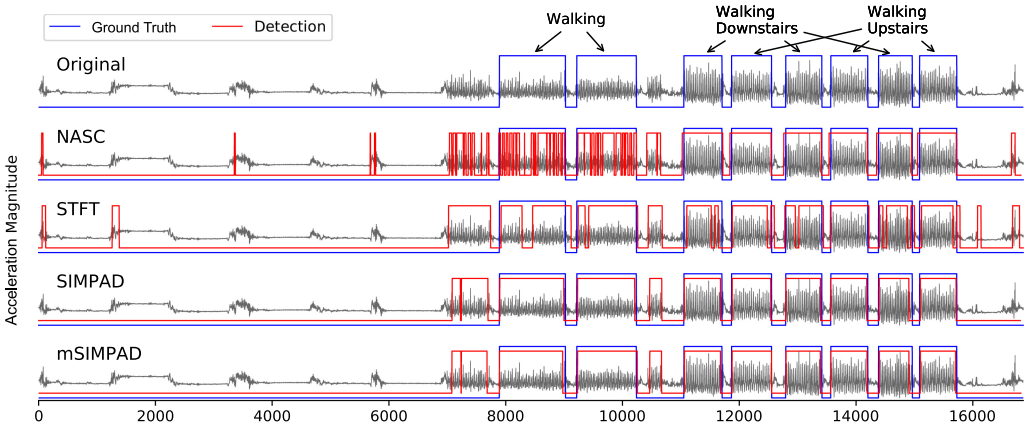


Fig. 4. Detection result on one of the traces with different algorithms. The top row indicates the ground truth of the trace with a blue line, and the detection results are indicated with a red line in the lower figures.

the smartphone using the gyroscope signal, then transform the linear acceleration to the coordinate with respect to the earth.

The parameters of the SOTA methods were then selected using a brute force search approach to provide the upper bound of performance for each of the methods. The selected values are reported in Table 1. For SIMPAD, we are only required to provide the parameters of l and m. We choose the length to be 1 second (≈50 samples) as discussed earlier. From the previous experiment, we notice that five times the subsequence length can tolerate the mismatch between l and l_{true}well. For mSIMPAD, we choose $L = [40, 50, 60]$ to cover the variations of walking speeds and the same scale for the searching range as $M = 5 \times L$.

An example of a detection result of each algorithm is shown in Figure 4. The red lines indicate the detection result generated by the methods where 0 is non-repetitive and otherwise is repetitive. The ground truth is indicated by blue lines. The graph shows that NASC suffers from a higher FNR and breaks one walking period into several segments due to the difficulty of defining a global threshold. STFT has a more continuous detection period but is not sensitive enough to cover the walking period in place that results in higher FPR. On the contrary, SIMPAD and mSIMPAD reveal better performance and cover the entire walking period and fit the walking period better compared to the other algorithms.

From the preceding example, we observed that the data contains two subsequences that have significantly higher acceleration magnitude for every trace (between samples 7050-7900 and 10250-11060 in the preceding example). We further investigated all traces and found that there exist repeating patterns that have not been reported in the annotations provided in the work of Anguita et al. [1]. This might be due to the relocation of the experiment, as it happens when the task changes from lying to walking, and from walking to climbing downstairs. Therefore, we excluded the two suspicious zones of data from the last transition to the first walking part, and from the last walking to the first climbing downstairs part for a more precise evaluation.

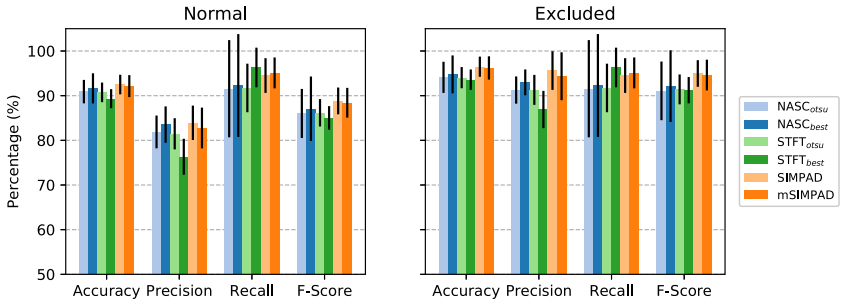


Fig. 5. Performance on the HAPT dataset.

The overall result is reported in Figure 5 including the normal evaluation where the entire dataset was used and the excluded evaluation where the suspicious zone has been removed. To provide a fair comparison, we modified NASC and STFT to automatically determine the threshold using Otsu's method as we use in our proposed approach, and selected 1 second as the window size for those methods. We notice that the modified NASC and STFT achieved a comparable result to the best parameters obtained from exhaustive search. Automatic threshold determination may lead to variation tolerance between different time series to the point where its incorporation into STFT even outperforms the best parameters. Although NASC is a two-step thresholding approach, we fixed r_{thresh} as 0.4 and determine std_{thresh} by using Otsu's method, which yields similar results to the best parameters. Both SIMPAD and mSIMPAD show improvements by 2.8 and 2.5%, respectively, with F-scores roughly equal to 95%. In addition, the high precision and recall of the proposed methods illustrate that the RCMP is a robust indicator for differentiating repeating and non-repeating patterns.

5.3.2 Evaluation on Generality. This section aims to evaluate the ability of the proposed method on general SSP detection. The evaluation is conducted using PAMAP2 [23], which is collected from nine subjects wearing three IMUs sampled at 100-Hz frequency while performing 18 different activities. The various types of activities aim to provide a range of different repeating frequencies for generality evaluation. We classify the following activities as repetitive: walking, running, cycling, Nordic walking, ascending stairs, descending stairs, and rope jumping. The rest are considered as non-repetitive activities. We leverage the IMU data on the subject's ankle for activity detection. We down-sample the data to 50 Hz, and the same transformation method mentioned in the previous section was applied for vertical acceleration estimation.

We compare the results to the modified NASC and STFT with the same parameters that were used in the previous section, as it is difficult to define a global threshold for various activities. For mSIMPAD, we choose $L = 40, 70, 100$ accordingly to the 0.8, 1.4, and 2 seconds to capture repeating patterns with different lengths. The lengths were selected as a reference to τ_{min} and τ_{max} where NASC searches within this time lag range.

The overall results are reported in Table 2 in which the best values are in bold. The proposed methods outperform NASC and STFT by 5.7% in F-score. It shows that the RCMP works well with different types of repetitive movements and different lengths of repeating pattern. The similar results of SIMPAD and mSIMPAD suggest that if the target patterns have similar lengths, SIMPAD can capture most of the repeating patterns with ease. The performance of mSIMPAD is still higher than SIMPAD, as it can find a better match within the patterns of different lengths. We expect the improvement would be much larger when the variability of the pattern lengths is huge.

Note that both SIMPAD and mSIMPAD are MP-based methods, in which the input can be replaced by the original MP. However, the regular MP has a higher probability of finding the nearest neighbor that is not a repeating pattern just by chance without the range constraint. We can see that from its performance on the HAPT dataset, where the F-scores are 92.15% and 91.23% for SIMPAD and mSIMPAD, respectively. We notice

Table 2. Performance on the HAPT Dataset Where the Values Are Given as Mean ± SD

Dataset	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
HAPT	NASC	94.10 ± 3.49	91.26 ± 3.05	91.56 ± 10.88	91.07 ± 6.57
	STFT	94.03 ± 2.37	91.30 ± 3.35	91.73 ± 5.44	91.41 ± 3.34
	SIMPAD	**96.44 ± 2.26**	**95.63 ± 4.33**	94.50 ± 3.89	**94.96 ± 2.98**
	mSIMPAD	96.16 ± 2.60	94.35 ± 5.36	**95.10 ± 3.45**	94.62 ± 3.44
PAMAP2	NASC	81.70 ± 12.32	**99.39 ± 0.85**	66.47 ± 21.95	77.12 ± 22.26
	STFT	78.79 ± 9.12	99.31 ± 0.90	62.45 ± 6.75	76.50 ± 4.76
	SIMPAD	84.11 ± 5.59	99.12 ± 1.14	71.24 ± 5.66	82.78 ± 3.74
	mSIMPAD	**84.62 ± 5.65**	98.28 ± 1.78	**72.90 ± 5.58**	**83.59 ± 3.62**

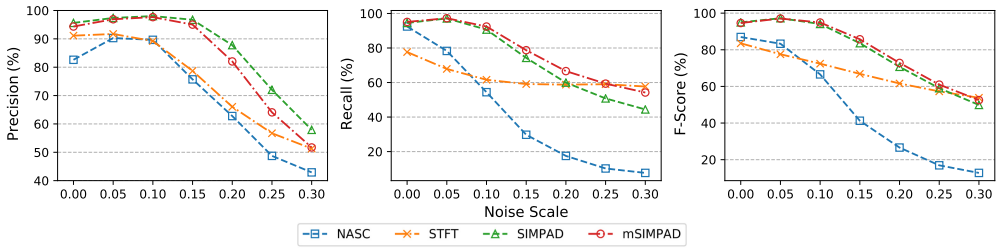


Fig. 6. Effect of sensor noise to performance.

a significantly lower precision (87.78% and 85.89%) that complies with the inference when the regular matrix profile is used. Alternatively, we could perform windowing on the time series by letting m be the window size to satisfy the range constraint. However, the sliding window fails to capture coherent information, which results in poor performance compared to the RCMP. The SIMPAD and mSIMPAD with sliding-window-based regular matrix profile achieved F-scores of 92.27% and 92.86%, respectively, on HAPT, and 79.96% and 81.28% on PAMAP2, which shows that both are worse than the proposed RCMP-based approach. The RCMP satisfied the range constraint while preserving the coherent information and greatly reduced the computational cost, which is a more suitable solution for the problem at hand.

5.3.3 Robustness on Low-Quality Data. Battery-free wearable devices rely only on harvested kinetic energy from the user, which has emerged as an alternative to power sensor nodes [25]. The wireless communication and sensing units consume much more power than a typical microcontroller, so the transmission and sampling rate of such devices is reduced to optimize power consumption [11]. We investigate the influence of low-quality accelerometer data by downsampling the traces in the HAPT dataset to 20 Hz and adjusted the parameters l and m by the downsampling ratio. As the performance of SIMPAD and mSIMPAD is similar in the HAPT dataset, we report the mSIMPAD result for simplicity. The performance of mSIMPAD recorded a decrease by 2.11% to 92.51%, whereas NASC and STFT also decreased by 2.2% and 0.12% to 88.87% and 91.29%, respectively. This shows that the existing approaches perform fairly well on sensor data having a low sampling frequency, as most human activities are lower than 10 Hz.

We also study how sensor quality influences performance by adding Gaussian noise to the traces. We first normalized the traces to the scale of 0 to 1, then inserted random noise of a normal distribution having 0 mean and scale as a standard deviation between 0 and 0.3. Surprisingly, the performance of mSIMPAD improves slightly when a small level of noise is inserted into the signal (scale = 0.05) as shown in Figure 6. The reason is that segments of relatively idle data could be identified as similar patterns (e.g., a slightly upward trend or slight

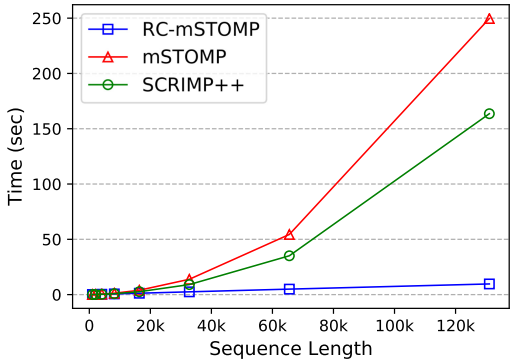


Fig. 7. A comparison on execution time of different sequence lengths.

downward trend), which actually should not be detected at all. The added noise increases the distance between these drifts and can help distinguish such data up to a certain noise level (10% of the maximum sensor value in this case), which suggests a future direction toward improving the performance of mSIMPAD by simply adding random noise. The performance of STFT is slightly higher than the proposed approach under very serious noise where the noise scale is 0.3—that is, 30% of the maximum value of the data, which is rather unrealistic. In general, the proposed approach is more robust than the existing methods, whereas STFT potentially performs better in the situation of extremely low quality data.

5.4 Comparison of Execution Times

The preceding evaluation demonstrated that mSIMPAD is a robust method for repetitive movement detection even on datasets that contain multiple activities. In this section, the empirical computational cost to obtain the *MP* is examined. Theoretically, the time complexity of RC-mSTOMP is $O(nm)$, in which m can be neglected as $m \ll n$, whereas mSTOMP is $O(n^2)$ and both ACF and FFT are $O(n \log n)$, where n is the sequence length, and m is the searching range (note that both algorithms increase linearly with respect to the number of dimensions). The time complexity of RC-mSTOMP is significantly lower than the other methods since it scales linearly to the sequence length. We evaluate this property by comparing RC-mSTOMP to its parent algorithms (mSTOMP and SCRIMP++) and also examine the effect on the execution time with different parameters. All experiments were performed on a conventional PC with an Intel Core i7-8850H CPU @ 2.60 GHz and 12 and 16 GB of RAM. The default values of the parameters are as follows when not specified: $n = 2^{14}$, $l = 100$, $m = 200$, and $d = 1$.

First, we examined the computational cost on different sequence length empirically to compare RC-mSTOMP to its parent algorithms mSTOMP and SCRIMP++. It is not intended to conclude that the proposed algorithms are superior than their parent algorithms, as the goal of these algorithms is different. Instead, we aim to demonstrate that the proposed method inherits the important properties of the parent algorithms while scaling linearly with respect to sequence length so that it can support real-time applications. In this evaluation, different lengths of sinusoidal signals were generated as the input sequences. The resulting execution times are shown in Figure 7. They coincided with our expectations where RC-mSTOMP produces the lowest execution time among all of the other algorithms and scales linearly, as we will show in the following. mSTOMP and SCRIMP++ are roughly scaling at $O(n^2)$ but still very scalable to large time series. SCRIMP++ completes slightly faster than mSTOMP, which might due to the sinusoidal data in this particular case.

Then, we examined the effect on execution time over different parameters: n, l, m, and d. The resulting execution times are shown in Figure 8. The figure shows that the proposed algorithm inherits the same property as discussed in the work of Yeh et al. [32] on subsequence lengths and dimensionality. Subsequence length has no effect on execution time when m is fixed, and it follows a linear relationship to dimensionality. Next, we

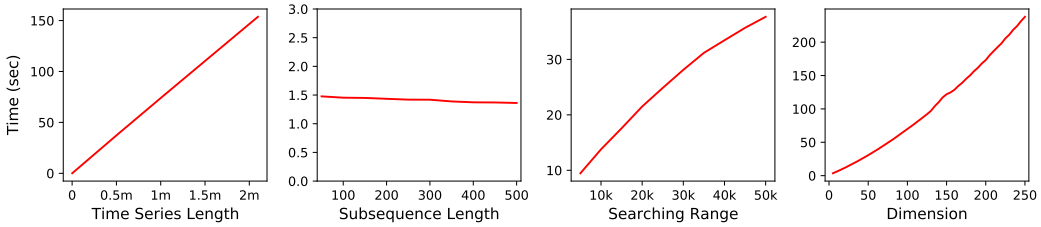


Fig. 8. An evaluation on execution time of different parameters.

examined the effect of the length n of the input sequence by fixing the other parameters as default values and execute RC-mSTOMP on the time series with increasing length. As expected, we found that the execution time increases linearly with respect to sequence length. Finally, we evaluated the effect of the length of the searching range m on execution time. We found that for small n, the effect of different sizes of m is negligible. Therefore, we increased n to 2^{16} and $l = 500$. The result at the right of Figure 8 shows that it also follows a linearithmic relationship with respect to m.

In this section, we demonstrated that RC-mSTOMP is capable of supporting real-time applications, as the execution time has no effect on l and linearly correlates to n, m, and d. In the next section, we will discuss the potential problems and application of this work.

6 DISCUSSION

We proposed the method for multiple-length SSPs mining using the matrix profile, and we evaluated the proposed methods in the use case of repetitive movement detection on three public datasets. Results show that the proposed method is efficient and robust for general repeated pattern mining without prior knowledge of the pattern, except the expected lengths of the target patterns. In this section, we discuss the potential problems and the applications of the proposed algorithm.

The underlying technique of the proposed method is related to matching the patterns of a time series on its own. This seems to coincide with ACF and FFT, which are the commonly used techniques in the previous work, but there are two major differences that explain the superiority of the proposed method. First, ACF and FFT have more restricted constraints on the time span so that the repeated patterns occur with regular intervals, whereas our proposed method is capable of detecting repeated patterns with a variable time span since we are searching for a local motif within a given range m. Second, efficient algorithms for calculating ACF and FFT are $O(n \log n)$, whereas our method achieved $O(nm)$. To the best of our knowledge, it is by far the fastest deterministic and exact algorithm for SSP detection.

The key limitation of the proposed method is to determine the parameters. The pattern length l and the displacement m could be designated based on the sampling frequency of the sensor data. We suggest that larger l and m would be more favorable to highly repeating patterns such as walking as shown in Section 5.2. In addition, the experiment shows that SIMPAD can work well even if l is quite different from the actual pattern length. In addition, the threshold σ is determined by Otsu's method, which assumed the time series is a composition of repeated and non-repeated components. If such assumption is not met, choosing a global threshold would be an option. This is not difficult because the distance is normalized both by the signal and the length of the pattern.

The experiments might not show an enormous improvement of our method compared to the SOTA as reported in Section 5.3.2. However, considering the modest number of parameters to set, and its robustness to novel situations and to poor-quality sensor data, the proposed method has great potential as a general approach that is devoid the effort devoted to studying specific scenarios. In addition, we investigated the detection results over different series and found two key reasons that degrade the detection performance. First, the dataset is generated mainly for activity recognition purposes, so the quality of the ground truth labels is rough. There are offsets on

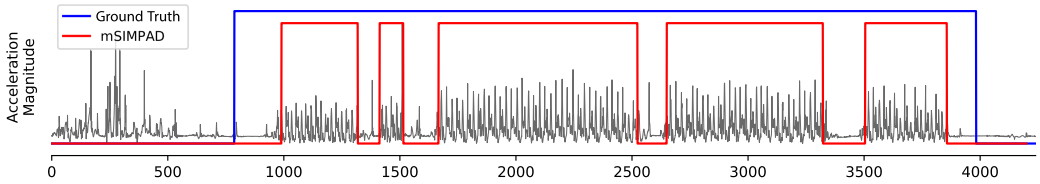


Fig. 9. An example of rope jumping data in PAMAP2 [23]. The magnitude of the acceleration signal shows that several pauses occur during the activity. The blue line indicates the ground truth of the repetitive movements, and the red line indicates the detection result of mSIMPAD: 0 as non-repeating and greater than 0 as repeating.

almost every repeated activity that result in many false negatives that actually are non-repeated components. As well, the repeated activities are not always continuous nor contiguous, but the ground truth labeling annotated the entire segment as one activity. For instance, participants may fail while rope jumping. An example can be found in Figure 9 where the participant has stopped several times during rope jumping. Those regions constitute idle data, as no body motion is captured, whereas the ground truth data was not handled to that level of detail.

Second, we only used the raw accelerometer data as input, which is a noisy signal. Various techniques would be applied to obtain quality data including filtering and sensor fusion to improve the detection performance. As an example, we included the gyroscope data as input to mSIMPAD, and the average F-score has increased from 83.6 to 87.5 for the PAMAP2 dataset. However, we aim to compare the result to the SOTA walking detectors within a consistent setting. We therefore apply the same configuration over the three datasets to demonstrate that the proposed methods are superior to the SOTA in general, but not by manipulating the data input nor the parameters of the algorithm. In addition, our method can better extract those repeated activities in place as shown in the example in Figures 4 and 9. This suggests that the proposed method is desired, and it can be applied as a subroutine of human activity recognition and analysis.

The potential applications of the proposed algorithm are twofold. For non-periodical time series, it can identify where repeated patterns occurred, especially for asynchronous periodic patterns and slowly changing patterns. Applications such as exercise tracking, which rely on repeated pattern detection, can utilize the proposed method for better performance. However, it can identify abnormalities for periodical time series such as ECGs of heartbeats as shown in Figure 1. The intuition is that the *MP* computed from periodical data is the composition of regular patterns and abnormal patterns. For the regular patterns, the distance of *MP* should be close to 0, where the abnormal patterns are the discords from the *MP*. It can also be applied as a subroutine of other data mining tasks such as activity recognition, activity segmentation, and routine discovery. For instance, RCMP can efficiently identify those segments containing repetitive movements, and machine learning techniques can then be applied only to those segments to eliminate unnecessary computation.

7 CONCLUSION

An SSP is a key feature of many kinds of interesting data. The detection of these repeating patterns without prior knowledge comes with significant challenges. In this study, we proposed mSIMPAD, an efficient algorithm for multiple-length SSP detection based on the matrix profile. We formally defined SSP based on the distance to the nearby subsequences and introduced the RCMP—a modification of the original matrix profile—to compute the distances efficiently. Then, SIMPAD was proposed, and we further extended it to handle multiple-length SSPs automatically, namely mSIMPAD. We studied the repetitive movement detection problem as a use case, and we conducted experiments on three public datasets to evaluate the proposed method in terms of robustness and time efficiency. The experimental evaluation shows that mSIMPAD achieved on par (or even better) results compared to the SOTA repeating pattern–based walking detectors on all three public datasets. Finally, we discussed the

potential problems and applications arising from the proposed method. In the future, we plan to extract the template of the SSPs using the RCMP and apply it to activity recognition and abnormality detection.

ACKNOWLEDGMENTS

We sincerely appreciate the efforts of the anonymous reviewers and their insightful comments for improving this article.

REFERENCES

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. *In Proceedings of ESANN 2013.*

[2] Donald S. Baim, Wilson S. Colucci, E. Scott Monrad, Harton S. Smith, Richard F. Wright, Alyce Lanoue, Diane F. Gauthier, Bernard J. Ransil, William Grossman, and Eugene Braunwald. 1986. Survival of patients with severe congestive heart failure treated with oral milrinone. *Journal of the American College of Cardiology* 7, 3 (1986), 661–670.

[3] P. Barralon, N. Vuillerme, and N. Noury. 2006. Walk detection with a kinematic sensor: Frequency and wavelet comparison. *In Proceedings of IEEE EMBS 2006.* DOI: https://doi.org/10.1109/iembs.2006.260770

[4] Agata Brajdic and Robert Harle. 2013. Walk detection and step counting on unconstrained smartphones. *In Proceedings of ACM UbiComp 2013.* ACM, New York, NY, 225–234. DOI: https://doi.org/10.1145/2493432.2493449

[5] Maria Cornacchia, Koray Ozcan, Yu Zheng, and Senem Velipasalar. 2016. A survey on activity detection and classification using wearable sensors. *IEEE Sensors Journal* 17 (2016), 386–403. DOI: https://doi.org/10.1109/jsen.2016.2628346

[6] Hoang Anh Dau and Eamonn Keogh. 2017. Matrix Profile V: A generic technique to incorporate domain knowledge into motif discovery. In *Proceedings of ACM SIGKDD 2017.* ACM, New York, NY, 125–134.

[7] Shaghayegh Gharghabi, Chin-Chia Michael Yeh, Yifei Ding, Wei Ding, Paul Hibbing, Samuel LaMunion, Andrew Kaplan, Scott E. Crouter, and Eamonn Keogh. 2019. Domain agnostic online semantic segmentation for multi-dimensional time series. *Data Mining and Knowledge Discovery* 33, 1 (2019), 96–130.

[8] Earl F. Glynn, Jie Chen, and Arcady R. Mushegian. 2006. Detecting periodic patterns in unevenly spaced gene expression time series using Lomb-Scargle periodograms. *Bioinformatics* 22 (2006), 310–316. DOI: https://doi.org/10.1093/bioinformatics/bti789

[9] Xiaonan Guo, Jian Liu, and Yingying Chen. 2017. FitCoach: Virtual fitness coach empowered by wearable mobile devices. *In Proceedings of IEEE INFOCOM 2017.* DOI: https://doi.org/10.1109/infocom.2017.8057208

[10] Tian Hao, Guoliang Xing, and Gang Zhou. 2015. RunBuddy: A smartphone system for running rhythm monitoring. In *Proceedings of ACM UbiComp 2015.* ACM, New York, NY, 133–144.

[11] Qianyi Huang, Yan Mei, Wei Wang, and Qian Zhang. 2016. Battery-free sensing platform for wearable devices: The synergy between two feet. *In Proceedings of IEEE INFOCOM 2016.*

[12] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2011. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter* 12 (2011), 74–82. DOI: https://doi.org/10.1145/1964897.1964918

[13] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. 2007. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery* 15 (2007), 107–144. DOI: https://doi.org/10.1007/s10618-007-0064-z

[14] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. 2018. Matrix Profile X: VALMOD—Scalable discovery of variable-length motifs in data series. *In Proceedings of ACM SIGMOD2018.* DOI: https://doi.org/10.1145/3183713.3183744

[15] Sebastian Madgwick. 2010. *An Efficient Orientation Filter for Inertial and Inertial/Magnetic Sensor Arrays.* Report X-IO. University of Bristol.

[16] Takuya Maekawa, Daisuke Nakai, Kazuya Ohara, and Yasuo Namioka. 2016. Toward practical factory activity recognition: Unsupervised understanding of repetitive assembly work in a factory. *In Proceedings of ACM UbiComp 2016.* ACM, New York, NY, 1088–1099. DOI: https://doi.org/10.1145/2971648.2971721

[17] Mahtab Mirmomeni, Yousef Kowsar, Lars Kulik, and James Bailey. 2018. An automated matrix profile for mining consecutive repeats in time series. *In Proceedings of PRICAI 2018.* DOI: https://doi.org/10.1007/978-3-319-97310-4_22

[18] Abdullah Mueen, Suman Nath, and Jie Liu. 2010. Fast approximate correlation for massive time-series data. *In Proceedings of ACM SIGMOD 2010.* DOI: https://doi.org/10.1145/1807167.1807188

[19] Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1979), 62–66. DOI: https://doi.org/10.1109/tsmc.1979.4310076

[20] Panos M. Pardalos and Nisha Desai. 1991. An algorithm for finding a maximum weighted independent set in an arbitrary graph. *International Journal of Computer Mathematics* 38 (1991), 163–175. DOI: https://doi.org/10.1080/00207169108803967

[21] Nastaran Mohammadian Rad, Seyed Mostafa Kia, Calogero Zarbo, Twan van Laarhoven, Giuseppe Jurman, Paola Venuti, Elena Marchiori, and Cesare Furlanello. 2018. Deep learning for automatic stereotypical motor movement detection using wearable sensors in autism spectrum disorders. *Signal Processing* 144 (2018), 180–191. DOI: https://doi.org/10.1016/j.sigpro.2017.10.011

[22] Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-effort crowdsourcing for indoor localization. *In Proceedings of ACM MobiCom 2012.* DOI: https://doi.org/10.1145/2348543.2348580

[23] Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. *In Proceedings of IEEE ISWC 2012.* DOI: https://doi.org/10.1109/iswc.2012.13

[24] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul J. M. Havinga. 2015. A survey of online activity recognition using mobile phones. *Sensors* 15 (2015), 2059–2085. DOI: https://doi.org/10.3390/s150102059

[25] Sujesha Sudevalayam and Purushottam Kulkarni. 2010. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys & Tutorials* 13, 3 (2010), 443–461.

[26] Michail Vlachos, Philip Yu, and Vittorio Castelli. 2005. On periodicity detection and structural periodic similarity. *In Proceedings of SDM 2005.* DOI: https://doi.org/10.1137/1.9781611972757.40

[27] Michail Vlachos, Philip S. Yu, Vittorio Castelli, and Christopher Meek. 2006. Structural periodic measures for time-series data. *Data Mining and Knowledge Discovery* 12 (2006), 1–28. DOI: https://doi.org/10.1007/s10618-005-0016-4

[28] Lei Xie, Xu Dong, Wei Wang, and Dawei Huang. 2017. Meta-activity recognition: A wearable approach for logic cognition-based activity sensing. *In Proceedings of IEEE INFOCOM 2017.* DOI: https://doi.org/10.1109/infocom.2017.8057209

[29] Jiong Yang, Wei Wang, and P. S. Yu. 2003. Mining asynchronous periodic patterns in time series data. *IEEE Transactions on Knowledge and Data Engineering* 15 (2003), 613–628. DOI: https://doi.org/10.1109/tkde.2003.1198394

[30] Kung-Jiuan Yang, Tzung-Pei Hong, Yuh-Min Chen, and Guo-Cheng Lan. 2013. Projection-based partial periodic pattern mining for event sequences. *Expert Systems with Applications* 40 (2013), 4232–4240. DOI: https://doi.org/10.1016/j.eswa.2013.01.021

[31] Chin-Chia Michael Yeh, Helga Van Herle, and Eamonn Keogh. 2016. Matrix Profile III: The matrix profile allows visualization of salient subsequences in massive time series. *In Proceedings of IEEE ICDM 2016.* DOI: https://doi.org/10.1109/icdm.2016.0069

[32] Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamonn Keogh. 2017. Matrix Profile VI: Meaningful multidimensional motif discovery. *In Proceedings of IEEE ICDM 2017.* DOI: https://doi.org/10.1109/icdm.2017.66

[33] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix Profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. *In Proceedings of IEEE ICDM 2016.* DOI: https://doi.org/10.1109/icdm.2016.0179

[34] Xiao Yu, Qing Li, and Jin Liu. 2019. Scalable and parallel sequential pattern mining using spark. *World Wide Web: Internet and Web Information Systems* 22 (2019), 295–324. DOI: https://doi.org/10.1007/s11280-018-0566-1

[35] Quan Yuan, Jingbo Shang, Xin Cao, Chao Zhang, Xinhe Geng, and Jiawei Han. 2017. Detecting multiple periods and periodic patterns in event time sequences. *In Proceedings of ACM CIKM 2017.* DOI: https://doi.org/10.1145/3132847.3133027

[36] Yan Zhu, Chin-Chia Michael Yeh, Zachary Zimmerman, Kaveh Kamgar, and Eamonn Keogh. 2018. Matrix Profile XI: SCRIMP++: Time series motif discovery at interactive speeds. *In Proceedings of IEEE ICDM 2018.*

Received August 2019; revised April 2020; accepted April 2020