Repetitive Activity Monitoring from Multivariate Time Series: A Generic and Efficient Approach

Chun-Tung Li*, Jiaxing Shen*, Yanni Yang*, Jiannong Cao*, Milos Stojmenovic†
*Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
†Department of Informatics and Computing, Singidunum University, Belgrade, Serbia
Email: chun-tung.li@connect.polyu.hk, {jiaxshen,csynyang,csjcao}@comp.polyu.edu.hk, mstojmenovic@singidunum.ac.rs

***Abstract*—Repetitive activities like breathing and walking account for a large fraction of human activities. Monitoring these activities with sensing technology plays a vital role in numerous applications ranging from health monitoring to manufacturing management. Over the last decade, traditional machine learning approaches and recent end-to-end deep learning paradigms have achieved massive successes in human activity recognition. However, these approaches are mostly scenario dependent and computationally expensive. Moreover, real-world repetitive activities may have varying time intervals between each repetition, which invalidate existing sliding window methods. In this paper, we propose STEM, a Scalable Template Extraction Method for scenario independent monitoring of repetitive activities with varying intervals. Instead of using sliding windows, we detect and locate the appearance of repeating patterns based on the Matrix Profile. Distributional features are then extracted from the identified patterns such that domain knowledge can be avoided. The approach is efficient and robust as shown by the evaluation on three public datasets, in which around 95% of the undesired computation were eliminated with up to 4% accuracy improvement. It is also generic as demonstrated by a use case of respiration rate estimation using wireless signals.**

***Index Terms*—Repetitive Activity Monitoring; Human Activity Recognition; Multivariate Time Series; Internet of Things;**

I. Introduction

Repetitive activities are building blocks of our daily lives including fundamental bodily functions and common human activities, like respiration, heartbeat, and exercises. Repetitive activity monitoring (RAM) refers to the ability to measure and distinguish among different successively repeated physical motions, enables numerous applications ranging from health monitoring [1], [2] to manufacturing management [3]. The recent advancement in Internet of Things (IoT) technologies allows efficient collection of sensor data, providing a great opportunity to capture repetitive activities continuously. For example, premature ventricular contractions as a risk factor to many heart diseases could be detected via RAM with commodity wearable devices [4].

Due to the significance of RAM, various approaches have been proposed utilizing signal processing and machine learning techniques to identify the repeating patterns or recognize the activity by analyzing the extracted features. Although those approaches have their merit, they suffer from the following key limitations. First, existing approaches are mostly sliding-window-based methods. It typically requires considerable domain knowledge to determine the size and step of the window as well as to design the features, which remains a major challenge in general activity recognition [5]. Second, the sliding window methods that perform the feature extraction and recognition pipeline on every window induce a prohibitive computational cost, which is impractical for IoT devices with limited resources. Third, most existing approaches assumed repetitive activities contain fixed periodicity which is unrealistic in many real-world scenarios. For instance, people who suffer from sleep apnea stop and start breathing repeatedly at an irregular rate. At the gym, the time intervals between consecutive movements may change with the levels of energy. The presence of irregular intervals of the repetitive activity invalidates most existing approaches.

In this paper, we focus on monitoring repetitive activities using multivariate time series data derived from IoT devices. The main idea is based on the Successive Similar Pattern (SSP) [4] - the recurring pattern with irregular intervals - generated by the repeating physical motions within the time series. However, the vision of applying SSP for RAM entails the following challenges.

- The SSPs could have variable lengths, shapes, and intervals making it difficult to identify the segments where SSPs occur.
- The start and end positions are ambiguous for recurrent patterns. As a result, identifying each SSP from the time series incur a high computational cost that becomes intractable even for small data.
- The SSPs may have various lengths and could be misaligned, making it difficult to compare the distances between different SSPs.

To address those challenges, we firstly proposed an algorithm called *R-mSIMPAD* to detect time series segments containing SSP and estimate the length of the pattern which is more robust and has fewer assumptions than the existing method [4]. Next, we introduced the concept of the *templates* that is the underlying "true" patterns being repeated with variations, in order to formally define a set of SSPs in the same segment without any assumption regarding the pattern. On this basis, we proposed a *Scalable Template Extraction Method* (STEM) to identify and extract the set of SSPs from the detected segment. Finally, we examined two approaches to classify the detected segments. On one hand, we investigated

Corresponding author: J. Shen

an elastic-measurement-based method to compute the pairwise distance between the extracted template and apply the Nearest Neighbor algorithm for the classification. On the other, we combined STEM with the Empirical Cumulative Distribution Function (ECDF) to extract distributional features from the segments to mitigate the computational cost incurred by the elastic measure.

We conducted extensive experimental evaluations on both public datasets and a synthetic dataset. The results suggest that our approach can efficiently reduce 95% unnecessary computation by ignoring time series that do not contain any repeating patterns. The combination of STEM and ECDF can effectively distinguish among different SSPs achieving an average improvement of 14.26% on synthetic data, and up to 4.4% on the public datasets. We found that STEM can better identify the patterns being repeated and exclude those patterns with abnormalities and variations that lead to a better result. Finally, we study a use case of respiration monitoring based on wireless signals. Without any modification, STEM easily achieved at least 3 times better performance compared to the baseline method. The use case illustrated that the proposed method has great potential as a general method for many other applications.

The main contributions of this paper are as follows:

- We investigated the problem of repetitive activity monitoring and proposed STEM, an efficient and effective method to identify and recognize SSPs from multivariate time series.
- We performed extensive evaluations on both public datasets and synthetic data to validate the performance of the proposed method and achieving on par or even superior performance.
- We demonstrated that the proposed approach is a general method that can be applied to many applications as we illustrated in the two use cases of repetitive activity recognition and respiration rate monitoring.

The rest of this article is organized as follows. Section II summarize the related literature. Section III presents the detailed design and rationale of the method. Section IV illustrate the result of the experiments. Section V discussed some issues that might be uncleared. Finally, we conclude this article in Section VI.

II. Related Work

Repetitive activity monitoring is closely related to activity recognition, which aims to classify different activities using the collected data. Although the previous work does not explicitly target repetitive activities, most of them are studied and evaluated mainly on repetitive activities [2], [6]–[9]. Some studies included recognition of non-repetitive activities but the performance there usually suffers due to the complex nature of the activity [10], making it out of the scope of our work here.

A large body of literature adopted time series classification techniques for activity recognition [11], in which unrealistic assumptions were made. They assumed that the start and end positions of a pattern can be accurately identified and that the lengths are equal for patterns of the same class [12]. Therefore, considerable work adopted the sliding window approach combined with machine learning models to perform activity recognition given its simplicity and robustness. One of the key contributions of the prior work focuses on extracting distinctive features from the data. Popular statistical features such as [13] and distributional features [14] achieved promising results on many activity recognition tasks even compared to state-of-the-art deep-learning-based approaches [10].

There is also existing work that focuses on a particular set of highly repetitive activities. Xia et al. [15] proposed an unsupervised method to recognize assembly work in a factory by finding the motif in the sensor data. [2], [9], [16] investigated the recognition of different gym exercises that are highly repetitive, and count the repetitions of each exercise for performance evaluation. The auto-correlation function that computes the self-similarity at different lags, is the most commonly used approach for repetition counting. The major drawback is that auto-correlation cannot handle repeating patterns with irregular intervals as shown recently in [4]. Although repetitive activity recognition has been widely studied, the existing approaches are either scenario dependant and require extensive domain knowledge to determine many parameter settings or make unrealistic assumptions that are not practical for real-world applications. We aim to propose a general method for repetitive activity monitoring that has barely any pattern assumptions regarding shape and periodicity and is efficient and robust to novel situations.

III. Methodology

In this section, we first introduce the notations and definitions essential to understanding the problem. Then we provide the general problem statement of repetitive activity monitoring. We then present the general idea and rationale of the proposed method and give examples.

A. Definitions

A multivariate time series $\mathcal{T}$ is a sequence of d-dimensional real-valued numbers. A subsequence $\mathcal{T}_{i,l}$ of $\mathcal{T}$ is a continuous subset of the values from $\mathcal{T}$ of length l starting from position i. Formally, $\mathcal{T}_{i,l} = [\mathbf{T}_i, ..., \mathbf{T}_{i+l-1}]$, where $\mathbf{T}_i$ is a d-dimensional vector. A Successive Similar Pattern (SSP) is a more general definition of a repeating pattern, which is a subsequence that occurs consecutively at non-regular intervals in time series [4]. It is defined as a subsequence $\mathcal{T}_{i,l}$ of $\mathcal{T}$ where a *similar* subsequence $\mathcal{T}_{j,l}$ appears within a *nearby range*. The range is a user-defined constraint of the displacement of the SSP and the similarity is defined by the z-normalized Euclidean distance as:

$$D(\mathcal{T}_{i,l}, \mathcal{T}_{j,l}) = \sqrt{\sum_{p=0}^{l}\sum_{k=0}^{d}(\frac{t_{i+p}^{(k)} - \mu_{i,l}^{(k)}}{\sigma_{i,l}^{(k)}} - \frac{t_{j+p}^{(k)} - \mu_{j,l}^{(k)}}{\sigma_{j,l}^{(k)}})^2} \quad (1)$$

where $t_i^{(k)}$ is the value of $\mathbf{T}_i$ at k-th dimension, $\mu_{i,l}^{(k)}$ and $\sigma_{i,l}^{(k)}$ are the mean and standard deviation of $[\mathbf{T}_i^{(k)}, \ldots, \mathbf{T}_{i+l}^{(k)}]$. The pair of $\mathcal{T}_{i,l}$ and $\mathcal{T}_{j,l}$ is considered in the same class if the above condition is satisfied. A segment is a subsequence $\hat{\mathcal{T}}$ of $\mathcal{T}$ contains either none or exactly one class of SSPs. Each segment belongs to either one class in the set of all possible classes $Y = [y_1, \ldots, y_m]$.

B. Problem Statement

Repetitive activity monitoring aims to classify and measure different repetitions of the same physical motion using data collected from IoT devices. In this paper, we focus on the identification and classification of SSPs in multivariate time series. Since the SSPs within a segment is ill-defined, we introduce the concept of templates to help formulate the problem. We assume there is a *template*, a d-dimensional sequence $\mathcal{T}_l$ of length l that is being repeated with variations at non-regular interval within a segment $\hat{\mathcal{T}}$. Given a multivariate time series $\mathcal{T}$, our objective is to find a set of non-overlapping subsequences $\mathbb{S} = \{\mathcal{T}_{i,l}\}$ as SSPs that minimize $\sum_{\mathcal{T}_{i,l} \in \mathbb{S}} D(\mathcal{T}_l, \mathcal{T}_{i,l})$, and predict $\mathbb{S}$ as $y' \in Y$ that minimize the error between y' and y.

C. Method Overview

The proposed method has four major components. Firstly, data is collected with IoT devices and preprocessing is performed on the acquired data. The preprocessing is simply an interpolation of missing data and low-pass filtering with 20Hz as the cutoff frequency, that can be computed in most of the light-weight IoT devices. Then the SSP detection method will find subsequences that contain SSPs and estimates the pattern length. SSP template extraction will then finds the patterns that are being repeated only on the detected segments and therefore significantly reduced unnecessary computation. Finally, SSP identification can be achieved by either matching the extracted template or combining it with existing distributional features.

D. Successive Similar Pattern Extraction

1) SSP Detection: Mining SSPs is computationally expensive as it covers a more general set of repeating patterns without assuming a fixed periodicity of the recurring interval. In this regard, mSIMPAD has been proposed recently for mining SSPs of multiple lengths in time series [4]. It scales linearly to the size of the input series and is robust to novel situations as well as to poor quality data. This method is developed based on the Matrix Profile [17], a method for all-pair-similarity-search across a time series. It modified the original matrix profile by introducing a temporal constraint, namely a Range-constrained Matrix Profile (RCMP). The intuition is that the distances between SSPs are comparatively lower than those of non-SSPs. A set of SSP candidates can then be identified from the RCMP as *valleys*, which is a continuous segment that has a lower distance to some threshold. With the SSP candidates obtained from different target lengths, mSIMPAD chooses a non-overlapping subset of candidates that maximize the sum of depths of the selected valleys as the SSP. This detects SSPs and provides a rough estimation of the pattern length.

The key limitation of mSIMPAD is that it assumes the input series must contain both repeating and non-repeating subsequences, such that it can apply the Otsu method [18] to determine the threshold θ. However, this might not be the case in some scenarios where data contains only repeating segments. Also, the detected segments might simply be rejected or separated by unexpected spikes due to random noise. To overcome these problems, we further improved mSIMPAD by learning the threshold θ and introducing the method for merging time series segments to avoid spikes or false rejections.

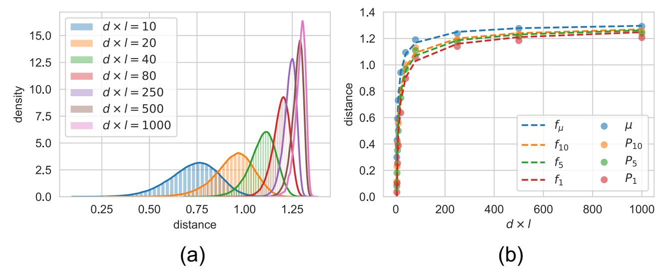

Fig. 1. The effect of sequence dimension and length to the distance of random signals. (a) shows the probability distribution of the distance of random noise with different d x l combinations. (b) shows the dotted lines are the estimation of power-law function and the dots are true values of mean, 10-percentile, 5-percentile, and 1-percentile respectively.

To learn the threshold θ, we study how the random signal contributes to Z-normalized Euclidean distances and determine a threshold that can eliminate most of the random signal. From equation 1, we notice that apart from the signal itself, the distance is attributed to the subsequence length l and dimension d of the series. Adding l by 1 will increase the elements by a factor of d, and adding d by 1 will increase the elements by a factor of l. Therefore, we model the relationship by varying the value of $d \times l$ from a set of candidates from [4, ..., 1000] and choosing d and l arbitrarily. For each candidate, we generate 100 time-series using the approach proposed in [19] and compute the RCMP of the series to estimate the distance of a random signal at different $d \times l$. The result is shown in Figure 1 which reveals that the distance follows the power-law distribution in terms of the rise of the average values, as well as the scale of the variation. The reason is that when there are more elements to compare, it is less likely to find two similar subsequences just by chance. Therefore, the distances will converge if the number of points is large enough. We could estimate the distance of a random signal given different $d \times l$ with the following equation, setting $\alpha = -2.46, k = -0.62; \epsilon = 1.33$:

$$\theta(d \times l) = \alpha(d \times l)^k + \epsilon \tag{2}$$

In this work, we aim at a 95% confidence interval in eliminating random signals by computing the threshold as the estimated mean subtracted by 1.645 times the estimated standard deviation. Then, we could identify regions that contain SSPs by choosing the subsequences where the distance values

TABLE I
PERFORMANCE COMPARISON ON REPEATING PATTERN DETECTION.

	Method	Acc.	Prec.	Rec.	F_1
HAPT	SIMPAD	0.970	**0.994**	0.945	0.968
	mSIMPAD	**0.971**	0.991	0.951	0.970
	R-SIMPAD	**0.971**	0.945	**1.000**	**0.972**
	R-mSIMPAD	0.966	0.935	**1.000**	0.966
mHealth	SIMPAD	0.692	**1.000**	0.579	0.731
	mSIMPAD	0.777	**1.000**	0.696	0.819
	R-SIMPAD	0.891	**1.000**	0.852	0.920
	R-mSIMPAD	**0.915**	**1.000**	**0.884**	**0.938**
PAMAP2	SIMPAD	0.808	**0.994**	0.712	0.829
	mSIMPAD	0.816	0.990	0.729	0.839
	R-SIMPAD	**0.933**	0.970	0.923	**0.946**
	R-mSIMPAD	0.928	0.952	**0.935**	0.943

are less than θ. Data from periods of idle activity may however contain drifts that have lower mutual distances causing false-positive just by chance. To overcome this issue, we modified the mSIMPAD to standardize the input series with 0 mean and standard deviation as 1. Then we insert a Gaussian noise with a scale of 0.1 to mitigate the effect of idle data. On the other hand, to avoid splitting the desired subsequence into smaller parts due to an abnormal spike or valley, we introduced a greedy, iterative merging approach to combine a split with its surrounding subsequences if the length of the split is less than l. It starts from the split with the smallest length and iteratively merging those splits until all of the splits have at least length l. Then the subsequence is verified for containing SSPs by majority voting.

The improved method relaxes the assumption that the input time series must contain both repeating and non-repeating patterns. It also provides more accurate detection results as the inserted Gaussian noise can better differentiate the idle and non-idle components in sensor data. We employed the same evaluation metric in [4] and we further discard irrelevant data such as subsequences labeled as a transition since they may contain any activities including repetitive activity (e.g. walking to another location) during the transition phase. The improved methods are denoted as R-SIMPAD and R-mSIMPAD respectively, that are more robust and have fewer assumptions on the input signal compared to their original forms. The results are reported in Table I. We notice that our algorithm performs similarly to existing work on the HAPT dataset but significant improvements can be observed on both the mHealth and PAMAP2 datasets. The number of false negatives has been drastically reduced as we can see from the much higher recall rate, resulting in an over 11% improvement in the F_1 score.

2) Scalable Template Extraction Method (STEM): With the above-given method, we identified subsequences that contain SSPs and provided a rough estimation of pattern lengths. Then for each subsequence, we compute the distance profile DP_i, and select the nearest neighbor $T(j, l)$ iteratively if the distance is less than θ. The distances from $j - pr$ to $j + pr$ are discarded for the selected nearest neighbor $T(j, l)$ with a pruning range $pr = \gamma \times l$, in which γ is the pruning

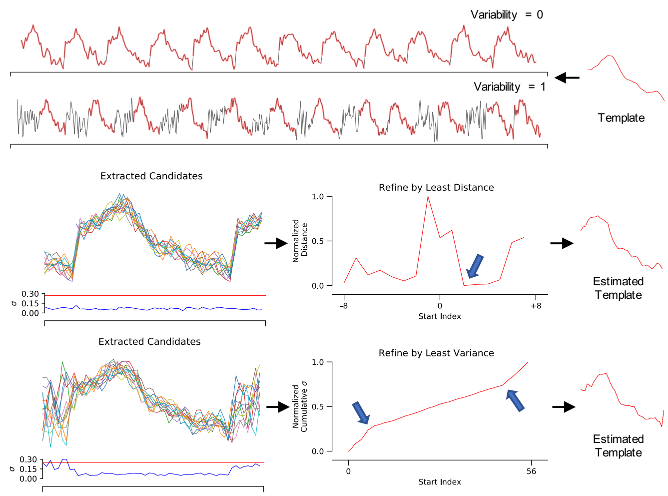

Fig. 2. Example of a repeating pattern with different variability and the feature extraction and refinement procedure. It finds a set of SSP candidates, then estimates the point-wise variation among them. It then refines the start and end position by either the least variance or the least distance, based on the maximum point-wise variation that is smaller or larger than some threshold σ accordingly.

factor of the pattern length. This process will repeat until all of the distances are discarded or are greater than θ. Then we estimate the quality of the match by averaging all of the distances of the chosen subsequences that are denoted as *template candidates* and select the set of candidates with the minimum average distances as the best match of the SSP. However, the estimated length of the pattern is relatively rough, which may over or underestimate the true length of the pattern. We proposed a two-step approach to better refines the length of the extracted template. The SSP should either be continuous or vary at different intervals. We could evaluate the variation of the positions around the start and end points of the candidates. If the SSP is continuous, we could refine the template by minimizing the distance between the start and end positions. The intuition is that the pattern should appear one after another such that the variation should remain relatively low as we can see from Figure 2; otherwise, the variation should be relatively high such that we can refine the length by minimizing the averaged variations. From the experiment, we notice a huge improvement in terms of recognition accuracy with this template refinement approach as we will show in section IV.

The recognition is then performed by comparing the templates. One can imagine that for continuous patterns, it is difficult to determine the start and end positions. The extracted template may be misaligned where the start and end positions lie around the middle. When comparing two templates, we align them by padding one by itself to cover all possible extracted cycles and compute the cross-correlation between the padded template with the other. Then, we roll the template by maximizing the cross-correlation.

On the other hand, the template generated from one sequence might be slightly different from another in terms of

Algorithm 1 Scalable Template Extraction Method

Require: T, int l, double θ, double γ=1
Ensure: TP
$C_{best} \leftarrow$ [];
for $i \leftarrow 1 : n - l + 1$ **do**
  $DP \leftarrow$ computeDP($T_{i,l}, T$);
  $C \leftarrow$ findCandidates(DP);
  if $D(C) < D(C_{best}$ **then**
    $C_{best} \leftarrow C$;
  end if
end for
$\sigma_C \leftarrow$ pointwiseSTD(C_{best});
if $\sigma_C < std(T)$ **then**
  $C_{best} \leftarrow$ refineByNearestPoint(C_{best});
else
  $C_{best} \leftarrow$ refineByVariability(C_{best});
end if
$TP \leftarrow$ median(C_{best});
return TP;

length and shape. We apply an elastic distance metric to handle these kinds of small differences. Dynamic Time Wrapping (DTW) has been the most widely used measure for time series. However, [20] suggested that the Time Warp Edit Distance (TWED) consistently achieves the best performance in their study. Compared to other distances like DTW, the TWED is a metric that can potentially speed up computation such as clustering and retrieval. To reduce the computational cost of TWED, we adopted the window constraint as discussed in [21] to the TWED to limit the maximum warping of the TWED.

The above-mentioned method, denoted as STEM-TWED, aims to identify the subsequences which minimize the internal distance as a representation and recognize the subsequence by comparing the TWED with the labeled templates. It can distinguish tiny differences among time series which is especially suitable for differentiating fairly similar, low dimensional series. However, it relies on accurate length estimation as the differences in lengths between time series incur higher costs. mSIMPAD only offers a rough estimation of the pattern length, and it depends on the input of the length candidates. Moreover, time series distance measures such as DTW and TWED are computationally expensive. Therefore, we introduce a variation of STEM we call STEM-ECDF that incorporates the existing feature extraction method, namely the Empirical Cumulative Distribution Function (ECDF) which is simple, yet very robust even compared with state-of-the-art deep-learning-based features [10].

We employ the same detection and candidates extraction approach as mentioned above. Instead of using a single template as a representation, we compute the ECDF of all the template candidates as a representation. Each detected subsequence will then be represented by the ECDF vector that preserves the distributional information of the template. Since the ECDF features have the same number of dimensions, we can leverage traditional machine learning models for recognizing the template. This allows recognition with much lower computational cost, while still being capable of handling repeating patterns with irregular intervals. We delayed the discussion on the merit of this approach until section V.

IV. Experimental Evaluation

In this section, we report the experimental evaluation of the proposed approach to SSP recognition and compare it to other activity recognition methods. One synthetic dataset and three sensor-based activity datasets were used for the evaluation. With the ground truth being available in the synthetic data, we specifically measured the performance of template extraction from three aspects: pattern length estimation, template candidate selection, and the similarity between the extracted template with the ground truth pattern. We then measure the performance of activity recognition on the three public datasets to illustrate the robustness of real-world applications. Finally, a use case of wireless-sensing-based respiration monitoring is provided to demonstrate that the proposed method as a general approach has a broad range of applications with great impact.

A. Experiment Setup

The methods for the comparison including: 1) one of the most widely used statistical features [13]; 2) the Empirical Cumulative Distribution Function (ECDF) [14] that preserve the underlying distributions with a fixed set of real-valued coefficients; 3) the DeepConvLSTM [22] which is an end-to-end deep learning method combining the convolutional and recurrent layers together with the attention mechanism in long-short term memory (LSTM) network [23]. It is able to learn a representation while optimizing the classifier, showing remarkable performance in multiple open datasets, and therefore is widely used as a baseline method for human activity recognition [10]. mSIMPAD and STEM perform directly on the input series, while the other methods are sliding window based. To provide a fair comparison, we divide the recognition result into equal-length segments, which is done in the other methods. Then we perform majority voting within each segment to decide whether it contains an SSP and choose the extracted template accordingly. To mitigate the effect of recognition model parameters, we apply the Nearest Neighbor (NN) classifier for both the STEM and other methods except DeepConvLSTM.

1) Evaluation Metric: We adopted the weighted F_1 score as the evaluation metrics defined as $\frac{1}{|X|}(\sum_{i \in C} 2|X_i| \times \frac{precision_i \times recall_i}{precision_i + recall_i})$ where C is the set of given classes, $|X|$ is the number of all testing instances, $|X_i|$, $precision_i$ and $recall_i$ refer to the number of testing instances, precision, and recall of a particular class i respectively. The $precision_i$ is defined as $\frac{TP_i}{TP_i + FP_i}$ and the $recall_i$ is defined as $\frac{TP_i}{TP_i + FN_i}$, where TP_i, FP_i, and FN_i refer to the true positive, false positive, and false negative of a particular class i respectively. The accuracy is also used for evaluation defined as $\sum_{i \in C} \frac{TP_i}{|X|}$.

2) Synthetic Data: To get a better grasp of the performance of the proposed approach, we need a dataset that we have full control over, including knowledge of the shape of the pattern, the number of repetitions, and the interval variability. Therefore, synthetic data is required for evaluation purposes, as well as to help determine the proper set of parameters for

TABLE II

LIST OF ACTIVITIES FOR EACH DATASET, IN WHICH THE ACTIVITY IDS ARE A0: NON-REPETITIVE ACTIVITIES, A1: WALKING, A2: WALKING UPSTAIRS, A3: WALKING DOWNSTAIRS, A4: JOGGING, A5: RUNNING, A6: CYCLING, A7: NORDIC WALKING.

	Repetitive Activities	Non-repetitive Activities
HAPT	A1, A2, A3	sitting, standing, lying
MHEALTH	A1, A2, A4, A5, A6	sitting, standing, lying
PAMAP2	A1, A2, A3, A5, A6, A7	sitting, standing, lying, watching TV, computer work

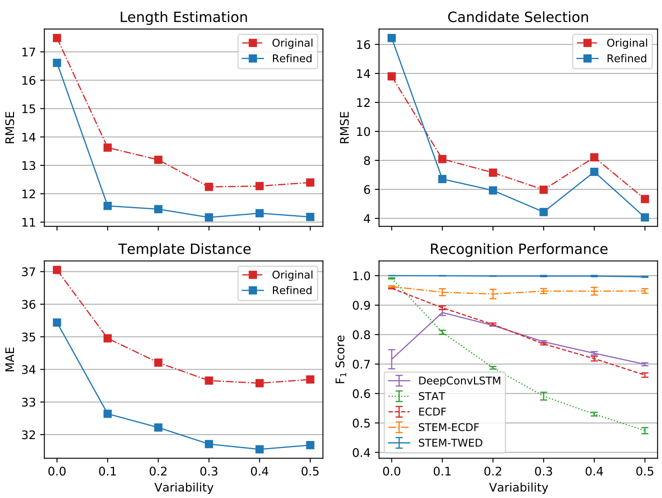

Fig. 3. Evaluation result on the synthetic dataset. It shows better performance for the refined template over the original template both in terms of length estimation, candidate selection, and template extraction. The bottom right figure shows that STEM-based methods consistently achieve better performance than other methods in activity recognition.

the algorithm itself. We first randomly generated 19 random walk time series with lengths between [40,...,80]. The 19 time series are treated as templates, and for each template, we generate 100 time-series repeating the template from 5 to 20 times randomly. We also generated 100 random time series with no repeating patterns as one negative class. Gaussian noise was then added to each of the generated series, resulting in 2000 synthetic time series.

We produced six datasets following the above-mentioned procedures with the same 19 templates, in which we introduce different variable intervals between [0,0.1,...,0.5]. The variable interval v is a factor of the pattern length l, which determines the standard deviation of the interval equal to $v \times l$ and the interval between patterns following the normal distribution. Figure 2 shows an example where variability equals 0 means the templates appear one after another, and variability equals 1 means the templates appear at varying intervals in which the variation follows a normal distribution with standard deviation as 1 $\times$ the pattern length.

3) Public Datasets: We choose three publicly available activity datasets for the evaluation since these datasets contain relatively more repetitive activities. The HAPT [24] collects data from 30 volunteers wearing a waist-mounted smartphone while performing various activates in laboratory conditions: walking, walking upstairs, walking downstairs, sitting, standing, and Lying down. MHEALTH [25] is composed of 12 activities in an out-of-lab environment, performed by 10 volunteers with 3 sensors placed on the subject's chest, right wrist, and left ankle. PAMAP2 [26] includes 18 activities performed by 9 subjects wearing 3 sensors on the subject's chest, and the dominant side's hand and ankle. The PAMAP2 is a commonly used public dataset in activity recognition, and the result of recently proposed methods can be found in [10].

We manually classify activities as repetitive activities and non-repetitive activities for each dataset, where non-repetitive activities are treated as one class. The details of the classification are mentioned in Table II. We choose only two IMUs (one from the hand and one from the ankle) from the MHEALTH and PAMAP2 datasets, as the chest data does not contribute much information on the listed activities. For each of the detected segments, we extract the template (denoted as STEM-TWED) and the ECDF features (denoted as STEM-ECDF) from the template candidates as mentioned in section III. Then the sliding window is applied to extract statistical and ECDF features directly from the window data. The STEM-TWED and STEM-ECDF features are also selected on the same window by majority voting. The size of the window is defined as 5 seconds with a step size of 2.5 seconds.

B. Evaluation of Repetitive Activity Recognition

1) Synthetic Data: We assess the quality of the extracted template by evaluating its performance on *length estimation*, *candidate selection*, and *template extraction* measured by the similarity between the template and the ground truth. The results can be found in Figure 3, and detailed evaluations are outlined in the following section.

Length estimation is evaluated by measuring the Root Mean Squared Error (RMSE) between the length of the template with the ground truth, where the quality of selected candidates is measured by the location differences. The location is defined as the center point of the pattern to mitigate the effects of the length variations. Noticing that the number of selected candidates might be different from the ground truth number, we match the largest common indices with a greedy approach. The most important measure of the extracted template is how accurately the approach recovers the underlying pattern from the candidates. It is measured by the distance with TWED between the template and the ground truth after the two sequences are aligned.

The result shows that in all measures, RMSE decreases with the increasing variability in general. Also, the refined template recorded consistently lower RMSE than the original template. It suggests that the proposed template extraction method is extremely robust when the patterns contain variable intervals, while still performing well for patterns with regular intervals. Nonetheless, the template refinement is necessary as it provides more accurate length estimation and candidate selection which leads to a more precise template extraction, as

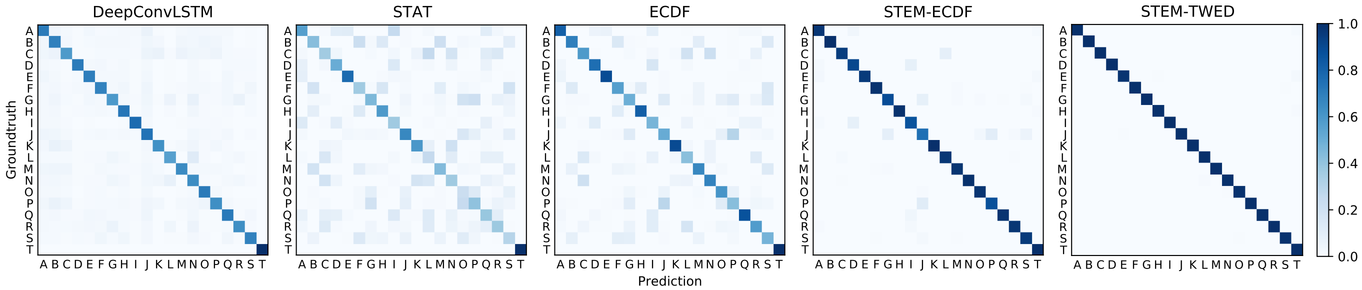

Fig. 4. Confusion matrix showing the average performance of different methods over all the synthetic datasets. The alphabets from A to S represent different patterns as a class, where T denotes the class without any repeating pattern. The values are normalized by each class rangeing from 0 to 1.

TABLE III
COMPARISON OF RECOGNITION PERFORMANCE FOR DIFFERENT METHODS ON THE PUBLIC DATASETS.

Dataset	Features	Acc.	Prec.	Rec.	F_1
HAPT	DeepConvLSTM	0.983	0.983	0.983	0.983
	STAT	0.951	0.952	0.951	0.952
	ECDF	**0.995**	**0.995**	**0.995**	**0.995**
	STEM-TWED	0.986	0.987	0.986	0.986
	STEM-ECDF	0.990	0.990	0.990	0.990
mHealth	DeepConvLSTM	0.830	0.882	0.830	0.855
	STAT	0.894	0.906	0.894	0.900
	ECDF	0.914	0.929	0.914	0.921
	STEM-TWED	0.764	0.786	0.764	0.773
	STEM-ECDF	**0.950**	**0.958**	**0.950**	**0.954**
PAMAP2	DeepConvLSTM	0.704	0.745	0.704	0.724
	STAT	0.853	0.855	0.853	0.854
	ECDF	0.848	0.865	0.848	0.856
	STEM-TWED	0.700	0.755	0.700	0.726
	STEM-ECDF	**0.885**	**0.916**	**0.885**	**0.900**

shown by the much lower distance compared with the original template.

The analysis above demonstrated how the proposed method performs on template extraction under different variability conditions of the repeating pattern. We then examine the performance of recognizing the pattern with the extracted templates. For each dataset of different variability, we perform 5-fold cross-validation to divide the dataset into 5 equal size partitions randomly. We then perform evaluations using each partition as the testing set and the remaining partitions as the training set. The performance is then calculated as the average over the 5 partitions as shown in Figure 4.

As expected, the ECDF and DeepConvLSTM methods achieved similar performance. While the proposed STEM-based methods achieved superior performance compared to all the other methods. When the variability is up to 0.5, both methods recorded at least 20% improvement comparing with the baseline method. The average improvement from ECDF to STEM-ECDF is 14.26% over all the synthetic datasets.

2) Public Datasets: We evaluate the performance of repetitive activity recognition from two aspects: *efficiency* and *classification accuracy*. Efficiency is evaluated by measuring the degree to which unnecessary computations are avoided. The unnecessary computations are considered non-repetitive activities. On the contrary, repetitive activities are the target that should be included in the computation. Therefore, successfully classifying all the subsequences of both the repetitive and non-repetitive activities indicates the efficiency of the method. We recorded an average true positive rate of repetitive activity classification of 92.4% and 93.9% respectively for SIMPAD and mSIMPAD over the three datasets; and 96.6% and 95% for non-repetitive activity classification. This suggests that the mSIMPAD is better at identifying repetitive activities at 93.9% while eliminating 95% of the non-repetitive segments.

For the classification accuracy, we perform 3-fold cross-validation for each of the public datasets, since some of the classes only have a few samples. The results are reported in Table III. Surprisingly, the DeepConvLSTM recorded a relatively lower performance than the other methods. The major reason would be the number of channels available in the data is limited. There were at most 24 channels of the input data while the previous study employed 113 channels as input [22]. While the other two methods namely ECDF and STAT produce similar results where the ECDF features perform slightly better in general. Note that the STEM-TWED recorded the worst performance, showing that the individual differences can largely degrade the recognition performance with an elastic distance measure approach. This implies comparing the raw pattern might not provide as good performance as the other feature-based approaches.

With the abstraction of the pattern using statistical features, STEM-ECDF achieved the best performance in two out of three datasets. It obtained similar but slightly lower results to the ECDF with just 0.5% margin only for the simplest dataset, whereas the better performance on the other two datasets is at least 3% higher than others. STEM-ECDF obtaining better results since it identifies internally similar patterns and ignores abnormalities to form better quality features out of the repetitive activity. As we have illustrated on the synthetic dataset, STEM-TWED is more suitable for repetitive activity monitoring where the patterns are very similar in most cases. We delay the discussion in choosing the STEM-ECDF and STEM-TWED, and the reason that STEM-ECDF outperforms the original ECDF in section V. In general, the proposed approach achieved superior performance compared to the other methods, recording up to 4.4% improvement for the STEM-ECDF.

The above evaluation suggests that for repeating patterns with irregular intervals, the proposed approach performs significantly better than the other methods. However, measuring the distance between two sequences with time-warping techniques is known to be computationally expensive. The superior performance comes with a trade-off of computational efficiency. By combining the template extraction with the distributional features, the efficiency can be largely improved while achieving on par or even better results on repetitive activity recognition in reality.

C. Use case: Respiration Monitoring

Wireless sensing is an emerging area in the IoT community. Numerous publications have shown the potential of wireless-sensing based vital sign detection, which enables various applications in healthcare as well as activity recognition. To illustrate the potential of the proposed method on other repeating pattern extraction problems, we adopted respiration monitoring using wireless signals as a use case. Specifically, we identify the repeating patterns within a wireless signal captured from an RFID transceiver to estimate the respiration rate of a subject with an RFID tag on their chest. We randomly selected 10 signals collected in [27] in which half of them contain normal breathing, and the other half contain periods where the participants were instructed to hold their breath to simulate the condition of Sleep Apnea.

Respiration can be identified by measuring the phase of the wireless signal [28]. Intuitively, the physical motion of the chest affects the signal strength of the tag as it expands and contracts while inhaling and exhaling. These miniature changes constitute periodic patterns in the phase values of the signal. These patterns can be detected by STEM and provide the estimated pattern length if it exists. The located template candidates are considered as a signal of breathing in which we can estimate the breathing rate by counting the number of candidates. We compare the performance with the baseline method introduced in [27]. It assumed that the breathing pattern is a simple waveform signal that can be identified by peak detection. It computes a threshold, determined by the mean of the phase value to eliminate false positives caused by small variations. A normal breathing rate is roughly 30 times per minute and is considered as a physical limitation. This avoids peaks that are closer than 2 seconds apart. Then, we compute the number of breaths within each signal and calculate the RMSE for each of the methods. The RMSE of STEM is 0.89, where the RMSE of the baseline method is 3.66, which shows that STEM achieved much better performance compared to the baseline method.

Figure 5 shows an example of normal breathing and simulated sleep apnea by holding one's breath. As we can see, STEM can accurately identify the waveform generated by breathing even with irregular intervals, periods of pause, and shape variations. In contrast, although the baseline method achieved similar results on normal breathing datasets, it fails

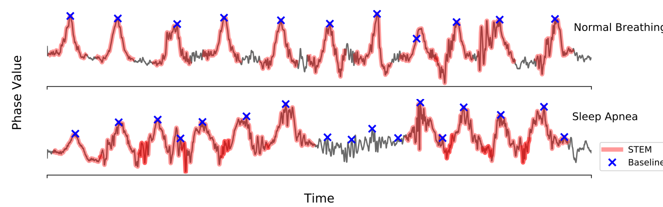

Fig. 5. Example of respiration estimation. The red line denotes the detected pattern using STEM and the blue cross marks denote the detected respiration using the baseline method.

if the signal contains periods of pause. The minimum distance between peaks simply does not work when the pattern contains irregular intervals as the breathing rate might vary from the average value. The drift of the signal can invalidate the threshold approach. In contrast, STEM first detects if the repeating pattern occurs by comparing the z-normalized distance between the subsequences, which can better handle data drift and shape variation. The inserted noise can better differentiate the truly repeating patterns from the non-repeating subsequences. Therefore, STEM achieved much better performance in estimating the breathing rate from the wireless signal. Note that this is just an example application of our approach. It can also facilitate many other applications such as heart rate detection, blink rate detection, and many repetitive motions such as hand-flapping, rocking, spinning, just to name a few.

V. Discussion

In this work, we focus on the classification of multivariate time series that may contain repeating patterns. These kinds of time series are prevalent in day to day life, and are especially interesting when considering repetitive activities [1]–[3], [15], [16], physiological signals [28], or the audio signals of music [29], just to name a few. We focus on the application of repetitive activities given their importance for physical health monitoring. The presented method is however general enough for other time series classification tasks with repeating patterns, as the proposed method is scenario independent where the only required parameter is the length of the target pattern.

Based on STEM, we proposed a recognition method using the nearest neighbor algorithm with time warp edit distance namely the STEM-TWED. It shows an ability for accurate recognition that however has a few drawbacks. First, the TWED relies on accurate length estimation as the differences between lengths incur a higher distance due to the warping penalty. Second, it is non-trivial to design a proper penalty for general template matching. Third, the computational cost increases with the number of training samples due to the distance measure for each sample, and calculating the TWED is much slower than calculating the Euclidean distance for the traditional features. To balance the recognition accuracy with the computational cost, we combine the STEM with the Empirical-Cumulative-Distribution-Function based features namely the STEM-ECDF. The advantage of the STEM-ECDF is that it can avoid unnecessary computation on non-repetitive series, while efficiently extracting template candidates and ignoring noisy data within the subsequences.

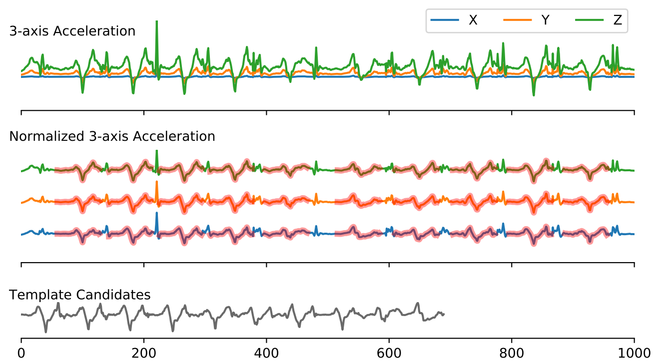

Fig. 6. Example of candidate selection in a snippet of mHealth dataset.

Figure 6 shows a snippet of data from the mHealth dataset, obtained from the accelerometer on a person's ankle while climbing stairs. From the raw accelerometer signal, the X-axis is almost flat compared to the Z- and Y-axis data. If we normalize the data on each axis, similar patterns can be observed among the three axes. Although there are slight differences between each repetition, we effectively extract internally similar subsequences using STEM as the subsequences highlighted. We only show the candidates extracted from the X-axis since the patterns over three axes are identical.

VI. Conclusion

In this work, we proposed STEM, a template extraction method to identify repeating patterns in multivariate time series and apply it to repetitive activity identification. STEM aims to substitute the commonly used sliding window technique, which is computationally expensive and usually requires extensive domain knowledge to work. STEM leverages the recently proposed successive similar pattern detection method to determine whether repeating patterns occur within a time series. For these detected subsequences, it identifies a template, which is a pattern that minimizes the internal distances within the subsequence as a representation. We evaluated our approach on synthetic data, as well as on three publicly available datasets. The experiment shows that the proposed method can efficiently avoid unnecessary computation on non-repeating series. It also provides more accurate recognition results especially when the periodicity of the repeating pattern is variable. By combining the STEM with the distributional feature, it achieved a more balanced trade-off between computational cost and recognition accuracy. The proposed method shows superior performance compared to the baseline methods on repetitive activity recognition, and can additionally be applied to other time series classification tasks with repeating patterns.

Acknowledgment

This work was supported by the HK RGC General Research Fund under Grant Number PolyU 15220020.

References

[1] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *in Proc. of the ACM UbiComp*, 2013.
[2] X. Guo, J. Liu, and Y. Chen, "Fitcoach: Virtual fitness coach empowered by wearable mobile devices," in *in Proc. of the IEEE INFOCOM*, 2017.
[3] T. Maekawa, D. Nakai *et al.*, "Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory," in *in Proc. of the ACM UbiComp*, 2016.
[4] C.-T. Li, J. Cao *et al.*, "msimpad: Efficient and robust mining of successive similar patterns of multiple lengths in time series," *ACM Transactions on Computing for Healthcare*, vol. 1, 2020.
[5] H. Li, G. D. Abowd, and T. Plötz, "On specialized window lengths and detector based human activity recognition," in *in Proc. of the ACM ISWC*, 2018.
[6] K. Chen, L. Yao *et al.*, "Multi-agent attentional activity recognition," in *in Proc. of the IJCAI*, 2019.
[7] W. Lu, F. Fan *et al.*, "Wearable computing for internet of things: A discriminant approach for human activity recognition," *IEEE Internet of Things Journal*, vol. 6.
[8] L. Yao, F. Nie *et al.*, "Learning from less for better: semi-supervised activity recognition via shared structure discovery," in *in Proc. of the ACM UbiComp*, 2016.
[9] D. Morris, T. S. Saponas *et al.*, "Recofit: using a wearable sensor to find, recognize, and count repetitive exercises," in *in Proc. of the ACM CHI*, 2014.
[10] H. Haresamudram, D. V. Anderson, and T. Plötz, "On the role of features in human activity recognition," in *in Proc. of the ACM ISWC*, 2019.
[11] Z. S. Abdallah, M. M. Gaber *et al.*, "Activity recognition with evolving data streams: A review," 2018.
[12] B. Hu, Y. Chen, and E. Keogh, "Time series classification under more realistic assumptions," in *in Proc. of the SIAM ICDM*, 2013.
[13] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Systems with Applications*, vol. 41.
[14] N. Y. Hammerla, R. Kirkham *et al.*, "On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution," in *in Proc. of the ACM ISWC*, 2013.
[15] X. Qingxin, A. Wada *et al.*, "Unsupervised factory activity recognition with wearable sensors using process instruction information," vol. 3.
[16] L. Xie, X. Dong *et al.*, "Meta-activity recognition: A wearable approach for logic cognition-based activity sensing," in *in Proc. of the IEEE INFOCOM*, 2017.
[17] C.-C. M. Yeh, Y. Zhu *et al.*, "Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets," in *in Proc. of the IEEE ICDM*, 2016.
[18] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, 1979.
[19] M. W. Kadous, "Learning comprehensible descriptions of multivariate time series." in *in Proc. of the ICML*.
[20] P.-F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31.
[21] A. Mueen and E. Keogh, "Extracting optimal performance from dynamic time warping," in *in Proc. of the ACM SIGKDD*, 2016.
[22] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16.
[23] V. S. Murahari and T. Plötz, "On attention models for human activity recognition," in *in Proc. of the ACM ISWC*, 2018.
[24] D. Anguita, A. Ghio *et al.*, "A public domain dataset for human activity recognition using smartphones." in *in Proc. of the ESANN*, 2013.
[25] O. Banos, R. Garcia *et al.*, "mhealthdroid: a novel framework for agile development of mobile health applications," in *in Proc. of the IWAAL*, 2014.
[26] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *in Proc. of the IEEE ISWC*, 2012.
[27] Y. Yang and J. Cao, "Robust rfid-based respiration monitoring in dynamic environments," in *in Proc. of the IEEE SECON*, 2020.
[28] X. Liu, J. Cao *et al.*, "Contactless respiration monitoring via off-the-shelf wifi devices," *IEEE Transactions on Mobile Computing*, vol. 15.
[29] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (repet): A simple method for music/voice separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21.