A Transferable Technique for Detecting and Localising Segments of Repeating Patterns in Time series

1st Mahtab Mirmomeni
The University of Melbourne
m.mirmomeni@student.unimelb.edu.au

2nd Lars Kulik
The University of Melbourne
lkulik@unimelb.edu.au

3rd James Bailey
The University of Melbourne
baileyj@unimelb.edu.au

***Abstract*—In time series data, consecutively repeated patterns occur in many applications, including activity recognition from wearable sensors. Repeating patterns may vary over time and present in various shapes and sizes, which makes their detection a challenging problem. We develop a novel technique, RP-Mask, that can detect and localise segments of consecutively repeated patterns, without prior knowledge about the shape and length of the repeats. Our technique represents time series using recurrence plots (RP), a method for visualising repetition in time series. We identify two key features of recurrence plots—checkerboard patterns and vertical/horizontal lines marking the start and end of checkerboard patterns. We use object recognition on RP images to detect and localise the checkerboard patterns, which are mapped to the segments of consecutively repeating patterns on the underlying time series. Since the collection and labeling of a real world dataset that exhibits all possible variations of a repetition is prohibitive, we demonstrate that our model is able to effectively learn from synthetically curated data and perform equally effective on a real world dataset, while it is noise tolerant. We compare our method to a number of state-of-the-art techniques and show that our method outperforms the state of the art both when trained using real activity recognition and synthetic data.**

***Index Terms*—Time series Segmentation, Deep learning, Object detection, Recurrence plots, Human activity recognition**

I. Introduction

Time-series data is being generated in many domains, including data generated by sensors. In many applications, such as detecting repetitive activities using wearable sensors, certain segments of a time series exhibit a periodic behaviour, a pattern being repeated consecutively over time. The repeats in these segments may vary over time and have various shapes and length. Deep learning approaches are emerging techniques for detecting such segments, where a labelled datasets is required for training [11], [32]. However, data collection and labelling in such applications is a lengthy, expensive and sometimes impossible process [16], [24] and thus methods that can transfer their learning to unseen datasets are required.

Fig 1 shows an example of time series data that is recorded by an accelerometer and collected from a patient performing rehabilitation exercises. The segments of the time series containing repeating patterns (shown using colored dotted lines) refer to exercises being performed by the patient, which are of the form of a pattern being repeated consecutively over time. The areas between the exercises indicate other activities or breaks between the exercises.

The task of detecting segments of repeating patterns in human activity tracking is difficult due to three main challenges [3]:

- Repeating patterns have varying shapes, lengths, and frequencies.
- Repeating patterns may be surrounded by other repeats and/or noise.
- The shape, frequency and length of repeating patterns may change over time.

As an example in Fig 1 the repeats differ from one exercise type (Exec1) to another (Exec2). The repeating patterns of an exercise can change during performance of an exercise, due to factors such as a patient's limitations in movement ability or a patient getting tired during the exercise. An example of such scenario has been given in Fig 1 in exercise number 2 (Exec 2). As seen in this figure, the repeating pattern changes over time. The repeating pattern highlighted with yellow color (segment 1) is similar to the pink repeating pattern (segment 2) adjacent to it, but the orange repeat (segment 3), which has happened further in time, has a different amplitude compared

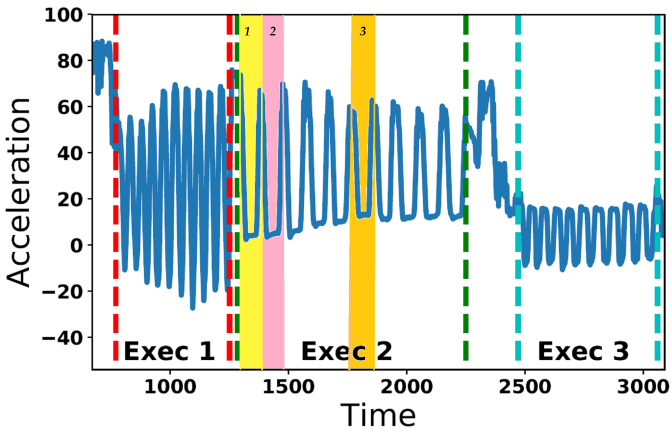


Fig. 1: A time series captured from accelerometer tracking exercises, where 3 different exercises have been performed by an individual.

to the yellow (segment 1) and pink repeats (segment 2).

The state of the art approach for automatically detecting and localising segments of repeating patterns is to use unsupervised or supervised techniques. Unsupervised methods search for a characteristic within the dataset to segment the time series. Supervised techniques use machine learning/deep learning and train a model that learns the properties of a segment of repetitive patterns [11]. These approaches require a collected, labeled dataset for training. However, collecting and labelling real data with such properties, such as a rich diverse dataset that represents the many different movement patterns of patients, is difficult, lengthy and expensive [24]. Thus methods that can learn from synthetically curated data and perform equally as effective on a real dataset are desirable.

We propose a novel transferable technique, RP-Mask, that can learn to detect and localise segments of consecutive repeating patterns. We demonstrate that the problem of detecting and localising segments of repeating patterns can be mapped to an object recognition activity in images. RP-Mask leverages recurrence plots (RPs) [7], a method that creates an image representation of the time series, capturing recurrence in the time series. We show that the periodic segments reveal themselves as blocks located on the diagonal of the RP. We can then analyse the behaviour of this diagonal by use of a well established object detection/localisation algorithm, Mask RCNN [14], a deep learning approach. This algorithm allows us to detect and localise patterns in the RP, corresponding to consecutive repeating patterns in the time series.

Our technique, RP-Mask, successfully detects and localises segments of consecutive repeating patterns from real data using a model trained entirely on a synthetic dataset. We compare RP-Mask with both state-of-the-art techniques that use deep learning and traditional unsupervised time series mining techniques. We show that RP-Mask outperforms the state of the art methods when learned from real data and tested on a unseen real data as well as when learned using a curated synthetic data and tested on an unseen real dataset. We further demonstrate that our technique is noise-tolerant.

II. Related work

In the literature, deep learning approaches and classical time series data mining approaches have been applied to detect repeating patterns from a time series.

Unsupervised time series segmentation techniques that divide the time series into intervals have been used for detecting and localising patterns from time series data [17], [22]. Zhao and Itti [38] introduced the TS-Decompose algorithm, which splits the time series using information gain. Kowsar et al. [16] introduced the OToR algorithm. OToR is designed as an online real time method for tracking short bursts of repeating patterns. The authors prove that periodic points of the auto-correlation from a segment of repeating patterns are periodic and linear. They used these two characteristics to design a windowing algorithm that tracks repetitions in a segment of repeating patterns in real-time.

A class of unsupervised algorithms that have been used for detecting repeating patterns in time series are motif discovery algorithms. Motifs are previously unknown subsequences that have been repeated in the time series [4], [30]. Matrix profile (MP) [34], [35] is the most recent time series pattern mining technique that can be used for motif discovery. The FLOSS algorithm [9] uses Matrix profile for semantic segmentation of time series data and has been used for activity recognition from sensor data.

Supervised methods that learn what the repeating patterns look like by defining a number of features extracted from the training dataset have been used for activity recognition. Some of the early approaches include Decision Trees [1], [18], and Nearest Neighbor [31].

Another class of supervised methods are Deep learning techniques that have been investigated for time series classification and prediction [2], [6], [15], [19], [25], [32], [33], [36]. In these methods a time series is fed to a Deep Neural Network (DNN) to either predict the future values or classify the time series. A common first pre-processing steps among these techniques is to use a sliding/tumbling window that splits the time series. However, as reported by Xia et al. [32] keeping the homogeneity of the data in these splits is an important factor in the performance of the DNN. A challenge for these methods is to find the correct size of sliding window for their task. In this regards, our work can act as the first step to correctly segments the time series in to homogeneous segments that can be fed to these DNN techniques for later classification/prediction processing.

Hammerla et al. [11] have evaluated various deep learning techniques for human activity recognition and conclude that for activities of short duration, (such as rehabilitation exercises performed by patients), Bi-directional LSTM (Bi-LSTM) is the preferable method. In this paper we use Bi-LSTM as one of our baseline methods.

Recurrence plots [7] have been used as an image representation for the time series by a number of authors [8], [12], [13], [28], [37]. These papers have focused on time series classification by using a DNN to classify the recurrence plot images. However, they do not localise where on the recurrence plots specific patterns of interest, such as repetitive activities, occur, which is the focus of this paper.

III. Problem Statement

We formulate our goal as localising segments of consecutive patterns in a time series. Assume we have a series generated from sampling a continuous function in chronological order. We receive a series $Y = \{y_1, ...y_n\}$ of n samples (a.k.a. time series) where each y_i is a real valued number associated with a read from the sampled function at time step i.

We want to define whether in the given time series Y, there exists sub-series $P = \{p_0, p_1, ...p_i, ...p_m\}$ where $p_i \subset Y$ are consecutive samples from a periodic function S with period s, i.e., $p_i = \{y_{c_i}, y_{c_i+1}, ..., y_{c_i+s}\}$ where $1 \leq c_i \leq n-s$, such that $\forall i | 0 \leq i \leq m-1, \textit{Sim}(p_i, p_{i+1}) \leq \theta$. *Sim* is a similarity

function that measures if two patterns p_i and p_{i+1} are similar within some bound θ.

The length, shape and period of patterns are unknown and may vary over time, which makes detecting the patterns a difficult problem. An example of how the shape of the repeats might change over time has been given in Fig 1 Exec 2. Our goal is to design a method that learns a model from a training dataset, and can be used on various test sets, possibly having differing distributional characteristics.

IV. Introducing RP-Mask

In order to automatically detect and localise segments of previously unknown repeating patterns from a time series with various shape, length and period, we need to design a method that is able to detect and localise recurrence in the time series and is robust to the changes in the underlying repeating patterns. We transform the time series to a RP [7] presentation, which creates a 2D image representation of the time series containing information about where the recurrence in the time series has occurred (Section IV-A). We demonstrate RP is robust to the changes in the repeating signal. We detect and localise patterns of recurrence from this image representation using object detection techniques (Section IV-D).

A. Recurrence plot(RP):

Eckmann et al. [7] designed recurrence plot (RP) as a way to visualise the periodic nature of a time series. Given a time series of length n ($Y = \{y_1, ...y_n\}$), the RP that represents Y is a 2-dimensional squared image generated from a $n * n$ matrix (D_{n*n}). In the original RP the value of the pixels at the position (i, j), d_{ij}, is defined by the following formula:

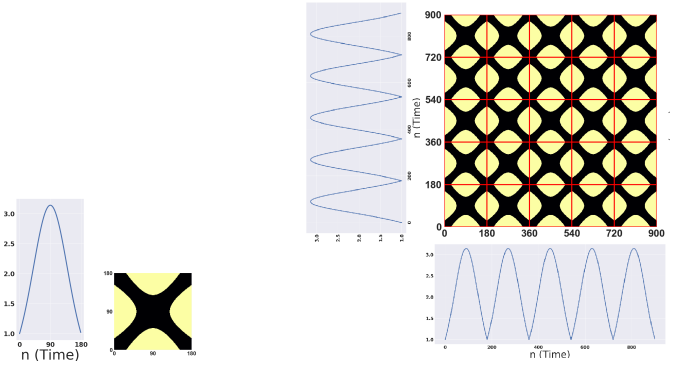
$$d_{ij} = \begin{cases} 1 & \text{if } |y_i - y_j| \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

where $||$ is the absolute value function. Each pixel shows whether the trajectory of the time series at time j has reached to a sufficiently close (ϵ) position of the time series at time i. In this plot a periodic trajectory shows itself as a diagonal line. However, as discussed by Norbert Marwan [20] a diagonal line does not always represent a periodic trajectory. Thus, it is more robust to search for repeating patterns as described below.

B. Checkerboard pattern:

Assume each $p_i \in P$ has length s, i.e., $|p_i| = s$. Consider S to be the signal that p_i is generated from through sampling. By definition P is a region in the time series where the periodic function S with period s is repeating, i.e., $\forall y_i \in P \;\; S(y_i + s) = S(y_i)$.

Assume the pattern τ_1 is the pattern generated by drawing the recurrence plot of the signal p_1. Thus τ_1 is an image of size $s * s$ (width=s and length=s). Now by the definition of a periodic function, the same pattern will be generated for every $p_i \in P$. As a result τ_1 acts like a tile that repeats in the entire diagonal line of the recurrence plot generated from P. Through the definition of periodic function $\forall i, j | p_i = p_j$ then the same



(a) Signal (b) RP for the signal.

(c) The checkerboard pattern generated from the periodic pattern in 2a repeated multiple times. The tiles are separated by red lines for visibility purposes.

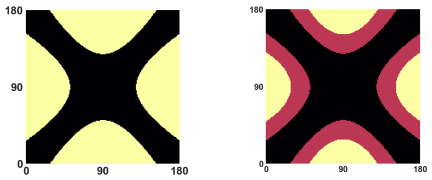
Fig. 2: The checkerboard pattern is generated as a result of periodic behaviour of the repeating signal

tile τ_1 is generated in the recurrence plot generated from p_i and p_j. Thus τ_1 is repeating through the entire recurrence plot for P. As a result of this repetition, the recurrence plot for P is generated from the periodic repetition of τ_1 in the entire P. Fig 2b shows the corresponding recurrence plot tile for the signal in Fig 2a. A signal that is consecutively repeated results in a segment of periodic repeats as shown in Fig 2c (Both segments of repeats are identical and given to demonstrate how recurrence plots is generated in a 2D dimension). The corresponding recurrence plot of the segment is also given in this figure. As shown in the figure, the recurrence plot consists of repeated tiles for the duration of the segment and it resembles a checkerboard pattern.

The epsilon parameter, in the RP formula stated above, determines how close two points on a time series need to be, to be considered similar. This parameter has a key influence in generating the periodic tiles. An epsilon that is too small results in tiles that are majorly bright color (yellow as seen in Figure 4) and a too big epsilon results in tiles that are majorly dark colour (black in Figure 4). The challenge is to find the most suitable ϵ that generates the most revealing tile inside the recurrence plot. This problem of finding the right ϵ magnifies in real world problems where the periodic signal may change over time. As a result each repetition of the p_i in P is equal to some variation of the periodic signal S. These variations manifest themselves as variations of each tile in the recurrence plot of the periodic segment.

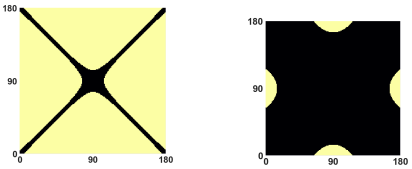
A solution to this problem is to still use thresholds, but sharpen the differences by quantizing the threshold for similarity. In this approach the dark part in the generated tile remains unchanged, but the bright part that is associated with dissimilarities is represented through multiple colors. An example of such tile with two threshold levels is shown in Fig 3.

We use a variation of RP with multiple thresholds as



(a) Single thresholded tile with $\epsilon = .7$ (b) Double thresholded tile with $\epsilon = .7$

Fig. 3: The effect of Δ on a recurrence plot tile.



(a) A tile with a very small threshold $\epsilon = .1$ (b) A tile with a very big threshold $\epsilon = 1.8$

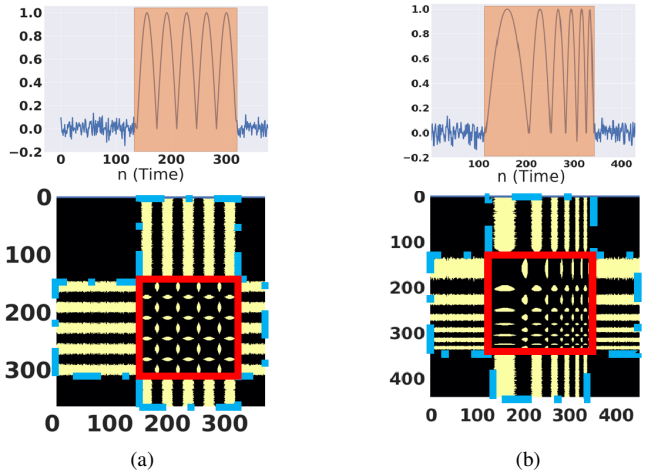
Fig. 4: The effect of ϵ on a recurrence plot tile

follows:

$$d_{ij} = \begin{cases} 1 & \text{if } |y_i - y_j| \leq \epsilon \\ \dots & \\ \Delta & \text{if } (\Delta - 1)\epsilon \leq |y_i - y_j| \leq \Delta\epsilon \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Using this technique, we create a RP representation of our time series and look for patterns that are associated with segments of repeating signals.

Fig 5a presents a polynomial function that has been repeated multiple times surrounded by white noise followed by its corresponding RP. As seen in this figure, the polynomial function, which contains perfect repeating patterns, is manifested as a checkerboard pattern on the diagonal of the RP, surrounded by vertical/horizontal lines. The checkerboard pattern is highlighted by red box and the vertical/horizontal lines are highlighted by dashed blue lines. Fig 5b presents a polynomial function whose frequency has changed gradually followed by its corresponding RP. As seen in this figure, the checkerboard pattern still captures the segment in the time series where repeat has occurred, although the repeats have a different frequency and shape. Examples of RP from a real rehabilitation exercises time series dataset [24] with only some of the time series containing exercises have been shown in Fig 6. As seen in this figure, the exercises, in the form of segments of consecutive repeats manifest themselves as checkerboard patterns that are on the diagonal of the RP.



(a) (b)

Fig. 5: (a) An example of a synthetic time series, consisting of a periodic function surrounded by noise and its corresponding RP. (b) A periodic function whose period has changed with time and the corresponding RP.

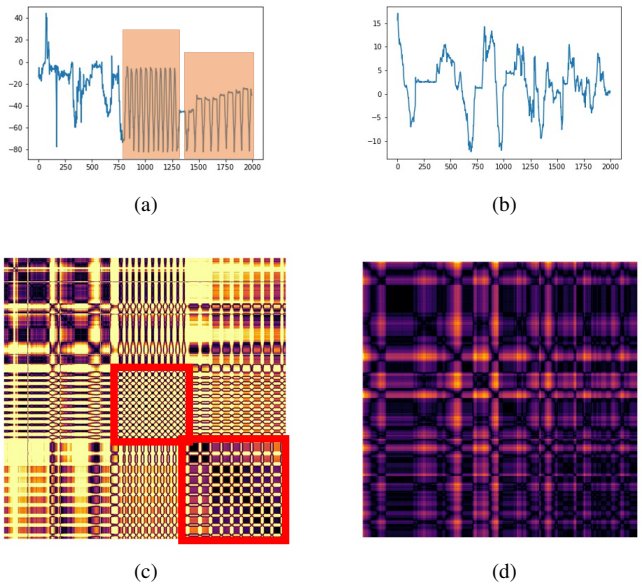
C. Vertical line:

The other distinctive feature from a segment of consecutive repeats that reveals itself in RP is the presence of periodic vertical and horizontal lines that surround the checkerboard pattern. Marwan et. al [21] proof that the presence of these patterns shows the transition from periodic to periodic or periodic to chaotic (non-periodic) states in the time series. That is they illustrate the transition that exist in the time series from a chaotic pattern to a periodic pattern. Examples of such lines are shown in Fig 5a and Fig 5b. In these figures the checkerboard pattern is showing the region where the segment of consecutive repeats is matching itself in the RP. The vertical and horizontal lines confirm that there exists a transition from a non-periodic behaviour to a periodic behaviour. In Section IX we discuss how these regions make our technique robust to noise.

RP-Mask searches for checkerboard patterns in a given RP residing on the diagonal. We consider the checkerboard pattern an object that we want to detect and localise in the RP image and leverage the advances made by the computer vision community in object detection.

D. Object detection:

In order to detect and localise the checkerboard pattern object from the RP image, we use the Mask RCNN algorithm [14]. Assume multiple objects of interest (with different sizes) exist inside an image. CNN networks cannot detect all these objects in one pass, because CNN looks at all regions in the image to determine whether a specific pattern or object exists and thus it needs to look at different regions of the image in different passes. However, we do not know in advance how many regions the image should be segmented, to feed them to a CNN. Regional-based CNN (R-CNN) solves this problem by introducing search algorithms to extract these region of interests (RoI) [10]. A successful approach to efficiently detect these RoI is introduced by Ren et al. [26], called Faster-RCNN, which finds RoI by first learning a convolutional feature map. This map then becomes the input for a neural network called a Region Proposal Network to identify possible RoIs. Mask



(a) (b) (c) (d)

Fig. 6: Example of a time series epochs with two exercises and the corresponding RP (a). Example of a time series epochs without any exercise and the corresponding RP (b)

RCNN extends this architecture one step further using the actual pixel mapping of the object [14]. Mask RCNN tries to find the actual boundary of the object of interest inside the RoI by considering the pixel level labels inside the RoI [14].

The algorithm of RP-Mask is presented in Algorithm 1. The algorithm starts by receiving a time series (T), the two parameters for generating recurrence plot ϵ, Δ, the pretrained Mask-RCNN model and the confidence threshold Θ. The algorithm generates the recurrence plot for the given T (line 2). It then passes the plot to the trained Mask-RCNN to detect the checkerboard pattern regions (line 3). In line 5 to 9 the algorithm filters the regions with confidence lower than the given threshold Θ.

Algorithm 1 RP-Mask

1: **procedure** DETECTREPEATSEGMENTS(T, ϵ, Δ, M, Θ)
▷ Time series T, Recurrence plot parameter ϵ, Δ, Mask-RCNN model M, confidence threshold Θ
2: RP $\leftarrow$GENERATERECURRENCEPLOT(T, ϵ, Δ)
3: Regions $\leftarrow$ M(RP)
4: S $\leftarrow \emptyset$
5: **for all** $r \in$ Regions **do**
6: **if** $r.confidence \geq \theta$
7: S.append(r) ▷ r is a region inside the image with a checkerboard pattern
8: **end if**
9: **end for**
10: **Return** S
11: **end procedure**

The workflow of RP-Mask for localisation of segments of consecutive repeating patterns from time series data is presented in Fig 7. As seen in Fig 7, the last step of RP-Mask is a post-processing filtering task. We use the Mask RCNN returned confidence for each detected region to filer the regions with low confidence. Each detected region is specified with a bounding box that corresponds to the start and end of the segment of repeating patterns on the actual time series. We study the impact of the confidence threshold in VIII-0b.

V. Experimental design

We experiment on how well our approach can detect and localise segments of consecutive repeating patterns using a real dataset and a synthetically curated dataset. We further experiment how well our learned model on the synthetic data works when tested on a real dataset and compare our approach to baseline methods.

A. Real dataset:

We use a dataset collected from tracking individuals' with chronic knee pain using an accelerometer by Nicholson et al. [24]. This dataset contains time series collected from the subjects over a two weeks period using an Xl6-mini triaxial accelerometer. The subjects have been given an ankle cuff with an embedded accelerometer and asked to wear the ankle cuff every time they are doing a set of rehabilitation exercises. Once the subjects wear the ankle cuff the accelerometer starts recording data. Given the type of exercises, the data collected by the accelerometer's X-axis is of interest. There are 4 different types of exercises that could be performed by the subjects. The data has been labelled by subject matter experts and the start and end of the exercises have been marked. For each timestamp in the time series, there is a label which indicates whether the timestamp belongs to an exercises region or not. We had access to the data of 27 patients from this study. The authors have noted the data collection has taken 3 months, which re-emphasizes the point that data collection from patients is a lengthy process. Data has been sampled at 12 samples per second. Exercises are expected to take 30 sec to 1 min roughly. We cut each time series at 2000 timestamps to make all the same size, which is equal to 2.7 minutes of the data recorded in each session.

We have divided the dataset into train, validation and held out test data, where 70% has been used for training (60%) and validation (10%) and 30% as held-out test data. The split has been done patient-wise, meaning each patient's data has been used only for training, validation or testing to avoid over-optimism bias. Our training data are time series of length 2000, which have a corresponding 2000-length label. Each label is 0 or 1, 0 meaning the corresponding time point in the time series is not part of an exercise and 1 meaning the opposite. Our training set contains 535 time series samples containing exercises and 2165 samples containing no exercises. Our validation set has 22 samples containing exercises and our test set contains 303 samples containing exercises and 996 samples with no exercises. We experiment how well RP-Mask and our baseline methods can detect whether each point on

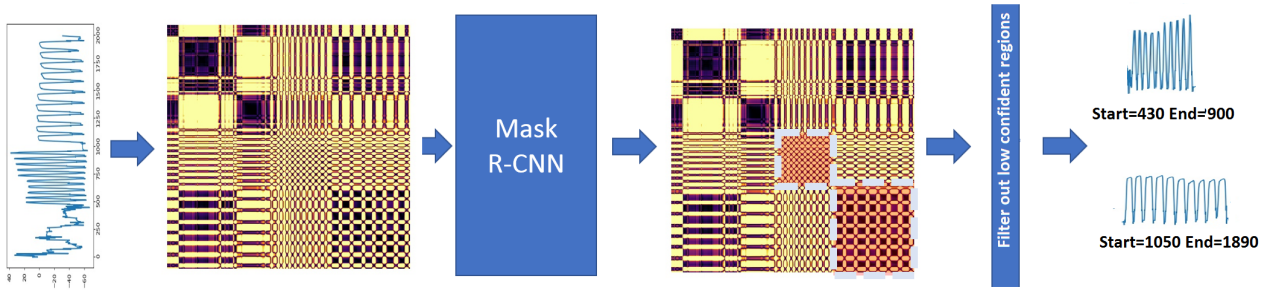


Fig. 7: The workflow of RP-Mask for detecting consecutive repeating patterns from a time series.

the time series of the held-out test data belongs to the right class, specified in the ground truth.

B. Synthetic dataset:

Our synthetic dataset contains curated time series which represent key features from real-world data with consecutive repeating patterns. As such, the repeating patterns might be repeated after each other with the same length and frequency or as a result in a change in the source of the data, they may change. Our synthetic dataset contains curated time series of length 2000 as follows:

1) Perfect signal: Each time series is generated from repetitions of a single signal surrounded by noise. The repeating signal was generated using $\sin(x), x \in [0, \pi]$. In each time series the signal was repeated for a random number of times (between 4 and 10) and surrounded with white noise with $\mu = 0, \sigma = 0.1$.
2) Degrading signal: Each time series was generated using a single signal whose frequency was increasing. The signal was generated using $\sin(x/m), x \in [0, \pi]$ and m was equal to the 1 and incremented after each repetition. Again each segment contained between 4 and 10 repetitions, surrounded by white noise with $\mu = 0, \sigma = 0.1$.
3) Multiple segments: A time series which contains two repeating segments randomly selected from the periodic segments from the first two items.

Noise in our synthetic dataset has been modeled using white Gaussian noise. Our training set contains 2986 time series which contain repeating patterns. Our validation set contains 300 time series with repeating patterns. We generate a 2000-long label for each synthetic time series similar to the real data labels.

C. Baseline methods:

We have implemented a number of state-of-the-art algorithms as our baselines and compared RP-Mask to these algorithms. We have parameter-tuned all the baseline methods based on our datasets.

The first baseline method we have implemented is the Bidirectional-LSTM (Bi-LSTM) method from Hammerla et al. [11]. The network includes 400 repeating modules which are connected to a Time distributed dense layer for sequence learning. We used the Adaptive Gradients Algorithm (AdaGrad) to optimize the network. The time series are fed in batches of 200 to the network. We balance the training data by under-sampling from no-exercises data before using training.

The second baseline method is OToR [16]. OToR detects regions of repeats by finding the auto-correlations of a time series and finding peaks that are supported by linearity and periodicity of points. OToR requires two parameters, ϵ: period error threshold and α: linearity error threshold. For our synthetic dataset, we found ϵ=50 and α=150 to be the best parameters, whereas for the real dataset ϵ=60 and α=200 produced the best result.

Our third baseline is the TSDecompose [38] algorithm, which contains two steps: 1- Time series symbolisation and string decomposition. The authors of TS-Decompose introduce a symbolisation method using the HOG-1 algorithm that preserves the shape of the time series. They use this presentation to segment a time series by splitting a time series into segments such that the left and right sub-strings have the maximal homogeneity. The split points for the decomposition are found by maximising information gain from information theory of the split points. We consider output segments by the algorithm that are larger than 100 to be segments of consecutive repeating patterns.

Our fourth baseline is the Matrix profile(MP) method [34]. We have used the matrixprofile-ts python package and used the STAMP algorithm. To find the best input parameter 'm' for our real and synthetic data, we used the method presented by Mirmomeni et al. [23]. We found m=100 to be the best value for our real data and m=250 to be the paramter for our synthetic data. We post-process the MPs and look at the average values of MPs within a window of length 20. If the average value dropped below a threshold (0.5), we label every point in this window as 1. The default label is 0. This approach is based on the property of MP in which it drops close to zero where a pattern has been repeated. [34]. We did an exhaustive search to find the optimal value for the window size and the threshold.

The reason for comparing our approach, RP-Mask, with MP, OToR and TS-Decompose, is to show how methods with no access to labelled data can detect segments of consecutive repeats. We report the result for both training with labelled data and synthetic data for RP-Mask to show how well the method can perform in the absence of labelled data.

D. Evaluation technique

We report F1-score, Adjusted Rand Index (ARI) and Intersection Over Union (IOU). For each time series, we define F1-score as a function of True positives, False positives and False negatives. For each prediction in the 2000-length label, if the predicated label is the same as the ground truth, that is a true positive. We define False negative and False positives accordingly. An interim F1 is calculated for each time series sample and final F1 is the result of averaging interim F1s over all time series samples. ARI measures the extend to which the labels in ground truth for each time series matches the labels predicated by the algorithm. The ARI maximizes at 1, when all the labels in two partitions agree. The IoU [27] is a normalised measure that calculates the overlap of two segments. Given this definition, in our context, IoU is only applicable when the ground truth time series contains exercises, or segments of repeating patterns.

VI. Results

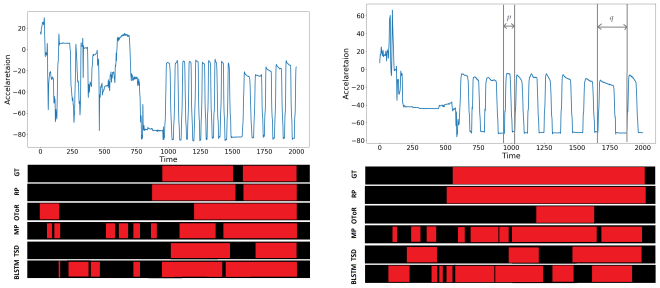
We investigate how well our approach and our baseline methods can detect the location of the segments of repeating patterns. The results of this comparison is shown in Table I. The Real dataset section refers to methods being trained on the real dataset described in Section V-A. The Synthetic dataset section refers to methods being trained on the synthetic dataset described in Section V-B. No exercise/No repeats section refers to time series without exercises/repeats, whereas the exercise/repeat section refers to time series with exercises/repeats. F1 is F1-score measure, ARI is Adjusted Rand Index and IoU is Intersection Over Union. We described the evaluation techniques in Section V-D.

As shown in this table, when trained on real data, RP-Mask can detect time series without exercises with an F1-score of 0.98 and ARI of 0.97. For time series with exercises, RP-Mask identifies the start and end of exercises, with an F1-score of 83%,Adjusted rand score of 58% and IoU of 76%.

When training on synthetic data and testing on real data, RP-Mask is able to detect time series with no exercise with an F1-score of 90% and ARI of 68%, and start and end of segments with exercises with an F1 of 78%, ARI of 43% and IoU of 68%, a slight drop from the values obtained in real training but highly comparable. Our results show that when training entirely on a synthetic dataset, our method is transferable and performs as effective on a real test data. As mentioned, this is particularly important in applications such as patient activity recognition, where collecting and labelling data is expensive and collecting variations of data is sometimes impossible.

TABLE I: Results for detecting and localising a segment of consecutive repeating patterns

	Real dataset					Synthetic dataset				
	No Exercise		Exercise			No Exercise		Exercise		
Method	**F1**	**ARI**	**F1**	**ARI**	**IoU**	**F1**	**ARI**	**F1**	**ARI**	**IoU**
RP-Mask	**.98**	**.97**	**.83**	**.58**	**.76**	.90	.68	.78	.43	.68
Bi-LSTM	.97	.59	.67	.28	.54	.95	.84	.02	.01	.01
OToR	.95	.79	.68	.34	.57	.65	.32	.80	.57	.68
TSDec	.63	.14	.67	.24	.54	.63	.14	.67	.24	.54
MP	.97	.67	.76	.4	.67	.99	.97	.34	.13	.26



(a) Sample of an exercise and how different methods perform in detecting the signals.

(b) Sample of an exercise with systematic noise and performance of different methods in detecting the region.

VII. Discussion

A closer investigation on the results of Table I shows that MaskRCNN when trained on real data can correctly label a time series with no exercise. The high F1-score and ARI for the No Exercise column confirms this finding. OToR and MP both perform reasonable in labelling a time series with no exercise, however, the big drop in ARI (from 0.97 in MaskRCNN to 0.67 in MP) of No Exercise data shows that MP detects large segments of exercise from a time series with no exercise. This problem also shows itself in the time series with exercise segments where non-exercise (non-repeating) segments are falsely detected as exercise segments. An example of such case is shown in Figure 8a. In this figure a time series with two exercise is shown in the top row. The ground truth is shown in the second row, where exercise regions are shown in red. In this figure, we can see MP has selected areas out of the exercise regions as exercise (the first six red areas from MP).

For OToR we observe the same problem as reported by the authors [16] where the starting or ending of an exercise session can be missed due to the sliding window size. OToR requires three consecutive repetitions before being able to detect an exercise region. Comparing the selected segments in Figure 8a from OToR to GT shows demonstrates the exercise detection delay problem. In situations where the sliding window breaks an exercise section, the exercise segment is broken into two segments. In this scenario, the first segment is often misclassified as non-exercise by OToR. We observe the same problem for TSDecompose, where it requires a long sequence of data before considering the segment an exercise region. The delay in detecting the second exercise in Figure 8a shows this problem.

The presence of systematic noise is another challenge that MaskRCNN can handle while other methods fail to detect correctly. An example of such data is shown in Figure 8b. In this figure the patient failed to perform the exercise consistently. As a result the duration of each repeat varies from one repetition to another repetition. For example the duration of the two samples labeled by p (duration 7sec) and q (duration 20sec) in the figure differs by more than 10 seconds. As reported by OToR [16] authors this algorithm cannot track regions with

systematic noise. We can see TSDecompose also fails to cover the entire exercise while MP incorrectly detects extra exercises at the beginning of the routine. Figure 8a and 8b both show that Bi-LSTM incorrectly selects regions where no exercise is performed.

In the case of training on synthetic data, the two deep learning methods (RP-Mask and Bi-LSTM) are trained on the synthetic dataset, described in Section V-B, while the traditional time series methods have been parameter-tuned using the Synthetic dataset. A closer investigation on the result of the evaluation of the stated methods on real data in Table I (Synthetic dataset) shows that the only method that can perform well on both detecting Exercise and No Exercise data is our approach, RP-Mask, (F1$>$ .75). In this scenario, on No exercise data, methods such as MP and Bi-LSTM outperform RP-Mask. However they perform poorly on detecting Exercise data (F1-score drops from 0.78 in RP-Mask by at least 0.40 in MP). On the other hand, although OToR outperforms RP-Mask on detecting Exercise regions, its F1-score performance on No Exercise regions drops by more than 0.30. Our results for traditional time series segmentation methods show that these methods require access to the labelled data for parameter tuning to achieve their best performance. Our results show that Bi-LSTM's performance significantly drops from real dataset to synthetic dataset which is in line to what was reported by Hammerla et. al [11]. In this paper, the authors report that Bi-LSTM is sensitive to its number of internal units and needs to be tuned for each dataset.

Our results show that the checkerboard pattern generated in the RP and the vertical lines are a distinctive feature for detecting and localising the segments of repeating patterns, in comparison to other methods. One reason is that these features are not sensitive to the frequencies of the signals in the time series. As shown by Marwan et al. the vertical lines in RP are robust to noise [21]. An example of RP from our synthetic dataset after adding white noise to the segment of consecutive repeat is shown in Fig 8. Although, the checkerboard pattern (red boxes) is fading (from left to right), the horizontal/vertical lines (blue boxes) associated with the chaos-periodic transitions are still visible in the RP.

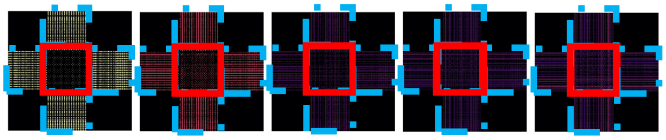
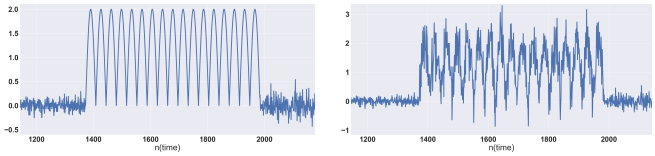


Fig. 8: The effect of noise on checkerboard patterns and the horizontal/vertical line. From left to right white noise is added.

Noise analysis: We discussed how RP-Mask can tolerate the presence of noise. We empirically test this using our synthetic dataset. We train Mask RCNN on the dataset with no noise. We then add different levels of white noise to the synthetic signals and run the trained model to detect segments of consecutive repeats. The result of this experiment is in Fig 10. On this graph the X-axis shows the amount white noise



(a) (b)

Fig. 9: The effect of adding 0-mean white noise, $\sigma = .41$ (Figure 9b) to a synthetic signal (Figure 9a).

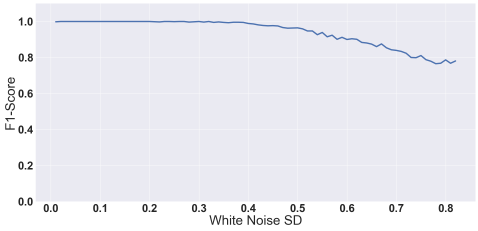


Fig. 10: The effect of white noise in detecting regions of consecutive repeats.

standard deviation and Y-axis shows the F1-score for finding the segments of consecutive repeats. From the graph we can see that RP-Mask's performance does not drop until the white noise standard deviation reaches 0.4. Afterwards, RP-Mask's performance declines slowly as the noise level increases. An example of a synthetic signal before and after adding the white noise is given in Fig 9 for reference. In this figure, we can see that the noise distorts the shape of the signal.

VIII. Parameter-tuning strategy for RP-Mask

To implement RP-Mask explained in Section IV, we use a Mask RCNN model which has been pre-trained on the COCO dataset [5], a large-scale object detection, segmentation, and captioning dataset with more than 200,000 labeled images. The training set for Mask RCNN only includes the RP of the time series in the real dataset explained in Section V-A with at least one segment of exercises, labeled 1. The RP of our training dataset is generated by finding the optimal ϵ and δ parameters using the validation dataset, as explained in Section VIII-0a. For each RP in the training data, we create an annotation image by mapping the label of the associated time series to itself. In this image, segments of repeating patterns show themselves as white squares, while the rest of the image is black. We use a stochastic gradient descent (SGD) optimizer for optimizing the network and a step-wise learning rate, where in each step we change the learning rate. We re-learn the last layer of the network over 30 epoch. The effect of hyper parameters has been given in Section VIII-0c.

a) RP parameters: As we mentioned in Equation 1 RP requires two parameters ϵ (similarity threshold) and Δ (levels of similarity). In order to determine the best value for ϵ and Δ, we did a heuristic parameter search. Figure 11 shows the value of F1-scores for each pair of ϵ and Δ when trained on the real train dataset and tested on the real validation dataset. As shown in this figure, $\epsilon = 0.9$ and $\Delta = 250$ produces the best results.

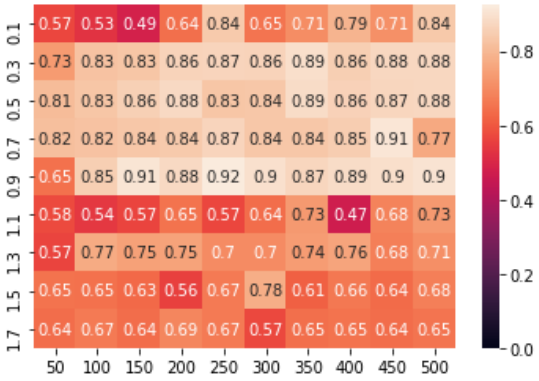


Fig. 11: The effect of epsilon and delta parameters of RP.

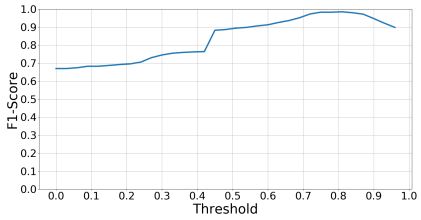


Fig. 12: The effect of filtering by network's confidence.

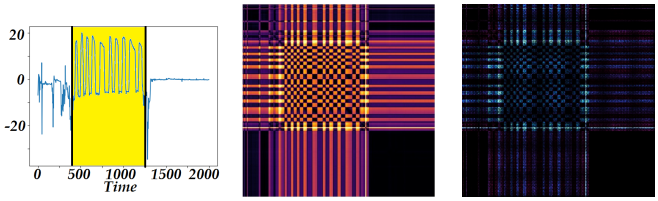
b) Confidence threshold: To show the impact of thresholding the network's confidence, we perform an experiment in which we change the network's confidence and evaluate the accuracy of the Mask-RCNN in finding the correct segments of repeating patterns. The result of this experiment is in Fig 12. The graph shows that low thresholding the network's confidence results in a low F1-score in the result. However, setting the threshold 0.7 and above results in the F1-score of 0.95 and higher.

c) Effect of hyper parameters: As previously mentioned, we use Stochastic Gradient Descent (SGD) optimizer with a step decay scheduler to train our model. SGD requires a learning rate which shows the amount that the DNN weights are updated during training. We observe that in training the real dataset, best results are achieved when learning rate is set close to 0.08. However, when training on the synthetic data, the best performance is achieved when the learning rate is set to 0.03. We used an step decay scheduler to update the SGD learning rate. The step decay scheduler updates the learning rate by multiplying the learning rate to a coefficient less than one.

We observe in our real dataset that is best to decay the learning rate every 10 steps, while when training on synthetic dataset, the best performance is achieved when decay rate is set to 5. The update coefficient for the scheduler was set to 0.8 for both scenarios. We train the MaskRCNN until its accuracy for the training data flattens, which in both scenarios happened after 30 iterations of training (epochs).

IX. Explaining the decision making

Machine learning and deep learning approaches often work in a black box fashion, in which it is difficult to determine which layers/pixels have contributed most to a particular decision making. The Integrated Gradients method [29], tackles this problem for deep learning techniques by identifying which pixels have contributed most to the decision making of the model and presenting them in a saliency map in which the pixels that have contributed most are presented in brighter colors according to their contributions. Fig 13a shows an example of a time series with a segment of repeating patterns (highlighted area). The corresponding RP has been given in Fig 13b and the corresponding saliency map in Fig 13c. As shown in Fig 13c, there are two dominant pattern that the Mask RCNN has focused on namely: 1) the checkerboard patterns, and 2) the vertical lines surrounding the checkerboard patterns. As discussed earlier, the checkerboard pattern is the main feature that reveals a segment of repeating patterns inside the RP and vertical lines are a side effect of these segments that demonstrate the transition from a periodic segment to a chaotic segment and vice versa.



(a) Time series (b) RP (c) Saliency map

Fig. 13: (a) A time series from our real dataset with a segment of repeating patterns. (b) The RP corresponding to the time series in (a), and (c) the saliency map corresponding to (b).

X. Conclusion

We have created a transferable approach that outperforms the state of the art algorithms for detecting and localising segments of consecutive repeating patterns from a time series. Our approach, RP-Mask, leverages the latest advancements in the field of image segmentation and object detection, Mask RCNN, in combination with a classical approach for detecting recurrence in a time series, recurrence plots, RP, for detecting repeating patterns in a time series. We demonstrate through a set of experiments that our model is agnostic to training data and generalisable to an unseen test data and is tolerant to noise. Our approach outperforms the state of the art algorithms, both deep learning and classical pattern mining approaches, when all models are trained and tested on the same training and test set from a real accelerometer dataset. Furthermore, our approach outperforms the state of the art algorithms in transferability of its learned model on a curated synthetic data to a real world test dataset.

REFERENCES

[1] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In A. Ferscha and F. Mattern, editors, *Pervasive Computing*, pages 1–17, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[2] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and T. Kechadi. Human activity recognition with convolutional neural networks. 09 2018.

[3] A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, 46(3):33:1–33:33, Jan. 2014.

[4] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Proc. of KDD '03*, pages 493–498, 2003.

[5] C. dataset. Coco dataset http://cocodataset.org, 2019.

[6] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He. Modeling extreme events in time series prediction. pages 1114–1122, 07 2019.

[7] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhysics Letters (EPL)*, 4(9):973–977, nov 1987.

[8] E. Garcia-Ceja, M. Z. Uddin, and J. Torresen. Classification of recurrence plots' distance matrices with a convolutional neural network for activity recognition. *Procedia Computer Science*, 130:157 – 163, 2018. The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018).

[9] S. Gharghabi, Y. Ding, C. C. M. Yeh, K. Kamgar, L. Ulanova, and E. Keogh. Matrix profile viii: Domain agnostic online semantic segmentation at superhuman performance levels. In *Proc. of ICDM 17*, pages 117–126, 2017.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of CVPR 14*, 2014.

[11] N. Y. Hammerla, S. Halloran, and T. Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proc. of IJCAI'16*, pages 1533–1540. AAAI Press, 2016.

[12] N. Hatami, Y. Gavet, and J. Debayle. Classification of time-series images using deep convolutional neural networks. In A. Verikas, P. Radeva, D. Nikolaev, and J. Zhou, editors, *Tenth International Conference on Machine Vision (ICMV 2017)*, volume 10696, pages 242 – 249. International Society for Optics and Photonics, SPIE, 2018.

[13] N. Hatami, Y. Gavet, and J. Debayle. Bag of recurrence patterns representation for time-series classification. *Pattern Analysis and Applications*, 22(3):877, 2019.

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *Proc. of ICCV'17*, Oct 2017.

[15] A. Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing Journal*, 62:915 – 922, 2018.

[16] Y. Kowsar, M. Moshtaghi, E. Velloso, C. Leckie, and L. Kulik. An online unsupervised dynamic window method to track repeating patterns from sensor data. *IEEE Transactions on Cybernetics*, pages 1–13, 2020.

[17] J.-S. Lin and D. Kulic. Online segmentation of human motion for automated rehabilitation exercise analysis. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 22(1):168 – 180, 2014.

[18] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille. A long-term evaluation of sensing modalities for activity recognition. *Lecture notes in computer science*, page 483, 2007.

[19] P. Malhotra, V. TV, L. Vig, P. Agarwal, and G. Shroff. Timenet: Pre-trained deep recurrent neural network for time series classification. *CoRR*, 2017.

[20] N. Marwan. How to avoid potential pitfalls in recurrence plot based data analysis. *International Journal of Bifurcation and Chaos*, 21, 07 2010.

[21] N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths. Recurrence-plot-based measures of complexity and their application to heart-rate-variability data. *Phys. Rev. E*, 66:026702, Aug 2002.

[22] S. Maya, A. Yamaguchi, K. Nishino, and K. Ueno. *Lag-Aware Multivariate Time-Series Segmentation*, pages 622–630. 01 2020.

[23] M. Mirmomeni, Y. Kowsar, L. Kulik, and J. Bailey. *An Automated Matrix Profile for Mining Consecutive Repeats in Time Series*, pages 192–200. 07 2018.

[24] P. J. A. Nicoloson, R. S. Hinman, T. V. Wrigley, P. W. Stratford, and K. L. BEnnel. Effects of covertly measured home exercise adherence on patient outcomes among older adults with chronic knee pain. *Journal of Orthopaedic Sports Physical Therapy*, pages 548 – 556, 2019.

[25] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. 04 2017.

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[27] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proc. of CVPR'19*, pages 658–666, 2019.

[28] D. F. Silva, V. M. D. Souza, and G. E. Batista. Time series classification using compression distance of recurrence plots. In *2013 IEEE 13th International Conference on Data Mining*, pages 687–696, 2013.

[29] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. 2017.

[30] S. Torkamani and V. Lohweg. Survey on time series motif discovery. *WIREs: Data Mining and Knowledge Discovery*, 7(2):n/a, 2017.

[31] K. D. S. B. Van Laerhoven, K. Using rhythm awareness in long-term activity recognition. *Proc. of ISWC 2008*, page 63, 2008.

[32] K. Xia, J. Huang, and H. Wang. Lstm-cnn architecture for human activity recognition. *IEEE Access*, 8:56855–56866, 2020.

[33] M. Xu, M. Han, C. Chen, and T. Qiu. Recurrent broad learning systems for time series prediction. *IEEE Transactions on Cybernetics, Cybernetics, IEEE Transactions on, IEEE Trans. Cybern*, 50(4):1405 – 1417, 2020.

[34] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh. Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *Proc. of ICDM 16*, 2016.

[35] C. M. Yeh, N. Kavantzas, and E. Keogh. Matrix profile vi: Meaningful multidimensional motif discovery. In *Proc. of ICDM'17)*, pages 565–574, Nov 2017.

[36] C. T. Yen, J. X. Liao, and Y. K. Huang. Human daily activity recognition performed using wearable inertial sensors combined with deep learning algorithms. *IEEE Access*, 8:174105–174114, 2020.

[37] Y. Zhang, Y. Hou, S. Zhou, and K. Ouyang. Encoding time series as multi-scale signed recurrence plots for classification using fully convolutional networks. *Sensors*, 20(14), 2020.

[38] J. Zhao and L. Itti. Decomposing time series with application to temporal segmentation. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016.