

# SAXJS: An Online Change Point Detection for Wearable Sensor Data

Giovanna A. Riqueti<sup>1</sup>, Pedro H. Barros<sup>1</sup>, João B. Borges<sup>2</sup>, Felipe D. Cunha<sup>3</sup>  
Osvaldo A. Rosso<sup>4</sup>, Heitor S. Ramos<sup>1</sup>

<sup>1</sup>Department of Computer Science  
Universidade Federal de Minas Gerais (UFMG)  
Belo Horizonte, MG 30123-970, Brazil

<sup>2</sup>Department of Computing and Technology  
Universidade Federal do Rio Grande do Norte (UFRN)  
Caicó, RN 59300-000, Brazil

<sup>3</sup>Department of Computer Science  
Pontifícia Universidade Católica de Minas Gerais (PUC-MG)  
Belo Horizonte, MG, 30535-610, Brazil

<sup>4</sup>Instituto de Física  
Universidade Federal de Alagoas (UFAL)  
Maceió, Alagoas, 57072900, Brazil

{giovanna.riqueti, ramosh, pedro.barros}@dcc.ufmg.br

joao.borges@ufrn.br, felipe@pucminas.br, aorosso@gmail.com

**Abstract.** *Wearable electronics are devices used by humans that can continuously and uninterruptedly monitor human activity through sensor data. The data collected by them have several applications, such as recommending running techniques and helping to monitor health status. Segmenting such data into chunks containing only a single human activity is challenging due to the wide variability of underlying process characteristics presented in the data. To deal with this problem, we propose a new change point detection algorithm based on the Symbolic Aggregate approxImation (SAX) transformation, the probability of transition between symbols, and the Jensen-Shannon distance.*

## 1. Introduction

The wearable electronics market is rising in 2022 and has reached significant success in the last decade. The total market worth was nearly \$80 billions in 2020, having tripled since 2014, and the expectation for 2025 is of \$140 billions [IDTechEx 2021]. Wearable electronics can monitor individual activities continuously and closely without interrupting or limiting user movements. These devices, and others like them, offered a range of features that made them appealing to consumers, including fitness tracking, health monitoring, and convenient access to information and communication [Gao et al. 2016].

Detecting breakpoints or transitions based on sensor data has many applications, such as the segmentation of running activity to evaluate the cause of an injury [Napier et al. 2017], identification human activity transitions [Thakur and Biswas 2022] and helping monitoring health status to detect emergencies [Nachiar et al. 2020].

This problem of detecting breakpoints can be formulated as change point detection [Aminikhanghahi and Cook 2017].

The change point problem consists of discovering the points in time at which the time series changes its behavior. Time series is an ordered data sequence, and we can divide the algorithms to detect these changes into two main groups: offline and online [Aminikhanghahi and Cook 2017]. Offline techniques only apply with the complete time series information [Truong et al. 2020]. Online techniques, otherwise, can be real-time or  $\varepsilon$ -real-time, where the  $\varepsilon$  is the number of points necessary to look ahead [Aminikhanghahi et al. 2018].

Online algorithms are one of the most crucial challenges in the study area of change point detection, and it is an essential characteristic to improve its usability in wearable sensors. Typically, wearable devices are small and portable, so they have limited space for a battery. As a result, they need to be efficient in their use of power to have a long battery life. Hence, another critical issue in this field is the low computational cost to run the algorithm in the devices [Benson et al. 2018]. In addition, the data collection can be uncompleted, noise distorted due to human error or environment changes [Shinmoto Torres et al. 2013]. Therefore, the algorithm must handle non-stationary data and be robust for different systems.

In order to create a change point algorithm that solves the problem of human activity segmentation in wearable sensors, we propose a new online method based on Symbolic Aggregate approXimation (SAX) [Lin et al. 2007] transformation and the Jensen-Shannon distance [Lin 1991] between the probabilities of transitions among the SAX symbols. The SAX is a non-parametric transformation, which is an advantage for constructing an adaptable algorithm for different systems. In addition, by being a distance, the Jensen-Shannon has the advantage of being symmetric compared to dissimilarity measures [Zunino et al. 2022].

Our approach uses two consecutive sliding windows of the time series, transforming them into the SAX representation, and then constructs a probability distribution matrix between symbol transitions. After constructing the matrix, we calculate the Jensen-Shannon distance between the probabilities of the two windows. For each Jensen-Shannon value, we applied the Savitzky–Golay filter [Press and Teukolsky 1990], which fits a low-degree polynomial based on its adjacent data to smooth the data. As the Jensen-Shannon value is smoothed we also apply a maximum value detection algorithm that analyzes a user-determined number of neighbors. The maximum value must be greater than a defined threshold value to be considered as a possible change point.

First, this proposed algorithm was tested on synthetic data sets to analyze the weaknesses and strengths of the approach in a change point ideal scenario. After validation on the synthetic data set, the algorithm was evaluated using the Human Activity Sensing Consortium (HASC) Challenge data [Kawaguchi et al. 2011a], which is a collaborative project to acquire wearable sensor data information. The HASC Challenge data is a benchmark for both wearable sensors [Cleland et al. 2018] as it is for change point detection methods [Aminikhanghahi and Cook 2017] such as RuLSIF [Liu et al. 2013] and TIRE [De Ryck et al. 2021].

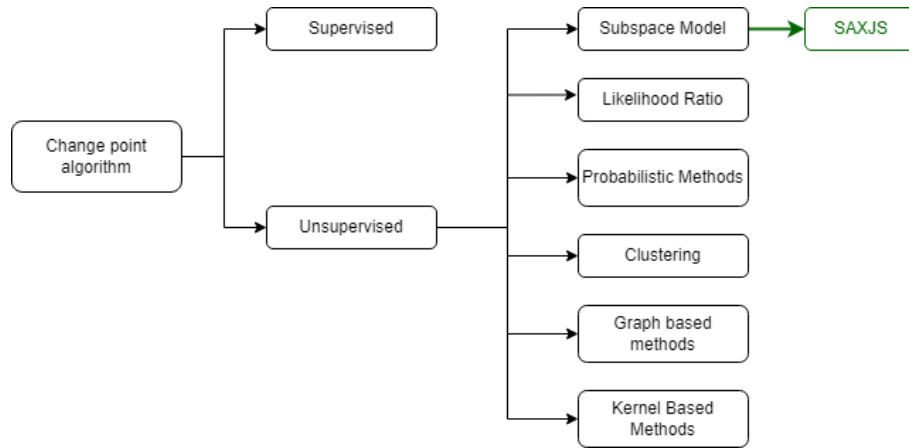
The rest of this paper is structured as follows: In Section 2, we review the literature

about change point detection approaches and the techniques used to segment wearable sensor data. In Section 3, we formulate our change point detection problem. In Section 4, we describe our proposed change point algorithm. In Section 5, we describe our evaluation measure and discuss the results for the synthetic data sets and the results for the HASC Challenge data. Finally, in Section 6, we concluded the work and showed future perspectives.

## 2. Related Work

This section reviews the existing research progress in change point detection and data segmentation in wearable sensors.

### 2.1. Change point detection



**Figure 1. Diagram of change point detection methods. Strongly based on [Aminikhanghahi et al. 2018].**

Figure 1 presents a taxonomy of the techniques used for change point detection. The supervised learning approaches need manual labeling change point examples to train the model to predict the correct output for new input times series data [Ko et al. 2015, Saatçi et al. 2010]. However, this strategy has some disadvantages, as it is challenging to have an extensive and diverse training set to address multiple state change alternatives.

Conversely, unsupervised learning (UL) strives to find patterns or relationships in a data set without being given any labeled examples. In a change point detection scenario, UL segments the data and thereby analyzes its statistical structure [Aminikhanghahi et al. 2018]. Space modeling is a UL paradigm where states represent the time series, and to detect change points, these approaches estimate the distance between these state spaces. For example, the Subspace Identification (SI) algorithm [Kawahara et al. 2007] builds an observation matrix based on the state space model generated through a sliding window. Based on this matrix, the authors calculate the distances between the subspaces to detect change points.

Another UL paradigm estimates the *likelihood* rate of two consecutive intervals and checks whether they come from the same probability distribution. Specifically, *Cumulative Sum* (CUSUM) accumulates the likelihood deviations to a reference point, and when its accumulated sum exceeds a threshold, the approach detects a change

point [Page 1955]. Other studies using CUSUM make several modifications on it, such as the *likelihood* ratio estimate [Verdier 2020]. Algorithms like CUSUM and Change-Finder [Yamanishi and Takeuchi 2002] rely on pre-designed parametric models and are less flexible in real-world scenarios.

Some recent studies try to solve this problem by estimating the density ratio without knowing the two probabilities' density directly. The RuLSIF algorithm [Liu et al. 2013] applies the Unconstrained least-squares importance fitting (uLSIF) [Kanamori et al. 2009] method that learns the density ratio using a least-square fitting framework. The uLSIF has a weakness that density ratios can be unbounded if the denominator is not defined. In order to solve this problem, RuLSIF considered the relative density ratio using a Gaussian kernel.

Recently, the algorithm partially time-invariant representation (TIRE) [De Ryck et al. 2021] uses an autoencoder to estimate time-invariant features. The authors use the Fourier transform to extract frequency domain information and combine both (time and frequency) features. Lastly, the autoencoder uses the Euclidean distance between the sliding window to detect changes in the dynamics. Still, probabilistic methods try to estimate the probability of a window according to the prior probabilities. One of the most used algorithms that follow this trend is the *Online Bayesian Network* (OBN) [Adams and MacKay 2007].

Following the Kernel-based approach, the KernelCPD [Celisse et al. 2018] [Arlot et al. 2019] is implemented with a Pruned Exact Linear Time (Pelt) [Killick et al. 2012]. Pelt detects change points by minimizing a cost function over possible locations and the number of change points. To reduce the computational cost, Pelt prunes the data set using the minimal distance between change points (a.k.a. *minsize*), a cost function regularizer (i.e., a larger regularized value induces the model to find more change points), and the grid of possible change points.

The SAXJS proposed in this work is a new approach to the Subspace modeling strategy using SAX symbols [Lin et al. 2007]. SAX is a non-parametric and low-cost operation seldom explored in literature in this area. Our major novelty is estimating the probability distribution of SAX symbols' transition. Moreover, we use the Jensen-Shannon [Lin 1991] distance to estimate the distance between two consecutive slices (windows) in a probability space to discriminate change points. By being a distance, the Jensen-Shannon has an advantage compared to the dissimilarity measures used in RuLSIF.

## 2.2. Data segmentation in wearable sensors

Wearable data usually comes into a continuous data flow. Segmenting this data can help distinguish valuable segments most suitable for the activity recognition [Ni et al. 2016] and help to improve the model accuracy in this task [Alhammad and Al-Dossari 2021].

According to the literature, the sliding window is a typical segmentation time series approach. These methods can be divided into fixed or dynamic windows. The main challenge is identifying the optimal window size since the activities can have different duration times. In such case, the window must contain at least one activity cycle [Ige and Noor 2022]. For example, [Fida et al. 2015] estimates the window size by using information about the sensor data, its environment, and the association of both. Other approaches use event [Selles et al. 2005] or sensor signal information [Bulling et al. 2014].

We can use change point algorithms to segment time series, for example, Bayesian change point detection [Booth and Smith 1982] and CUSUM [Zhang et al. 2010]. However, [Prajapati and Mahapatra 2009] CUSUM has difficulty dealing with sudden changes since they are not of the same probability distribution. Multivariate Online Change detection Algorithm (MOCA) [Patterson et al. 2016] detects changes in a multivariate data stream. MOCA detects a change point candidate in an analyzed window, which maximizes the heuristic function, and after this stage, MOCA verifies the hypothesis using a statistical test.

As SAXJS is a non-parametric approach, it has the advantage of being adaptable to different systems and changing environments. SAXJS is also more robust compared to the fixed or dynamic window approach. Unlike MOCA, our focus is on univariate rather than multivariate problems.

### 3. Problem formulation

Before we present our proposal, we will briefly define and formalize the problem. These definitions will be used throughout the paper.

**Definition 1** *A time series is a series of data points indexed (or listed or represented graphically) by some time representation. More commonly, a time series is a sequence of successive points typically equally spaced in time; thus, it is a discrete-time data sequence.*

**Definition 2** *Let  $\mathbf{S}$  be a time series with  $s_t \in \mathbf{S}$  in a particular timestamp  $t$  with size  $n$ . We assume the existence of some probability distribution function  $\mathcal{P}$  for  $s_t \in \mathbf{S}$ . Therefore, with this definition, the time series can be considered a stochastic process [Traversaro et al. 2018]. Since  $\mathcal{P}$  can be described by a set of parameters  $\pi_t$  for a given point  $s_t$ , we can formalize that in the existence of a change point in the times series  $\mathbf{S} = \{s_1, s_2, s_3, \dots, s_n\}$  we have*

$$\pi_{s_1} = \pi_{s_2} = \dots = \pi_{s_{T^*}} \neq \pi_{s_{T^*+1}} = \dots = \pi_{s_n},$$

*where we consider the change point between  $\pi_{T^*}$  and  $\pi_{T^*+1}$  because  $\pi_{T^*} \neq \pi_{T^*+1}$ . More formally,  $\exists T$  where we have  $T^* \leq T < T^* + 1$ . We define  $T^*$  as being the time of the change point of  $\mathbf{S}$  in the interval  $\{s_1, s_2, s_3, \dots, s_n\}$ .*

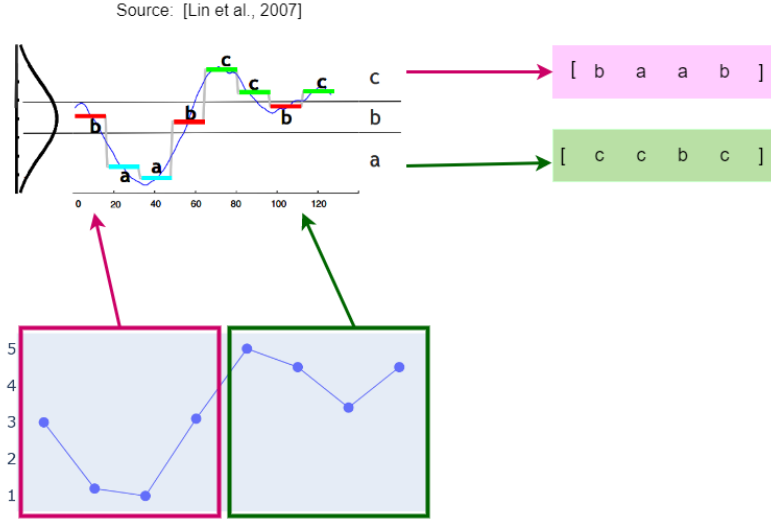
Detecting a change point is a challenging task as it tries to identify a change in the intrinsic characteristics of the data rather than an anomaly. Some of the approaches consider finding outliers (anomalies) and, by doing so, investigate how these outliers are related to the change points [Takeuchi and Yamanishi 2006]. The change points are related to a change in the dynamics of the underlying model. So, change points have characteristics of a new model and can not be considered an outlier. Assuming a model change, detecting a change point can be seen as a change of dynamics (or phase transition) detection.

**Definition 3** *Let  $\mathcal{A}$  be an algorithm that finds a change point  $T$  in the interval  $\{s_1, s_2, s_3, \dots, s_n\}$  for a time series  $\mathbf{S}$ . We define that  $\mathcal{A}$ -performance is given in  $\lambda$ -real time as the interval needed to detect the change point  $T$ . So, if an algorithm is said to be  $\lambda$ -real time we have it find the change point  $T$  in the interval  $[t - \lambda, t + \lambda]$ , where  $t - \lambda \leq T < t + \lambda$ . The offline approach considers the whole time series  $\mathbf{S}$  having a  $\lambda$  value approximated to  $n$ , as for the online approach the  $\lambda$  can be a smaller number compared to  $n$ .*

## 4. SAXJS algorithm

In this section, we define our symbolization method, our probability distribution approach, and the distance measure used in this work.

### 4.1. SAX transformation



**Figure 2. SAX whole process using two sequential windows [Lin et al. 2007].**

First, we discretize the time series window  $w_t$  into a symbolic string sequence using the Symbolic Aggregate Approximation (SAX) transformation [Lin et al. 2007]. The SAX process with two sequential windows is shown in Figure 2.

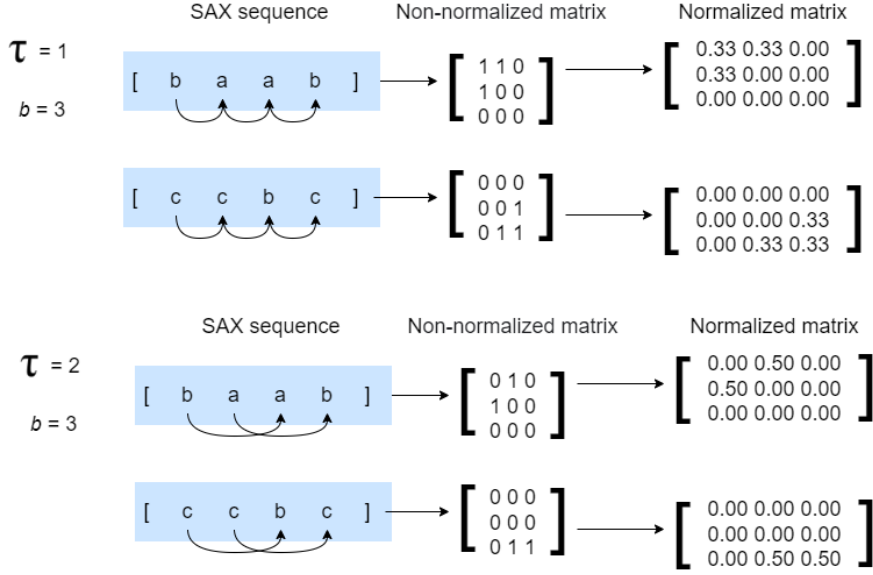
The SAX algorithm considers that since a normalized time series has a Gaussian Distribution [Larzen and Marx 1981], we can determine the breakpoints that divide the time series into  $b$  equal-sizes areas under the Gaussian curve, being  $b$  a user-defined value. The breakpoints are a sorted list of numbers  $B = \{\beta_1, \dots, \beta_{\alpha-1}\}$  such that the area under a  $\mathcal{N}(0, 1)$  Gaussian curve from  $\beta_i$  to  $\beta_{i+1} = \frac{1}{b}$  are all equal-sized.

Therefore, all the time series elements are represented by letters according to their value compared to the breakpoints. For example, if the value is under the smallest breakpoint, it is mapped to the symbol “a”.

Let  $\alpha_i$  represent the  $i^{th}$  element of the alphabet, for example,  $\alpha_1 = a$  and  $\alpha_3 = c$ . Thus, considering the time series  $X \equiv \{x_t\}_{t=1}^n$  of size  $n$  and time  $t$ , the mapping from a time series representation to a word  $\hat{C} = \{\hat{c}_1, \dots, \hat{c}_n\}$  is obtained by  $\hat{C}_i = \alpha_j$ , if  $\beta_{j-1} \leq \bar{c}_i < \beta_j$ .

### 4.2. Transition frequency matrix

After we had the SAX symbolization with  $b$  symbols, we constructed a matrix of the transition frequency of each symbol. As it is shown in Figure 3 for a small window, considering the SAX sequence  $\hat{C} = \{\hat{c}_1, \dots, \hat{c}_n\}$ , we count the sequential connections from one symbol  $\alpha_t$  to another symbol  $\alpha_{t+1}$ . The user-defined  $\tau$  variable determines the jump in the connections; for example, the symbol  $c_t$  will connect with the symbol  $c_{t+\tau}$ . The matrix is converted into a probability matrix by normalizing the values to sum 1. Then, the probability matrix is converted to a 1-dimensional array by row order.



**Figure 3. Example of how we construct the symbols' transition frequency matrix and the effects using different  $\tau$ . The SAX sequences are the same use in Figure 2.**

### 4.3. Jensen-Shannon distance calculation

For each window  $w_t$  and  $w_{t+N}$ , we have the  $\mathcal{P}_t$  and  $\mathcal{P}_{t+N}$  probability distribution of the occurrence of each connection between symbols. The Jensen-Shannon distance between  $\mathcal{P}_t$  and  $\mathcal{P}_{t+N}$  is calculated.

We calculate the Jensen-Shannon distance measure in time stamp  $t$  between the consecutive windows  $w_t = (X_{t-N}, \dots, X_{t-1})$  and  $d(X_t, \dots, X_{t+N})$ , where  $N$  is the user-defined window size. Both consecutive windows move to time stamp  $t + 1$ , and the Jensen-Shannon distance measure is calculated again.

### 4.4. Data smoothing and peak detection

After we calculate the Jensen-Shannon distance we apply the Savitzky–Golay filter. The Savitzky–Golay filter is a digital filter that can be applied to time series in order to smooth the series by fitting a low-degree polynomial with its adjacent data and using a linear least squares method [Press and Teukolsky 1990]. The number of adjacency points  $G$  is a parameter chosen by user definition and the polynomial degree is fixed at 3. In order to detect the peaks we use the `argrelextrema` function in python [Virtanen et al. 2020], which calculates the relative extrema of data considering a neighborhood. For the size of the neighborhood, we set the value of  $N/2$  for both sizes. Therefore, to evaluate if a point  $t$  is a change point, our method needs  $\frac{3N}{2} + G$  points.

## 5. Experiments

In this section, we experimentally investigate the performance of the proposed and other existing change point methods such as KernelCPD [Célisse et al. 2018] [Arlot et al. 2019], RuLSIF [Liu et al. 2013] and TIRE [De Ryck et al. 2021]. The evaluation strategy is

detailed in Section 5.1 and the parameter settings are defined in Section 5.2. First, we investigate the performance of the proposed algorithm on a synthetic data set in Section 5.3 and, then, on the HASC Challenge 2011 database [Kawaguchi et al. 2011b] in Section 5.4.

### 5.1. Evaluation Measure

Following the strategy used in [Liu et al. 2013], the Receiver Operating Characteristic (ROC) curve and the area under the curve (AUC) are calculated using the definition of true positive rate (TPR) and false positive rate (FPR) described in [Kawahara and Sugiyama 2009] as

- True positive rate (TPR):  $\frac{n_{cc}}{n_{rc}}$
- False positive rate (FPR):  $\frac{n_{al} - n_{cc}}{n_{al}}$

where  $n_{cc}$  is the number of times the change point alarm was correct,  $n_{rc}$  is the number of all ground truth change points, and  $n_{al}$  is the number of all change points alarms.

For the synthetic data set, a detection alarm at time  $t$  is considered correct if a true change point exists at step  $t^*$  such that  $[t^* - I, t^* + I]$ , being  $I$  equal to 10 as it is used in [Liu et al. 2013]. In order to avoid duplication, the  $k$ th alarm at the step  $t_k$  if  $t_k - t_{k-1} < 2I$  is removed. The HASC data set was tested with  $I = 10$  and with  $I = 100$ .

In addition to the ROC curve and AUC value, we also evaluate the detection delay, which is the measure of how close each predicted change point is to the actual change point by the following Equation 1 based on [Aminikhanghahi et al. 2018]

$$delay = \frac{\sum_{i=1}^{n_{cc}} Predicted(CP)_i - Actual(CP)_i}{n_{cc}}. \quad (1)$$

### 5.2. Parameter settings

For **SAXJS**, we report the results for two different parameter settings. For both evaluations, synthetic data sets and HASC Challenge 2011, the SAX number of symbols  $b$  varies one by one from 3 to 9, the  $\tau$  value varies one by one from 2 to 15, and the threshold varies from 0.15 to 0.55 by 0.01. The window size  $N$  and the Savitzky–Golay window  $G$  were different in the two-parameter settings, for the synthetic data set  $N$  varies from 50 to 100 by ten, and  $G$  varies from 11 to 41 by 10. For the HASC Challenge,  $N$  varies from 300 to 800 by 100 and  $G = [81, 101, 201]$ . The difference in  $N$  was due to the sizes of both databases and change points distance, as the maximum distance between each change point in the synthetic data sets is 100.

The first baseline method we used was the **KernelCPD**. For both data sets, we tested the three main kernels: linear, cosine, and Gaussian, and the *jump* varied from 1 to 10 by 1. For the synthetic data set, the *minsize* varied from 10 to 50 by 10, and for HASC, it varied from 50 to 150 by 10. Second, we considered the **RuLSIF** method. Because the validation data set in [Liu et al. 2013] largely overlaps with ours, we used the same parameter settings, except for the window size in the HASC evaluation that we consider a variation from 300 to 800 by 100 due to the bigger size of the data set. The data smoothing applied and the set of its parameters was the same as **SAXJS**, which makes the method online. The threshold varies from 0.1 to 2 by 0.1. Next, due to significant data overlap, we use **TIRE** with the same configurations as [De Ryck et al. 2021].



### 5.3. Synthetic data sets

First, we evaluate the algorithm using three synthetic data sets with manually inserted change points suggested in [Liu et al. 2013].

- **Dataset 1 (Jumping mean):** The following 1-dimensional model borrowed from [Yamanishi and Takeuchi 2002] is used to generate 5000 samples (i.e.,  $t = 1, \dots, 5000$ ):

$$y(t) = 0.6y(t-1) - 0.5y(t-2) + \epsilon_t$$

where  $\epsilon_t$  is a Gaussian noise with mean  $\mu$  and standard deviation 1.5. The initial values are set as  $y(1) = y(2) = 0$ , and a change point is inserted every 100-time steps by changing  $\mu$  at time  $t$  according to

$$\mu_N = \begin{cases} 0, & N = 1, \\ \mu_{N-1} + \frac{N}{16}, & N = \{2, \dots, 49\}, \end{cases}$$

where  $N$  is a natural number such that  $100(N-1) + 1 \leq t \leq 100N$ .

- **Dataset 2 (Scaling variance):** The same model as Dataset 1 is used, but a change point is inserted every 100 steps by changing the standard deviation  $\alpha$  at time  $t$  as

$$\alpha = \begin{cases} 1, & N = \{1, 3, \dots, 49\} \\ \ln(\epsilon + \frac{N}{4}), & N = \{2, 4, \dots, 48\} \end{cases}$$

- **Dataset 3 (Changing Frequency):** The model used to generate the series follows the equation  $y(t) = \sin(\omega x) + \epsilon_t$ , where  $\epsilon_t$  Gaussian noise with standard deviation 0.8. A change point is inserted at every 100 points by changing  $\omega$  at time  $t$  as:

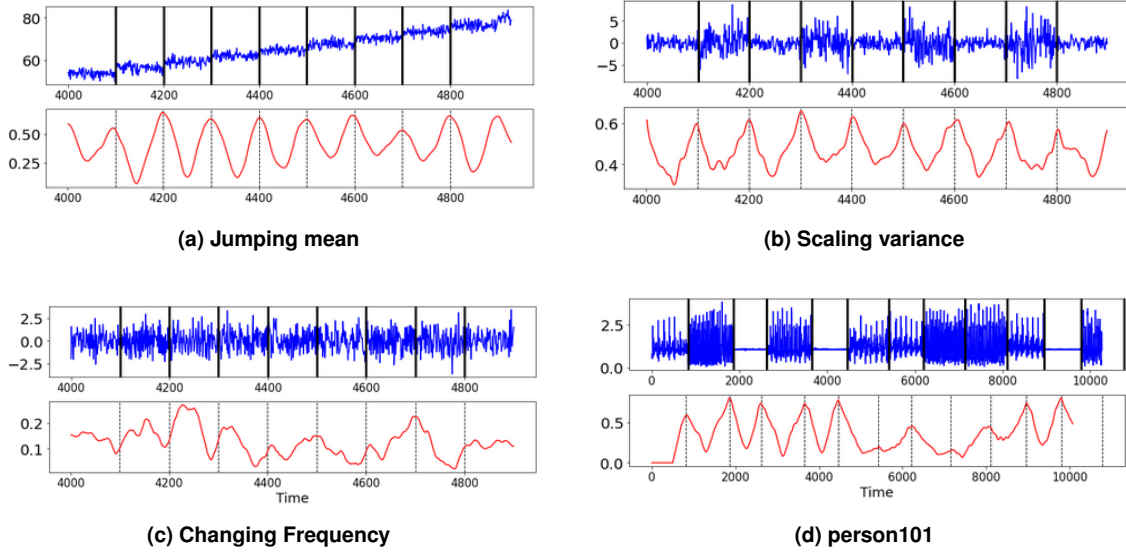
$$\omega = \begin{cases} 1, & N = 1 \\ \omega_{N-1} \ln(\epsilon + \frac{N}{2}), & N = \{2, \dots, 49\} \end{cases}$$

**Table 1. AUC values for synthetic data sets**

	SAXJS	KernelCPD	RuSIF	TIRE
Jumping Mean	<b>0.908</b>	0.830	0.805	0.870
Scaling variance	<b>0.952</b>	0.307	0.739	0.868
Changing Frequency	0.566	0.407	0.299	<b>0.583</b>

In Figures 4a, 4b, and 4c, we plotted samples of the synthetic data sets (blue curves), true change points (black vertical lines), and the change point score (red curves) obtained by SAXJS method. Analyzing the plots, we can see that the algorithm captures the mean and variance changes, while there are some false positives for the frequency dataset, as it is shown in Figure 4c.

In Table 1, we can see the AUC results for SAXJS and other change point methods. The results show that SAXJS has a better performance in the variance and jumping mean dataset and a fair performance detecting frequency changes, which achieves second place under our evaluation, slightly behind the TIRE method. Another advantage of our algorithm is the online approach that does not require to have the whole series information such as KernelCPD and TIRE. The RuSIF implementation used in our work is also an online approach.



**Figure 4. Illustrative time-series samples (upper) and the change point score obtained by our proposed method (lower). The true change-points are marked by black vertical lines.**

**Table 2. AUC values for HASC Challenge using  $I = 10$ .**

	SAXJS	KernelCPD	RulSIF	TIRE
person101	<b>0.884</b>	0.664	0.602	0.074
person102	<b>0.746</b>	0.573	0.175	0.008
person103	0.812	0.710	<b>0.830</b>	0.536
person104	<b>0.817</b>	0.605	0.430	0.010
person105	<b>0.909</b>	0.544	0.574	0.054
<b>Mean</b>	<b>0.834</b>	0.619	0.522	0.136

#### 5.4. HASC Challenge 2011

Next, we evaluate the performance of SAXJS in comparison to other existing change point methods cited in [Liu et al. 2013] using one wearable sensors database: HASC (Human Activity Sensing Consortium) Challenge 2011 [Kawaguchi et al. 2011a].

HASC Challenge is a collaborative project to collect a large-scale human activity database. The project is based on collecting data from accelerometer wearable sensor data of the following activities: "stay", "walk", "jogging", "skip", "stair-up" and "stair-down" [Kawaguchi et al. 2011a]. We tested the algorithm using the first 10 data sets and we take the  $l_2$  - norm of the 3-dimensional accelerometer information following the example in [Liu et al. 2013].

In Figure 4d, we plotted examples of HASC original time series (blue curve), true change points (vertical black lines), and the change point score obtained by SAXJS (red curves). It shows that the algorithm clearly captures all changing behaviors but has difficulty capturing the frequency changes between 5000 to 6000 and 7000 to 8000, which is difficult to visualize even for human eyes.

Observing the results, it is possible to conclude that SAXJS has competitive results although it is not better than KernelCPD in most cases. Table 2 shows the AUC results

**Table 3. AUC values for HASC Challenge using  $I = 100$ .**

	SAXJS	KernelCPD	RuSIF	TIRE
person101	0.949	<b>0.978</b>	0.852	0.863
person102	0.939	<b>0.954</b>	0.939	0.848
person103	0.946	<b>0.997</b>	0.905	0.917
person104	0.952	<b>0.967</b>	0.875	0.917
person105	<b>0.958</b>	0.929	0.863	0.917
<b>Mean</b>	0.949	<b>0.965</b>	0.887	0.892

**Table 4. Delay time comparison for HASC Challenge of each method with best parameters setting according to AUC.**

	SAXJS	KernelCPD	RuSIF	TIRE
person101	18.27	43.25	<b>15.12</b>	38.56
person102	18.90	<b>16.82</b>	32.40	33.12
person103	21.09	12.42	<b>12.20</b>	22.60
person104	15.50	47.75	<b>14.89</b>	31.80
person105	<b>8.55</b>	51.58	9.56	33.12
<b>Mean</b>	<b>16.46</b>	34.36	16.83	31.84

for SAXJS and other change point methods considering  $I = 100$ . But considering a more restrictive interval such as  $I = 10$  showed in Table 3, the SAXJS has better results than other methods due to its low delay to find the change point as shown in the delay time mean in Table 4.

## 6. Conclusion

This paper proposes a novel online change point method that compares the distance between the probability of SAX symbol transitions of two sliding windows. To improve the results, we smoothed the distance results using a Savitzky–Golay filter. First, we evaluate synthetic data sets. Our algorithm showed the best results when we evaluated mean and variance changes, although it did not capture the frequency with the same accuracy. The results for HASC Challenge 2011 were also satisfactory and competitive. Compared to other methods, our results were best when we used a more restrictive delay, which is a neat characteristic. We will investigate the model’s sensitivity to different hyper-parameters and add a complexity analysis comparison for future work. Also, we will extend our analysis to more data sets.

## Acknowledgments

This work was supported in part by São Paulo Research Foundation (FAPESP) grant #2020/05121-4, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brazil (CAPES), in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil (CNPq), and in part by Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) grant #APQ-00426-22.

## References

Adams, R. P. and MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

- Alhammad, N. and Al-Dossari, H. (2021). Dynamic segmentation for physical activity recognition using a single wearable sensor. *Applied Sciences*, 11(6):2633.
- Aminikhanghahi, S. and Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367.
- Aminikhanghahi, S., Wang, T., and Cook, D. J. (2018). Real-time change point detection with application to smart home time series data. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):1010–1023.
- Arlot, S., Celisse, A., and Harchaoui, Z. (2019). A kernel multiple change-point algorithm via model selection. *Journal of machine learning research*, 20(162).
- Benson, L. C., Clermont, C. A., Osis, S. T., Kobsar, D., and Ferber, R. (2018). Classifying running speed conditions using a single wearable sensor: Optimal segmentation and feature extraction methods. *Journal of biomechanics*, 71:94–99.
- Booth, N. and Smith, A. (1982). A bayesian approach to retrospective identification of change-points. *Journal of Econometrics*, 19(1):7–22.
- Bulling, A., Blanke, U., and Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):1–33.
- Celisse, A., Marot, G., Pierre-Jean, M., and Rigail, G. (2018). New efficient algorithms for multiple change-point detection with reproducing kernels. *Computational Statistics & Data Analysis*, 128:200–220.
- Cleland, I., Donnelly, M. P., Nugent, C. D., Hallberg, J., Espinilla, M., and Garcia-Constantino, M. (2018). Collection of a diverse, realistic and annotated dataset for wearable activity recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 555–560. IEEE.
- De Ryck, T., De Vos, M., and Bertrand, A. (2021). Change point detection in time series data using autoencoders with a time-invariant representation. *IEEE Transactions on Signal Processing*, 69:3513–3524.
- Fida, B., Bernabucci, I., Bibbo, D., Conforto, S., and Schmid, M. (2015). Varying behavior of different window sizes on the classification of static and dynamic physical activities from a single accelerometer. *Medical engineering & physics*, 37(7):705–711.
- Gao, W., Emaminejad, S., Nyein, H. Y. Y., Challa, S., Chen, K., Peck, A., Fahad, H. M., Ota, H., Shiraki, H., Kiriya, D., et al. (2016). Fully integrated wearable sensor arrays for multiplexed in situ perspiration analysis. *Nature*, 529(7587):509–514.
- IDTechEx (2021). Wearable technology forecasts 2021-2031.
- Ige, A. O. and Noor, M. H. M. (2022). A survey on unsupervised learning for wearable sensor-based activity recognition. *Applied Soft Computing*, page 109363.
- Kanamori, T., Hido, S., and Sugiyama, M. (2009). A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445.
- Kawaguchi, N., Ogawa, N., Iwasaki, Y., Kaji, K., Terada, T., Murao, K., Inoue, S., Kawahara, Y., Sumi, Y., and Nishio, N. (2011a). Hasc challenge: gathering large scale

- human activity corpus for the real-world activity understandings. In *Proceedings of the 2nd augmented human international conference*, pages 1–5.
- Kawaguchi, N., Yang, Y., Yang, T., Ogawa, N., Iwasaki, Y., Kaji, K., Terada, T., Murao, K., Inoue, S., Kawahara, Y., et al. (2011b). Hasc2011 corpus: towards the common ground of human activity recognition. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 571–572.
- Kawahara, Y. and Sugiyama, M. (2009). Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of the 2009 SIAM international conference on data mining*, pages 389–400. SIAM.
- Kawahara, Y., Yairi, T., and Machida, K. (2007). Change-point detection in time-series data based on subspace identification. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 559–564. IEEE.
- Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.
- Ko, S. I., Chong, T. T., and Ghosh, P. (2015). Dirichlet process hidden markov multiple change-point model. *Bayesian Analysis*, 10(2):275–296.
- Larzen, R. and Marx, M. (1981). An introduction to mathematical statistics and its applications.
- Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.
- Lin, J., Keogh, E., Wei, L., and Lonardi, S. (2007). Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144.
- Liu, S., Yamada, M., Collier, N., and Sugiyama, M. (2013). Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83.
- Nachiar, C. C., Ambika, N., Moulika, R., and Poovendran, R. (2020). Design of cost-effective wearable sensors with integrated health monitoring system. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 1289–1292. IEEE.
- Napier, C., Esculier, J.-F., and Hunt, M. A. (2017). Gait retraining: out of the lab and onto the streets with the benefit of wearables.
- Ni, Q., Patterson, T., Cleland, I., and Nugent, C. (2016). Dynamic detection of window starting positions and its implementation within an activity recognition framework. *Journal of biomedical informatics*, 62:171–180.
- Page, E. (1955). A test for a change in a parameter occurring at an unknown point. *Biometrika*, 42(3/4):523–527.
- Patterson, T., Khan, N., McClean, S., Nugent, C., Zhang, S., Cleland, I., and Ni, Q. (2016). Sensor-based change detection for timely solicitation of user engagement. *IEEE Transactions on Mobile Computing*, 16(10):2889–2900.
- Prajapati, D. and Mahapatra, P. (2009). A new x chart comparable to cusum and ewma charts. *International Journal of Productivity and Quality Management*, 4(1):103–128.

- Press, W. H. and Teukolsky, S. A. (1990). Savitzky-golay smoothing filters. *Computers in Physics*, 4(6):669–672.
- Saatçi, Y., Turner, R. D., and Rasmussen, C. E. (2010). Gaussian process change point models. In *ICML*.
- Selles, R. W., Formanoy, M. A., Bussmann, J. B., Janssens, P. J., and Stam, H. J. (2005). Automated estimation of initial and terminal contact timing using accelerometers; development and validation in transtibial amputees and controls. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(1):81–88.
- Shinmoto Torres, R. L., Ranasinghe, D. C., and Shi, Q. (2013). Evaluation of wearable sensor tag data segmentation approaches for real time activity classification in elderly. In *International conference on mobile and ubiquitous systems: computing, networking, and services*, pages 384–395. Springer.
- Takeuchi, J.-i. and Yamanishi, K. (2006). A unifying framework for detecting outliers and change points from time series. *IEEE transactions on Knowledge and Data Engineering*, 18(4):482–492.
- Thakur, D. and Biswas, S. (2022). Online change point detection in application with transition-aware activity recognition. *IEEE Transactions on Human-Machine Systems*, 52(6):1176–1185.
- Traversaro, F., Redelico, F. O., Risk, M. R., Frery, A. C., and Rosso, O. A. (2018). Bandt-pompe symbolization dynamics for time series with tied values: A data-driven approach. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075502.
- Truong, C., Oudre, L., and Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, 167:107299.
- Verdier, G. (2020). An empirical likelihood-based cusum for on-line model change detection. *Communications in Statistics-Theory and Methods*, 49(8):1818–1839.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, e. a., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.
- Yamanishi, K. and Takeuchi, J.-i. (2002). A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 676–681.
- Zhang, S., Galway, L., McClean, S., Scotney, B., Finlay, D., and Nugent, C. D. (2010). Deriving relationships between physiological change and activities of daily living using wearable sensors. In *International Conference on Sensor Systems and Software*, pages 235–250. Springer.
- Zunino, L., Olivares, F., Ribeiro, H. V., and Rosso, O. A. (2022). Permutation jensen-shannon distance: A versatile and fast symbolic tool for complex time-series analysis. *Physical Review E*, 105(4):045310.