Computers in Biology and Medicine 167 (2023) 107595

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Dimensionality reduction and features visual representation based on conditional probabilities applied to activity classification

Alihuén García-Pavioni *, Beatriz López

Exit Grup, University of Girona, Carrer Universitat de Girona, 6, Girona, 17003, Girona, Spain

ARTICLE INFO

Keywords:
Time series
Feature extraction
Dimensionality reduction
Length-independent
Features visual representation
Time series classification
Activity recognition
Accelerometers
Markov model features
Conditional probabilities
Time series distribution

ABSTRACT

A large part of the information emitted by contemporary technological devices comes in the form of time series. The massive commercialization of these kinds of devices has made the study of time series feature extraction techniques acquire a vital relevance in last years. Two main things are essential when applying feature extraction techniques to time series: to reduce the dimensionality so it occupies the least amount of storage memory possible, and to make features that contain the relevant information regarding the nature of the data set and the goals to be achieved. For this purpose, we propose in this work a brand new technique called the State Changes Representation for Time Series (SCRTS), which relies on the relevant data associated with the conditional probabilities of the time series (also known in the literature as Markov model's features), and the distribution of its values. This method is length-independent, which means that we can apply it to time series of different dimensions obtaining the same number of features for each one. Also, it provides a visual representation of the input data, so it is possible to interpret what makes a certain time series different from the other. After explaining how it works, we apply it to 3 different wearable accelerometer data sets. This algorithm reduces the original dimension of the time series considerably (in the best case from 5499 values to 31), having a good performance in the classification results (in the best chance with an accuracy of 98%).

1. Introduction

The massive commercial usage of wearable devices in the last few years has provided a wealth of data that can be used in many applications, such as activity recognition or health monitoring. Since then, several studies have been carried out applying machine learning algorithms to classify the data that these devices provide to recognize the activities made by the users, as well as to predict emotions, stress, epileptic seizures, heart attacks [1–4], and other diseases such as Parkinson [5], or fall detection in the elderly [6–8], among many others.

All the data output by these devices is in the form of time series, this is a succession of values measured in time and arranged chronologically. Since many of these devices are full-day used, the time series output can be very complex and long, and take up considerable storage space while adding some complications to the performance of the machine learning algorithm. So finding features that can take the relevant information of the time series, though reducing its dimensionality, is a task of concern [9]. However, the sequential and numeric nature of the time series makes this task non-trivial. There are no universal feature selection techniques that work well for all the time series data sets. Which information is relevant and which is not depends on the context of the experiment where the data set comes from and the desired goals to be achieved.

Another important thing when analyzing and hypothesizing about the outputs of an experiment or the nature of a data set is to have a way to visualize and interpret the elements that make a time series different or similar to another. Most machine learning algorithms do not return any interpretable information that allows one to understand their performance, or to come to conclusions about the meaningful elements of the data set. This makes it of great importance to have a method that not only selects the relevant features but also provides a technique to visualize their differences and similarities.

In this paper, we present a simple method of feature extraction and feature visual representation, which we call State Changes Representation for Time Series (SCRTS). Unlike other techniques in the literature, the SCRTS relies on the relevant data associated with the time series "state changes" and the distribution of its values in their respective discretization. These state changes are identified according to the conditional probabilities of passing from one state to another during the time (also known in the literature by the name of Markov Model Parameters [10–12]), that together with what we call "states relevance features", which contains the information regarding the importance of

* Corresponding author.
E-mail addresses: alihuen.garcia@udg.edu (A. García-Pavioni), beatriz.lopez@udg.edu (B. López).

https://doi.org/10.1016/j.compbiomed.2023.107595
Received 20 July 2023; Received in revised form 5 October 2023; Accepted 17 October 2023
Available online 20 October 2023

each state and the distribution of its values, provide the information needed to represent or characterize a time series. This method is length-independent, which means that no matter what the dimensions of the different time series are, we can always apply this technique, and after that, all the vectors made will be of the same size.

We developed this technique thinking mainly in terms of working with time series accelerometer data, so we tested our method with activity recognition data sets. The results obtained are good. In the best cases, we were able to reduce the dimension of the frames from 5499 to 41, having an accuracy of 88%, which confirms that this method extracts relevant information regarding this kind of time series, in this particular context, and with a considerable reduction of the dimensionality, which could be of great utility when dealing with storage capacity problems. But we also believe that the time series coming from accelerometer devices is not the only time series that this method could work well with. This method could be applied in any context where time series plays a fundamental role, as when wearable sensors are used to predict heart attacks, epileptic seizures, stress, or anxiety, among others. The reader can find in this article all the explanations of how this method works and why, so he or she can understand it and conclude if it could be useful or not in some other contexts of interest.

Additionally, an important characteristic of this technique is that enables one to make a comparison of the time series via a visual interpretation of the features. This important characteristic of the SCRTS method can contribute to a better understanding of the time series data.

This work is divided into 5 main sections. In Section 2 we present the related works that we consider most relevant and we explain how we think that our method could contribute. In Section 3 we explain our methodology detailing the reasons for the feature selection made. In Section 4 we present the data sets used to test our method and show and analyze the results. In Section 5 we explain the features visual representation technique and test it with one of the data sets presented in the last section. In Section 6 we discuss the possible scopes and limitations of our method, and we talk about which other different contexts the SCRTS could be applied in addition to activity recognition. Finally, in Section 7, we present the final conclusions.

2. Related work

Time series feature extraction (TSFE) is essential for machine learning effectiveness when applied to time series problems, for two main reasons: it reduces the storage space, and it allows the machine learning algorithm to work only with the relevant data so that it can improve its performance.

The problem is how to know which is the relevant data. What is relevant and what is not depends on what is wanted to be achieved and the context related to the data set. There exist many ways to do TSFE, and each one selects different kinds of features, so their performance depends on how well these features represent the information considered relevant in the context of the experiment.

The Fourier Transform [13] and the Wavelet Transform [14], which are very classical, could be very useful applied to time series composed mostly of periodic waves, as it happens with the EEG signals [15] or the signals related to the light, the electricity, the image, or the sound, among others. But when it comes to analyzing time series that do not present a periodic behavior, such as the data extracted from accelerometers, these methods may not take into account features that could be of importance for the machine learning algorithm performance.

There exist other classical statistical methods for feature extraction like the Singular Value Decomposition (SVD) [16], the Principal Component Analysis (PCA) [17], or the Linear Discriminant Analysis (LDA) [18]. These methods use Linear Algebra tools for reducing the information in the data set. They work well for static data, but when it comes to time series, they may also lose some relevant information depending on the context.

Other techniques create new features to represent the time series but combine different mathematics elements that are typically used to measure some particular properties that are not often related to feature extraction techniques. Using mathematical properties as features has been shown to achieve good performances in several cases. In [19], distance measures like dynamic time warping are combined with feature-based methods like SAX to create new features, showing good results.

Mathematical tools such as probabilities are often used for understanding situations related to the possible changes in events, but if we think of a time series as some sort of situation where its values are the possible events, then the probabilities can describe the possible changes and, therefore, the behavior of the time series. In [20] conditional probabilities are used to create a measure that allows to discover some intra and inter-temporal patterns. These patterns are used as features in a machine learning algorithm, having a good performance.

Using conditional probabilities as features has shown to work well in a number of contexts: in [12] conditional probabilities are used as features in classifications using logistic regression to separate schizophrenia patients from healthy patients; in [11] this technique is used to take the relevant information from waist-worn accelerometers from 22 toddlers to recognize the activity they performed, using random forest; and in [10] conditional probabilities are used to select the relevant information from many days of wearable device data from users for monitoring their circadian rhythmicity.

These works achieved good results in the final performance, showing the relevance that the conditional probabilities could have when used as the feature inputs. They refer to the conditional probabilities' vector as hidden Markov models because a hidden Markov matrix is composed of all the conditional probabilities of the system, so it is the same thing. We refer to them as conditional probabilities for simplicity reasons related to the vocabulary used in this work.

With the SCRTS technique, we explore the scope of using conditional probabilities as features together with some other features related to the distribution of the time series values along the states made in the discretization. We explain the mathematics of this technique and why we use them, so the reader can then draw his or her own conclusions about what other contexts this technique might or might not work well with. This technique comes also with a visual interpretation of the features which allows us to find differences and similarities between the time series, which we believe could be very useful for understanding the nature of the data set, even if other feature selection technique is used.

3. Methodology

The SCRTS algorithm consists of 6 different steps (shown in Fig. 1) that we describe in this section. We leave the details of the method for the visualization and interpretation of the results for Section 5.

3.1. Data collection

Accelerometers are devices that measure the acceleration of movement along the x, y, and z axes. This means that each accelerometer returns 3 values with a frequency of τ seconds. There are many ways in which we could use these data as inputs to our algorithm, but in this paper, we limit ourselves to taking a single value for each of these three values, and we refer to it as the *vector magnitude*, which is defined as

$$v = \sqrt{(a_x)^2 + (a_y)^2 + (a_z)^2}, \tag{1}$$

were a_x, a_y, a_z are the accelerations measured by the device in axis x, y, z respectively.

In case we had more than 3 values every τ seconds, it could be used the same vector magnitude, but instead of using just 3 values, we use more. This is the case of the AReM data set that we describe in Section 4.1.3, where we use 3 dispositives outputting 3 values each. In this case, we had 9 values, so the vector magnitude used is

$$v = \sqrt{(a_1)^2 + (a_2)^2 + \cdots + (a_9)^2}. \tag{2}$$

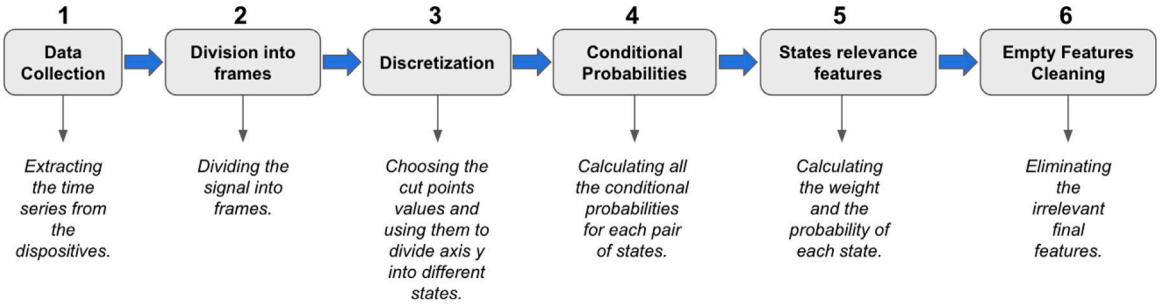

Fig. 1. SCRTS steps.

3.2. Division into frames

We divide every signal into k frames $F_1, F_2, \ldots, F_k$, so we use them for the training-test set classification algorithm. It is to note that, with the SCRTS it is not necessary for all the frames to be of the same size. As we already mentioned, our method is length-independent, which means that the final classification result will not be affected if one or more frames are of different lengths from the rest.

If the data set used is not too big it is possible to use overlapping, so the final number of frames is higher.

For every frame F we refer to its dimension with the letter d, which is equal to the number of samples in it. Then, we can denote each frame F as a vector composed by its vector magnitudes v_i in chronological order, to be more precise,

$$F = (v_i)_{i \leq d}. \tag{3}$$

3.3. Discretization

In this step of our method, we label the vector magnitudes values, obtaining a sequence of domain-dependent *states* from the numerical time series, so then we can represent each frame as a sequence of states. This technique is known as *time series discretization*, and there are many different ways to do it [21,22].

Any discretization technique could work with the SCRTS. The important thing is to represent each frame as a sequence of states. We do not have an optimized way of selecting the discretization or the number of states. A good choice of these elements can lead to a good performance of this technique, and vice versa. In activity recognition, for example, a discretization that works well will be one where every state represents a different movement because every activity is a combination of different movements.

For simplicity, in this work we do a partition of axis y so then we assign one of these partitions to every vector magnitude. In other words, if we look at the time series, with axis y being the vector magnitude values and axis x the time, then we divide the y axis in different *cut points*, so they represent the beginning or the end of an interval. So each interval represents a state, as shown in Fig. 2. There are as many ways to make this division of the y axis as there are possible discretizations, and it depends on the essence of the data set which could output better results.

Given a set of cut points

$$CP = \{cp_0, cp_1, \ldots, cp_n\}, \tag{4}$$

such that $cp_0 < cp_1 < \cdots < cp_n$, we can generate a set Σ of n states, each state $S_i \in \Sigma$ representing an interval of the vector magnitudes values made by the cut points, as follows:

$$\begin{aligned} S_1 &= [cp_0, cp_1), \\ S_2 &= [cp_1, cp_2), \\ &\vdots \\ S_n &= [cp_{n-1}, cp_n]. \end{aligned} \tag{5}$$

Then, we say that a vector magnitude v *is of state* S_i if $v \in S_i$, in other words, if $cp_{i-1} \leq v \leq cp_i$. Thus, given a set of states Σ, we can represent each frame F by a sequence of states

$$S(F) = \{S^1, S^2, \ldots, S^d\}, \tag{6}$$

where $S^t \in \Sigma$, and the supra-index t indicates the chronological position of the state in the frame and d the frame's dimension (as we said in Eq. (3)).

3.4. Conditional probabilities

In probability theory, the *conditional probability* is a measure of the probability of an event occurring knowing that another event has already occurred [23]. In this method, we use conditional probabilities as part of the features representing the frames. The reason for that is because the conditional probabilities reflect the "jumps" from one state to another, giving a description of the changes or the "stays", showing which jumps were more common in each frame, and which states stay longer without changing. This provides good information about the "behavior" of the time signal because it describes how the intensity of the movements changed over the period, thus allowing the machine learning algorithm to easily find differences between the activities performed. It also provides a graphical and easy way to visualize these differences, as we will see in Section 5.

Given a frame F of our time-signal represented by $S(F) = \{S^1, S^2, \ldots, S^d\}$, the *conditional probability* of getting state S_b after being in state S_a in F, with $S_a, S_b \in \Sigma$, is defined as

$$\mathrm{P}_F(S_b \mid S_a) = \frac{\mathrm{Car}_F(S_a, S_b)}{\mathrm{Car'}_F(S_a)}, \tag{7}$$

being $\mathrm{Car}_F(S_a, S_b)$ the cardinal of (S_a, S_b) in $S(F) = \{S^1, S^2, \ldots, S^d\}$, that is, the number of times that S_a is followed by S_b in $S(F)$; and $\mathrm{Car'}_F(S_a)$ the number of times that S_a appears in $\{S^1, S^2, \ldots, S^{d-1}\} \subset S(F)$.

Therefore, we calculate all the conditional probabilities for each frame, which gives us a total of n^2 features per frame in each case (being n the total of states, as we mentioned in Eq. (4)). Thus, we use these features to make a new vector for representing the information contained in frame F. We call $C(F)$ to the vector made with all the conditional probabilities of F. That is,

$$C(F) = \{\mathrm{P}_F(S_i \mid S_j) : i, j \leq n\}. \tag{8}$$

$C(F)$ is also known as Stochastic Matrix, or Markov Matrix. But in this case, for organizational reasons, instead of using a matrix, we represent $C(F)$ as a vector.

3.5. States relevance features

Although $C(F)$ has the information about the "jumps" and "stays" of the states in the time series, it does not say anything about the "relevance" of each state in the frame. In other words, the conditional probabilities have the information about which state changes are more

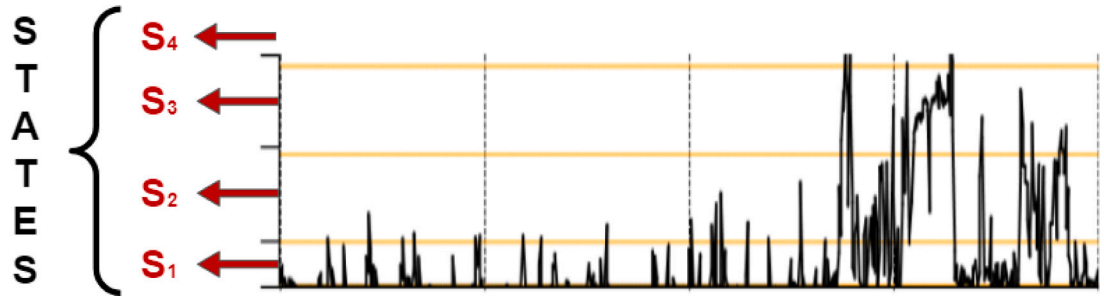

Fig. 2. A piece of a time signal, with axis y being the size of v and axis x the time. In this case, the axis y is divided into 4 different states, but any number of states is possible.

likely to happen in $S(F)$, but they do not have the information about which states are more likely to appear in it. However, if we want to create a vector that contains most of the state's changes relevant information so that the machine learning algorithm can have a good performance, then we should probably include relevant data regarding the states' appearance in the frame. To that end, we make use of two features that are related to each other but have some considerable differences: the *state's probabilities*, and the *state's weights*.

3.5.1. State's probabilities

As is well known by probability theory [23], for each $S_i \in \Sigma$, the *probability of state* S_i to come out in frame F is:

$$\mathrm{P}_F(S_i) = \frac{\mathrm{Car}_F(S_i)}{d}, \tag{9}$$

with $\mathrm{Car}_F(S_i)$ being the number of times that S_i appears in $S(F)$, and d being the dimension of $S(F)$. Therefore, we refer to the set of all the state's probabilities of a frame F as $P(F)$, that is to say

$$P(F) = \{\mathrm{P}_F(S_1), \mathrm{P}_F(S_2), \dots, \mathrm{P}_F(S_n)\}. \tag{10}$$

Then, $P(F)$ has the information about how many times each state has appeared in $S(F)$.

3.5.2. State's weights

We are also interested in the information regarding the distribution of the vector magnitudes along their respective states for each frame F. To this end, first of all, we need a function that works as a measurement of how close a vector magnitude is to the midpoint of its respective state. In other words, we are looking for a function $f_i : [cp_{i-1}, cp_i] \to [0, 1]$ such that the closer a vector magnitude v is to the midpoint of its respective state S_i, the closer $f_i(v)$ is to 1; and the closer v is to the border of S_i, the closer $f_i(v)$ is to 0. So then, for each state S_i we can sum all the values $f_i(v)$ for all v in S_i and normalize the result. On one hand, this sum would express the distribution of the vector magnitudes in S_i; and on the other hand, if we make a comparison between all the states, it would express the importance, or "weight" that state S_i has in frame F.

In other words, for every state $S_i \in \Sigma$, we look for a *radial function* [18] $f_i : [cp_{i-1}, cp_i] \to [0, 1]$, such that

$$\begin{cases} f_i(cp_{i-1}) = f_i(cp_i) = 0, \\ f_i(\mathrm{mid}_i) = 1, \end{cases} \tag{11}$$

being cp_{i-1} and cp_i the boarders of the state S_i (as we already mentioned in Eq. (5)), and mid_i the midpoint of S_i, to be more precise,

$$\mathrm{mid}_i = \frac{(cp_i + cp_{i-1})}{2}. \tag{12}$$

Although any radial function would work well, for simplicity we use the one that we call the *normalized inverted distance* (NID), which we define next. First of all, we define the *distance from the midpoint to the top* of state S_i as

$$\mathrm{dis}_i = |\mathrm{mid}_i - cp_i|, \tag{13}$$

or, what is the same,

$$\mathrm{dis}_i = |\mathrm{mid}_i - cp_{i-1}|. \tag{14}$$

Now, for every v belonging to a state S_i, the normalized inverted distance of v to its respective midpoint mid_i of S_i, is

$$\mathrm{NID}_i(v) = 1 - \frac{|\mathrm{mid}_i - v|}{\mathrm{dis}_i}. \tag{15}$$

So, it is easy to see that the normal inverted distance is a radial function that satisfies Eq. (11), in other words,

$$\begin{cases} \mathrm{NID}_i(cp_{i-1}) = \mathrm{NID}_i(cp_i) = 0, \\ \mathrm{NID}_i(\mathrm{mid}_i) = 1. \end{cases} \tag{16}$$

Thus, let us say that $Q_F(S_i) = \{v_1, v_2, \dots, v_q\}$ is the set of all the vector magnitudes of the frame F laying in state S_i, then, we finally define the *weight* of state S_i in F as follows:

$$\mathrm{W}_F(S_i) = \begin{cases} \dfrac{\sum_{j=1}^{q} \mathrm{NID}_i(v_j)}{d}, & \text{if } Q_F(S_i) \neq \emptyset; \\ 0, & \text{if } Q_F(S_i) = \emptyset. \end{cases} \tag{17}$$

Therefore, we refer to all the state weights of a frame F as $W(F)$, that is to say,

$$W(F) = \{\mathrm{W}_F(S_1), \mathrm{W}_F(S_2), \dots, \mathrm{W}_F(S_n)\}. \tag{18}$$

3.6. Empty features cleaning

The SCRTS is, as its name indicates, a method for representing time series. In other words, our goal is to extract all the relevant data of the frames so they can be represented as vectors for a machine learning algorithm. We construct these vectors with the conditional probabilities $C(F)$, the state's probabilities $P(F)$, and the state's weights $W(F)$. We call the *representation vector* $R(F)$ to the vector made with $P(F), C(F)$ and $W(F)$. The dimension of these vectors depends on the number of states, so let us call $dim(V)$ to the function that returns the dimension of a vector V, then:

$$dim(C(F)) = n^2; \tag{19}$$

$$dim(P(F)) = dim(W(F)) = n. \tag{20}$$

So, till now, the dimension of $R(F)$ is:

$$dim(R(F)) = n^2 + 2n. \tag{21}$$

As we can see, the dimension of $R(F)$ depends only on the number of states n, and is independent of d, the original dimension of F, which means that our method is length-independent.

Therefore, the training-test matrix for the machine learning will have all the $R(P)$ vectors of each frame F as rows. This means there will be a column for each conditional probability, for each probability, for each weight, etc. So if one of these columns has more than 75% of zeros, it means that the feature represented in this column is probably irrelevant for representing the time series, and it could also bring some noise to the machine learning performance. Therefore, what we do in

Table 1
Data sets descriptions.

Data set	Sample frequency (Hz)	Num. of participants	Num. of activities	Num. of states
Wristband acc.	0.1	8	2	5
Chest acc.	52	15	5	6, 7, 8
AReM	60	15	5	7, 9, 11, 13

Table 2
Data sets with their respective ANN architectures: number of hidden layers, number of nodes, activation functions, optimizers, learning rates, loss functions, and number of epochs.

Data set	Hidden layers	Nodes	Act. f.	Opt.	Learning r.	Loss f.	Epochs
Wristband acc.	8	12	Relu/Adam	Adam	0.001	binary c.	40
Chest acc.	3	64/16	ReLU/Sotmax	Adam	0.001	binary c.	40
AReM	3	64/16	ReLU/Sotmax	Adam	0.001	binary c.	40

this process called "empty features cleaning", is to seek these columns (the ones with more than 75% of zeros) and eliminate them.

This process would probably reduce the dimension of the training-test matrix even more. As a result, the representation vector of each frame would probably reduce its dimensionality. In other words, we can rewrite Eq. (21) as

$$dim(R(F)) \leq n^2 + 2n. \tag{22}$$

In most cases the reduction of the dimension of the training matrix after this process is considerable, as we show in Section 4.2. The magnitude of this reduction depends on the distribution of the vector magnitudes along the different states, so it varies according to the context.

4. Results

To test our method, we used three different accelerometer databases. The good thing about accelerometers data is that every activity is made up of different motions, so each motion can be seen as a state, so the features made with the SCRTS method express how these motions were combined in each activity, what can lead to the machine learning algorithm to find clear differences between them.

In all the cases we labeled the frames of the activity to be classified with 1, and the rest of the activities with 0, and once we had done the training and the test set, we applied random oversampling [24] in the training set to level the number of frames labeled with 1 with the ones with 0. We trained an artificial neural network (ANN), and the accuracy, the true positive rate (TPR), and the true negative rate (TNR), were calculated. This procedure was executed 20 in each case, and the final results were calculated as the average of the results obtained in each of the 20 performances.

4.1. Data sets

In this section we present the data sets used in this work for experimental reasons (shown in Table 1) and the ANN used for the classifications (shown in Table 2).

4.1.1. Wristband accelerometers data set

This data set [25] was collected from an experiment we conducted, where 8 participants used an Actigraph accelerometer wristband for approximately 9 days, having a total of 1582 h of time series data. The sampling frequency used in the accelerometers was $\tau = 0.1$ Hz. The goal was to identify the frames where the person worked at the office.

We made 3 different experiments to test different ways of using our technique in this data set. For the first experiment classification, we divided the time series into 15-minute frames (where the dimensions of the frames are $d = 90$), the second into 30-minute frames ($d = 180$), and the third into 60-minute frames ($d = 360$).

For the discretization, we tried with the different cut points provided by Actilife, which resulted in the usage of different numbers of states. After comparing the final results, we chose the Freedson Adult 1998 cut points [26], because of its performance, which gives us a total of 5 states (i.e., $n = 5$).

We applied our method to represent each frame with its respective vector $R(F)$, and then we randomly shuffled them together and split them into the training set (75%) and the test set (25%).

The ANN used had 8 hidden layers of 12 nodes and the Relu activation function in each layer. The output layer was dense with 2 nodes and the Sigmoid activation function. The optimizer was Adam with a learning rate of 0.001 units; the loss function was the binary crossentropy and the number of epochs was 40.

4.1.2. Chest accelerometers data set

This data set [27] collects data from single wearable accelerometers mounted on the participants' chests. 15 participants performed 5 different activities to be classified: working at the computer, standing, walking, climbing stairs, and talking while standing. The sampling frequency used in the accelerometers was $\tau = 52$ Hz.

We divided each activity's time series into 3 frames. The frames turned out to have different dimensions from each other in most cases (which is not a problem, because, as we already said, our algorithm is length-independent), with an average of $\overline{d} = 5499$.

We performed a k-means clustering with all the vector magnitudes of all the time series to choose the states' cut points. After trying with different numbers of states, we chose 6, 7, and 8 states for the time series discretization (i.e., $n = 6$, $n = 7$, and $n = 8$) because of its performance in the final results.

We made one leave-one-out classification for each activity, leaving one participant for the test and using the rest to train the ANN.

The ANN used had 3 hidden layers: the first one with 64 neurons and the ReLU activation function, and the second and third ones with 16 neurons, also with the ReLU activation function. After the last hidden layer, there is a dropout of 0.2 units. The output layer was a dense layer with 2 nodes and the Softmax activation function. The optimizer was Adam with a learning rate of 0.001 units; the loss function was the binary crossentropy and the number of epochs was 40.

4.1.3. AReM data set

This data set [28] collects the data from approximately 15 individuals using three accelerometers: one at each ankle, and one in the chest. Unlike the wristband accelerometers data set and the chest accelerometers data set, in this data set 9 values were used for the vector magnitude, as shown in Eq. (2). A total of 5 activities were performed to be classified: bending, cycling, lying, sitting, and walking. The sampling frequency used in the accelerometers was $\tau = 60$ Hz.

As with the chest accelerometers data set, we performed k-means clusterings for generating the states for the time series discretization, but in this case, we tried from 3 to 13 clusters (i.e., $n = 3$ to 13). After this, we chose the number of states with the best performance, which are 7, 9, 11, and 13.

We ran 10 experiments for each activity made, one for each discretization. We did not apply any division to the original frames,

Table 3
Results using the SCRTS method with the wristband accelerometers data set.

T (min.)	States	d_i	d_f	Acc. (%)	TPR (%)	TNR (%)
15	5	90	8	79	83	78
30	5	180	8	81	82	81
60	**5**	**360**	**12**	**84**	**81**	**85**

Table 4
Results using the SCRTS method with chest accelerometers data set.

States	$\overline{d}_i$	d_f		A1	A2	**A3**	A4	A5
			Acc. (%)	66	67	**88**	69	69
6	5499	**31**	TPR (%)	46	52	**71**	45	38
			TNR (%)	71	67	**91**	75	77
			Acc. (%)	68	71	**85**	74	74
7	5499	34	TPR (%)	46	43	**73**	44	41
			TNR (%)	74	77	**88**	81	82
			Acc. (%)	**67**	**72**	**88**	**74**	**76**
8	**5499**	**41**	**TPR (%)**	**41**	**40**	**77**	**38**	**39**
			TNR (%)	**74**	**80**	**90**	**83**	**85**

instead, we used each frame as originally was. The original dimension of the time series of each activity was $d = 480$.

For each classification, we applied our algorithm to represent each frame with its respective vector $R(F)$, and then we randomly shuffled them together and split them into the training set (75%) and the test set (25%).

For each classification, we used the same ANN architecture used with the chest accelerometers data set, already detailed.

4.2. Performance results

In this section, we show the downstream impact of our method when using the features obtained in a classification task (activity recognition). As all the data sets are different from each other, we decided to try different ways of testing our algorithm depending on the data set, so in each experiment, the results are presented in different ways.

In Table 3 the results of the wristband accelerometers data set are provided. We compare the dimension of the frames before applying the SCRTS method (d_i), the dimension of the frames after applying the SCRTS method (d_f), the accuracy (Acc.), the true positive rate (TPR), and the true negative rate (TNR) according to different values of the frame lengths $T = 15$, $T = 30$, and $T = 60$ min. We can see that, with our method, frames of $T = 60$ min perform better than doing it with $T = 15$ and $T = 30$ min.

In Table 4 we show the average of the frame dimensions before applying our method ($\overline{d}_i$), the dimension of the frames after applying our method (d_f), the accuracy, the TPR and the TNR for each activity classification made with chest accelerometers data set. These activities are: working at the computer (A1), standing (A2), walking (A3), going up/down stairs (A4), and talking while standing (A5). We present the results of the classifications made with 6, 7, and 8 states, which achieved the best performance after trying with 3 to 9 states. As we can see, using 8 states performs better than 6 and 7 states, and the dimensionality reduction is also remarkable. The activity that has performed better with all the discretizations is walking (A3).

In Table 5 we show the activities of the classification made with AReM data set, the number of states used that had the best performance, the dimension of the frames before applying our method (d_i), the final dimension obtained after applying our method (d_f), the TPR, the TNR and the accuracy of the classification performance. The best results are with the activity "cycling", followed by "walking" and "laying". The difference between the performance of this data set and the last two is remarkable, and this may be due to several reasons, among them: the usage of 3 accelerometers instead of one, outputting 9 values instead of just 3; wearing the dispositive on the ankle instead of the wrists, which have shown to have a better performance in other experiments as well [29]; or maybe because the experiment presents less noise.

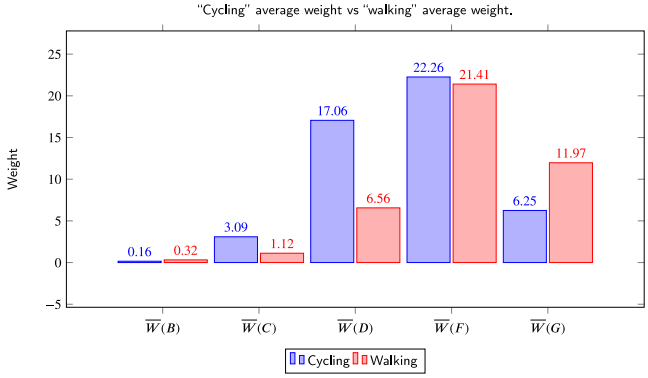

Fig. 3. Comparison of average weights for cycling (left) and walking (right).

5. SCRTS features visual representation

The problem with making a classification using ANNs is that ANNs do not provide an explanation or an easy interpretation of what kind of aspects make a class different from the others. But using the SCRTS algorithm allows us to visualize some attributes that could be very useful when drawing conclusions. Next, we explain how this features visual representation technique works.

5.1. Average weights comparison

Let us $S_r \in \Sigma$ being a state, and $F_1, F_2, \ldots, F_k$, the frames division of some activity in the data set, then we can define the *average weight* of state S_r as

$$\overline{\mathrm{W}}(S_r) = \frac{\mathrm{W}_{F_1}(S_r) + \mathrm{W}_{F_2}(S_r) + \cdots + \mathrm{W}_{F_k}(S_r)}{k}.$$

So, $\overline{\mathrm{W}}(S_r)$ represents the average of the weights of all the frames in their respective activity, and it has a value between 0 and 1 for all $S_r \in \Sigma$. So we can compare these averages and guess which states were more weighty in each activity.

In Fig. 3 we show the comparison between the average weights of the activities "cycling" and "walking" from AReM data set, using 7 states (A, B, C, D, E, F, G). For simplicity we multiplied the original average weights by 100, then the values in the graphic are not between 0 and 1 as originally stated, but between 0 and 100, so the differences and similarities are bigger and easier to be seen.

The vector magnitude cut points of these 7 states are: A: [0, 15.88]; B: [24.77, 34.9); C: [34.91, 40.68); D: [40.68, 45.08); E: [45.08,

Table 5
Results using the SCRTS method with AReM data set.

Activity	States	d_i	d_f	TPR (%)	TNR (%)	Acc. (%)
Bending	7	480	33	87	97	95
Cycling	**13**	**480**	**71**	**100**	**100**	**100**
Laying	**13**	**480**	**71**	**93**	**98**	**97**
Sitting	9	480	47	68	90	85
Walking	**11**	**480**	**63**	**100**	**98**	**98**

49.26); F: [49.26, 54.84); G: [54.84, 67.19). In Table 5 we only showed the results of classifying the activity "bending" using these cut points, but we showed the results of classifying "cycling" and "walking" using 13 and 11 states respectively, instead of 7. The reason we use 7 states for showing the visual representation of the features in this section is that for the comparison we need to use the same cut points in the activities frames, and using 7 states we obtained a TPR of the 88%, a TNR of the 98% and an accuracy of the 97% for "cycling"; and a TPR of a 100%, a TNR of the 97% and an accuracy of the 98% for "walking", which in the classification of both "cycling" and "walking" as a whole, has a better performance than using 13 or 11 states.

As we can see in Fig. 3, after the empty features cleaning process (Section 3.6), only 5 states' weights were relevant for the training-test matrix: $\mathrm{W}(B), \mathrm{W}(C), \mathrm{W}(D), \mathrm{W}(F)$ and $\mathrm{W}(G)$.

We can see that both "cycling" and "walking" have similar weights for state F, but there are several differences between the weights of states D and G. These similarities and differences can help us to understand and make conclusions about the essence of the activities. For example, if we had to make a classification having only these two activities, we could look at the weights of each frame and conclude:

- if the frame's activity has more than a value of 11 in $\overline{\mathrm{W}}(D)$ and less than 9 in $\overline{\mathrm{W}}(G)$, then it is probably "cycling";
- if the frame's activity has less than a value of 11 in $\overline{\mathrm{W}}(D)$ and more than 9 in $\overline{\mathrm{W}}(G)$, then it is probably "walking".

In this example, we only use the weight features of states D and G for making the differentiation, because it is where the differences are most visible, but to differentiate from other activities of the data set, the other weight states features would probably be needed.

We can also look at this graphic and interpret why one activity is different from the other. For example, if we think as states B and C as light movements, D as moderate movements, F as aggressive movements, and G as very aggressive movements, we can conclude that:

- both activities have very few moments doing light movements;
- cycling has more moderate movements than walking;
- both have many moments doing aggressive movements;
- walking has more "very agressivve" movements than cycling.

It should be noted that accelerometers do not measure the speed of the movement but the acceleration, which is the change of the speed or direction of the movement. This is probably why walking has more "very aggressive movements" than cycling, because the ankles when walking change the direction of the movement faster, while, when cycling, the movement is more constant.

5.2. Probabilities comparison

With this method we can also compare the probability features of the different activities, looking for similarities and differences that help us to understand the classification made.

Let us $S_r \in \Sigma$ being a state, and $F_1, F_2, \ldots, F_k$, the frames division of some activity in the data set, then we can define the *mean probability* of state S_r as

$$\overline{\mathrm{P}}(S_r) = \frac{\mathrm{P}_{F_1}(S_r) + \mathrm{P}_{F_2}(S_r) + \cdots + \mathrm{P}_{F_k}(S_r)}{k}.$$

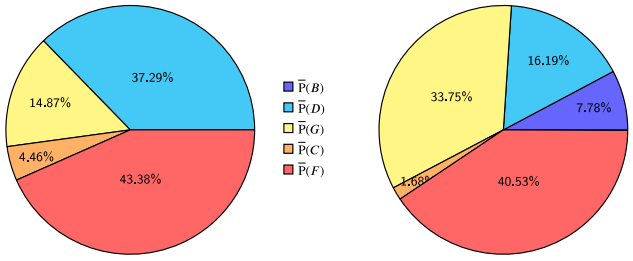

Fig. 4. Mean probabilities of cycling (left) vs. walking (right).

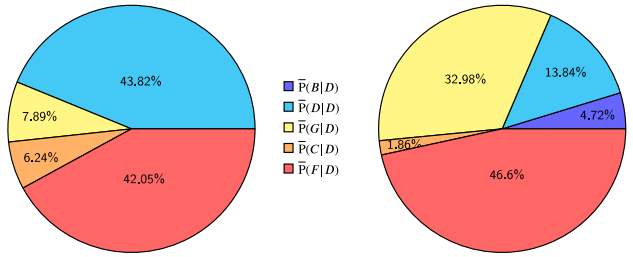

Fig. 5. Comparison of the D-state conditional probabilities of cycling (left) vs. walking (right).

It is easy to mathematically prove that, if we sum the mean probabilities of all the n states in Σ, this is equal to 1, that is

$$\overline{\mathrm{P}}(S_1) + \overline{\mathrm{P}}(S_2) + \cdots + \overline{\mathrm{P}}(S_n) = 1. \tag{23}$$

This allows us to visualize all the mean probabilities in a pie chart. Then, we can compare the mean probabilities of different activities to get conclusions about their differences and similarities, as we already did with the average weight.

In Fig. 4 we show the pie chart of the mean probabilities from the activities "cycling" and "walking". Instead of using the values between 0 and 1, we multiplied the probabilities by 100 so that we could put them in the pie chart as a percentage.

As we can see, there is a correlation between the values of the average weights and the mean probabilities, but while the second one gives us an idea of the times that a state appears in the frames, the second one gives us an idea of the dispersion of the vector magnitudes along the different cut points.

The same can be done with the conditional probabilities. Let us $S_r, S_q \in \Sigma$ being states, and $F_1, F_2, \ldots, F_k$, the frames division of some activity in the data set, then we define the *mean conditional probability* of getting state S_q after being in state S_r as

$$\overline{\mathrm{P}}(S_q|S_r) = \frac{\mathrm{P}_{F_1}(S_q|S_r) + \mathrm{P}_{F_2}(S_q|S_r) + \cdots + \mathrm{P}_{F_k}(S_q|S_r)}{k}.$$

We can also mathematically prove that the sum of all the mean conditional probabilities is equal to 1, to put it in another way,

$$\overline{\mathrm{P}}(S_1|S_r) + \overline{\mathrm{P}}(S_2|S_r) + \cdots + \overline{\mathrm{P}}(S_n|S_r) = 1, \tag{24}$$

for all $S_r \in \Sigma$. So, as we did with the mean probabilities, we can compare the mean conditional probabilities of the different activities using pie charts.

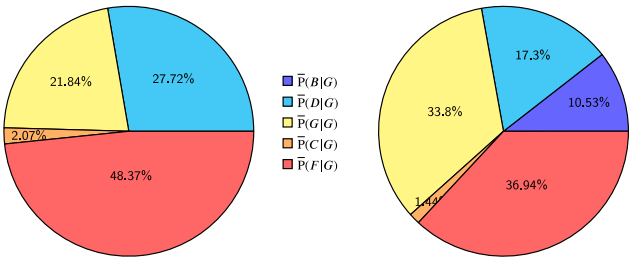

Fig. 6. Comparison of the G-state conditional probabilities of cycling (left) vs. walking (right).

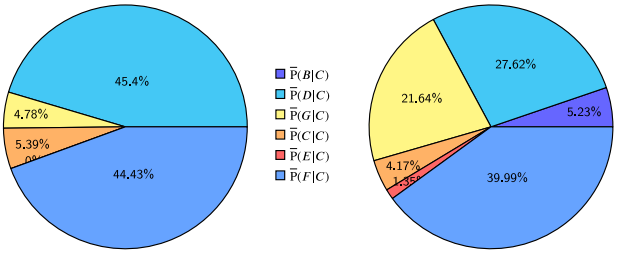

Fig. 7. Comparison of the C-state conditional probabilities of cycling (left) vs. walking (right).

In Figs. 5, 6, 7, and 8 we show the pie charts of the mean conditional probabilities of the activities "cycling" and "walking". As we did with the mean probabilities, we multiply the results by 100 so we can see them as percentages.

In some of these pie charts, there were some probabilities with values minor than 1%, so since these values are negligible with respect to the rest of the probabilities, adding difficulties to visualizing the data, we have decided to remove them and distribute these small values among the rest of the probabilities of its respective pie charts, so then we have a better visual representation.

We could do the same that we did with the example of the weights, in other words, we could make a comparison between each pie chart and conclude some of the percentages that a frame should have to be classified as "cycling" or "walking".

For example, we could see in Fig. 6 all the conditional probabilities $\overline{P}(\ \cdot\ |G)$ and say:

- when cycling, after a "very aggressive" movement there is a 21.84% probability that the next movement is also "very aggressive", and a 78.16% (the remainder probability) that the next movement will be smoother;
- when walking, after a "very aggressive" movement there is a 33.8% probability that the next movement is also "very aggressive", and a 66.2% that the next movement will be smoother;
- In both cases, after a "very aggressive" movement the next movement would be probably "aggressive".

There are many ways of interpreting these graphics depending on the study being done.

6. Discussion

The SCRTS algorithm has been shown to perform well in extracting the essential features for the activity classification over accelerometer data, while also considerably reducing the frame's dimensionality. This method also allows us to visualize the selected features' differences and similarities in graphics and pie charts, giving rise to the possibility of analysis and drawing conclusions about the downstream classification task (i.e., the activities classified in our case studies).

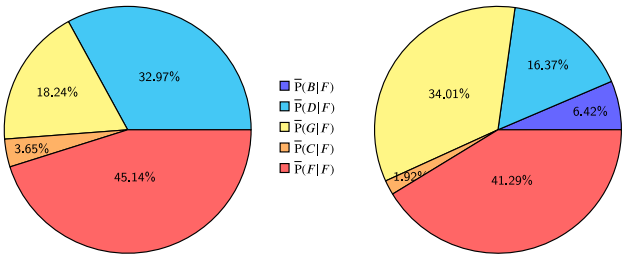

Fig. 8. Comparison of the F-state conditional probabilities of cycling (left) vs. walking (right).

In this work, we used the vector magnitude (Eqs. (1) and (2)) for the clustering, thus obtaining the states. If we only use the vector magnitude we are only considering the magnitude of the resultant acceleration of the movement made in the activity, ignoring the direction of the movement. In the activities classified in this paper, the magnitude of the resultant acceleration was relevant, and not so much the direction of the motion, so the classifications were good. But there may be other types of classifications where it is important to take into account where the movement was going, so it would be important to consider all the outputs of the device as a single vector before doing the discretization into states. This is left for future work.

We believe that our method can be of great use especially for long-time series, not only because the dimensionality reduction is considerable, but also because the features $C(F)$, $P(F)$, and $W(F)$ are increasingly accurate in representing each frame as more vector magnitudes are used to compute them.

In some cases it may be that $P(F)$ and $W(F)$ do not contribute much in comparison to $C(F)$, this we believe could happen in classifications where there has been some complexity in the movements made by the users, that is, in classifications where there has been an important variation among the different movements (states). We believe that, on the contrary, in classifications with little variation in the movements, the features $P(F)$ and $W(F)$ should play a more important role, as is the case of the data sets we worked with in this paper.

The SCRTS features visual representation technique can be very useful since it shows important differences and similarities between the classes that otherwise, in most cases, it is very difficult, or not possible, to note. This allows us to make interpretations and draw conclusions that could be very useful for our work.

Although this algorithm is particularly intended for activity classification, we think it can be used in completely different contexts. We will try our technique in different contexts in future works.

7. Conclusions

The massive commercialization of wearable devices has made the study of time series feature extraction gain much attention in the last few years. As these devices are normally used daily, they require storing huge amounts of data. Then, finding the proper features that can take the relevant information of the time series, thus reducing its dimensionality, and keeping storage needs under sustainable levels, is a task of concern. With this aim, this work proposes the SCRTS method, which is based on three kinds of time series features: conditional probabilities, state probabilities, and state weights.

We tried our method with 3 different accelerometer data sets. We applied clustering for the discretization in 2 of these data sets, and Actilife's cut points in the other. We used ANN for the classification in each case. In the best performances, we had a dimensionality reduction from 5499 to 31, with an accuracy of 88%, a TPR of 71%, and a TNR of 91%; or a dimensionality reduction from 480 to 63 with an accuracy of 98%, a TPR of 100%, and a TNR of 98%. Therefore, the features extracted with the SCRTS had been shown to represent

correctly the important elements of a time series in the context of activity classification using wearable accelerometers.

Our method includes a visual representation tool of the main differences of the time series features, which could be very useful when analyzing the data sets, or when driving conclusions about the classification process.

We have shown the scope of the SCRTS method applied to accelerometer data sets. In future works we would like to try this method with other kinds of time series data sets to see its scope and limits in other contexts, as well as to expand the SCRTS method in other to consider all the dimensions of the input vectors as a whole.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was carried out with the support of the Generalitat de Catalunya 2021 SGR 01125, and funded by the Grants for the Recruitment of New Research Staff (FI), provided by the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR).

References

[1] V. Montesinos, F. Dell'Agnola, A. Arza, A. Aminifar, D. Atienza, Multi-modal acute stress recognition using off-the-shelf wearable devices, in: 2019 41st annual international conference of the ieee engineering in medicine and biology society, EMBC, IEEE, 2019, pp. 2196–2201.

[2] A.H. Shoeb, J.V. Guttag, Application of machine learning to epileptic seizure detection, in: Proceedings of the 27th International Conference on Machine Learning, ICML-10, 2010, pp. 975–982.

[3] X.W. Wang, D. Nie, B.L. Lu, Emotional state classification from EEG data using machine learning approach, Neurocomputing 129 (2014) 94–106.

[4] D. Ravish, K. Shanthi, N.R. Shenoy, S. Nisargh, Heart function monitoring, prediction and prevention of heart attacks: Using artificial neural networks, in: 2014 International Conference on Contemporary Computing and Informatics, IC3I, IEEE, 2014, pp. 1–6.

[5] E. Rastegari, S. Azizian, H. Ali, Machine learning and similarity network approaches to support automatic classification of parkinson's diseases using accelerometer-based gait analysis, in: Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019, p. 1.

[6] J.A.U. Sanchez, D.M. Muñoz, Fall detection using accelerometer on the user's wrist and artificial neural networks, in: XXVI Brazilian Congress on Biomedical Engineering, Springer, 2019, pp. 641–647.

[7] H. Li, A. Shrestha, F. Fioranelli, J. Le Kernec, H. Heidari, M. Pepa, E. Cippitelli, E. Gambi, S. Spinsante, Multisensor data fusion for human activities classification and fall detection, in: 2017 IEEE SENSORS, IEEE, 2017, pp. 1–3.

[8] J. Howcroft, J. Kofman, E.D. Lemaire, Feature selection for elderly faller classification based on wearable sensors, J. Neuroeng. Rehabil. 14 (1) (2017) 1–11.

[9] P. Rajpurkar, E. Chen, O. Banerjee, E.J. Topol, AI in health and medicine, Nat. Med. 28 (1) (2022) 31–38.

[10] Q. Huang, D. Cohen, S. Komarzynski, X.M. Li, P. Innominato, F. Lévi, B. Finkenstädt, Hidden Markov models for monitoring circadian rhythmicity in telemetric activity data, J. R. Soc. Interface 15 (139) (2018) 20170885.

[11] M.V. Albert, A. Sugianto, K. Nickele, P. Zavos, P. Sindu, M. Ali, S. Kwon, Hidden Markov model-based activity recognition for toddlers, Physiol. Meas. 41 (2) (2020) 025003.

[12] M. Boeker, H.L. Hammer, M.A. Riegler, P. Halvorsen, P. Jakobsen, Prediction of schizophrenia from activity data using hidden Markov model parameters, Neural Comput. Appl. 35 (8) (2023) 5619–5630.

[13] R.N. Bracewell, R.N. Bracewell, The Fourier Transform and Its Applications, Vol. 31999, McGraw-Hill New York, 1986.

[14] M.J. Shensa, The discrete wavelet transform: wedding the a trous and Mallat algorithms, IEEE Trans. Signal Process. 40 (10) (1992) 2464–2482.

[15] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, F. Yger, A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update, J. Neural Eng. 15 (3) (2018) 031005.

[16] J.A. Cadzow, B. Baseghi, T. Hsu, Singular-value decomposition approach to time series modelling, in: IEE Proceedings F (Communications, Radar and Signal Processing), Vol. 130, IET, 1983, pp. 202–210.

[17] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometr. Intell. Lab. Syst. 2 (1–3) (1987) 37–52.

[18] A.J. Izenman, Linear discriminant analysis, in: Modern Multivariate Statistical Techniques, Springer, 2013, pp. 237–280.

[19] R.J. Kate, Using dynamic time warping distances as features for improved time series classification, Data Min. Knowl. Discov. 30 (2) (2016) 283–312.

[20] P.Y. Zhou, K.C. Chan, A feature extraction method for multivariate time series classification using temporal patterns, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2015, pp. 409–421.

[21] R. Moskovitch, Y. Shahar, Classification-driven temporal discretization of multivariate time series, Data Min. Knowl. Discov. 29 (4) (2015) 871–913.

[22] R. Azulay, R. Moskovitch, D. Stopel, M. Verduijn, E. De Jonge, Y. Shahar, Temporal discretization of medical time series-A comparative study, in: IDAMAP 2007 Workshop, 2007, p. 1.

[23] S. Ross, A First Course in Probability, Pearson, 2010.

[24] C.X. Ling, C. Li, Data mining for direct marketing: Problems and solutions, vol. 98, in: KDD, 1998, pp. 73–79.

[25] A. García-Pavioni, B. López, A new method of dimensionality reduction for large time series applied to accelerometer wristbands' signals, in: Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies - BIOSIGNALS, Scite Press, 2022, pp. 103–110, http://dx.doi.org/10.5220/0010672800003123.

[26] P. Freedson, E. Melanson, J. Sirard, Calibration of the computer science and applications, inc. accelerometer, Med. Sci. Sports Exerc. 30 (5) (1998) 777–781.

[27] P. Casale, O. Pujol, P. Radeva, Human activity recognition from accelerometer data using a wearable device, in: Pattern Recognition and Image Analysis: 5th Iberian Conference, IbPRIA 2011, Las Palmas de Gran Canaria, Spain, June 8–10, Springer, 2011, pp. 289–296.

[28] F. Palumbo, C. Gallicchio, R. Pucci, A. Micheli, Human activity recognition using multisensor data fusion based on reservoir computing, J. Ambient Intell. Smart Environ. 8 (2) (2016) 87–107.

[29] J.E. Sasaki, A. Hickey, J. Staudenmayer, D. John, J.A. Kent, P.S. Freedson, Performance of activity classification algorithms in free-living older adults, Med. Sci. Sports Exerc. 48 (5) (2016) 941.