Noname manuscript No.
(will be inserted by the editor)

Design issues in Time Series dataset balancing algorithms

Enrique de la Cal · José R. Villar · Paula M. Vergara · Alvaro Herrero · Javier Sedano

Received: date / Accepted: date

Abstract Nowadays, the Internet of Things and the e-Health are producing huge collections of Time Series that are analyzed in order to classify current status or to detect certain events, among others. In two class problems, when the positive events to detect are infrequent, the gathered data lacks balance. Even in unsupervised learning, this imbalance causes models to decrease their generalization capability. In order to solve such problem, Time Series balancing algorithms have been proposed.

Time Series balancing algorithms have barely been studied; the different approaches make use of either a single bag of Time Series extracting some of them in order to generate a synthetic new one or ghost points in the distance space. These solutions are suitable when there is one only data source and they are univariate datasets. However, in the context of the Internet of Things, where multiple data sources are available, these approaches may not perform

This research has been funded by the Spanish Ministry of Economy, Industry and Competitiveness (MINECO), under Grants TIN2014-56967-R and TIN2017-84804-R.

Enrique de la Cal
Computer Science Department, University of Oviedo, EIMEM, Oviedo 33004, Spain
Tel.: +34-985182520
Fax: +34-985181986
E-mail: delacal@uniovi.es

José R. Villar
Computer Science Department, University of Oviedo, EIMEM, Oviedo 33004, Spain
E-mail: villarjose@uniovi.es
Paula M. Vergara
Computer Science Department, University of Oviedo, EIMEM, Oviedo 33004, Spain
E-mail: paulavg09@gmail.com
Álvaro Herrero
Civil Engineering Department, University of Burgos, EPS, Burgos 09006, Spain
E-mail: ahcosio@ubu.es
Javier Sedano
Instituto Tecnológico de Castilla y León, Burgos 09001, Spain
E-mail: javier.sedano@itcl.es

coherently. Besides, up to our knowledge there is not multiple datasources and multivariate TS balancing algorithms in the literature.

In this research, we study two main concerns that should be considered when designing balancing Time Series algorithms: on the one hand, the TS balancing algorithms should deal with multiple multivariate data sources; on the other hand, the balancing algorithms should be shape-preserving. A new algorithm is proposed for balancing multivariate Time Series datasets, as part of our work.

A complete evaluation of the algorithm is performed dealing with two real world multivariate Time Series datasets coming from the e-Health domain: one about epilepsy crisis identification and the other on fall detection. A thorough analysis of the performance is discussed, showing the advantages of considering the Time Series issues within the balancing algorithm.

Keywords Imbalanced Time Series · Correlation measures · Human Activity Recognition · Epilepsy Onset Recognition · Fall Detection

1 Introduction

The Internet of Things, e-Health, Bio-informatics and Bio-medicine are examples of how the technology is changing current problems or introducing new ones in machine learning: from static and high-cost data sets to big data problems [18]. In many cases, these datasets include Time Series (TS); for instance, the problems of human activity recognition and the human abnormal movement detection include this type of data [5,27,29,32]. Moreover, nowadays TS problems include not only one time sequence but several, so it is possible to have missed atomic features, each of them being a TS. And not to mention the possibility of mixing atomic features and TS with different sampling rates.

Nevertheless, gathering balanced TS datasets is not always feasible in several contexts, for instance, in Bio-medicine. To illustrate this assertion just consider the problem of seizure detection [33, 34], where the occurrence of a seizure in every-day life must be detected. Depending on the severity of the illness, the seizures can occur even once a month or less. Even in the case of unsupervised learning, the generalization capabilities of the models are in compromised due to the imbalance in the dataset.

The main part of the literature concerning the dataset balancing problem is focused on classical datasets, where a sample includes several features and each feature contains a single value not a sequence like in a TS. These techniques can rely on oversampling the minority classes or undersampling the majority classes; it depends on the kind of problem, but both techniques have their own advantages as well as drawbacks [12]. Furthermore, some other valid approaches have been reported including developing algorithms dealing with imbalance problems [11,22,23] or introducing ensembles for the minority classes together with a sort of majority classes undersampling [17].

Examples of oversampling techniques include well-known algorithms as SMOTE (Synthetic Minority Over-sampling Technique, [3, 13]), ADASYN

(ADAptive SYNthetic Sampling, [19]), ADOMS (Adjusting the Direction Of the synthetic Minority clasS examples, [31]) or SPIDER (Selective Preprocessing of Imbalanced Data, [28]).

Unfortunately, TS datasets have not received much attention from the research community as the dataset balancing problem has barely been studied. The most similar approaches are focused on having a dataset consisting of just one single univariate TS, where a single variable is available as a time sequence, each value in the sequence is labelled with a class. The problem is addressed as how to classify the values in the incoming sequence [16, 24–26], where the known data sequence labels are clearly biased towards the majority class. Therefore, the solutions rely on drawing new synthetic atomic values based on any of the above mentioned algorithms.

In [21], the problem of balancing a TS dataset, that is, each sample in the considered dataset is a univariate TS, is faced. In this approach, instead of working on the original domain, ghost points (new synthetic points) are suggested on the distance between TS domain. The matrix of distances (a matrix containing the distance between each pair of TS using a certain TS measure distance function) is then introduced in a SVM classifier, producing an improved model. Different TS distance measurements were proposed, one of the most popular is Dynamic Time Warping.

In our previous work [9] a simple adaptation of the SMOTE algorithm for multivariate TS balancing (TS_SMOTE) was presented. The current study addresses some design issues for TS dataset balancing algorithms extending our previous work: i) introducing the Dynamic Time Wrapping TS distance measurement in the TS merging mechanism, ii) and a more detailed explanation of the TS_SMOTE new algorithm including a more formal definition of the multivariate TS balancing problem iii) a complete description of the our proposal for the generation of new synthetic TSs, and finally iv) the experimentation section has been improved, with extra statistical analysis as well as new experimentation comprising three state-of-art classifiers comparing the performance of the model taking as input the original and the balanced TS datasets as input.

The study is structured as follows. The next section outlines the SMOTE, while the design issues are explained and possible solutions are given in Sect. 3. Experimentation and the discussion on the results are covered in Sect. 4. Finally, the main conclusions are drawn in section 5.

2 The SMOTE algorithm

The SMOTE algorithm is an oversampling method [13], where each sample from the minority class is randomly combined with each of its nearest neighbors to balance the dataset.This method assumes a two-class problem. However, it can be easily extended to a multi-class problem [30].

Algorithm 1 reproduces the algorithm from the original paper for the sake of completeness. The parameters of this method include the number of nearest

neighbors to consider (k, by default $k = 5$ has been proposed), the number of samples belonging to the minority class (T) and the number of synthetic samples to generate for each original sample from the minority class (N). This parameter N is given as a percentage; values smaller than 100% reduce the original minority subset and produces a new dataset of the same size as the original. Whenever $N > 100$ means that $N/100$ synthetic samples are to be generated for each sample from the minority class.

SMOTE takes a sample and searches for k neighbors, considered as the parents, using the Euclidean distance; each synthetic sample is generated as a random linear combination of the parents. This method has been successfully tested on different domains; and many different versions have been published [3]. Some improvements on the SMOTE include i) cleaning the new dataset of minority-class' *Tomek links* producing the *SMOTE+Tomek links* and ii) cleaning the whole dataset of *Tomek links*, known as *SMOTE-ENN*. A Tomek link is a sample from one class that is included in the counterpart class space[1]. Formally speaking, a pair of samples E_i and E_j forms a Tomek link if there exists no sample E_k such that $d(E_i, E_k) < d(E_i, E_j)$ or $d(E_j, E_k) < d(E_i, Ej)$, with d being the distance function. However, these methods are a sort of denoising stage, and we will not consider these versions.

Algorithm 1 The SMOTE algorithm. Three parameters are needed: Number of minority class samples T, the amount of SMOTE N% and the number of nearest neighbors k. The outcome includes (N/100) * T synthetic minority class samples. **SMOTE(T, N, k)**

```
1: if N < 100 then
2:     Randomize the T minority class samples
3:     T = (N / 100) * T
4:     N = 100
5: end if
6: N = int( N / 100 )
7: numattrs = Number of attributes
8: Sample[][]: array for original minority class samples
9: newindex: counts the number of generated synthetic samples
10: Synthetic[][]= array for synthetic samples
11: for i = 1 : T do
12:     Compute the k nearest neighbors for i, saving the indexes in nnarray
13:     Populate(N, i, nnarray)
14: end for
15: function Populate(N, i, nnarray)
16:     while N ≠ 0 do
17:         Choose a random number nn in {1, ..., k}
18:         for  attr = 1 : numattrs do
19:             dif = Sample[nnarray[nn]][attr]-Sample[i][attr]
20:             gap = random number in {0, 1}
21:             Synthetic[newindex][attr] = Sample[i][attr] + gap * dif
22:         end for
23:         newindex ++
24:         N = N - 1
25:     end while
26: end function
```

[1] In medical record databases regarding a rare disease, were there is a large number of patients who do not have that disease; continuous fault-monitoring tasks where non-faulty examples heavily outnumber faulty examples, and others. The counterpart class is the one corresponding to patients without the desease.

3 TS dataset balancing algorithm's design issues

The main drawback with the balancing algorithms in TS datasets is that the proposed solutions have been analyzed when the samples include atomic attributes. For instance, if we apply SMOTE to datasets where each sample includes attributes that are TS, the new synthetic sample will introduce a mere combination of the TS, as shown in Fig. 1, not to mention what to do when the two TS have different length.

This section is devoted to explaining some issues to consider when designing balancing algorithms in case of TS datasets. Firstly, the notation and the definition of TS datasets is presented in the next subsection, introducing the main issues to solve. Next, a solution based on SMOTE is detailed in subsection 3.2 solving these main issues.

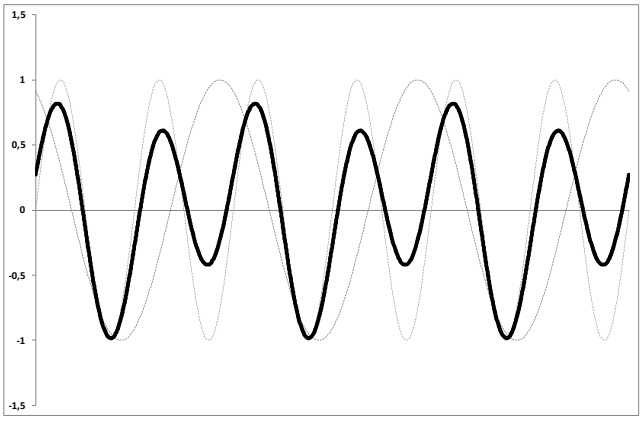

Fig. 1 A random combination of two TS -the two thinner gray lines- produce a totally different TS -the thicker line-.

3.1 Issues in balancing TS datasets

3.1.1 Motivation

In the context of balancing algorithms, a dataset can be denoted as $\{\overline{x}_i\}$, where $\overline{x}_i$ is called a sample. Each sample includes values for each of the M attributes; therefore $\overline{x}_i = \{x_{i1}, \ldots, x_{iM}\}$, and $x_{ij} \in \Re \; \forall j \in \{1, \ldots, M\}$. This is the type of dataset considered for the SMOTE algorithm [13].

When univariate TS datasets are considered [21], then some modifications are needed in the definition of a TS dataset: M is set to 1, and each sample is a TS per se. Therefore, the dataset is still $\{\overline{ts}_i\}$, but each sample $\overline{ts}_i$ is an univari-

ate TS: $\overline{ts}_{it} = \{ts_{it_1}, \ldots, ts_{it}, \ldots, ts_{it_{T_i}}, \}$, t is the timestamp $\in \{t_1, \ldots, t_{T_i}\}$ and T_i is the number of values in this sequence i.

These datasets representation do not cope with what is currently encountered in some research fields. As mentioned earlier, TS datasets, where each sample is a multi-variable TS are present nowadays in many Bio-medicine problems [4]. Therefore, for these cases, a new definition of a TS dataset is needed solving some specific issues that need representation.

The first issue to consider is the availability of multiples sources. For instance, TS gathered from wearable sensors present variability due to the sensor different behavior and the participants specific and unique way of moving or specific signals, etc. While the bias due to the sensors are bounded by the technical specifications, the differences due to the participants characteristics are rather more complex.

Therefore, grouping the TS according to the participant is required. In order to generate a new synthetic TS sample, only data from the same participant must be considered to avoid mixing different behaviors. To make this concept clear, refer to Fig. 1. Let us suppose that the gray lines may represent a biometric signal from two participants; each one with his/her own frequency and bias. Clearly, mixing these two TS leads to a undesirable result as the outcome does not reflect the expected behavior.

The second issue to consider is that when balancing a dataset by means of oversampling, the new synthetic samples must be shape preserving -see Fig. 2. The point is that merging two TS needs a method that must preserve the shape of each of the merged TS with its own differences. In the mentioned figure, the TS d) keeps the same shape as the input TSs, a wider alarm set interval that starts and ends in two plausible timestamps.

These two issues are tackled in the next subsection, proposing extensions on the SMOTE algorithm to cope with both of them. But first, a new and complete definition of a TS dataset is provided.

3.1.2 Formal definition of a TS dataset

Let S be the number of data sources, then the dataset is $\{\overline{TS}_s\}$, $\forall s \in [1, S]$. We call a source of data each of the possible wells where data can be extracted. In the examples given before, where data is gathered from each participant, each participant acts as a different data source.

Each $\overline{TS}_s$ is a TS dataset sharing the same data source: $\overline{TS}_s = \{\overline{ts}_i^s\}$ $\forall i \in [1, |\overline{TS}_s|]$, with $|\overline{TS}_s|$ being the number of TS gathered from data source s. The i-th multivariate TS sample from source s is denoted as $\overline{ts}_i^s = \{\overline{ts}_{i,1}^s, \ldots, \overline{ts}_{i,j}^s, \ldots, \overline{ts}_{i,M}^s\}$; while $\overline{ts}_{i,j}^s = \{ts_{ij,t_1}^s, ts_{ij,t_2}^s, \ldots, ts_{ij,t_{T_i^s}}^s\}$ refers to the feature j for the i-th TS sample from source s: here j indexes the input feature or attribute, $j \in [1, \ldots, M]$, M is the total number of input features, $ts_{ij,t}^s$ is the value at timestamp t in $\overline{ts}_i^s$, while T_i^s is the number of values in this sequence and the same for each feature.

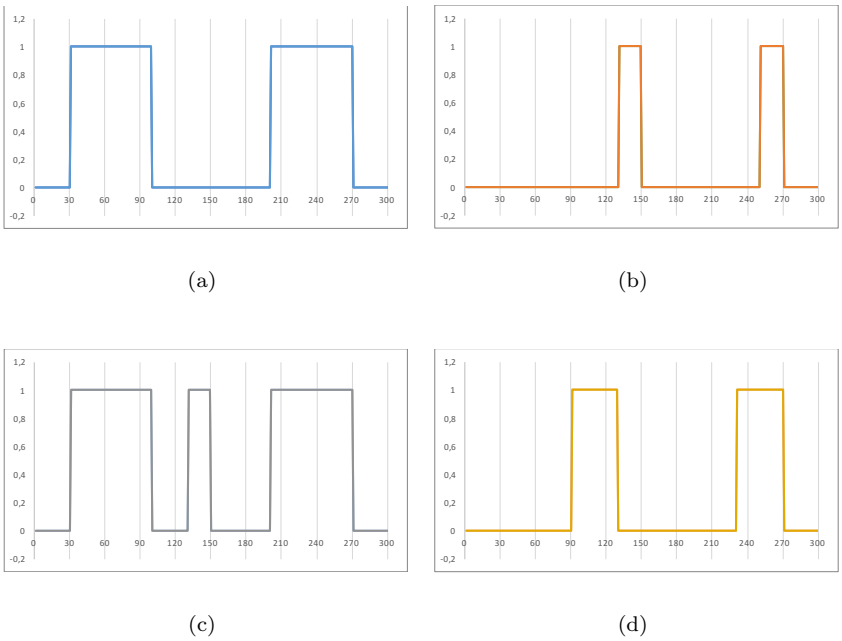

Fig. 2 Example of shape preserving merging: a) and b) are the two input TSs to merge, c) is a new synthetic TS obtained by merging the TSs a) and b) using the logical formula MAX and d) is a new synthetic TS generated using a shape preserving method for the input TSs.

Considering the label of a TS, we can regard two scenarios. The first one is labeling a TS $\overline{ts}_i^s$ with a single atomic class c_i^s that represents the class of the whole TS (e.g. any kind of ADL like Walking, Running, or typical positive classes like Epilepsy or Fall). The counterpart is that a TS $\overline{ts}_i^s$ is labelled with T_i^s timestamp labels $\overline{C}_i^s$ representing what timestamps of the TS belongs to certain class (e.g. a TS regarding a epilepsy crisis has part of it that corresponds to the crisis and the remaining to the pre and post stages). In this study, we propose a feasible solution for both cases. The first case is obvious as when in the generation of a synthetic TS sample for the minority class we already have the label for this new sample. For the second case, we will generate the C_i^s for the new TS sample as well. Then, a TS dataset $\{\overline{TS}_s\}$ has associated the suitable atomic label $\{c_s\}$ and the TS dataset $\{\overline{C}_s\}$.

Besides, let mC and MC be the labels for the minority and majority classes labeling the TS samples within the dataset, respectively. The number of TS with label mC for source s is denoted as $|mC_s|$, while the number of TS with label MC for source s is denoted as $|MC_s|$.

It must be specified that only a TS with atomic label $\{c_s\} = mC$ can be labeled with different timestamp labels of $\{\overline{C}_s\}(mC$ for the part of TS that

Attribute	Formula	Value
3DACC Sampling Frequency	-	16Hz
Number of timestamps of a TS	T_i^s	30-70 secs
Number of Subjects	S	6
Sliding Windows Size	-	2 seconds.
Sliding Windows Shift	-	1 second
Features	M	3 (SMA[2], AoM[3], TbP[4])
Atomic Labels	mC or MC	EPILEPSY, NO_EPILEPSY
Number of TSs per Subject	$\lvert\overline{TS}_s\rvert$)	10-15 (mC)/30-47(MC)

Table 1 Summary of the UNIOVI-Epilepsy problem attributes, considering the formulas of our TS data definition

corresponds to the crisis and *MC* for the remaining part of the TS). See Figure 3.

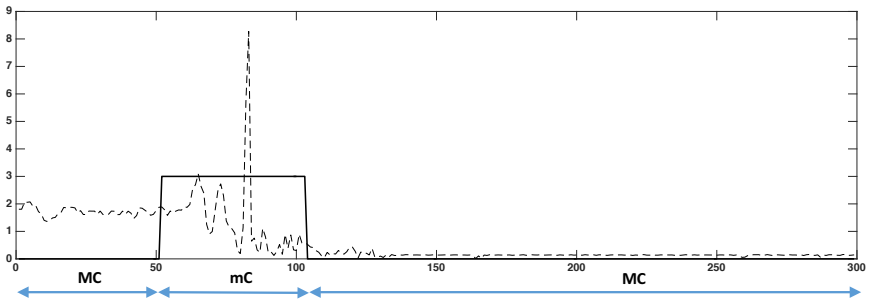

Fig. 3 Label a TS with the two kinds of labels: atomic label (this TS is a mC sample) and timestamp labels (mC or MC)

3.1.3 An example

Lets illustrate our ideas by means of an example. In the problem of epilepsy identification tackled in [34], six healthy people were enrolled to carry on simulation of epilepsy onsets, label EPILEPSY , and some different activities of daily living, label NON_EPILEPSY (see table 1, attribute Atomic Label). The six people wore an smartwatch with a 3DACC sensor, so each person represents a source in the formalization of our problem (S=6). Three transformations were calculated from the three componentes of the acceleration (see table 1, attribute Features).

Thus regarding a sampling rate of 16Hz, a sliding window size of two seconds and a window shift of one second, the length of the raw acceleration TSs are reduced by 16 when features TSs are computed (see Figure 4). For example, a RAW ACC TS of 1120 samples(70 seconds) is transformed in a multivariate

TS of 70 samples (4.375 seconds). As a consequence, we transform the domain of the acceleration TS to a different TS domain, with a multivariate TS of the three features and the atomic label.

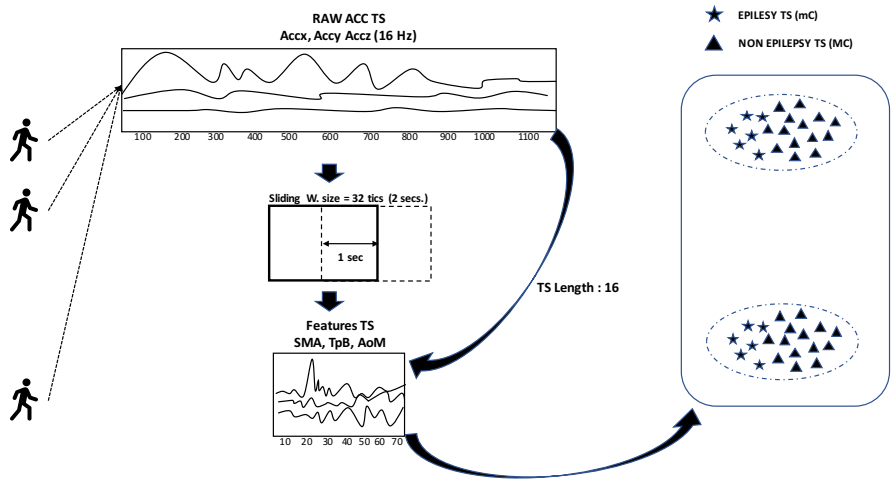

Fig. 4 Epilepsy identification problem

3.2 Generating synthetic artificial TS

In this proposal we still consider the two class problem {mC, MC}. So, Then, the mC needs to be SMOTEd provided that we are dealing with a TS dataset with different data sources. The proposal is described in Algorithm 2. The first difference is that the new samples are generated without mixing the data sources -which is introduced as a different generation per source s in line 1.

Then, two TSs are chosen from the population of TSs. The first one $\overline{ts}_i^s$ is selected randomly from the s dataset. And a second one $\overline{ts}_k^s$ that must be in the nearest neighbor considering the original SMOTE algorithm and using the Euclidean distance function as neighborhood distance.

Thirdly, the generation of the new TS is performed, which represents the main novelty of this paper. To do so, a random merging value -α is also chosen. Merging these two TS can be done in the euclidean space -as in SMOTE, this α is used for the weighted sum of the two TS- or can be done considering the shape of the TS. After some experimentation we've found out that merging the TS in the euclidean space might lead to unsuitable outcomes. Therefore, we propose the use of shape factors in the merging of TS. The underlying idea is to find corresponding points in the two TS, merging in the euclidean space in the related intervals.

For each input feature we apply DTW [6, 8, 9] to obtain the sequence of matching pairs, each pair is the corresponding timestamps from each of the

Algorithm 2 The extension of the SMOTE algorithm for dealing with TS datasets. Three parameters are needed: the minority class to smote, the number of samples to generate, and the TS dataset. **TS_SMOTE(mC, max_iter, $\{\overline{TS}_s\}$)**

for s=1:S **do**
  $e \leftarrow 0$
  $knn\_distances \leftarrow knn(\overline{ts}^s)$
  while e < max_iter **do**
    $i \leftarrow random[1 \ldots |mC_s|]$ points to $\overline{ts}^s_i$
    $n\_k \leftarrow |MC_s| \ / \ |mC_s| - 1$
    $j \leftarrow 0$
    while $e < n\_iter$ and $j < n\_k$ **do**
      $k \leftarrow random[1 \ldots |knn\_distances(ts^s_i)|]$ points to $\overline{ts}^s_k$
      $\alpha = random[-1.0..1.0]$
      $\overline{new\_ts}^s \leftarrow []$
      for $j = 1 : N$ **do**
        $warp\_pairs \leftarrow DTW(\overline{ts}^s_{i,j}, \overline{ts}^s_{k,j})$
        for $t = 1 : |warp\_pairs|$ **do**
          $(x, y) \leftarrow TimePoint(warp\_pairs_{t,i}, warp\_pairs_{j,k}, \overline{ts}^s_{i,j}, \overline{ts}^s_{k,j}, \alpha)$
          $\overline{new\_ts}^s_{-,j,x} \leftarrow y$
        end for
      end for
      $warp\_pairs \leftarrow DTW(\overline{C}^s_i, \overline{C}^s_k)$
      for $t = 1 : |warp\_pairs|$ **do**
        $(x, y) \leftarrow TimePoint(warp\_pairs_{t,i}, warp\_pairs_{j,k}, \overline{C}^s_i, \overline{C}^s_k, \alpha)$
        $\overline{C}^s_{new,x} \leftarrow y$
      end for
      $\{\overline{TS}_s\} \leftarrow \{\overline{TS}_s\} \cup \overline{new\_ts}$
      $\{\overline{C}_s\} \leftarrow \{\overline{C}_s\} \cup \overline{C}^s_{new}$
      $\{c_s\} \leftarrow \{c_s\} \cup mC$
      $j \leftarrow j + 1$
    end while
    $e \leftarrow e + 1$
  end while
end for
function TimePoint(t_i, t_k, $\overline{ts}^s_{i,j}$, $\overline{ts}^s_{k,j}$,α)
  $gpx \leftarrow min(t_i, t_k) + (t_i + t_k)/2$
  $d \leftarrow |\overline{ts}^s_{i,j,t_i} - \overline{ts}^s_{k,j,t_k}|$
  $m \leftarrow min(\overline{ts}^s_{i,j,t_i}, \overline{ts}^s_{k,j,t_k}) + (\overline{ts}^s_{i,j,t_i} + \overline{ts}^s_{k,j,t_k})/2$
  $gpy = m + \alpha \times d$ **return** (gpx, gpy)
end function

two TS to merge. For each pair a new value is estimated as shown in function *TimePoint*. Once a TS is obtained for each feature, the class TS is also computed and the TS dataset and the class data set are updated.

Figure 5 graphically explains how to compute each of the time points to add to each TS. Figure 6 shows an example of synthetic new TS (solid line) based on other two real TS belonging to two epilepsy seizures of the same participant (dotted and dashed lines).

It is worth mentioning that special attention should be paid to the length of the TS, as they should be coherent. In this study, we decide to limit the length of each synthetic TS to the shortest one between the original pair of TSs selected to generate it, but this is just a solution for the sake of simplicity. However, this is an issue that requires further study.

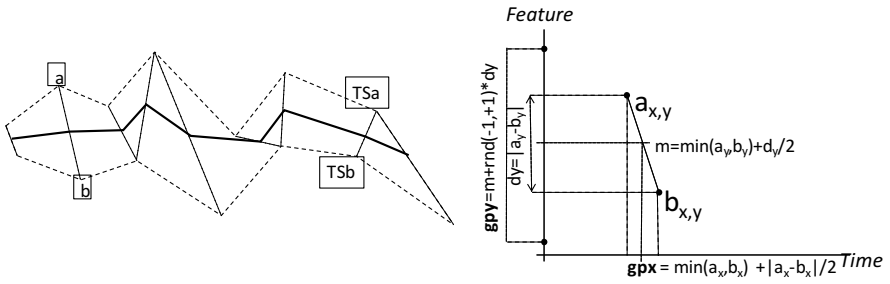

Fig. 5 Random time point proposal: DTW distance example (Left), Time Point calculation proposal (Right).

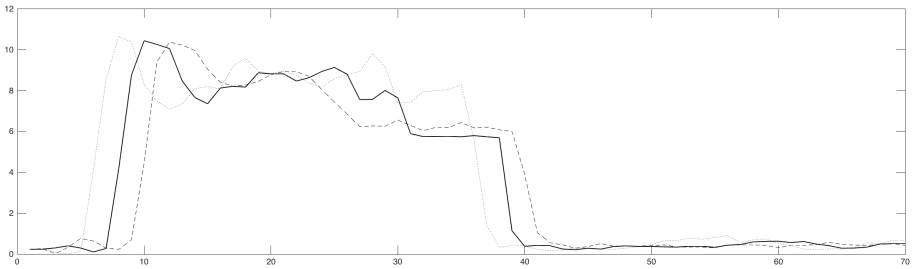

Fig. 6 Example of Synthetic TS: the new synthetic TS (solid line) and the parents' TSs (dotted and dashed lines)

4 Experiments and results

This research has used two public multivariate TS datasets: one comes from our own public UNIOVI-Epilepsy repository, with data from a real data set obtained from the simulation of epileptic seizures [34], and the other one taken from the fall detection UMAFALL repository published by the University of Malaga, in [10].

As the way to process both datasets are quite different and at least for the current contribution a general procedure for multivariate datasets is not ready, a complete section with all the materials and experiments will be included to cope with each dataset.

4.1 UNIOVI Epilepsy dataset

4.1.1 Materials

This TS dataset was gathered following a previously defined and very strict protocol, defining a set of activities, namely, the simulation of the epileptic convulsions and three activities: running, sawing and walking -either gesturing while walking slowly or normal walking at different paces. A wearable triaxial

accelerometer sensor (3DACM) included in a bracelet placed on the affected wrist measured the participant movements. The protocol was expected to cover the wide variety of scenarios: sawing and running are activities similar to the epileptic seizure, while walking is rather different. The length of the walking test is about three times the length of the others due to the chosen design.

The bracelets have wireless data sampling capabilities at a rate of 16 Hz, the 3DACM have a range of $2 \times g$. Up to 6 healthy participants, all of whom remained anonymous, successfully completed this experiment, each running 10 trials of each activity. The ages of the participants ranged from 22 to 47, with four participants of around 40 years old. One participant out of six was female, and the eldest was left-handed. An identification number was given to each Time Series (TS), including information fields on participant ID, the number of trials, the activity, etc. The data were gathered and stored for subsequent filtering and preprocessing, as described below. The complete data set is publicly available at [34] provided the citation acknowledgements are used.

Furthermore, the acceleration components have been transformed with the functions depicted in Table 2: the Signal-Magnitude Area (SMA), the Amount of Movement (AoM) and the Time Between Peaks (TBP). These transformations use the body acceleration, and were calculated on a sliding window of size 32 samples (2 secons) and a shift of 16 samples (1 second). To extract the body acceleration from the 3DACC values, a third-order elliptic Cauer digital filter was proposed. This filter used a sliding window of 4 samples and 1 sample shift; the parameters for obtaining these filters using the Matlab toolbox were as follows: a filter order of 3, a passband ripple of 1 dB, a stop-band attenuation of 80 dB and a normalized cutoff of 0.25. Each value within a TS in the transformed space was labelled with the majority class within the sliding window of size 32 samples and a shift of 16 samples but the windows corresponding to the period of crisis that was labeled with the minority class label. At the same time, each TS sample was labeled as belonging to the corresponding activity -EPILEPSY, NO_EPILEPSY- as a whole.

This TS dataset, consisting of TS samples of three TS each -SMA, AoM and TBP- $\{\overline{TS}_s\}$, with the label for each activity $\{c_s\}$ and with the TS for each timestamp label $\{\overline{C}_s\}$, has been used in this experimentation. We denote this TS dataset as ORIG, while the TS dataset after applying TS_SMOTE is denoted as SMT.

4.1.2 Experimental setup

The TS were manually segmented and labelled according to the procedure proposed in [2]. Three variables are considered, one for each acceleration component; therefore, the number of features is $M = 3$. The minority class mC is EPILEPSY, while the majority class is NO_EPILEPSY. There are 6 subjects, $S = 6$; the current ratio of imbalance is three non-epileptic activities per each epileptic activity. However, this ratio is really worse if we consider the length of the TS.

Transformation	Calculation
$SMA_t(\mathbf{s})$	$\frac{1}{w}\sum_{i=1}^{w-1}(\sum_{c\in\{x,y,z\}}\|b_{c,t-1}\|)$
$AoM_t(\mathbf{s})$	$\sum_{i=0}^{i=w-1}\sum_{c\in\{x,y,z\}}\|max(b_{c,t-i}) -min(b_{c,t-i})\|$
$TBP_t(\mathbf{s})$	Computed with the following algorithm: 1.- Find the sequences with value higher than mean+K*std within the window ($K = 0.9$) 2.- Keep the rising points from each of these sequences 3.- Measure the mean time between them

Table 2 The transformations of the components of the acceleration, where $b_{c,i}$ stands for the body acceleration.

To select the number of samples to introduce in the dataset we used the following criteria. In an imbalanced dataset, for example for subject 1 in table 3, there exists $R = 2.47$ times more examples belonging to the MC class than to the mC class for the s data source. So, to balance the number of samples for both classes means injecting $(R-1) \times |mC_s|$ new TS samples.

	Subject											
	1		2		3		4		5		6	
Dataset	**Size**	**R**	**Size**	**R**	**Size**	**R**	**Size**	**R**	**Size**	**R**	**Size**	**R**
EPILEPSY	15	2.47	10	4.7	10	2.47	10	3.1	13	2.46	10	3.0
NO EPILEPSY	37		47		31		31		32		30	
TS_SMOTE	22	-	37	-	21	-	21	-	11	-	20	-

Table 3 Number of TSs for each subject from the UNIOVI-Epilepsy dataset. Dataset column refers to the dataset: EPILEPSY and NO EPILEPSY (ADL) refer to the TSs from the UNIOVI-Epilepsy dataset, and TS_SMOTE refer to the new synthetic TSs computed from the EPILEPSY original dataset after running the TS_SMOTE algorithm, Size is the number of TSs in the corresponding dataset and R stands for Imbalance Ratio.

Finally, the α parameter was allowed to drift in the interval [-1.0, 1.0]. Although this is a rather wide interval, it was used to evaluate the robustness of the algorithm when the generation of the synthetic TS samples generate disperse samples. As stated in [3], this scenario highly penalizes the performance of balancing dataset algorithms; therefore, the conclusions can be extracted on adverse scenarios. Furthermore, to compare the correlation between the datasets before and after applying TS_SMOTE different measures of correlation have been calculated between each feature and the class for each dataset.

The next experimentation focuses on: i) analyzing the correlation before and after applying TS_SMOTE, and ii) comparing the performance of well-known classification algorithms (DT and KNN) when training with the original and with the balanced TS datasets.

4.1.3 Correlation between each feature and the class

Analyzing the correlation between the features and the class with and without the TS_SMOTE, shows whether the relationships between the features

and the class remain the same after the new synthetic TS samples have been introduced. In order to assess the similarity between the distribution of the original TS dataset and the balanced TS dataset, we made use of the Pearson Correlation ($\rho_{X,Y}$, Eq. 1) and the Mutual Information ($MI(X,Y)$, Eq. 2); where cov is the covariance, σ_X is the standard deviation of X, $p(x)$ is the probability of the event x and $p(x,y)$ is the conditional probability of x given y.

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

$$MI(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\, p(y)} \right) \tag{2}$$

These measurements have been calculated between each feature and the class, for both the ORIG and the SMT datasets. Aggregated results for the six participants are shown in Table 4; only the boxplot obtained for the participant number 1 is depicted in Fig. 7.

These results show that the TS_SMOTE generates a TS dataset rather similar to the original one in terms of the correlation between each feature and the class.

Nevertheless, the Wilcoxon signed-rank test has been performed to test if the Pearson and Mutual Information correlation values from the ORIG and from the SMT datasets belong to the same distribution. Clearly, there is no evidence against the null hypothesis of the two series belonging to the same distribution of the cases studied. Only three out of the 36 Wilcoxon tests ORIG-SMT (figures in bold face in tables 5 and 6) using Pearson as well as Mutual Information reject the null hypothesis. Therefore, the behavior of the TS_SMOTE seems to be valid to calculate new samples in order to balance the TS dataset and keeps the distribution of the original dataset.

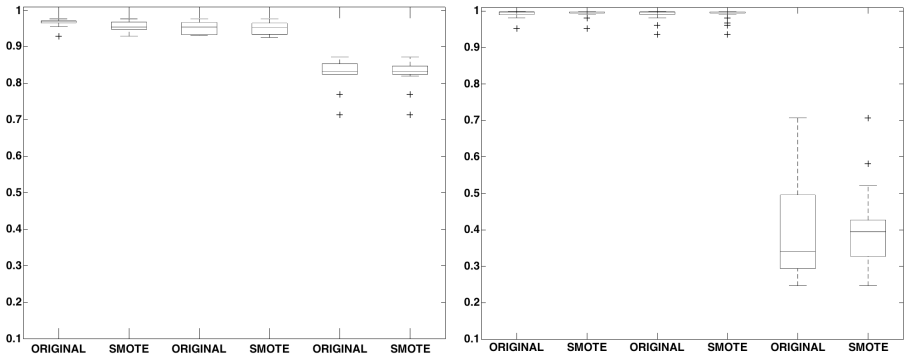

Fig. 7 Boxplot of the correlation measurements for participant 1. The boxplot on the left represents the $\rho_{X,Y}$ results and the one on the right refers to the $MI(X,Y)$ results. For both figures, each pair ORIGINAL-SMOTE corresponds to the following features from left to right: SMA, TbP and AoM.

	$\rho_{X,Y}$					
	SMA		TbP		AoM	
Subject	**ORIG**	**SMT**	**ORIG**	**SMT**	**ORIG**	**SMT**
1	0.97/0.01	0.95/0.01	0.95/0.02	0.95/0.02	0.82/0.05	0.83/0.03
2	0.97/0.01	0.97/0.01	0.94/0.03	0.95/0.02	0.83/0.03	0.86/0.03
3	0.96/0.02	0.96/0.01	0.92/0.04	0.93/0.03	0.73/0.06	0.76/0.06
4	0.98/0.00	0.98/0.01	0.96/0.01	0.96/0.02	0.81/0.07	0.76/0.05
5	0.96/0.01	0.95/0.01	0.94/0.02	0.94/0.01	0.78/0.11	0.74/0.09
6	0.97/0.01	0.96/0.01	0.97/0.02	0.98/0.01	0.82/0.03	0.83/0.02
	$MI(X,Y)$					
	SMA		TbP		AoM	
Subject	**ORIG**	**SMT**	**ORIG**	**SMT**	**ORIG**	**SMT**
1	0.99/0.02	0.99/0.01	0.99/0.02	0.99/0.01	0.40/0.13	0.39/0.09
2	0.98/0.01	0.98/0.02	0.92/0.04	0.92/0.03	0.20/0.09	0.24/0.08
3	0.98/0.04	0.98/0.02	0.94/0.06	0.95/0.04	0.23/0.07	0.20/0.06
4	0.99/0.00	0.99/0.01	0.96/0.04	0.96/0.03	0.35/0.18	0.26/0.14
5	0.97/0.03	0.98/0.03	0.95/0.05	0.97/0.04	0.12/0.06	0.12/0.04
6	0.99/0.01	0.99/0.01	0.97/0.05	0.97/0.04	0.66/0.08	0.69/0.07

Table 4 Correlation results between each feature and the class. The first column on the left refers to the participant id, that is, the data source correlative identification. Each cell contains the mean and the standard statistics for the values of the correlation measurements.

Participant	SMA	AoM	TbP
1	**0.0051**	0.8004	0.8477
2	0.2893	0.5498	0.0523
3	0.4209	0.6597	0.1670
4	0.7498	0.3867	**0.0112**
5	**0.0141**	0.1761	0.1394
6	0.1841	0.6061	0.5423

Table 5 Wilcoxon signed-rank test P-values at a significance level of 0.05. The null hypothesis is that the data obtained for the $\rho(feature, class)$ calculated for the ORIG dataset and for the SMT dataset belong to the same distribution.

Participant	SMA	AoM	TbP
1	0.4883	0.3254	0.9275
2	0.9414	0.6823	0.1503
3	0.8077	0.9394	0.3701
4	0.2298	0.9394	0.1071
5	0.2184	0.2539	0.5391
6	0.4499	0.7304	0.1739

Table 6 Wilcoxon signed-rank test P-values at a significance level of 0.05. The null hypothesis is that the data obtained for the $MI(feature, class)$ calculated for the ORIG dataset and for the SMT dataset belong to the same distribution.

4.1.4 Performance comparison

In order to compare the performance of TS_SMOTE, we will reproduce the experimentation on the ORIG dataset performed in [34] with the SMT dataset. The mentioned experimentation performed a 5x2 cross-validation on the participants, learning several classifiers, the Decision Trees (DT) [7, 14] and K-Nearest Neighbors (KNN) [15] among them -all of them from Matlab toolboxes. It's very important to highlight that this experimentation will obtain generalized models for the whole population of subjects, since the 5x2 cv technique is mixing TSs from the different subjects selected randomly in the same folds. In this case, TS_SMOTE will take place on the training fold before learning the models, while the test subset remains the same. With this con-

figuration we can train the DT and KNN with balanced dataset, while at the same time the comparison can be done, as long as the test dataset is exactly the same. For the KNN, three and five neighbors are considered. To compare the performances, the same measurements are used: the Mean Absolute classification Error (MAE) and the Geometric Mean (GM). These measurements are calculated according Eq. 3 and Eq. 4, respectively, where P, TS_p, N_t, $\hat{x_i}$, x_i stand for the number of subjects, the number of Time series per subject p, the number of samples for TS p, the expected value for sample i and the forecasted value for sample i respectively; and TP, FN, TN and FP stand for the True Positive, False Negative, True Negative and False Positive rates, respectively.

$$MAE = \frac{\sum_{p=1}^{P}\sum_{t=1}^{TS_p}\sum_{i=1}^{N_t} abs(\hat{x_i} - x_i)}{\sum_{p=1}^{P}\sum_{t=1}^{TS_p} N_t} \ . \tag{3}$$

$$\begin{aligned} Sensitivity &= \frac{TP}{TP+FN} \\ Specifity &= \frac{TN}{TN+FP} \\ GM &= \sqrt{Sensitivity.Specifity} \end{aligned} \tag{4}$$

For this experimentation, only the test dataset results are included, not the train dataset results, for the sake of brevity. Table 7 shows the obtained values for GM and the MAE of each fold from the cross-validation. Figure 8 depicts the box plots of the GM and MAE for the test dataset. It is worth mentioning that the learning process is driven by the MAE. Although the results show that the TS_SMOTE does not improve the MAE results, the GM values are much better for the TS_SMOTE, suggesting that the robustness of the models has been enhanced. The results obtained for the GM shows a good reduction in the spread of the results after applying TS_SMOTE.

Finally, the Wilcoxson ranked test has been calculated to compare the results of the combination of different models, datasets and measurements (see Table 8). In all the cases there is no evidence against the null hypothesis, so we can not state statistically that TS_SMOTE outperforms the Original dataset results. Anyway, based on our experience, we can say that models learnt from simulated datasets don't usually work in real situations.

4.2 UMAFALL dataset

4.2.1 Materials

Usually the FD (Fall Detection) studies develop a dataset comprising simulated falls and sessions of different ADL (Activities of Daily Living). All of these TS are labeled and become the test bed for the corresponding study. In

	Performance Results: GM					
	KNN3		KNN5		DT	
Fold	ORIG	SMT	ORIG	SMT	ORIG	SMT
1	0.9183	0.9324	0.9178	0.9345	0.9335	0.9405
2	0.9150	0.9236	0.9074	0.9193	0.9261	0.9417
3	0.9242	0.9322	0.9264	0.9308	0.9380	0.9388
4	0.7894	0.8176	0.7866	0.8097	0.7971	0.8753
5	0.9130	0.9202	0.9138	0.9255	0.9325	0.9436
6	0.8004	0.8619	0.7921	0.8546	0.8020	0.9063
7	0.8234	0.8386	0.8180	0.8368	0.8289	0.8685
8	0.8176	0.8488	0.8030	0.8433	0.8356	0.8625
9	0.8025	0.8170	0.8014	0.8139	0.8010	0.8593
10	0.8888	0.9024	0.8913	0.9086	0.8832	0.9181
Mean	0.8593	0.8795	0.8558	0.8777	0.8678	0.9055
Median	0.8561	0.8822	0.8546	0.8816	0.8594	0.9122
Dev	0.0569	0.0476	0.0598	0.0507	0.0609	0.0358
	Performance Results: MAE					
1	0.0518	0.0583	0.0503	0.0574	0.0552	0.0560
2	0.0415	0.0525	0.0399	0.0501	0.0440	0.0556
3	0.0560	0.0599	0.0543	0.0596	0.0419	0.0519
4	0.0459	0.0474	0.0441	0.0470	0.0524	0.0481
5	0.0486	0.0582	0.0480	0.0545	0.0449	0.0481
6	0.0406	0.0391	0.0397	0.0387	0.0423	0.0399
7	0.0391	0.0539	0.0374	0.0504	0.0448	0.0577
8	0.0480	0.0499	0.0468	0.0461	0.0544	0.0497
9	0.0412	0.0432	0.0396	0.0424	0.0478	0.0442
10	0.0456	0.0482	0.0428	0.0468	0.0453	0.0446
Mean	0.0459	0.0511	0.0443	0.0493	0.0473	0.0496
Median	0.0458	0.0512	0.0434	0.0486	0.0451	0.0489
Dev	0.0054	0.0068	0.0055	0.0065	0.0049	0.0058

Table 7 Cross validation test results using the GM measurement (GM) and Error measurement (MAE)

	Wilcox Results: GM Test					
	KNN3		KNN5		DT	
	ORIG	SMT	ORIG	SMT	ORIG	SMT
ORIG	1.000	0.273	1.000	0.186	1.000	0.104
SMT	0.273	1.000	0.186	1.000	0.104	1.000
	Wilcox Results: MAE Test					
ORIG	1.000	0.089	1.000	0.121	1.000	0.308
SMT	0.089	1.000	0.121	1.000	0.308	1.000

Table 8 Wilcoxon signed-rank test P-values at a significance level of 0.05. The null hypothesis is that the GM or MAE obtained for the different models, KNN3, KNN5 or DT, taking the ORIG dataset as input, belongs to the same distribution as the corresponding models obtained for the SMT dataset

this context, a simulated fall is performed by a set of healthy young participants wearing the sensory system, each of them letting him/herself fall towards a mattress from a standing still position.

The vast majority of these datasets were gathered with the sensor attached to the main body, either on the chest, waist, lumbar area, or tight. To our knowledge, the UMAFall dataset is the only one that includes data gathered from sensors placed on a wrist while performing simulated falls; this is the type of data needed in this research as long as the main hypothesis of this study is to perform FD with a sensor worn on a wrist. Furthermore, this dataset includes several sensors placed on different parts of the body -a 3DACC on the waist among others-. Finally, there is no pattern in the number of repetitions of each activity or fall simulation. Some participants did not simulate any fall, some performed 6 or 9 while a participant simulated 60 falls.

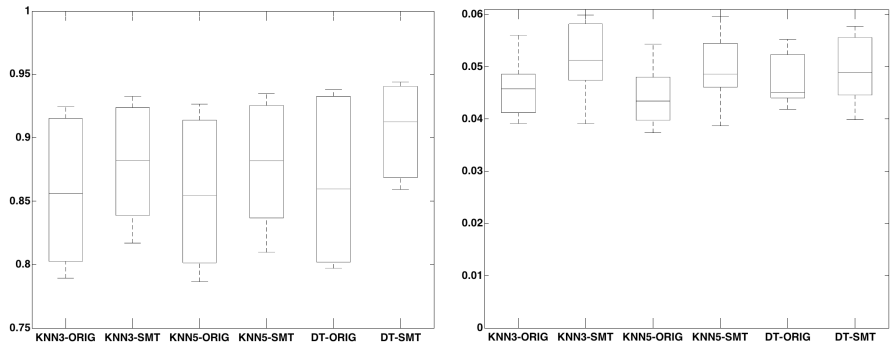

Fig. 8 Boxplots for the GM (left) and MAE (right) on the test dataset.

NP	NR	NF	Fqcy	Description
17	-	208	20	Includes forward, backward and lateral falls (FALL), running, hopping, walking and sitting (NO FALL). Neither all the participants have every type of activities nor the same number of goes. Sensors on the wrist, waist, ankle, chest and trouser pocket. Accelerometer, Gyroscope and Magnetometer

Table 9 Columns NP, NF and NR stand for the number of participants, the number of falls in the dataset and the number of goes for each ADL, respectively. A hyphen (-) means that it is not a regular value. The sampling frequency used in gathering the dataset is stated in Hz in column Fqcy.

Table 9 includes the basic information from this dataset where it can be seen that UMAFALL comprise different sensors and locations for them. The current study only uses the TS from the accelerometer located in the wrist since the other dataset included previously (UNIOVI-Epilepsy) uses a sensor in the same part of the body (the wrist). Besides this is the challenge in our current research line related with Human Activity Recognition [20, 33, 34].

4.2.2 Experimental setup

For this experiment we have selected two participants to check the validity of the TS_SMOTE proposal as an alternative to the analysis of a complete TS dataset (UNIOVI dataset) already presented.

In the latter, we tested whether it was possible to enhance the process of obtaining generalized models using the TS from different participants; results with and without TS_SMOTE were compared. On the other hand, with the new dataset we are not focused on generalized models but on specific models for each participant. In this case, the question is to evaluate whether it could be possible to apply TS_SMOTE in problems with a very reduced number of TS from one of the classes and still enhance the outcome as well. Furthermore, to complete the experimentation, the data from both participants have been merged, so an idea of the generalization capabilities for these two participants also hold.

The minority class mC is FALL, while the majority class is NO FALL. The current ratio of imbalance (see formula 8) is 2.33 non-fall activities per each fall activity for the subject 1, and 2.55 non-fall activities per each fall activity for the subject 2.

$$D = [ts_1, ts_2, ..., ts_n] \tag{5}$$

$$MCdata = ts_j \in D / ts_j.class = NOFALL \tag{6}$$

$$mCdata = ts_k \in D / ts_k.class = FALL \tag{7}$$

$$R = \frac{|MCdata|}{|mCdata|} \tag{8}$$

where D is the Dataset for one participant in UMAFall dataset, ts is a multivariate time series with four features: the three componentes of the acceleration and the magnitude, $MCdata$ and $mCdata$ are the partitions of D belonging to the majority and minority classes and R is the formula of the ratio of imbalance.

To select the number of samples to introduce in the dataset we used the following criteria. In an imbalanced dataset, there exists R times more examples belonging to the MC class than to the mC class for the subject. So, to balance the number of samples for both classes means injecting $(R-1) \times |mC_s|$ new TS samples. Thus, considering a goal R of 1.0, the number of new TS_SMOTEd[5] can be seen in table 10.

Subject	Dataset	Size	R
1	FALL	6	2.33
	NO FALL	14	
	TS_SMOTE	8	
2	FALL	9	2.55
	NO FALL	23	
	TS_SMOTE	14	

Table 10 Number of TSs for each subject from the UMAFall dataset. Dataset column refers to the dataset: FALL and NO FALL (ADL) refer to the TSs from the UMAFall original dataset, and TS_SMOTE refer to the new synthetic TSs computed from the FALL original dataset after running the TS_SMOTE algorithm, Size is the number of TSs in the corresponding dataset and R stands for Imbalance Ratio.

In this experiment, four variables are considered, one per each acceleration component and the magnitude calculated on the three previous components; therefore, the number of features is M = 4. The TS were manually segmented and labelled according to the procedure proposed in the well-known Abbate Algorithm [1] (see figure 9).

As in UNIOVI-Epilepsy the α parameter was used to drift in the interval [-1.0, 1.0]. Although this is a rather wide interval, it was used to evaluate the robustness of the algorithm when the generation of the synthetic TS samples generate disperse samples. As stated in [3], this scenario highly penalizes the performance of balancing dataset algorithms; therefore, the conclusions can be extracted on adverse scenarios. In figure 10 we can be seen a new synthetic TS (solid line) calculated from two original TSs using the TS_SMOTE algorithm.

[5] Remember that only the FALL TSs are TS_SMOTEd

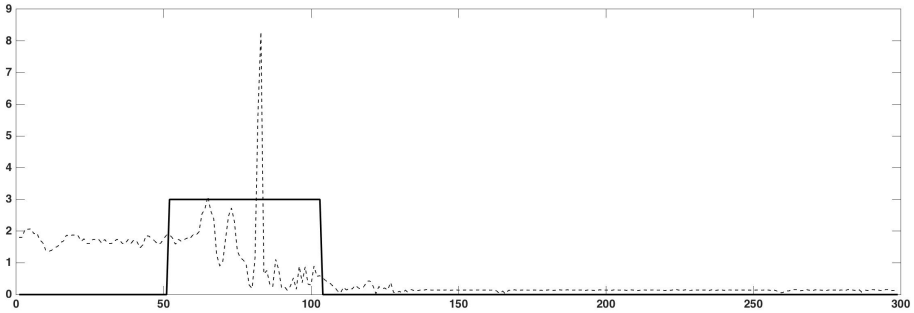

Fig. 9 Fall Output using the Abbate algorithm: the dashed line is the magnitude of the three components of the acceleration and the solid line is the Output of the Abbate algorithm for the class (3=FALL, 0=NO FALL) calculated manually. The data was gathered from a 3DACC sensor on the waist. Plots show the magnitude of the acceleration's evolution with the time.

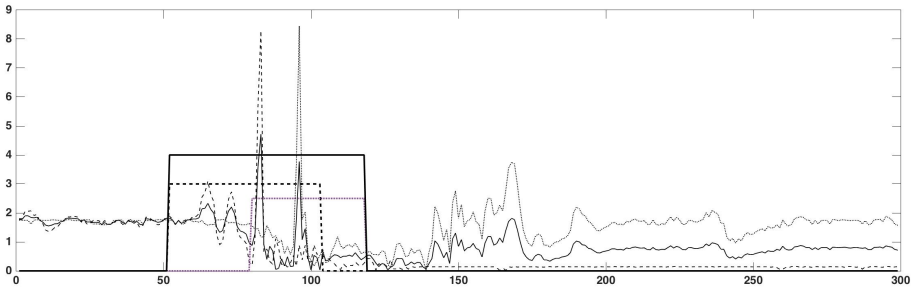

Fig. 10 Solid line: new Synthetic FALL TS Smote, Dashed line/Dotted Line: Original parents FALL TSs. The data was gathered from a 3DACC sensor on the waist. Plots show the magnitude of the acceleration's evolution with the time.

Correlation between each feature and the class As in the previous dataset, we have analyzed the correlation between the features and the class, with and without the TS_SMOTE. For this new problem we have used the Pearson Correlation ($\rho_{X,Y}$, Eq. 1) coefficient and the Mutual Information ($MI(X,Y)$, Eq. 2) as correlation measures too. Although the correlation results are not relevant enough to obtain any kind of conclusion, we have included the table 11 with the same kind of calculus as in UNIOVI-Epilepsy. However, the most relevant figures are the Wilcoxon test results that we have included in table 12. We have found that in this problem the new TS_SMOTEd dataset keeps the same distribution of the original dataset in all the cases.

Performance comparison This experimentation performed a train-test-validation process of the NN model, considering the datasets from the table 10. Six different experiments have been designed, grouped in three different scenarios (see table 13): subject 1, subject 2 and subjects 1&2. Each scenario has two cases: one using the TS_SMOTE and another without the TS_SMOTE dataset. The cases are identified with a experiment ID number shown in table 13. Each case splits the available dataset into two subdatasets: the Train-Test and the Vali-

	$ACCX$		$ACCY$		$ACCY$		MAG	
Subject	**ORIG**	**SMOTE**	**ORIG**	**SMOTE**	**ORIG**	**SMOTE**	**ORIG**	**SMOTE**
	$\rho_{X,Y}$							
1	-0.00/0.28	-0.06/0.24	-0.06/0.19	-0.06/0.19	-0.03/0.17	-0.03/0.17	0.04/0.19	0.04/0.19
2	-0.17/0.15	-0.20/0.14	0.31/0.20	0.31/0.20	-0.32/0.17	-0.32/0.17	0.24/0.14	0.24/0.14
	$MI(X,Y)$							
1	0.13/0.09	0.10/0.07	0.08/0.03	0.08/0.03	0.06/0.03	0.06/0.03	0.11/0.09	0.11/0.09
2	0.09/0.03	0.10/0.03	0.16/0.07	0.16/0.07	0.09/0.05	0.09/0.05	0.09/0.05	0.09/0.05

Table 11 Correlation results between each feature and the class. The first column on the left refers to the participant id, that is, the data source correlative identification. Each cell contains the mean and the standard statistics for $\rho_{X,Y}$ and $MI(feature, class)$.

		$ACCX$		$ACCY$		$ACCY$		MAG	
Subject	**Dataset**	**ORIG**	**SMOTE**	**ORIG**	**SMOTE**	**ORIG**	**SMOTE**	**ORIG**	**SMOTE**
		$\rho_{X,Y}$							
1	**ORIG**	1.0000	0.5910	1.0000	0.8363	1.0000	0.5353	1.0000	0.0521
	SMOTE	0.5910	1.0000	0.8363	1.0000	0.5353	1.0000	0.0521	1.0000
2	**ORIG**	1.0000	0.6596	1.0000	0.5156	1.0000	0.8503	1.0000	0.1365
	SMOTE	0.6596	1.0000	0.5156	1.0000	0.8503	1.0000	0.1365	1.0000
		$MI(X,Y)$							
1	**ORIG**	1.0000	0.4821	1.0000	0.7097	1.0000	0.3414	1.0000	0.3418
	SMOTE	0.4821	1.0000	0.7097	1.0000	0.3414	1.0000	0.3418	1.0000
2	**ORIG**	1.0000	0.5568	1.0000	0.3671	1.0000	0.7214	1.0000	0.3040
	SMOTE	0.5568	1.0000	0.3671	1.0000	0.7214	1.0000	0.3040	1.0000

Table 12 Wilcoxon signed-rank test P-values at a significance level of 0.05. The null hypothesis is that the data obtained for $\rho_{X,Y}$ and $MI(feature, class)$ calculated for the ORIG dataset and for the SMOTEd dataset belong to the same distribution.

dation. The best parameter subset for the NN model has been obtained with the Train subdataset using a 10-fold CV, while the final model is obtained using the complete subdataset Train-Test through a 5x2CV for the NN best parameters subset.

Exp. Id.	Subject	Class	R	Size	Train-Test	Validation
1	1	FALL	2.33	6	50%	50%
		NO FALL		14	50%	50%
2	1	FALL	1.00	6	50%	50%
		NO FALL		14	50%	50%
		FALL TS_SMOTE		8	100%	0%
3	2	FALL	2.55	9	50%	50%
		NO FALL		23	50%	50%
4	2	FALL	1.00	9	50%	50%
		NO FALL		23	50%	50%
		FALL TS_SMOTE		14	100%	0%
5	1	FALL	2.46	6	50%	50%
	1	NO FALL		14	50%	50%
	2	FALL		9	50%	50%
	2	NO FALL		23	50%	50%
6	1	FALL	1.00	6	50%	50%
	1	NO FALL		14	50%	50%
	1	FALL TS_SMOTE		8	100%	0%
	2	FALL		9	50%	50%
	2	NO FALL		23	50%	50%
	2	FALL TS_SMOTE		14	100%	0%

Table 13 Experiments: Exp. Id. is the Identificator we have used for this experiment, Subject is the id of the subjects considered, Class column refers to the kind of dataset: FALL, NO FALL (ADL) or FALL TS_SMOTE. R stands for Imbalance Ratio. Size is the number of TSs that the dataset include. Train-Test represents the percentage of TSs that have been used for the train-test process, and Validation the percentage of TSs used for validation

The Abbate algorithm needs a machine learning model to learn the fall detection event, and in this case we have used the classical model used in the original algorithm, a feed forward NN [1]. And the other parameter for Abate algorithm is the minimum threshold for the maximum peak during the fall. In terms of Abbate this peak is called the **Peak Time** and the threshold is **th1**.

The best subset of parameters was obtained for different values of **th1** and the NN model using the grid search[6]. The obtained parameters are shown in Table 14 for a feed forward NN and they are the same for the three thresholds tested (2.50g, 3.00g and 3.10g), so **2.5g** has been chosen as th1 for all the experiments.

Threshold	Size	Decay	Max. Iter.	Abs. Tol.	Rel. Tol.
2.50	20	100	0.001	1.0e-08	1.0e-06
3.00	20	100	0.001	1.0e-08	1.0e-06
3.10	20	100	0.001	1.0e-08	1.0e-06

Table 14 Best parameter set found for the feed forward NN and for different values of th1 for Experiment #1 (Subject 1 without TS_SMOTE)

Table 15 shows the performance of the experiments defined in table 13, comparing the two cases for the three scenarios. The comparison has been computed using the following formula:

$$\%\Delta = \frac{Performace_{TS\_SMOTE} - Performace_{ORIGINAL}}{Performace_{ORIGINAL}} \quad (9)$$

[6] R Caret package

We can state that the models obtained from the balanced validation dataset (Exp. Id. =#id TS_SMOTE) outperforms most of the classification statistics of the models obtained with the original dataset (Exp. Id. = #id Original) in the three comparisons but for the subject 2. Specifically in the comparison for subject 2 the TS_SMOTE outperforms the Accuracy, the Kappa factor, the Specificity and the Precision of the classifier obtained from the original Dataset while the performance classification of positive events (Sensitivity) and factor G worsens.

$$G = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \tag{10}$$

where TP, FN, FP stands for True Positives, False Negatives and False Positives respectively.

Subject	Exp. Id	ACC	Kp	Se	Sp	Pr	G
1	#1 Original	0.8000	0.4118	0.3333	1.0000	1.0000	0.5774
1	#2 TS_SMOTE	0.9000	0.7368	0.6667	1.0000	1.0000	0.8165
	%Δ	12,50	78,92	100,03	0,00	0,00	41,41
2	#3 Original	0.7333	0.4118	0.7500	0.7273	0.500	0.7386
2	#4 TS_SMOTE	0.8667	0.5946	0.5000	1.0000	1.0000	0.7071
	%Δ	18.19	44,39	-33,33	37,49	100,00	-4,26
1&2	#5 Original	0.7200	0.1117	0.1429	0.9444	0.5000	0.3673
1&2	#6 TS_SMOTE	0.8000	0.3655	0.2857	1.0000	1.0000	0.5345
	%Δ	11,11	227.22	99.93	5,89	100,00	45,52

Table 15 Results obtained from the UMAFALL validation dataset for a threshold th1 of $2.5 \times g$. The different columns are: the Exp. Id (is the id of the experiment and the descripcin of the dataset, Original or TS_SMOTE, that stands respectively for the Original Validation dataset and the balanced dataset with the TS_SMOTE algorithm), the threshold (th1), the Accuracy (Acc), Kappa factor (Kp), Sensitivity (Se), Specificity (Sp), Precision (Pr), and the geometric mean G computed using Eq. 10. The model is a feed forward NN.

5 Conclusions

This study focuses on the design issues that need addressing when designing a balancing algorithm for TS datasets; since this issue has barely been studied, it's a novelty in the literature. This problem is now faced in emerging real world problems in IoT and Biomedicine. An extension of the well-known balancing method SMOTE is proposed, including an stage where the shape of the TS parents merged is kept. To do so, DTW is used to determine the changes in the shape, and the corresponding intervals from each parent are then interpolated as in the original SMOTE.

Two datasets concerning two different problems have been tackled: one about epilepsy crisis identification and the other on fall detection. In the former, we have chosen the complete dataset producing a generalized model valid for different subjects, and for the latter, two similar participants were selected to obtain a very specific model for each one.

For both problems, a correlation study was carried out and the results showed that the correlations among each feature and the class are kept con-

stant after balancing the original datasets, so the statistical distribution of the datasets balanced with TS_SMOTE is the same as the imbalanced dataset.

As regards the classification results for the epilepsy problem, the results show that the robustness of the obtained classifiers is much better using the TS_SMOTE, although the MAE results were clearly not better or were even worse. As the TSs in this problem has been labeled manually, we think this issue might have affected the results. Regarding the second problem, we can state that the models trained with the datasets balanced with TS_SMOTE outperform the results of the models trained with the original dataset in all the experiments, although it has very poor results for the Sensitivity and G in the subject 2 scenario. Besides, it has to be considered that the number of positive TSs in this dataset is quite low and outlier TSs in such small datasets can dramatically affect the new synthetic TSs obtained with TS_SMOTE. In future studies we will address this issue.

However, the issue concerning how to merge related input features remains unstudied.

Compliance with Ethical Standards

The authors declare that they have no conflict of interest.

References

1. Abbate, S., Avvenuti, M., Corsini, P., Light, J., Vecchio, A.: Wireless Sensor Networks: Application - Centric Design, chap. Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: a survey, p. 22. Intech (2010)
2. Alvarez-Alvarez, A., Triviño, G., Cordón, O.: Human gait modeling using a genetic fuzzy finite state machine. IEEE Transactions on Fuzzy Systems **20**(2), 205–223 (2012)
3. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter pp. 20–29 (2004)
4. Baydogan, M.G., Runger, G.: Learning a symbolic representation for multivariate time series classification. Data Mining and Knowledge Discovery **29**(2), 400–422 (2015)
5. Beniczky, S., Polster, T., Kjaer, T., Hjalgrim, H.: Detection of generalized tonic-clonic seizures by a wireless wrist accelerometer: a prospective, multicenter study. Epilepsia **4**(54), e58–61 (2013)
6. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, AAAIWS'94, pp. 359–370. AAAI Press (1994)
7. Breiman, L., Friedman, J., Stone Charles, J., Olshen Richard, A.: Classification and regression trees. CRC press (1984)
8. de la Cal, E., Villar, J., Vergara, P., Sedano., J.: An study on the distances of an extension of the smote algorithm for time series. In: Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE 2017), pp. 722–733 (2017)
9. de la Cal, E., Villar, J., Vergara, P., Sedano, J., Herrero, A.: A smote extension for balancing multivariate epilepsy-related time series datasets. In: Proceedings of 12th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2017), pp. 439–448 (2017)

10. Casilari, E., Santoyo-Ramn, J.A., Cano-Garca, J.M.: Umafall: A multisensor dataset for the research on automatic fall detection. Procedia Computer Science **110**(Supplement C), 32 – 39 (2017)
11. Chan, T.K., Chin, C.S.: Health stages diagnostics of underwater thruster using sound features with imbalanced dataset. Neural Computing and Applications (2018)
12. Chawla, N.V.: Data Mining and Knowledge Discovery Handbook, chap. Data Mining for Imbalanced Datasets: An Overview. Springer (2010)
13. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research pp. 321–357 (2002)
14. Coppersmith, D., Hong, S.J., Hosking, J.R.M.: Partitioning nominal attributes in decision trees. Data Mining and Knowledge Discovery **3**(8), 197–217 (1999)
15. Friedman, J.H., A., J.B.R., Finkel: An algorithm for finding best matches in logarithmic expected time. ACM Transactions on Mathematical Software **3**(3), 209–226 (1977)
16. Fu, T.: A review on time series data mining. Engineering Applications of Artificial Intelligence **24**(1), 164–181 (2011)
17. Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognition **46**(12), 3460–3471 (2013)
18. Hardjono, T., Pentland, A.S.: Preserving data privacy in the iot world. Tech. rep., Massachusetts Institute of Technology (Connection Science & Engineering) (2016)
19. He, H., Bai, Y., Garcia, E., Li, S., et al.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: IEEE International Joint Conference on Neural Networks, pp. 1322–1328. IEEE (2008)
20. Khojasteh, S., Villar, J., Chira, C., González, V., de la Cal, E.: Improving fall detection using an on-wrist wearable accelerometer. Journal of Sensors **18**(5), 1350 (2018)
21. Köknar-Tezel, S., Latecki, L.J.: Improving svm classification on imbalanced time series data sets with ghost points. Knowledge and information systems **28**(1), 1–23 (2011)
22. Lopes Vinicius, M., Barradas Filho Oliveira, A., , Barros Kardec, A., Viegas Moraes Amorim, I., Silva Claudio O., L., Marques Pereira, E., Marques Lopes B., A.: Attesting compliance of biodiesel quality using composition data and classification methods. Neural Computing and Applications (2017)
23. López, V., Fernández, A., del Jesus, M., Herrera, F.: A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets. Knowledge-Based Systems **38**, 85–104 (2013)
24. Mishra, S., Saravanan, C., Dwivedi, V., Pathak, K.: Discovering flood rising pattern in hydrological time series data mining during the pre monsoon period. Indian Journal of Marine Sciences **44**(3), 3 (2015)
25. Montgomery, D.C., Jennings, C.L., Kulahci, M.: Introduction to time series analysis and forecasting. John Wiley & Sons (2015)
26. Moses, D., et al.: A survey of data mining algorithms used in cardiovascular disease diagnosis from multi-lead ecg data. Kuwait Journal of Science **42**(2) (2015)
27. Nooralishahi, P., Seera, M., Loo, C.K.: Online semi-supervised multi-channel time series classifier based on growing neural gas. Neural Computing and Applications **28**(11), 3491–3505 (2017)
28. Stefanowski, J., Wilk, S.: Selective pre-processing of imbalanced data for improving classification performance. In: Proceedings of the 10th International Conference in Data Warehousing and Knowledge Discovery (DaWaK 2008), pp. 283–292 (2008)
29. Suto, J., Oniga, S., Lung, C., Orha, I.: Comparison of offline and real-time human activity recognition results using machine learning techniques. Neural Computing and Applications (2018)
30. Sez, J.A., Krawczyk, B., Woniak, M.: Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. Pattern Recognition **57**, 164 – 178 (2016)
31. Tang, S., Chen, S.: The generation mechanism of synthetic minority class examples. In: Proceedings of 5th International Conference on Information Technology and Applications in Biomedicine (ITAB 2008), pp. 444–447 (2008)
32. Villar, J.R., González, S., Sedano, J., Chira, C., Trejo-Gabriel-Galán, J.M.: Improving human activity recognition and its application in early stroke diagnosis. International Journal of Neural Systems **25**(4), 1450,036–1450,055 (2015)

33. Villar, J.R., Menéndez, M., de la Cal, E., González, V.M., Sedano, J.: Identification of abnormal movements with 3d accelerometer sensors for its application to seizure recognition. International Journal of Applied Logic pp. 54–61 (2017)
34. Villar, J.R., Vergara, P., Menéndez, M., de la Cal, E., González, V.M., Sedano, J.: Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition. International Journal of Neural Systems **26**(6), 1650,037–1650,052 (2016)