Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Transfer Learning and Information Retrieval applied to Fall Detection

Mirko Fañez[1] | José R. Villar *[1] | Enrique de la Cal[1] | Javier Sedano[2] | V. M. González[3]

[1] Computer Science Department, University of Oviedo, Asturias, Spain
[2] Applied Electronics and Artificial Intelligence, Instituto Tecnológico de Castilla y León, Burgos, Spain
[3] Automatica and Control Department, University of Oviedo, Asturias, Spain

Correspondence
*José R. Villar, Departamento de Informática, Facultad de Geología, Despacho 5.1, Campus de Llamaquique, Oviedo 33005 (SPAIN). Email: villarjose@uniovi.es

Summary

Detecting falls in the elderly population is a very important issue that is related with the time of recovery. This study focuses on using wearable smart-watches to monitor the movements of the user in order to detect patterns that might be related to fall events. The proposed solution explores Symbolic Aggregate approXimation (SAX) Time Series representation, together with two Information Retrieval techniques enriched with Transfer Learning (TL). The solution is user-centred, that is, a model is developed for each specific user. Basically, the fall detection approach makes use of a finite state machine to detect peaks; the time series window embedding these peaks are represented using SAX. Assuming the data from the public fall detection data sets as valid, a dictionary is prepared using the most relevant words. This dictionary is then introduced as previous knowledge to a on-line learning classifier that is trained with normal Activities of Daily Living. The two classifiers are evaluated and compared with two classical approaches. Before this comparison, two clustering approaches are studied to produce the bag of relevant words. A complete experimentation is included, which makes use of several publicly available data sets and also with a data set developed by the research group. Comparisons are performed for all the data sets, showing how the TL stage empowers the classifier. Results show this solution produces high detection rates, while at the same time performed similarly for all the individuals tested. Furthermore, the positive effects of TL in this context are clearly remarked.

KEYWORDS:
Fall detection, Time Series, Machine Learning, Transfer Learning

1 | INTRODUCTION

Fall detection is still a challenge concerning to the elderly population Willy and Osterberg (2014). The performance of the fall detection devices, in their different nature and applications, shows a relatively high number of false alarms or, even worse, unidentified fall events. As a consequence, these alarming devices are considered annoying, costly and worthless.

Fall Detection is a very challenging research topic that has attracted the focus for several years. Solutions are very specific to the focused population as the level of activity varies from one to another. In general, the smaller the amount of movement, the more difficult the FD is. One of these groups is the elderly population; the activities are, in general, weaker and softer. For instance, as the elderly people walk slowly, the

[0] **Abbreviations:** 3DACC, triaxial accelerometer; Acc, Accuracy; ADL, Activities of Daily Living; CART, Classification and Regression Trees; DA, Discriminant Analysis; DT, Decision Trees; FD, Fall Detection; FN, False Negative; FP, False Positive;FSM, Finite State Machine; IDF, Inverse Document Frequency; KNN, K-Nearest Neighbour; LR, Logistic Regression; NN, Neural Network; OSVM, One-class Support Vector Machines; RS, Rule set; SAX, Symbolic Aggregation Approximation; Sen, Sensitivity; SVM, Support Vector Machines; TF, Term Frequency; TH, threshold; TL, Transfer Learning; TN, True Negative; TP, True Positive; TS, Time Series;

measurements in the waist get reduces, and the arms movement range is shorter. Even the commercial devices for elder perform with a relatively high false positive rate Roberts (2018).

There are many different techniques and solutions for fall detection. The wide variety of solutions includes in-mattress sensors to detect falls when uprising from the bed, smart tiles in intelligent homes, on-waist wearable sensors, video surveillance, etc. Interested readers are referred to Chaudhuri, Thompson, and Demiris (2014); Delahoz and Labrador (2014); Igual, Medrano, and Plaza (2013); Khan and Hoey (2017) for a complete review on the topic.

This study focuses on on-wrist 3DACC wearable devices to detect the FD. In this context, the majority of the published studies on this topic focus on either the method (combination of pre-processing and modelling techniques) or in the design of the sensor or sensor network, or in both of them (sensor network and FD method). Basically, the FD is performed using a peak detection method followed by a classification stage that labels a set of transformations extracted from the current data window. Alternatively, a classifier continuously monitors the TS to detect a possible fall event.

By far, 3DACC is the most common sensor used in wearable FD systems, in some cases combined with other sensors (such as gyroscopes or inertial sensors) or independently. The most common locations are the waist (which is not affected by the relative movements of the arms and mainly used with reduced movement people) or a wrist (which can be deployed on smart bands); some of the approaches also make use of the front pocket of trousers. The most relevant studies with 3DACC wearable FD solutions are included in Table 1. This table specify the sensory system and its location, the event detection method proposed in the study and the ML method. I-P refers to the proposal shown in Kangas et al. (2012).

In this table a distinction is made between generalized and used-oriented solutions. User-oriented solutions only make use of the data gathered from the current user. They all need a specific training and testing stage to evaluate the performance of the fall detection method. On the other hand, generalized solutions make use of all the data gathered from a set of users and propose a solution that can cope with FD for a user from the focused population. User-oriented solutions have the drawback that it is almost impossible to train with fall event data but learning the behaviour of the user might lead to a better set up of the models in order to detect a fall event.

In this research we propose the use of Transfer Learning to enhance an on-line user-centred learning solution's performance. On-line learning allows the classifier to continuously learn from errors, while TL incorporates the knowledge gather from previous experiences of other users. We propose using SAX to represent the TS, and a classifier uses the Manhattan distance instead of the cosine distances as proposed in Senin and Malinchik (2013). The evaluation of the TL and the classifier is done with two publicly available realistic fall detection data sets. Furthermore, a comparison with up-to-date methods have been done using a new data set gathered with ADLs from three participants and falls of a life-saving training mannequin.

This study is organized as follows. The next Section describes i) the data sets that will be used in this research, ii) the FD method that is proposed in this research and iii) the experimental setups that will be used to test the proposal. Section 3 is devoted to show and discuss the obtained results from the experimental setups. Finally, the document ends with the Conclusions.

2 | MATERIAL AND METHODS

2.1 | Data

Two main sources of data sets have been used in this research. On the one hand, standard publicly available data sets were used to evaluate the performance of the methods evaluated in this study. On the second hand, a new ad-hoc data set was gather mixing data gathered from participant and from a dummy.

The data sets include instances with TS for each of the acceleration components (a_x, a_y, a_z). The magnitude of the acceleration was computed $a = \sqrt{a_x^2 + a_y^2 + a_z^2}$. Each magnitude TS was labelled as FALL or NOT_FALL according to whether the TS includes data from a real fall or not.

2.1.1 | Publicly available data sets

There exists a compilation of data sets related with FD and wearable devices that was published in Casilari, Santoyo-Ramón, and Cano-García (2017a). In this research, up to twelve publicly available data sets were introduced; all of them included ADL and simulated FD when the participants wore 3DACC sensors located on different body parts. Moreover, a new data set of similar characteristics has also been published Sucerquia, López, and Vargas-Bonilla (2017).

In this study we have chosen two of these TS datasets. These data sets have been chosen because they include 3DACC sensor placed on a wrist, which is the solution that is being studied in this research. The data sets are:

UMA Fall data set Casilari, Santoyo-Ramón, and Cano-García (2017b) : 17 participants for a total of 531 TS (208 of them are labelled as FALL). The sampling frequency is 20 Hz. Includes forward, backward and lateral falls, running, hopping, walking and sitting.

TST Gasparrini et al. (2016) : 11 participants for a total of 264 TS (132 of them labelled as FALL). The sampling frequency is 100 Hz. Includes forward, backward and lateral falls. Two 3DACC sensors are used, one on the waist and one on the right wrist.

The design of these data sets differs. On the one hand, the performance of the ADLs for the UMA Fall are stronger than for TST. On the other hand, the distribution of the repeated ADL and simulated falls were different: in the case of the UMA Fall, there is neither a common number of activities and fall simulations nor the same number of repetitions; while in the case of the TST, all the participants performed the same activities more or less the same number of repetitions.

Because the variability in the number of TS associated to the participants in each of the data sets, participants with less than 20 TS or that did not include 9 simulated falls at least were omitted from the experimentation.

2.1.2 | A new mannequin-based fall detection data set

Analyzing the information published about the available FD data sets and how the data were gathered, there is a reasonable doubt that the fall events can represent real falls. That is the reason in this research we call these events as simulated falls. The point is that the participants fell on a mattress and tried to remain as quiet as possible. Also, the way of falling differs from the typical falls. Therefore, a new data set using a 3DACC has been elaborated. To gather the data, a marketed smart-watch with a 3DACC sensor and a sampling frequency of 100 Hz was used.

Three members of the research team wore this smart-watch for a period of one day in their normal life; the aim was to capture data from ADL. These time series correspond to activities of daily life (ADLs) like: Office work, daily household activities, driving, walking, running and other types of exercises (push-ups, etc.).

Moreover, a standard rescue-training mannequin has been used to produce this data set; the dummy has the dimensions and weight equivalent to an adult person -see Fig. 1)-. The reason of using this dummy is to mimic real falls, without the fear of injuring; however, this data set has a drawback: it is impossible to mimic the erratic movements a human typically makes after a fall.

This data set focuses on two types of falls, although both falls were labelled as FALL:

Fainting : where the person vanishes or faints. The following procedure was performed in order to capture this type of falls: i) the dummy starts in sitting position on a chair; ii) it is lifted by two members of the research team; iii) when fully up in front of the chair, it is dropped to the ground, producing an acceleration peak, iv) five seconds later the procedure ends.

Falls : mimicking an accidental fall. The following procedure was performed in order to capture this type of falls: i) the dummy starts in sitting position on a chair; ii) it is lifted by two members of the research team; iii) they walk the dummy one and a half meter, while moving the wrist with the sensor as similar as human do; iv) the dummy is plunged forward, falling to the ground; v) five seconds later the procedure ends.

The data gathered from each detected acceleration peak consist of a 7500 ms TS windows with the 3-axis acceleration data and the corresponding label. After the experimentation, the obtained dataset consists of 1072 time series of NOT_FALL 3DACC values and 87 time series of FALL 3DACC values, of which 45 are fainting and 42 are falls. Examples of these windows are depicted in Fig. 2. From now on, we will refer to this data set as *FallOVI*.

2.2 | Methods

2.2.1 | Peak detection

The same peak detection and feature extraction from Abbate et al. (2012); Khojasteh et al. (2018) is proposed in this study. A very simple finite state machine is used to detect the falls - see Figure 3 -. The data gathered from a 3DACC located on the wrist is processed using a sliding window. A peak detection is performed, and if a peak for a fall-like event is found, the data within the sliding window is analyzed to extract several features which are ultimately classified as FALL or NOT_FALL. The FD block is performed with a clustering + classification approach. In Khojasteh et al. (2018) it was claimed that the lower the computational cost the better as it must be run in the WD.

Each sliding window is transformed to a set of variables.-refer to Fig. 4-. Let's assume that the gravity is $g = 9.8 m/s$. Given the current timestamp t (please, refer to Fig. 4), we search a peak at **peak time** $pt = t - 2500ms$ (**point 1**). If at time $t = pt$ the magnitude of the acceleration a_t (computed as $a_t = \sqrt{a_{tx}^2 + a_{ty}^2 + a_{tz}^2}$) is higher than $th_1 = 3 \times g$ and there is no other peak in the period $(t - 2500ms; t]$ (no other a_t value higher than th_1), then it is stated that a peak occurred at pt.

The **impact end** ie (**point 2**) denotes the end of the fall event; it is the last time for which the a_t value is higher than $th_2 = 1.5 \times g$. Finally, the **impact start** is (**point 3**) denotes the starting time of the fall event, computed as the time of the first sequence of an $a_t <= th_3$ ($th_3 = 0.8 \times g$)

FIGURE 1 Left) Rescue-training mannequin before fall, right-botton) Rescue-training mannequin after fall, right-top) the marketed smart-watch

followed by a value of $a_t >= th_2$. The *impact start* must belong to the interval $[ie - 1200\text{ ms}, \text{peak time}]$. If no *impact end* is found, then it is fixed to *peak time* plus 1000 ms. If no *impact start* is found, it is fixed to *peak time*.

The following features are calculated whenever a peak is found:

- Average Absolute Acceleration Magnitude Variation, AAMV $=\sum_{t=is}^{ie} \frac{|a_{t+1} - a_t|}{N}$, with N the number of samples in the interval.
- Impact Duration Index, IDI $=$ impact end $-$ impact start.
- Maximum Peak Index, MPI $= \max_{t \in [is,ie]}(a_t)$.
- Minimum Valley Index, MVI $= \min_{t \in [is-500,ie]}(a_t)$.
- Peak Duration Index, PDI $=$ peak endtime $-$ peak starttime, with peak start defined as the time of the last magnitude sample below $th_{PDI} = 1.8 \times g$ occurred before pt, and peak end defined as the time of the first magnitude sample below $th_{PDI} = 1.8 \times g$ occurred after pt.
- Activity Ratio Index, ARI, calculated as the ratio between the number of samples not in $[th_{ARIlow} = 0.85 \times g, th_{ARIhigh} = 1.3 \times g]$ and the total number of samples in the 700 ms interval centred in $(is + ie)/2$.
- Free Fall Index, FFI, the average acceleration magnitude in the interval $[t_{FFI}, pt]$. The value of t_{FFI} is the time between the first acceleration magnitude below $th_{FFI} = 0.8 \times g$ occurring up to 200 ms before pt; if not found, it is set to $pt - 200$ms.

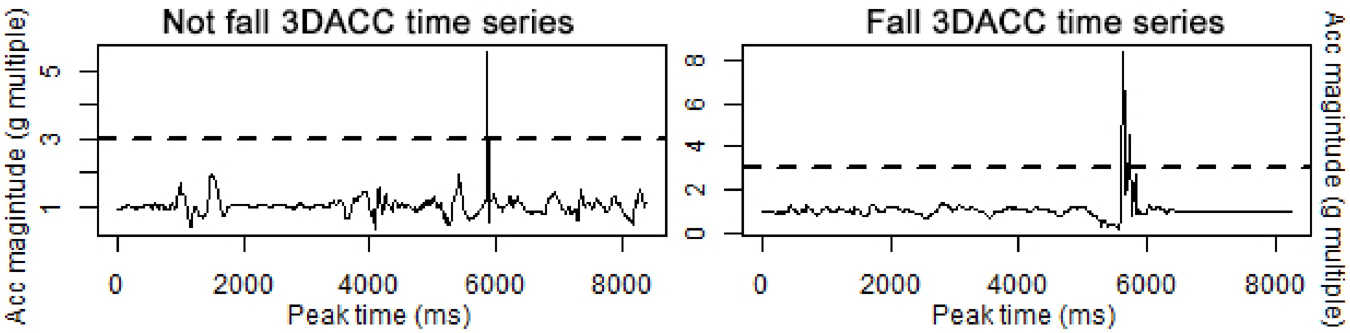

FIGURE 2 FALL vs NOT_FALL time series comparison

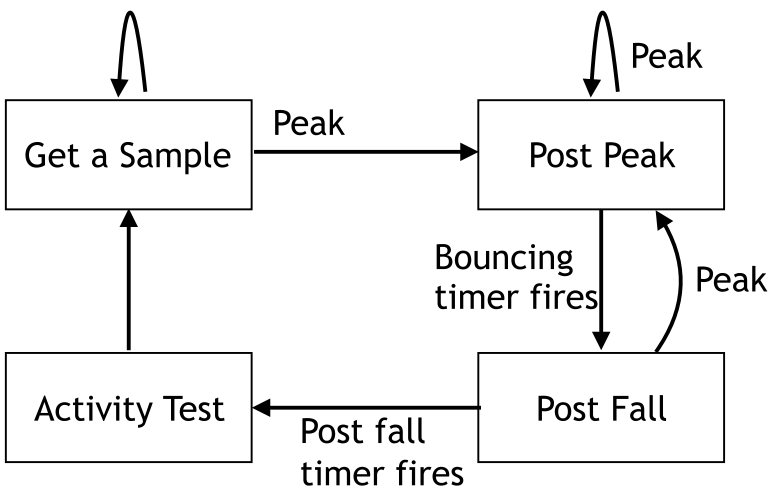

FIGURE 3 The finite state machine proposed in Abbate et al. (2012). Once a peak is detected, then a calm period must follow; any peak restarts the timer. After this calm period is found, if a new period of relatively calm occurs that means the data needs to be analyzed in order to determine if the peak is a fall.

- Step Count Index, SCI, measured as the number of peaks in the interval $[pt - 2200, pt]$.

Once a peak is found, then the features are computed for the current sliding window and these features are the inputs to a classifier, which should determine if it is a fall or not.

2.2.2 | Time Series representation and threshold determination

Following the literature in FD, this study makes use of the acceleration raw data Abbate et al. (2012); Khojasteh et al. (2018), despite a standardization of the data might be required at any moment when modelling.

Furthermore, we propose the use of Symbolic Aggregation approXimation (Lin, Keogh, Lonardi, and Chiu (2003)) to obtain a word representation of the peak window. The SAX words are the basis of the classifying method and of the transfer learning proposed later on this study.

Finally, each TS representation needs its specific set of thresholds. Consequently, three different strategies are proposed in this study to determine the thresholds and to compute the SAX representation:

- **origTSorigTH**: raw TS and the original thresholds proposed in Abbate et al. (2012) are used to detect the peaks and to compute the features. When modelling with SAX the peak window is standardized using its mean and standard deviation.

- **normTSnormTH**: the TS and the acceleration thresholds are standardized using the mean and standard deviation calculated with the ADL's TS for the current participant. The original thresholds proposed in Abbate et al. (2012) are, thus, normalized with these statistics.

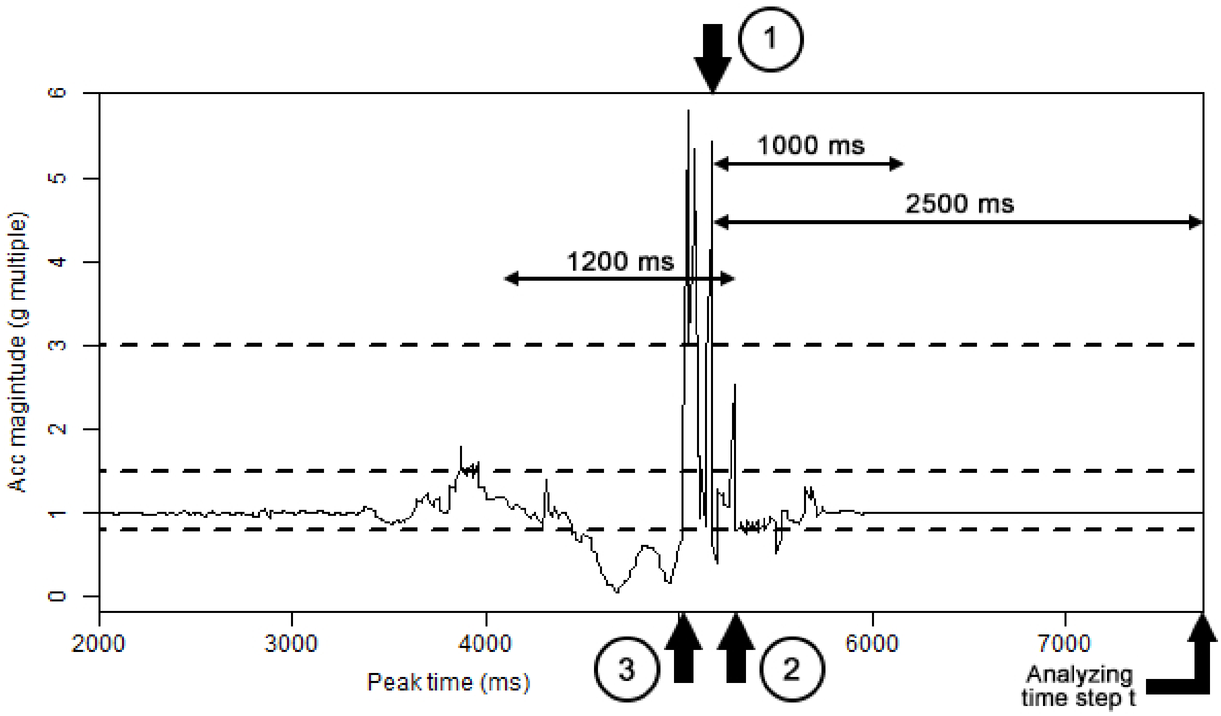

FIGURE 4 The peak detection process. A peak is detected as the last acceleration measurement higher than th_1 followed by a period of 2500 ms without any further peak.

- **normTSscldTH**: the TS are standardized using the mean and standard deviation calculated with the ADL's TS for the current participant but the thresholds are scaled. In this case, the peak threshold th_p^{norm} is determined as a percentage of the maximum peak value for any fall in the dataset (more specifically, $0.9 \times \min(\max_{i \in \mathsf{FALL}}(\mathsf{TS_i}))$); the remaining acceleration-based thresholds are obtained by scaling the thresholds in Abbate et al. (2012) using the peak threshold as reference: $\mathsf{th}_i^{norm} = \mathsf{th}_i^{Abbate} * \mathsf{th}_p^{norm} / \mathsf{th}_p^{Abbate}$.

As a summary, for each strategy, a detected peak is characterized with the 8 features and a SAX word and a label (FALL or NOT_FALL). The 8 features will be used later by the state-of-the-art classifiers that will be used for comparison purposes. The SAX word is used in the proposal of this study.

2.2.3 | The SAX based TF-IDF classifier

The classification is based on the classical Information Retrieval TF-IDF measurements and SAX; the basis of the classification were proposed in Senin and Malinchik (2013). The term frequency is computed using Eq. 1, while the inverse document frequency is calculated using Eq. 2. Basically, there exists a dictionary of relevant words, each word has the TF-IDF value for each of the labels FALL and NOT_FALL.

$$TF(w, label) = \begin{cases} 0 & \mathsf{freq(w, label)} = 0 \\ 1 + \log \mathsf{freq(w, label)} & \text{otherwise} \end{cases} \tag{1}$$

$$IDF(w, D) = \log \frac{|D|}{|\{label \in D / w \in label\}|}. \tag{2}$$

In this case, there exist a dictionary of relevant words obtained from training with the current user. This training should consist of TS coming from ADL's the user carries on during the training stage. If available, data will be transferred from previous experience of other users (see the next section covering this topic). To classify an incoming word w, if the w is included in the dictionary then the label with higher TF-IDF measurement is proposed. Otherwise, the closest word from the dictionary to w is retrieved and the label is assigned according to its TD-IDF values. To measure the distances between words the Manhattan distance is used (Eq. 3); where w_1 and w_2 are two SAX words of the same length. Thus, the Manhattan distance function measures the distance between each pair of SAX symbols, so the distance between a and b is 1, while the distance between a and d is 3. The distance between two SAX words is measured as the sum of the distances between the symbols, so the distance between aabba

and daaba is $3 + 0 + 1 + 0 + 0 = 4$. In this algorithm, the centroids are determined as the word with the minimum distance to the words of the cluster. From now on, we will refer to this classifier as **SAX-MAN**.

$$dist(w_1, w_2) = \sum_i |w_{1,i} - w_{2,i}|. \tag{3}$$

A second classifier is proposed as well. In this case, the min_dist distance proposed in SAX. To classify an incoming word w, if the w is included in the dictionary then the label with higher TF-IDF measurement is proposed. Otherwise, all the words in the dictionary with a distance to w smaller than a given threshold ($\mathrm{th}_{\mathrm{distance}}$) are retrieved. The TF-IDF measurements for each class are aggregated using the maximum; as before, the label with the final higher TF-IDF value is proposed. Finally, if no word is retrieved the FALL label is proposed. The distance measurement used in this study is the $\mathrm{min}_{\mathrm{d}}\mathrm{ist}$ distance from SAX Lin et al. (2003); the $\mathrm{th}_{\mathrm{distance}}$ is set to 15. We will refer to this classifier as **TF-IDF**.

2.2.4 | Learning the classifier and Transfer Learning

This study assumes the user of this FD solution must carry on with a short training stage in which the user performs with his/her normal life during a period of time; no fall event is considered during this short period. The gathered data from this ADLs is processed in order to detect peaks; these peaks generates the SAX words for the training. All these words are used in the generation of the bag of relevant words to compute the TF-IDF, which in turn is used for the classification.

If no more data is available, the classifier has to work with the small size bag of words. However, the performance of the classifier can be greatly improved if data is available from other users ADLs and falls, especially if the falls come from the same population (say, normal healthy elderly people).

For this reason, this study proposes to use TL from currently available data. The idea is to prepare a bag of relevant words from all the participants in the data set. Each detected peak generates a word, which is assigned with a label (either FALL or NOT_FALL). Together with the training data from the current user, an adaptation of the K-means algorithm is used to determine the most relevant words: the words that represent the centroids are those relevant words.

It is worth noticing that this bag of relevant words is obtained once the training stage has been completed, considering the data from the current user. For sure, the bag of words could have been calculated in advance and then the new words could have been added, but this method might introduce noise to the clustering and, consequently, to the TF-IDF classifier.

Algorithm 1 shows the clustering algorithm used in determining the bag of relevant words. The Manhattan distance function has been use in measuring the differences among two words (see Eq. 3). It is worth noticing that the Manhattan distance is used in both the TL and the SAX-MAN classifier; these relationships must be kept to keep the coherence between the both parts. From now on, this clustering is referred as **ALL-LABELS**.

Algorithm 1 Pseudocode of the adaptation of K-means to cluster SAX words

```
Set the value of k
Set the initial assignment of each word w to a clusters c_w
Determine the k centroids
Set the maximum number of iterations maxIter
Set reassignment to TRUE
while reassignment is TRUE and maxIter > 0 do
    Set reassignment to FALSE
    Decrement maxIter in 1
    for each word w do
        Determine the nearest centroid to w, c_w^new
        if c_w different than c_w^new then
            Set reassignment to TRUE
        end if
        c_w = c_w^new
    end for
    Determine the k centroids
end while
```

When determining the centroids, the values of the counters should also be updated. To do so, the counters for each label are determined as the aggregation of all the words belonging to the cluster. However, this aggregation of the counters could lead to a masquerading of the minority label. For instance, if we consider three words (with counters $< \mathsf{F} : 10, \mathsf{NF} : 1 >$, $< \mathsf{F} : 7, \mathsf{NF} : 0 >$ and $< \mathsf{F} : 0, \mathsf{NF} : 3 >$) that form a cluster, the aggregations would become $< \mathsf{F} : 17, \mathsf{NF} : 4 >$. In this case, the NOT_FALL label is masqueraded. To overcome with this problem we propose a second clustering arrangement: the idea is to group the words belonging to each label independently.

Therefore, the TL available words are split according to their label, and then grouped using the same algorithm. Once finished, the centroids are aggregated in a single bag of relevant words, the counters of common words are aggregated. For instance, if a word w_{FALL} is only included in the FALL clustering, then its counters will be $< \mathsf{F} : \mathsf{X}, \mathsf{NF} : 0 >$. Conversely, a word $w_{\mathsf{NOT\_FALL}}$ included in the NOT_FALL clusters would have counters as $< \mathsf{F} : 0, \mathsf{NF} : \mathsf{Y} >$. However, a word that has appeared as a centroid in both clusters will have counters as $< \mathsf{F} : \mathsf{X}, \mathsf{NF} : \mathsf{Y} >$. In all the cases, X and Y represent positive values. This clustering schema is referred as **BY-LABEL**.

2.3 | Experimental set up

There are two main experiments: i) a set of experiments to evaluate the performance of the method, this set makes use of the publicly available data sets; ii) a set of experiments to evaluate the method with totally new data, so to mimic real scenarios. For this latter experimentation set the new data set delivered in this study was used.

2.3.1 | Evaluation of the performance of the method

To evaluate and to compare each proposed method with other similar solutions the UMA Fall and the TST data sets are used, considering all the participants in a single data set. Furthermore, in order to compare the proposed solutions (SAX-MAN and TF-IDF, with and without TL) two state-of-the-art methods are used: KNN (following the ideas proposed in Abbate et al. (2012)) and SVM (following the ideas propose in Khojasteh et al. (2018)).

Because of the TL, the clustering described in the previous section should be performed. Therefore, it is needed to choose the best clustering approach and the number of clusters. To define the best number of clusters and the best clustering approach a set of values was evaluated: {30, 40, 50, 60, 70, 80, 90, 100, 110, 120} clusters. For each number of clusters we performed the ALL-LABELS clustering; Algorithm 2 describes the experimentation carried out to evaluate the method. This algorithm i) prepares the TL for the current participant, ii) learns the classifier using the training data, iii) evaluates the classifier on the test data set. The results with the SAX-MAN classifier are used to evaluate the candidate number of cluster.

Algorithm 2 Pseudocode for the evaluation of the method's performance

Obtain classical comparison models: SVM & KNN
for each participant p **do**
  Determine the TL without considering participant p
  for each fold f **do**
    Extract the training instances $\{TS_i^{train_f}\}_1^{N^p_{NOT\_FALL}}$
    Learn the model M with the instances
    Extract the testing instances $D = \{TS_i^{test_f}\}_1^{N^p_{NOT\_FALL}} \bigcup \{TS_i\}_1^{N^p_{FALL}}$
    for each instance I in D **do**
      Classify I using the learned model
      Update M with the correct label of I
    end for
    Evaluate SVM & KNN on D
  end for
end for

To evaluate a participant p from the data set, first, the TL stage is carried out with the data from all the participants in the data set but participant p. Let's call N^p_{FALL} and $N^p_{NOT\_FALL}$ the number of TS labelled as FALL and NOT_FALL, correspondingly, gathered for participant p. Secondly, 5x2 cross validation (5x2 cv) is performed only for the TS labelled as NOT_FALL ($\{TS_i^p\}_1^{N^p_{NOT\_FALL}}$), while all the TS labelled as FALL ($\{TS_i^p\}_1^{N^p_{FALL}}$) are kept for testing. Therefore, the training folds include only ADL data, while the test folds include both ADL (labelled as NO_FALL) and fall events (labelled as FALL).

Finally, the training takes place. This training can be carried out with or without the TL knowledge, so it is possible to compare how the TL affects the model. Afterwards, the learned model is evaluated with each instance in the test fold. The results for each instance of the test fold will be stored to compute the standard classification measurements: Accuracy, Kappa Factor, Sensitivity, Specificity and F1. As a summary, each participant will have a 5x2 cv on the ADL data, training without and with the TL. The testing fold will include all the participant's FALL instances.

After choosing the optimum number of clusters, the remaining options were compared. Firstly, the same algorithm and number of clusters was run with the BY-LABEL clustering and the SAX-MAN classifier. This experiment would lead to choose among the two clustering strategies. Secondly, the same algorithm and number of clusters using the best clustering strategy was performed for the TF-IDF classifier. After all these experiments, the best configuration will be obtained.

2.3.2 | Evaluation using the FallOVI data set

The point of this experimentation is to evaluate how the proposal performs when facing different participants, with no established protocol of ADLs and falls, with different sensors, etc. Because the current approach is user-centred, the comparison can only be performed with published solutions considering this type of modelling. To our knowledge, only an already published study developed a user-centred solution A. K. Bourke et al. (2016), while the remaining studies developed a generalized model obtained with data from all the participants. Nevertheless, the method published in A. K. Bourke et al. (2016) made use of 3DACC together with a gyroscope. Unfortunately, none of the available data sets includes data for the gyroscope; therefore, this solution cannot be used in the comparison.

Here, also the two well-known methods KNN and SVM are computed to compare the results of TL. The FallOVI data set will be used, splitting the samples in as follows. The three participants will be considered independently, the data from these participants include only ADL. The simulated falls will be considered as the fall data for each participant. Let's denote as $\{TS_{FALL}\}$ the TS gathered from the simulated falls. Let's also denote $\{TS_p\}$ the TS gathered for participant p. Then, the Algorithm 3 describes the process. The best configuration found in the previous experimentation will be used.

Algorithm 3 Pseudocode for the evaluation with FallOVI

```
Determine the TL with UMA Fall ⋃ TST
Obtain classical comparison models: SVM & KNN
for each participant p do
    for each fold f do
        Extract the training instances {TS_p^{train_f}}
        Learn the model M with the instances
        Extract the testing instances D = {TS_p^{test_f}} ⋃ {TS_FALL}
        for each instance I in D do
            Classify I using the learned model
            Update M with the correct label of I
        end for
        Evaluate SVM & KNN on D
    end for
end for
```

The evaluation of each model with each fold will be compared using the same standard classification measurements used before: Accuracy, Kappa Factor, Sensitivity, Specificity and F1. As a summary, each participant will have a 5x2 cv on the ADL data, the testing fold will include the FALL instances from the mannequin. For each participant two learning cases will be considered: learning the model without TL and learning with TL.

3 | RESULTS AND DISCUSSION

The results are divided in the two experimental setups describe before: the evaluation of the classifier using standard data sets and, on the other hand, the evaluation of the classifier with the FallOVI data set.

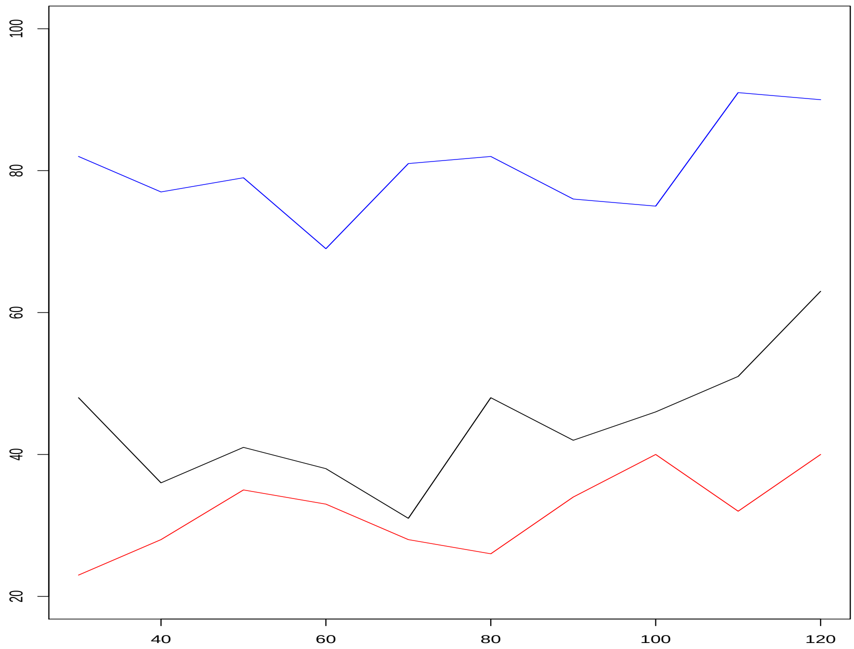

FIGURE 5 Screen plot using the aggregation of the FALSE NEGATIVE (FN) values. In black/blue/red the performance for the normTSnormTH/normTSscldTH/origTSorigTH scenarios, respectively.

3.1 | Classification results using standard data sets

Results from this experimental set up includes:

- the aggregation of the confusion matrix for all the participants in the data sets to determine the best number of clusters in Table 2 and Fig. 5,
- comparison of the two clustering approaches using the aggregation of the confusion matrices in Table 3,
- for each scenario, the aggregation of the confusion matrix among the folds in Table 4 and Fig. 6 using the SAX-MAN with and without the transfer learning, KNN and SVM classifiers,
- the mean and standard deviation of the Accuracy and the Sensitivity of the SAX-MAN classifier with and without the transfer learning, KNN and SVM classifiers for each scenario and participant from the TST and UMA Fall data sets in Tables 5 (normTSnormTH), 6 (normTSscldTH) and 7(origTSorigTH).
- for each scenario, the aggregation of the confusion matrix among the folds in Table 8 and Fig. 7 using the TF-IDF classifier with and without the TL, KNN and SVM classifiers,
- the mean and standard deviation of the Accuracy and the Sensitivity of the TF-IDF classifier with and without the transfer learning, KNN and SVM classifiers for each scenario and participant from the TST and UMA Fall data sets in Tables 9 (normTSnormTH), 10 (normTSscldTH) and 11(origTSorigTH).

The criteria to choose the best number of clusters is the number of undetected alarms (measured with the aggregated number of FN) as long as this value is critical in deploying a FD system. Therefore, we analysed the results in Table 2 for three scenarios at the same time using the SAX-MAN classifier and the ALL-LABELS clustering method (the screen plot in Fig. 5). We choose the value of 60 as the best number of clusters candidate as this value as long as it enhances the normTHscldTH performance while doing well enough in the remaining scenarios.

With this number of clusters, we proceeded to compare the two clustering methods (ALL-LABELS and BY-LABEL) in Table 3. Clearly, the ALL-LABELS clustering outperforms the BY-LABEL method; therefore, from now on we continue the experimentation using the ALL-LABELS clustering with 60 clusters.

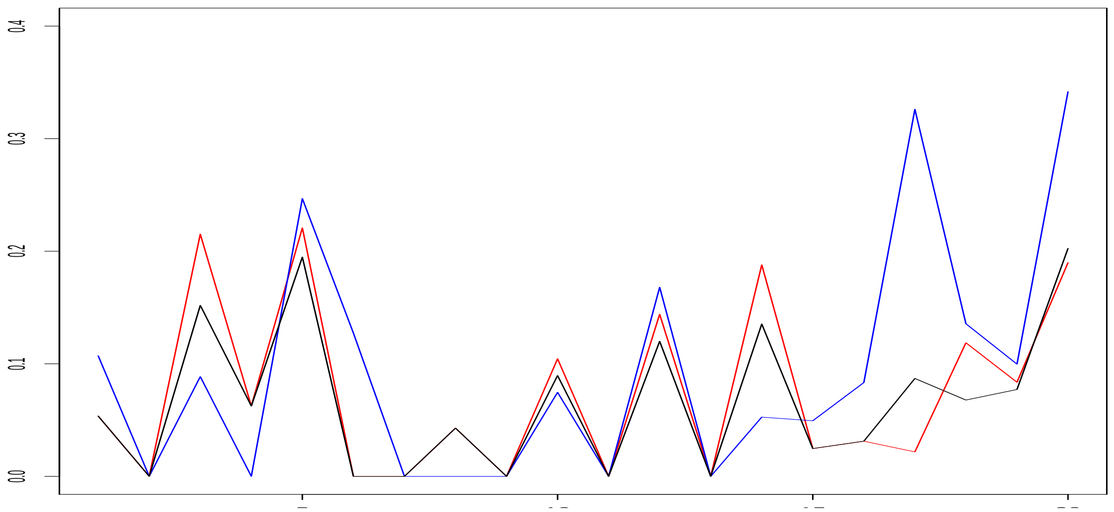

FIGURE 6 The evolution of the ratio $FP/(TP+TN+FP+FN)$ when using the SAX-MAN classifier for each participant. In black/blue/red the ratio for the normTSnormTH/normTSscldTH/origTSorigTH scenarios, respectively.

It is worth noticing from Table 4 the effect of TL in the performance of the SAX-MAX classifier. Although there are some exceptions, it is clear the improvement in the results of the classifier when TL is used. And what is more surprising is the competitive results obtained when using raw data (origTSorigTH scenario), with reduced aggregated FN and similar aggregated FP than the also successful normTSnormTH scenario. Besides, the KNN and SVM show quite similar performance in the first and second scenarios. WRT KNN, the following noteworthy points can be commented: i) the high rate of false alarms (FP), ii) the detection of nearly all the fall events, but in scenario normTSnormTH where it can be found several fails in UMA Fall dataset. On the other side, SVM outperforms KNN respect the negative samples (TN and FP). In addition, it can be stated that KNN shows similar number of false alarms (FN) than SAX-MAN+TL, while SVM outputs also the similar number of false ADLs (FP) than SAX-MAN+TL.

TABLE 1 Summary of fall detection approaches in the literature. Ref: the reference, Sensor: type of the sensor, SP: sensor placement, ED: event detection method, MM: modelling method, G/UB: generalized versus user-based modelling.

Ref	Sensor	SP	ED	MM	G/UB
Abbate et al. (2012)	3DACC	Waist	FSM	NN	Generalized
Bianchi, Redmond, Narayanan, Cerutti, and Lovell (2010)	3DACC + Air pressure	Wrist	TH + TIME	If-Then rules	Generalized
A. Bourke, O'Brien, and Lyons (2007)	3DACC	Wrist + Thigh	TH	If-Then rules	Generalized
A. K. Bourke et al. (2016)	3DACC + Gyroscope	Waist	TH	C4.5	Generalized
Casilari and Oviedo-Jimínez (2015)	3DACC + Gyroscope	Wrist + Pocket	TH	RS	User-Based
Deutsch and Burgsteiner (2016)	3DACC	Wrist	TH	NN	Generalized
Gjoreski, Bizjak, and Gams (2016)	3DACC	Wrist	TH	RS	Generalized
Hakim, Huq, Shanta, and Ibrahim (2017)	3DACC	Wrist	TH	kNN, DT SVM, DA	Generalized
Huynh, Nguyen, Irazabal, Ghassemian, and Tran (2015)	3DACC + Gyroscope	Wrist	TH	If-Then rules	Generalized
Igual, Medrano, and Plaza (2015)	3DACC	Waist or Pocket		NN or SVM	Generalized
Jatesiktat and Ang (2017)	3DACC + Gyroscope + Barometer	Waist	TH	SVM	Generalized
Kangas, Konttila, Lindgren, Winblad, and Jämsaä (2008)	3DACC	Waist	TH	RS	Generalized
Kangas et al. (2012)	3DACC	Waist	TH	RS	Genealized
Khojasteh, Villar, Chira, González, and de la Cal (2018)	3DACC	Wrist	TH	NN, SVM DT, RBS	Generalized
Kostopoulos, Nunes, Salvi, Deriaz, and Torrent (2015)	3DACC	Wrist	TH	RS	Generalized
Medrano, Plaza, Igual, Sánchez, and Castr0 (2016)	3DACC	Pocket	-	NN, SVM kNN	Generalized + model fitting
Ngu et al. (2017)	3DACC	Wrist	-	SVM kNN	Generalized
Putra, Brusey, Gaura, and Vesilo (2018)	3DACC	Chest + Thigh	FSM	SVM, LR, kNN, CART, I+P	Generalized
Sabatini, Ligorio, Mannini, Genovese, and Pinna (2016)	3DACC + Gyroscope + Barometer	Wrist	-	RS	Generalized
Sorvala, Alasaarela, Sorvoja, and Myllyla (2012)	3DACC + Gyroscope +	Waist + Ankle	-	RS	Generalized
Tsinganos and Skodras (2017)	3DACC	Wrist Waist	FSM + TH	kNN	Generalized
F. Wu, Zhao, Zhao, and Zhong (2015)	3DACC	Wrist	-	RS	Generalized
Zhang, Wang, Xu, and Liu (2006)	3DACC	Waist	TH	OSVM	Generalized

TABLE 2 Selection of the number of clusters using the ALL-LABELS clustering and the SAX-MAN classifier. Aggregation of the confusion matrix results for all the participants.

	normTSnormTH			
Num Cluster	**TN**	**TP**	**FP**	**FN**
30	105	1412	125	48
40	97	1424	133	36
50	101	1419	129	41
60	**101**	**1422**	**129**	**38**
70	93	1429	137	31
80	101	1412	129	48
90	99	1418	131	42
100	99	1414	131	46
110	99	1409	131	51
120	97	1397	133	63
mean	99.2	1415.6	130.8	44.4
	normTSscldTH			
Num Cluster	**TN**	**TP**	**FP**	**FN**
30	84	1457	159	82
40	73	1462	170	77
50	76	1460	167	79
60	78	1470	165	69
70	90	1458	153	81
80	80	1457	163	82
90	86	1463	157	76
100	78	1464	165	75
110	79	1448	164	91
120	81	1449	162	90
mean	80.5	1458.8	162.5	80.2
	origTSorigTH			
Num Cluster	**TN**	**TP**	**FP**	**FN**
30	75	1437	155	23
40	76	1432	154	28
50	84	1425	146	35
60	82	1427	148	33
70	83	1432	147	28
80	83	1434	147	26
90	83	1426	147	34
100	88	1420	142	40
110	88	1428	142	32
120	84	1420	146	40
mean	82.6	1428.1	147.4	31.9

TABLE 3 Selection of the clusters method using the SAX-MAN classifier. Aggregation of the confusion matrix results for all the participants.

	normTSnormTH			
Cluster Method	**TN**	**TP**	**FP**	**FN**
ALL-LABELS	**101**	**1422**	**129**	**38**
BY-LABEL	122	1355	108	105
	normTSscldTH			
Cluster Method	**TN**	**TP**	**FP**	**FN**
ALL-LABELS	**78**	**1470**	**165**	**69**
BY-LABEL	107	1388	136	151
	origTSorigTH			
Cluster Method	**TN**	**TP**	**FP**	**FN**
ALL-LABELS	**82**	**1427**	**148**	**33**
BY-LABEL	90	1392	140	68

TABLE 4 Results obtained for the standard data sets. Evaluation of the SAX-MAN, SAX-MAN + TL, KNN and SVM classifiers. Aggregation of the results from the confusion matrix for all the scenarios and methods. TL stands for transfer learning.

		normTSnormTH															
		SAX-MAN				**SAX-MAN + TL**				**KNN**				**SVM**			
source	**parID**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**
TST	1	0	40	3	13	0	53	3	0	0	53	3	0	0	51	3	2
TST	2	0	60	0	0	0	56	0	4	0	57	0	3	0	60	0	0
TST	3	17	40	2	20	7	60	12	0	2	60	17	0	8	58	11	2
TST	4	0	42	4	18	0	60	4	0	0	60	4	0	0	57	4	3
TST	5	27	34	0	16	12	49	15	1	3	50	24	0	11	48	16	2
TST	6	0	55	0	0	0	55	0	0	0	55	0	0	0	55	0	0
TST	7	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	8	0	57	3	10	0	67	3	0	0	67	3	0	0	66	3	1
TST	9	0	52	0	0	0	52	0	0	0	52	0	0	0	52	0	0
TST	10	0	35	7	25	1	59	6	1	0	60	7	0	0	57	7	3
TST	11	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
UMA Fall	1	30	57	0	38	15	94	15	1	3	95	27	0	12	91	18	4
UMA Fall	2	0	60	0	0	0	59	0	1	0	60	0	0	0	60	0	0
UMA Fall	3	43	45	0	45	25	87	18	3	6	88	37	2	21	84	22	6
UMA Fall	4	5	47	2	27	5	68	2	6	0	69	7	5	2	69	5	5
UMA Fall	9	0	56	3	37	0	90	3	3	0	91	3	2	0	88	3	5
UMA Fall	12	14	11	1	20	11	30	4	1	1	31	14	0	6	28	9	3
UMA Fall	15	10	30	4	15	10	44	4	1	1	45	13	0	4	44	10	1
UMA Fall	16	39	221	0	51	15	262	24	10	5	263	34	9	19	266	20	6
UMA Fall	17	14	28	2	35	0	57	16	6	1	58	15	5	5	58	11	5
		normTSscldTH															
TST	1	6	30	0	30	0	57	6	3	0	60	6	0	2	33	4	27
TST	2	0	60	0	0	0	55	0	5	0	60	0	0	0	60	0	0
TST	3	0	29	7	31	0	54	7	6	0	60	7	0	0	31	7	29
TST	4	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	5	32	42	0	18	13	59	19	1	3	60	29	0	13	43	19	17
TST	6	5	38	2	22	0	60	7	0	0	60	7	0	2	40	5	20
TST	7	0	60	0	0	0	59	0	1	0	60	0	0	0	60	0	0
TST	8	0	67	0	0	0	67	0	0	0	67	0	0	0	67	0	0
TST	9	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	10	0	31	7	29	2	59	5	1	0	60	7	0	0	33	7	27
TST	11	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
UMA Fall	1	30	67	0	33	9	95	21	5	3	100	27	0	14	70	16	30
UMA Fall	2	0	60	0	0	0	57	0	3	0	60	0	0	0	60	0	0
UMA Fall	3	7	59	2	31	2	88	7	2	0	90	9	0	3	62	6	28
UMA Fall	4	5	56	2	18	3	73	4	1	0	74	7	0	2	58	5	16
UMA Fall	9	22	65	0	28	14	91	8	2	2	93	20	0	9	66	13	27
UMA Fall	12	24	10	0	30	9	39	15	1	3	40	21	0	11	11	13	29
UMA Fall	15	10	33	4	17	6	50	8	0	1	50	13	0	4	34	10	16
UMA Fall	16	48	192	0	92	17	257	31	27	7	280	41	4	23	197	25	87
UMA Fall	17	30	27	0	54	3	70	27	11	3	80	27	1	13	33	17	48
		origTSorigTH															
TST	1	0	40	3	13	0	52	3	1	0	53	3	0	0	51	3	2
TST	2	0	60	0	0	0	58	0	2	0	60	0	0	0	60	0	0
TST	3	17	35	2	25	2	57	17	3	2	60	17	0	8	57	11	3
TST	4	0	46	4	14	0	58	4	2	0	60	4	0	0	59	4	1
TST	5	27	33	0	17	10	48	17	2	3	50	24	0	13	48	14	2
TST	6	0	55	0	0	0	55	0	0	0	55	0	0	0	55	0	0
TST	7	0	60	0	0	0	58	0	2	0	60	0	0	0	60	0	0
TST	8	0	55	3	12	0	67	3	0	0	67	3	0	0	66	3	1
TST	9	0	52	0	0	0	52	0	0	0	52	0	0	0	52	0	0
TST	10	0	39	7	21	0	59	7	1	0	60	7	0	0	57	7	3
TST	11	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
UMA Fall	1	30	71	0	24	12	93	18	2	3	95	27	0	12	92	18	3
UMA Fall	2	0	60	0	0	0	58	0	2	0	60	0	0	0	60	0	0
UMA Fall	3	43	49	0	41	18	89	25	1	6	90	37	0	21	86	22	4
UMA Fall	4	5	55	2	19	5	71	2	3	0	74	7	0	2	72	5	2
UMA Fall	9	0	67	3	26	0	91	3	2	0	93	3	0	0	90	3	3
UMA Fall	12	14	20	1	11	14	31	1	0	1	31	14	0	5	30	10	1
UMA Fall	15	10	28	4	17	7	43	7	2	1	45	13	0	4	42	10	3
UMA Fall	16	39	239	0	33	13	266	26	6	4	271	35	1	18	266	21	6
UMA Fall	17	14	37	2	26	1	61	15	2	1	63	15	0	6	60	10	3

TABLE 5 Evaluation of the SAX-MAN, SAX-MAN+TL, KNN and SVM classifiers for normTSnormTH scenario. Participants from the TST and UMA Fall datasets. Mean and standard deviation values of the Accuracy and Sensitivity for all the scenarios and methods. TL stands for transfer learning.

		normTSnormTH															
		SAX-MAN				SAX-MAN + TL				KNN				SVM			
		Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
source	**parID**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**
TST	1	0.7219	0.1201	0.7533	0.0932	0.9548	0.0731	1.0000	0.0000	0.9512	0.1044	1.0000	0.0000	0.9051	0.0546	0.9562	0.0611
TST	2	1.0000	0.0000	1.0000	0.0000	0.9333	0.0861	0.9333	0.0861	0.9531	0.0786	0.9441	0.1286	1.0000	0.0000	1.0000	0.0000
TST	3	0.7196	0.1002	0.6667	0.1111	0.8552	0.0849	1.0000	0.0000	0.7908	0.0818	1.0000	0.0000	0.8282	0.1230	0.9724	0.1207
TST	4	0.6619	0.1009	0.7000	0.0703	0.9429	0.0738	1.0000	0.0000	0.9463	0.1283	1.0000	0.0000	0.8965	0.0418	0.9573	0.0663
TST	5	0.7847	0.0851	0.6683	0.1287	0.7966	0.0977	0.9750	0.0791	0.6884	0.0757	1.0000	0.0000	0.7736	0.1372	0.9694	0.0608
TST	6	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	7	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	8	0.8179	0.0480	0.8500	0.0115	0.9625	0.0604	1.0000	0.0000	0.9538	0.0949	1.0000	0.0000	0.9470	0.0754	0.9928	0.0952
TST	9	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	10	0.5286	0.1463	0.5833	0.1416	0.9000	0.0964	0.9833	0.0527	0.9015	0.0773	1.0000	0.0000	0.8435	0.0489	0.9593	0.1054
TST	11	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	1	0.6959	0.1296	0.6000	0.1743	0.8748	0.0827	0.9900	0.0316	0.7814	0.0791	1.0000	0.0000	0.8203	0.0482	0.9632	0.0772
UMA Fall	2	1.0000	0.0000	1.0000	0.0000	0.9833	0.0527	0.9833	0.0527	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	3	0.6622	0.0391	0.5000	0.0786	0.8432	0.0713	0.9667	0.0750	0.7108	0.0887	0.9724	0.0018	0.7828	0.0189	0.9410	0.1083
UMA Fall	4	0.6467	0.1644	0.6321	0.1760	0.9045	0.1245	0.9179	0.1188	0.8551	0.0503	0.9269	0.0627	0.8850	0.0861	0.9344	0.0858
UMA Fall	9	0.5833	0.0442	0.6022	0.0485	0.9389	0.0707	0.9678	0.0520	0.9473	0.0311	0.9822	0.0831	0.9101	0.1321	0.9463	0.0947
UMA Fall	12	0.5233	0.1894	0.3417	0.2306	0.8900	0.1238	0.9750	0.0791	0.7016	0.1022	1.0000	0.0000	0.7333	0.0351	0.9051	0.0689
UMA Fall	15	0.6695	0.1247	0.6550	0.1877	0.9181	0.1357	0.9800	0.0632	0.7839	0.0824	1.0000	0.0000	0.8115	0.1045	0.9758	0.1101
UMA Fall	16	0.8362	0.0511	0.8124	0.0592	0.8921	0.0396	0.9631	0.0303	0.8544	0.0027	0.9748	0.0696	0.9192	0.0190	0.9746	0.0096
UMA Fall	17	0.5236	0.0942	0.4326	0.1494	0.7303	0.1122	0.9048	0.1323	0.7474	0.0331	0.9146	0.0842	0.8030	0.1217	0.9255	0.1141

TABLE 6 Evaluation of the SAX-MAN, SAX-MAN+TL, KNN and SVM classifiers for normTSscldTH scenario. Participants from the TST and UMA Fall datasets. Mean and standard deviation values of the Accuracy and Sensitivity for all the scenarios and methods. TL stands for transfer learning.

		normTSscldTH															
		SAX-MAN				SAX-MAN + TL				KNN				SVM			
		Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
source	parID	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
TST	1	0.5405	0.1468	0.5000	0.1361	0.8714	0.1251	0.9500	0.0805	0.9102	0.0941	1.0000	0.0000	0.5284	0.1245	0.5447	0.0215
TST	2	1.0000	0.0000	1.0000	0.0000	0.9167	0.0878	0.9167	0.0878	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	3	0.4333	0.1058	0.4833	0.1230	0.8048	0.1341	0.9000	0.1610	0.8871	0.1066	1.0000	0.0000	0.4635	0.0617	0.5169	0.0756
TST	4	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	5	0.7979	0.0980	0.7000	0.1315	0.7906	0.0892	0.9833	0.0527	0.6910	0.1121	1.0000	0.0000	0.6127	0.0048	0.7155	0.0437
TST	6	0.6548	0.1353	0.6333	0.1532	0.9071	0.1054	1.0000	0.0000	0.8947	0.0773	1.0000	0.0000	0.6260	0.0012	0.6725	0.1151
TST	7	1.0000	0.0000	1.0000	0.0000	0.9833	0.0527	0.9833	0.0527	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	8	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	9	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	10	0.4643	0.1334	0.5167	0.1459	0.9143	0.0738	0.9833	0.0527	0.8868	0.1373	1.0000	0.0000	0.4956	0.0467	0.5521	0.0269
TST	11	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	1	0.7467	0.0679	0.6700	0.0949	0.8018	0.1088	0.9500	0.0707	0.7853	0.0950	1.0000	0.0000	0.6444	0.1149	0.6955	0.0898
UMA Fall	2	1.0000	0.0000	1.0000	0.0000	0.9500	0.1125	0.9500	0.1125	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	3	0.6680	0.1163	0.6556	0.1105	0.9116	0.0533	0.9778	0.0468	0.9071	0.0869	1.0000	0.0000	0.6563	0.0290	0.6896	0.0928
UMA Fall	4	0.7565	0.0633	0.7571	0.0553	0.9442	0.0790	0.9875	0.0395	0.9157	0.1360	1.0000	0.0000	0.7342	0.1101	0.7905	0.0596
UMA Fall	9	0.7543	0.0733	0.6989	0.0832	0.9142	0.0683	0.9778	0.0468	0.8182	0.0407	1.0000	0.0000	0.6567	0.1306	0.7035	0.1095
UMA Fall	12	0.5038	0.1445	0.2500	0.1179	0.7683	0.1152	0.9750	0.0791	0.6685	0.0727	1.0000	0.0000	0.3427	0.0654	0.2754	0.1211
UMA Fall	15	0.6786	0.1635	0.6600	0.1897	0.8762	0.0662	1.0000	0.0000	0.7996	0.1361	1.0000	0.0000	0.5972	0.0172	0.6816	0.0271
UMA Fall	16	0.7222	0.0578	0.6761	0.0626	0.8265	0.0613	0.9050	0.0529	0.8654	0.0932	0.9953	0.1349	0.6689	0.0780	0.6981	0.1297
UMA Fall	17	0.5079	0.1161	0.3347	0.1212	0.6635	0.0989	0.8625	0.1376	0.7433	0.1149	0.9952	0.0939	0.4113	0.0571	0.4084	0.0546

TABLE 7 Evaluation of the SAX-MAN, SAX-MAN+TL, KNN and SVM classifiers for origTSorigTH scenario. Participants from the TST and UMA Fall datasets. Mean and standard deviation values of the Accuracy and Sensitivity for all the scenarios and methods. TL stands for transfer learning.

		origTSorigTH															
		SAX-MAN				SAX-MAN + TL				KNN				SVM			
		Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
source	parID	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
TST	1	0.7219	0.1201	0.7533	0.0932	0.9405	0.1024	0.9833	0.0527	0.9472	0.0787	1.0000	0.0000	0.9149	0.1081	0.9704	0.0368
TST	2	1.0000	0.0000	1.0000	0.0000	0.9667	0.0703	0.9667	0.0703	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	3	0.6571	0.1168	0.5833	0.1620	0.7534	0.0983	0.9500	0.0805	0.7814	0.1024	1.0000	0.0000	0.8232	0.0623	0.9502	0.1013
TST	4	0.7238	0.1256	0.7667	0.1165	0.9095	0.0784	0.9667	0.0703	0.9297	0.0072	1.0000	0.0000	0.9263	0.0836	0.9772	0.0479
TST	5	0.7722	0.0793	0.6517	0.1151	0.7524	0.0783	0.9633	0.0777	0.6861	0.0128	1.0000	0.0000	0.7999	0.0298	0.9605	0.0025
TST	6	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	7	1.0000	0.0000	1.0000	0.0000	0.9667	0.0703	0.9667	0.0703	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	8	0.7845	0.0781	0.8167	0.0794	0.9625	0.0604	1.0000	0.0000	0.9501	0.1279	1.0000	0.0000	0.9476	0.0230	0.9886	0.0974
TST	9	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	10	0.5905	0.1505	0.6500	0.1230	0.8857	0.0904	0.9833	0.0527	0.9027	0.0885	1.0000	0.0000	0.8511	0.0273	0.9519	0.0452
TST	11	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	1	0.8050	0.0937	0.7456	0.1174	0.8420	0.0649	0.9800	0.0422	0.7901	0.0037	1.0000	0.0000	0.8298	0.0926	0.9601	0.0914
UMA Fall	2	1.0000	0.0000	1.0000	0.0000	0.9667	0.0703	0.9667	0.0703	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	3	0.6904	0.0773	0.5444	0.1105	0.8056	0.0676	0.9889	0.0351	0.7231	0.0390	1.0000	0.0000	0.7997	0.0246	0.9535	0.0120
UMA Fall	4	0.7450	0.1225	0.7429	0.1203	0.9396	0.0874	0.9607	0.0866	0.9206	0.0680	1.0000	0.0000	0.9066	0.1369	0.9723	0.0762
UMA Fall	9	0.6989	0.1124	0.7222	0.1209	0.9500	0.0707	0.9800	0.0632	0.9754	0.0267	1.0000	0.0000	0.9375	0.0762	0.9691	0.0329
UMA Fall	12	0.7200	0.1474	0.6167	0.2194	0.9750	0.0791	1.0000	0.0000	0.7023	0.0636	1.0000	0.0000	0.7554	0.0452	0.9668	0.0503
UMA Fall	15	0.6419	0.1046	0.6200	0.1619	0.8448	0.0980	0.9550	0.0956	0.7853	0.1093	1.0000	0.0000	0.7804	0.0829	0.9337	0.1091
UMA Fall	16	0.8944	0.0408	0.8787	0.0491	0.8982	0.0420	0.9779	0.0258	0.8878	0.0362	0.9954	0.0494	0.9073	0.1374	0.9874	0.0740
UMA Fall	17	0.6458	0.1122	0.5782	0.1474	0.8019	0.1431	0.9675	0.0708	0.8161	0.0789	1.0000	0.0000	0.8419	0.1052	0.9445	0.0860

TABLE 8 Results obtained for the standard data sets. Evaluation of the TF-IDF, TF-IDF + TL, KNN and SVM classifiers. Aggregation of the results from the confusion matrix for all the scenarios and methods. TL stands for transfer learning.

		normTSnormTH															
		TF-IDF				**TF-IDF + TL**				**KNN**				**SVM**			
source	**parID**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**
TST	1	0	43	3	10	0	53	3	0	0	53	3	0	0	51	3	2
TST	2	0	60	0	0	0	60	0	0	0	57	0	3	0	60	0	0
TST	3	17	30	2	30	0	60	19	0	2	60	17	0	8	58	11	2
TST	4	0	50	4	10	0	60	4	0	0	60	4	0	0	57	4	3
TST	5	27	4	0	46	0	50	27	0	3	50	24	0	11	48	16	2
TST	6	0	55	0	0	0	55	0	0	0	55	0	0	0	55	0	0
TST	7	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	8	0	57	3	10	0	67	3	0	67	3	0	0	66	3	1	
TST	9	0	52	0	0	0	52	0	0	0	52	0	0	0	52	0	0
TST	10	0	50	7	10	0	60	7	0	0	60	7	0	0	57	7	3
TST	11	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
UMA Fall	1	30	29	0	66	0	95	30	0	3	95	27	0	12	91	18	4
UMA Fall	2	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
UMA Fall	3	43	0	0	90	0	90	43	0	6	88	37	2	21	84	22	6
UMA Fall	4	5	44	2	30	0	74	7	0	0	69	7	5	2	69	5	5
UMA Fall	9	0	83	3	10	0	93	3	0	0	91	3	2	0	88	3	5
UMA Fall	12	14	2	1	29	0	31	15	0	1	31	14	0	6	28	9	3
UMA Fall	15	10	15	4	30	0	45	14	0	1	45	13	0	4	44	10	1
UMA Fall	16	39	192	0	80	0	272	39	0	5	263	34	9	19	266	20	6
UMA Fall	17	14	33	2	30	0	63	16	0	1	58	15	5	5	58	11	5
		normTSscldTH															
TST	1	6	30	0	30	0	60	6	0	0	60	6	0	2	33	4	27
TST	2	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	3	0	50	7	10	0	60	7	0	0	60	7	0	0	31	7	29
TST	4	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	5	32	0	0	60	0	60	32	0	3	60	29	0	13	43	19	17
TST	6	5	30	2	30	0	60	7	0	0	60	7	0	2	40	5	20
TST	7	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	8	0	67	0	0	0	67	0	0	0	67	0	0	0	67	0	0
TST	9	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	10	0	50	7	10	0	60	7	0	0	60	7	0	0	33	7	27
TST	11	0	60	0	0	0	60	0	0	60	0	0	0	60	0	0	
UMA Fall	1	30	32	0	68	0	100	30	0	3	100	27	0	14	70	16	30
UMA Fall	2	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
UMA Fall	3	7	60	2	30	0	90	9	0	0	90	9	0	3	62	6	28
UMA Fall	4	5	44	2	30	0	74	7	0	0	74	7	0	2	58	5	16
UMA Fall	9	22	53	0	40	0	93	22	0	2	93	20	0	9	66	13	27
UMA Fall	12	24	0	0	40	0	40	24	0	3	40	21	0	11	11	13	29
UMA Fall	15	10	20	4	30	0	50	14	0	1	50	13	0	4	34	10	16
UMA Fall	16	48	184	0	100	0	284	48	0	7	280	41	4	23	197	25	87
UMA Fall	17	30	21	0	60	0	81	30	0	3	80	27	1	13	33	17	48
		origTSorigTH															
TST	1	0	43	3	10	0	53	3	0	0	53	3	0	0	51	3	2
TST	2	0	60	0	0	0	60	0	0	0	57	0	3	0	60	0	0
TST	3	17	30	2	30	0	60	19	0	2	60	17	0	8	58	11	2
TST	4	0	50	4	10	0	60	4	0	0	60	4	0	0	57	4	3
TST	5	27	20	0	30	0	50	27	0	3	50	24	0	11	48	16	2
TST	6	0	55	0	0	0	55	0	0	0	55	0	0	0	55	0	0
TST	7	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
TST	8	0	57	3	10	0	67	3	0	0	67	3	0	0	66	3	1
TST	9	0	52	0	0	0	52	0	0	0	52	0	0	0	52	0	0
TST	10	0	50	7	10	0	60	7	0	0	60	7	0	0	57	7	3
TST	11	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
UMA Fall	1	30	29	0	66	0	95	30	0	3	95	27	0	12	91	18	4
UMA Fall	2	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0
UMA Fall	3	43	0	0	90	0	90	43	0	6	88	37	2	21	84	22	6
UMA Fall	4	5	44	2	30	0	74	7	0	0	69	7	5	2	69	5	5
UMA Fall	9	0	83	3	10	0	93	3	0	0	91	3	2	0	88	3	5
UMA Fall	12	14	2	1	29	0	31	15	0	1	31	14	0	6	28	9	3
UMA Fall	15	10	15	4	30	0	45	14	0	1	45	13	0	4	44	10	1
UMA Fall	16	39	181	0	91	0	272	39	0	5	263	34	9	19	266	20	6
UMA Fall	17	14	33	2	30	0	63	16	0	1	58	15	5	5	58	11	5

TABLE 9 Evaluation of the TF-IDF, TF-IDF,+TL, KNN and SVM classifiers for normTSnormTH scenario. Participants from the TST and UMA Fall datasets. Mean and standard deviation values of the Accuracy and Sensitivity for all the scenarios and methods. TL stands for transfer learning.

		normTSnormTH															
		TF-IDF				TF-IDF + TL				KNN				SVM			
		Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
source	parID	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
TST	1	0.7729	0.0540	0.8100	0.0161	0.9548	0.0731	1.0000	0.0000	0.9512	0.1044	1.0000	0.0000	0.9051	0.0546	0.9562	0.0611
TST	2	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	0.9531	0.0786	0.9441	0.1286	1.0000	0.0000	1.0000	0.0000
TST	3	0.5933	0.0448	0.5000	0.0000	0.7690	0.0967	1.0000	0.0000	0.7908	0.0818	1.0000	0.0000	0.8282	0.1230	0.9724	0.1207
TST	4	0.7857	0.0615	0.8333	0.0000	0.9429	0.0738	1.0000	0.0000	0.9463	0.1283	1.0000	0.0000	0.8965	0.0418	0.9573	0.0663
TST	5	0.3970	0.0735	0.0700	0.0909	0.6502	0.0675	1.0000	0.0000	0.6884	0.0757	1.0000	0.0000	0.7736	0.1372	0.9694	0.0608
TST	6	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	7	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	8	0.8179	0.0480	0.8500	0.0115	0.9625	0.0604	1.0000	0.0000	0.9538	0.0949	1.0000	0.0000	0.9470	0.0754	0.9928	0.0952
TST	9	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	10	0.7500	0.0575	0.8333	0.0000	0.9000	0.0690	1.0000	0.0000	0.9015	0.0773	1.0000	0.0000	0.8435	0.0489	0.9593	0.1054
TST	11	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	1	0.4645	0.1366	0.3000	0.1610	0.7623	0.0357	1.0000	0.0000	0.7814	0.0791	1.0000	0.0000	0.8203	0.0482	0.9632	0.0772
UMA Fall	2	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	3	0.3196	0.0524	0.0000	0.0000	0.6804	0.0524	1.0000	0.0000	0.7108	0.0887	0.9724	0.0018	0.7828	0.0189	0.9410	0.1083
UMA Fall	4	0.6025	0.0413	0.5929	0.0277	0.9231	0.0885	1.0000	0.0000	0.8551	0.0503	0.9269	0.0627	0.8850	0.0861	0.9344	0.0858
UMA Fall	9	0.8656	0.0455	0.8922	0.0054	0.9700	0.0483	1.0000	0.0000	0.9473	0.0311	0.9822	0.0831	0.9101	0.1321	0.9463	0.0947
UMA Fall	12	0.3267	0.1375	0.0500	0.1054	0.6933	0.1401	1.0000	0.0000	0.7016	0.1022	1.0000	0.0000	0.7333	0.0351	0.9051	0.0689
UMA Fall	15	0.4224	0.0628	0.3250	0.0791	0.7662	0.0684	1.0000	0.0000	0.7839	0.0824	1.0000	0.0000	0.8115	0.1045	0.9758	0.1101
UMA Fall	16	0.7422	0.0132	0.7057	0.0069	0.8763	0.0409	1.0000	0.0000	0.8544	0.0027	0.9748	0.0696	0.9192	0.0190	0.9746	0.0096
UMA Fall	17	0.5849	0.0634	0.5139	0.0738	0.8128	0.1122	1.0000	0.0000	0.7474	0.0331	0.9146	0.0842	0.8030	0.1217	0.9255	0.1141

TABLE 10 Evaluation of the TF-IDF, TF-IDF,+TL, KNN and SVM classifiers for normTSscldTH scenario. Participants from the TST and UMA Fall datasets. Mean and standard deviation values of the Accuracy and Sensitivity for all the scenarios and methods. TL stands for transfer learning.

		normTSscldTH															
		TF-IDF				TF-IDF + TL				KNN				SVM			
		Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
source	parID	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
TST	1	0.5429	0.0369	0.5000	0.0000	0.9143	0.0738	1.0000	0.0000	0.9102	0.0941	1.0000	0.0000	0.5284	0.1245	0.5447	0.0215
TST	2	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	3	0.7500	0.0575	0.8333	0.0000	0.9000	0.0690	1.0000	0.0000	0.8871	0.1066	1.0000	0.0000	0.4635	0.0617	0.5169	0.0756
TST	4	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	5	0.3405	0.0733	0.0000	0.0000	0.6595	0.0733	1.0000	0.0000	0.6910	0.1121	1.0000	0.0000	0.6127	0.0048	0.7155	0.0437
TST	6	0.5214	0.0345	0.5000	0.0000	0.9071	0.1054	1.0000	0.0000	0.8947	0.0773	1.0000	0.0000	0.6260	0.0012	0.6725	0.1151
TST	7	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	8	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	9	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	10	0.7500	0.0575	0.8333	0.0000	0.9000	0.0690	1.0000	0.0000	0.8868	0.1373	1.0000	0.0000	0.4956	0.0467	0.5521	0.0269
TST	11	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	1	0.4733	0.1468	0.3200	0.1687	0.7711	0.0398	1.0000	0.0000	0.7853	0.0950	1.0000	0.0000	0.6444	0.1149	0.6955	0.0898
UMA Fall	2	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	3	0.6773	0.0264	0.6667	0.0000	0.9136	0.0679	1.0000	0.0000	0.9071	0.0869	1.0000	0.0000	0.6563	0.0290	0.6896	0.0928
UMA Fall	4	0.6025	0.0413	0.5929	0.0277	0.9231	0.0885	1.0000	0.0000	0.9157	0.1360	1.0000	0.0000	0.7342	0.1101	0.7905	0.0596
UMA Fall	9	0.6498	0.0308	0.5689	0.0215	0.8130	0.0688	1.0000	0.0000	0.8182	0.0407	1.0000	0.0000	0.6567	0.1306	0.7035	0.1095
UMA Fall	12	0.3408	0.1668	0.0000	0.0000	0.6592	0.1668	1.0000	0.0000	0.6685	0.0727	1.0000	0.0000	0.3427	0.0654	0.2754	0.1211
UMA Fall	15	0.4714	0.0369	0.4000	0.0000	0.7857	0.0615	1.0000	0.0000	0.7996	0.1361	1.0000	0.0000	0.5972	0.0172	0.6816	0.0271
UMA Fall	16	0.6985	0.0093	0.6478	0.0064	0.8562	0.0327	1.0000	0.0000	0.8654	0.0932	0.9953	0.1349	0.6689	0.0780	0.6981	0.1297
UMA Fall	17	0.4516	0.0677	0.2583	0.0264	0.7383	0.0783	1.0000	0.0000	0.7433	0.1149	0.9952	0.0939	0.4113	0.0571	0.4084	0.0546

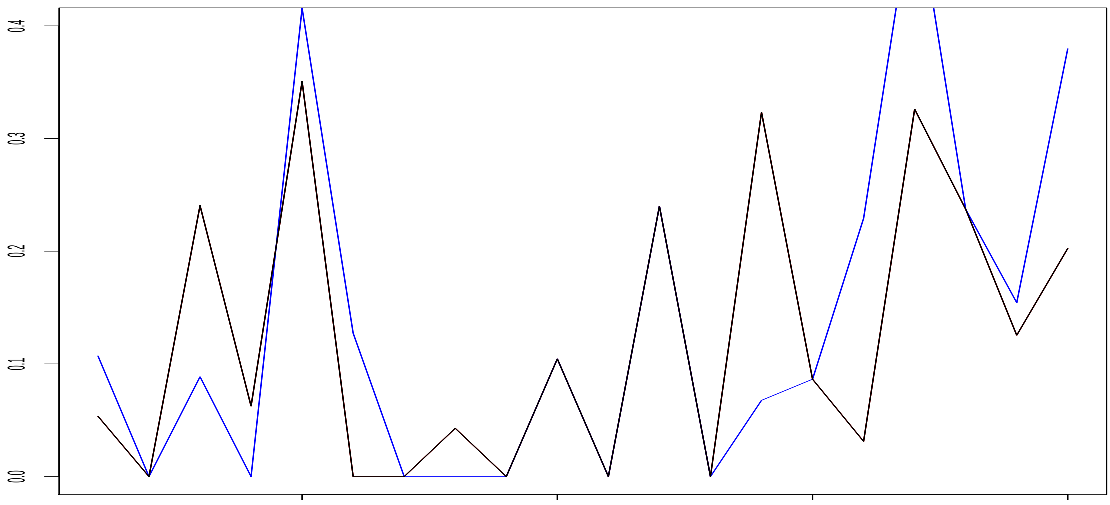

FIGURE 7 The evolution of the ratio $\mathrm{FP}/(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})$ when using the TF-IDF classifier for each participant. In black/blue/red the ratio for the normTSnormTH/normTSscldTH/origTSorigTH scenarios, respectively.

With all the previous results, we observed that TL induces a great improvement in both SAX-MAN and TF-IDF classifiers as long as the undetected falls almost disappeared. It is worth noticing that the number of undetected falls is really impressive, beating any other method we have studied so far. The SAX-MAN classifier, together with the TL stage, behaves really fine if we consider the four counters and the measurements of the Accuracy and Sensitivity. On the other hand, the TF-IDF (see table 8) identifies all the falls but at a cost of too many false alarms (too many FP). Basically, it seems that the TF-IDF with the TL is always assigning the FALL label, which is not desired. The conclusion about the TF-IDF is that either the distance function or the retrieving algorithm need enhancement to become as competitive as the SAX-MAN.

3.2 | Deployment of the Fall Detection system with new and independent participants

In this experiment we used the best configuration find so far: 80 clusters using the ALL-LABELS clustering and the SAX-MAN classifier. Results from this experimental set up includes:

- for each scenario, the aggregation of the confusion matrix among the folds in Table 12, and
- the mean and standard deviation of the Accuracy and the Sensitivity of the classifier with and without the transfer learning for each scenario and participant in Table 13.

The SAX-MAN performance without the TL stage is totally worthless as long no fall event is detected. When TL is included in the learning of this classifier, then its behaviour changes greatly. Nevertheless, it also lacks in a extremely high number of false alarms (FP); clearly, more research is needed to improve the SAX-MAN classification execution. The relevant point is that SAX-MAN performs uniformly on the three different scenarios, which clearly suggests that a better TL design would lead to most promising results. Furthermore, the on-line learning needs to be more aggressive, that is, gaining prevalence versus the knowledge from the TL.

On the other hand, the KNN and SVM do not show the same performance in the three different scenarios. There are two remarkable issues with these results: i) the high rate of false alarms and ii) the detection of all of the fall events. WRT the false alarms, the KNN shows similar number than SAX-MAN, while SVM behaves with a smaller amount of FP. In the case of KNN, with some on-line learning stage this number might be reduced, in the case of SVM this means that unless a better training data set is produced there would be no option to enhance the classifier.

The second point that is remarkable is that, in two out of three scenarios, both KNN and SVM are able to detect all the fall events. This is really shocking as no method using either KNN or SVM produced such impressive results Khojasteh et al. (2018). More surprisingly is the fact that there is a mismatch in the scenarios for which both method detect all the alarms. This fact suggests the robustness of these two solutions needs improvement.

In any case, we have to consider the nature of this FallOVI dataset. Three different participants used the smart-watch during one day in their normal lives, during the three periods there were no fall reported. Thus, the ADL's data comes from this three-day stage. The falls were performed

TABLE 11 Evaluation of the TF-IDF, TF-IDF+TL, KNN and SVM classifiers for origTSorigTH scenario. Participants from the TST and UMA Fall datasets. Mean and standard deviation values of the Accuracy and Sensitivity for all the scenarios and methods. TL stands for transfer learning.

		origTSorigTH															
		TF-IDF				TF-IDF + TL				KNN				SVM			
		Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
source	parID	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
TST	1	0.7729	0.0540	0.8100	0.0161	0.9548	0.0731	1.0000	0.0000	0.9472	0.0787	1.0000	0.0000	0.9149	0.1081	0.9704	0.0368
TST	2	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	3	0.5933	0.0448	0.5000	0.0000	0.7690	0.0967	1.0000	0.0000	0.7814	0.1024	1.0000	0.0000	0.8232	0.0623	0.9502	0.1013
TST	4	0.7857	0.0615	0.8333	0.0000	0.9429	0.0738	1.0000	0.0000	0.9297	0.0072	1.0000	0.0000	0.9263	0.0836	0.9772	0.0479
TST	5	0.6036	0.0555	0.3850	0.1029	0.6502	0.0675	1.0000	0.0000	0.6861	0.0128	1.0000	0.0000	0.7999	0.0298	0.9605	0.0025
TST	6	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	7	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	8	0.8179	0.0480	0.8500	0.0115	0.9625	0.0604	1.0000	0.0000	0.9501	0.1279	1.0000	0.0000	0.9476	0.0230	0.9886	0.0974
TST	9	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
TST	10	0.7500	0.0575	0.8333	0.0000	0.9000	0.0690	1.0000	0.0000	0.9027	0.0885	1.0000	0.0000	0.8511	0.0273	0.9519	0.0452
TST	11	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	1	0.4645	0.1366	0.3000	0.1610	0.7623	0.0357	1.0000	0.0000	0.7901	0.0037	1.0000	0.0000	0.8298	0.0926	0.9601	0.0914
UMA Fall	2	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
UMA Fall	3	0.3196	0.0524	0.0000	0.0000	0.6804	0.0524	1.0000	0.0000	0.7231	0.0390	1.0000	0.0000	0.7997	0.0246	0.9535	0.0120
UMA Fall	4	0.6025	0.0413	0.5929	0.0277	0.9231	0.0885	1.0000	0.0000	0.9206	0.0680	1.0000	0.0000	0.9066	0.1369	0.9723	0.0762
UMA Fall	9	0.8656	0.0455	0.8922	0.0054	0.9700	0.0483	1.0000	0.0000	0.9754	0.0267	1.0000	0.0000	0.9375	0.0762	0.9691	0.0329
UMA Fall	12	0.3267	0.1375	0.0500	0.1054	0.6933	0.1401	1.0000	0.0000	0.7023	0.0636	1.0000	0.0000	0.7554	0.0452	0.9668	0.0503
UMA Fall	15	0.4224	0.0628	0.3250	0.0791	0.7662	0.0684	1.0000	0.0000	0.7853	0.1093	1.0000	0.0000	0.7804	0.0829	0.9337	0.1091
UMA Fall	16	0.7075	0.0648	0.6663	0.0720	0.8763	0.0409	1.0000	0.0000	0.8878	0.0362	0.9954	0.0494	0.9073	0.1374	0.9874	0.0740
UMA Fall	17	0.5849	0.0634	0.5139	0.0738	0.8128	0.1122	1.0000	0.0000	0.8161	0.0789	1.0000	0.0000	0.8419	0.1052	0.9445	0.0860

TABLE 12 Results obtained for the FallOVI data set. Evaluation of the classifiers. Aggregation of the results from the confusion matrix for all the scenarios and methods. TL stands for transfer learning.

		normTSnormTH															
		SAX-MAN				SAX-MAN + TL				KNN				SVM			
source	**parID**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**
FallOVI	1	317	0	3	50	94	21	226	29	93	20	227	30	151	50	169	0
FallOVI	2	538	0	2	50	251	21	289	29	190	27	350	23	290	50	250	0
FallOVI	3	256	0	4	50	64	48	196	2	100	50	160	0	130	50	130	0
TOTAL		1111	0	9	150	409	90	711	60	383	97	737	53	571	150	549	0
		normTSscldTH															
		SAX-MAN				**SAX-MAN + TL**				**KNN**				**SVM**			
source	**parID**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**
FallOVI	1	328	18	2	32	132	50	198	0	58	50	272	0	152	20	178	30
FallOVI	2	508	10	2	40	158	31	352	19	158	50	352	0	220	30	290	20
FallOVI	3	256	20	4	30	77	40	183	10	100	50	160	0	160	30	100	20
TOTAL		1092	48	8	102	367	121	733	29	316	150	784	0	532	80	568	70
		origTSorigTH															
		SAX-MAN				**SAX-MAN + TL**				**KNN**				**SVM**			
source	**parID**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**	**TN**	**TP**	**FP**	**FN**
FallOVI	1	316	2	4	48	92	40	228	10	112	50	208	0	137	50	183	0
FallOVI	2	538	0	2	50	205	40	335	10	134	50	406	0	290	50	250	0
FallOVI	3	256	9	4	41	50	30	210	20	73	50	187	0	150	50	110	0
TOTAL		1110	11	10	139	347	110	773	40	319	150	801	0	577	150	543	0

with a dummy, either simulating a forward fall when walking or a fall due to a faint. And the data from these falls were assumed the falls for each of the three individuals.

From the results it seems that the falls are less related with the activities of each of the participants from FallOVI but more related with the falls in the TST and UMA Fall. Besides, there are no correlation with the expectations we had. We knew that participant number 1 did a lot of sports and high acceleration ADLs during the day wearing the smart-watch, while the two remaining participants performed low intensity ADLs. We were expecting to obtain a difference in the results.

In our opinion, this experimentation shows that there is a need of gathering data from real cases when participants from the focused population, unfortunately, suffer a fall. We are working on this issue nowadays; we are recording data in a senior house during, at least, three months in order to capture enough information concerning falls. We do expect then be able to propose solutions adapted to what would be really observed.

Introducing ensembles could be the first attempt to improve the overall FD performance; nevertheless, introducing the three classifiers might lead to problems with the methods to merge the uncertainty each method. We will continue this study finding the suitable ensemble. Besides, the performance of SVM is not bad in the sense that the number of TN high and the number of FN is small. The question of the performance of one-class SVM to detect the NOT_FALL cases followed by a classifier like the SAX-MAN+TL or KNN detecting the FALL events might be interesting, these two latters including on-line learning.

3.3 | Discussion about the results

Concerning the performance of the TL, it can be stated that in all the studied cases (SAX-MAN with TST/UMAFall, TF-IDF with TST/UMA Fall and SAX-MAN with FallOvi), TL clearly improves the original method; moreover, any of the classifiers together with TL shows similar results than KNN but with a reduced number of false alarms (FN).

KNN and SVM have been used as reference methods in the comparison and have not been extended with TL. However, the KNN can be easily enhanced with TL by considering only the instances from participants with similar behaviour than the current user. Besides, the studies in LM., SB., and et al. (2016); P. Wu and Dietterich (2004) focused on introducing TL for these two classifiers.

In this paper, a well-known peak detection algorithm variant of the Abbate et al. (2012) proposal is used but adapted to have the 3DACC placed on a wrist Khojasteh et al. (2018). There are several proposals in the literature based on manually configured thresholds, A. Bourke et al. (2007); Fang and Dzeng (2014 2017). The Abbate proposal makes use of a very simple finite state machine to detect the fall events based on a set of configuration thresholds. To study the relevance of these thresholds we used the same experimentation scenarios proposed in Khojasteh

TABLE 13 Results obtained for the FallOVI data set. Evaluation of the classifiers. Aggregation of the results from the confusion matrix for all the scenarios and methods. TL stands for transfer learning.

	normTSnormTH															
	SAX-MAN				SAX-MAN + TL				KNN				SVM			
	Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
parID	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**
1	0.8567	0.0182	0.0000	0.0000	0.3108	0.0191	0.4200	0.0632	0.3054	0.0130	0.0810	0.0016	0.5432	0.0085	0.2283	0.0034
2	0.9118	0.0071	0.0000	0.0000	0.4610	0.0175	0.4200	0.0632	0.3677	0.0407	0.0720	0.0145	0.5762	0.0000	0.1666	0.0000
3	0.8258	0.0225	0.0000	0.0000	0.3612	0.0366	0.9600	0.0843	0.4838	0.0000	0.2380	0.0000	0.5806	0.0000	0.2777	0.0000

	normTSscldTH															
	SAX-MAN				SAX-MAN + TL				KNN				SVM			
	Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
parID	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**
1	0.9105	0.0135	0.3600	0.0843	0.4789	0.0110	1.0000	0.0000	0.2842	0.0166	0.1553	0.0030	0.4526	0.0110	0.1010	0.0022
2	0.9250	0.0075	0.2000	0.0000	0.3375	0.0284	0.6200	0.0632	0.3714	0.0112	0.1244	0.0018	0.4464	0.0000	0.0937	0.0000
3	0.8903	0.0225	0.4000	0.0000	0.3774	0.0374	0.8000	0.0000	0.4838	0.0000	0.2380	0.0000	0.6129	0.0000	0.2307	0.0000

	origTSorigTH															
	SAX-MAN				SAX-MAN + TL				KNN				SVM			
	Acc		Sen		Acc		Sen		Acc		Sen		Acc		Sen	
parID	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**
1	0.8594	0.0248	0.0400	0.0843	0.3567	0.0213	0.8000	0.0000	0.4378	0.0170	0.1939	0.0047	0.5054	0.0130	0.2146	0.0043
2	0.9118	0.0071	0.0000	0.0000	0.4152	0.0183	0.8000	0.0000	0.3118	0.0087	0.1096	0.0012	0.5762	0.0000	0.1666	0.0000
3	0.8548	0.0228	0.1800	0.0632	0.2580	0.0263	0.6000	0.0000	0.3967	0.0155	0.2110	0.0043	0.6451	0.0000	0.3125	0.0000

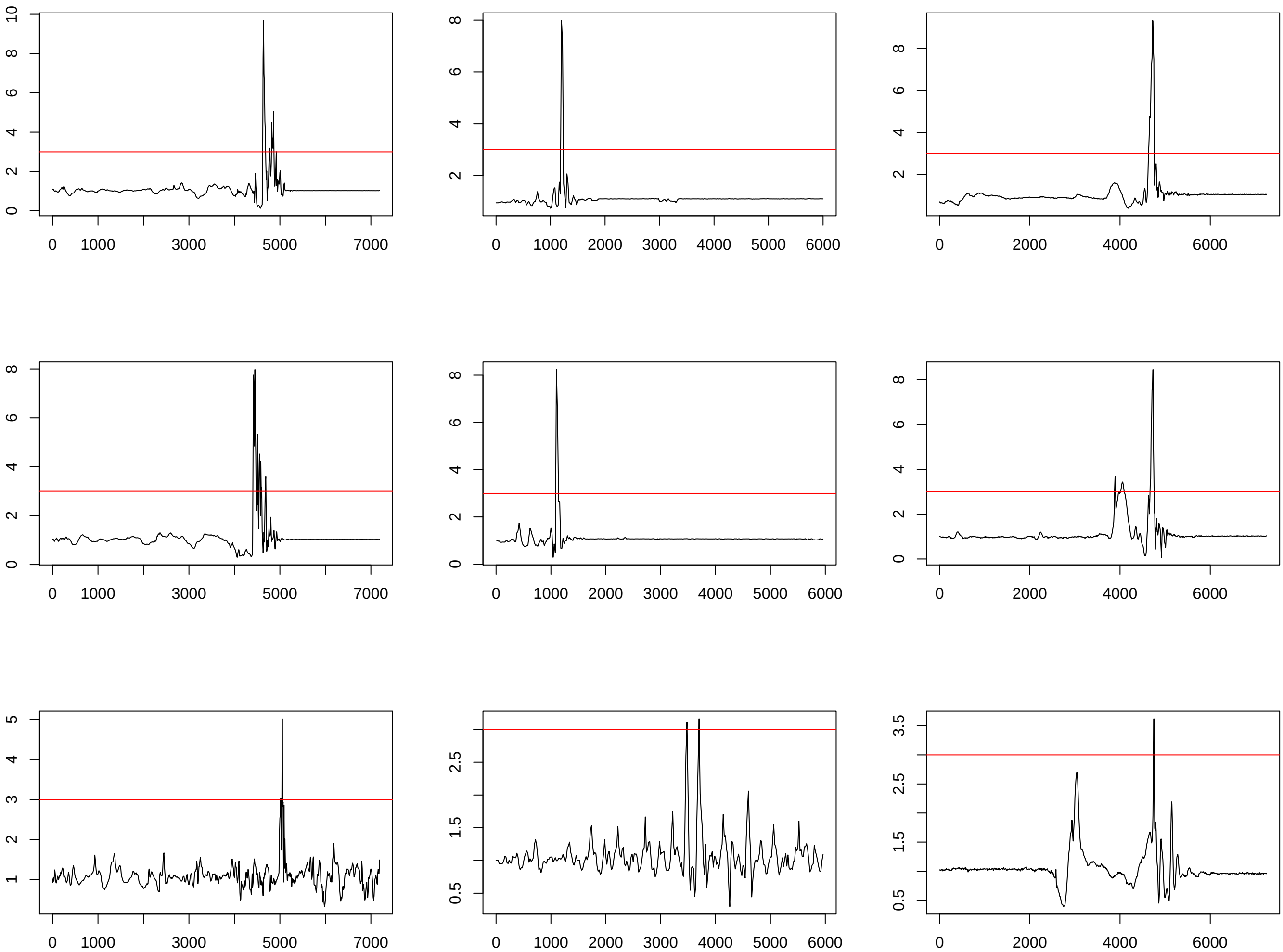

FIGURE 8 Examples of TS. Left-most column includes TS from FallOVI, the column in the center includes TS from UMAFall, while the right-most column focuses on TST. The two upper rows includes FALL TS, while the bottom one shows NOT_FALL TS.

et al. (2018) (see subsection 2.2.2). As in that research, the final results vary with the threshold configuration, concluding that the event detection method must be as free of parameters as possible. Alternatives, such the one presented in Villar and Villar (2019) could reduce the number of false peaks and improve the performance of the classification process.

Finally, we can state that the use of a dummy to mimic real falls has been a good election, since the TSs obtained for the dummy are very similar to the ones in the public data sets. Fig 8 includes several pictures from the three data sets: FallOVI, UMA Fall and TST, where it can be observed that FallOvi-FALL TSs are quite similar to UMA and TST ones. So, this confirm the good performance of the TL process for the FallOVI dataset.

4 | CONCLUSIONS

This study proposes a fall detection method focused on senior citizens using smart-watches with 3DACC. This approach differs from the majority of the solutions in the literature because it is user-centred: a model is developed for each user based on a training stage performing ADLs. The solution includes a peak detection stage, where the acceleration magnitude in the vicinity of the peak is represented using SAX. The SAX words are classified using two different techniques: using the SAX-SVM classifier but with the Manhattan distance instead of the cosine distance call

SAX-MAN, and a TF-IDF classifier. Transfer Learning has also been considered, so the obtained classifiers are enriched with previous knowledge (i.e., data gathered from other similar users) by means of a bag of relevant SAX words.

The experimentation stage includes the quality evaluation of the clustering methods and parameters to create the bag of relevant words, the evaluation of the two classifiers defined in this study using two publicly available realistic falls and ADLs data sets, and a comparison of these two classifiers versus two state-of-the-art solutions (KNN and SVM). In this latter comparison, preparing the bag of relevant words and the learning of the KNN and SVM is done using the two publicly available realistic falls; the evaluation is measured using a new data set prepared by this research team using a life saving mannequin. This data set will be published accompanying this manuscript.

From the results using the two publicly available data sets, it is worth noticing the enhancement effect the TL produces in the performance of the SAX-MAX and the TF-IDF classifiers. In the case of the SAX-MAX, there are almost no undetected falls, while for the TF-IDF there are no undetected falls. The SAX-MAN classifier behaves with a very reduced number of false alarms, a very interesting solution as long as this fact improves the experience of the user. Also, it is relevant the competitive results obtained when using raw data (origTSorigTH scenario), with reduced aggregated FN and similar aggregated FP than the also successful normTSnormTH scenario.

Results using the FallOVI data set shows the improvement in the SAX-MAN when using the TL, although too many undetected falls and false alarms are still present. This problem seems to be present despite the online learning; therefore, the on-line learning might be more aggressive than it is now. Also, the TL stage should be redesigned in order to reduce the FN and FP. The SVM and KNN performed extremely well in two out of three scenarios if we consider the FN. Nevertheless, the fact that they did not concur in these two successful scenarios induces these methods highly depend on the configuration of the experiment. Anyway, the number of FP is also high, which in the case of the SVM means that it could be barely enhanced. The results suggest that it might be interesting to evaluate whether using ensembles improves the results and also to introduce an one-class NOT_FALL SVM detection followed by a FALL/NOT_FALL classifier (which could be either the improved SAX-MAN or the KNN, both with TL and on-line learning). These two leads are the base of future work.

ACKNOWLEDGMENTS

This research has been funded by the Spanish Ministry of Science and Innovation, under project MINECO-TIN2017-84804-R, and by the Grant FCGRUPIN-IDI/2018/000226 project from the Asturias Regional Government.

Author contributions

All the authors have equally contributed to this research.

Financial disclosure

This research has been funded by the Spanish Ministry of Science and Innovation, under project MINECO-TIN2017-84804-R.

Conflict of interest

The authors declare no potential conflict of interests.

References

Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., & AlessioVecchio. (2012, December). A smartphone-based fall detection system. *Pervasive and Mobile Computing*, *8*(6), 883-899.

Bianchi, F., Redmond, S. J., Narayanan, M. R., Cerutti, S., & Lovell, N. H. (2010). Barometric pressure and triaxial accelerometry-based falls event detection. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *18*(6), 619-627. doi: 10.1109/TNSRE.2010.2070807

Bourke, A., O'Brien, J., & Lyons, G. (2007). Evaluation of a threshold-based triaxial accelerometer fall detection algorithm. *Gait and Posture*, *26*, 194-199.

Bourke, A. K., Klenk, J., Schwickert, L., Aminian, K., Ihlen, E. A. F., Mellone, S., … Becker, C. (2016). Fall detection algorithms for real-world falls harvested from lumbar sensors in the elderly population: A machine learning approach. In *2016 38th annual international conference of the ieee engineering in medicine and biology society (embc)* (p. 3712-3715). doi: 10.1109/EMBC.2016.7591534

Casilari, E., & Oviedo-Jimínez, M. A. (2015, 11). Automatic fall detection system based on the combined use of a smartphone and a smartwatch. *PLOS ONE*, *10*(11), 1-11. Retrieved from https://doi.org/10.1371/journal.pone.0140929 doi: 10.1371/journal.pone.0140929

Casilari, E., Santoyo-Ramón, J.-A., & Cano-García, J.-M. (2017a). Analysis of public datasets for wearable fall detection systems. *Sensors*, *17*(1513), 4324 - 4338. Retrieved from http://www.mdpi.com/1424-8220/17/7/1513 doi: https://10.3390/s17071513

Casilari, E., Santoyo-Ramón, J. A., & Cano-García, J. M. (2017b). Umafall: A multisensor dataset for the research on automatic fall detection. *Procedia Computer Science*, *110*(Supplement C), 32 - 39. Retrieved from http://www.sciencedirect.com/science/article/pii/S1877050917312899 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017) / 12th International Conference on Future Networks and Communications (FNC 2017) / Affiliated Workshops.

Chaudhuri, S., Thompson, H., & Demiris, G. (2014). Fall detection devices and their use with older adults: A systematic review. *J Geriatr Phys Ther*, *37*, 178-196.

Delahoz, Y. S., & Labrador, M. A. (2014). Survey on fall detection and fall prevention using wearable and external sensors. *Sensors*, *14*(10), 19806-19842. Retrieved from http://www.mdpi.com/1424-8220/14/10/19806/htm doi: doi:10.3390/s141019806

Deutsch, M., & Burgsteiner, H. (2016). Health informatics meets ehealth. In G. Schreier, E. Ammenwerth, A. Hörbst, & D. Hayn (Eds.), (Vol. 223, p. 259-266). IOS Press. doi: 10.3233/978-1-61499-645-3-259

Fang, Y. C., & Dzeng, R. J. (2014). A smartphone-based detection of fall portents for construction workers. *Procedia Eng.*, *85*, 147-156.

Fang, Y. C., & Dzeng, R. J. (2017). Accelerometer-based fall-portent detection algorithm for construction tiling operation. *Autom. Constr.*, *84*, 214-230.

Gasparrini, S., Cippitelli, E., Gambi, E., Spinsante, S., Wahslen, J., Orhan, I., & Lindh, T. (2016, October). Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion. In *Ict innovations 2015, advances in intelligent systems and computing* (Vol. 399, p. 99-108). Springer. Retrieved from http://www.tlc.dii.univpm.it/blog/databases4kinectandhttps://ieee-dataport.org/documents/tst-fall-detection-dataset-v2 doi: https://doi.org/10.1007/978-3-319-25733-4_11

Gjoreski, H., Bizjak, J., & Gams, M. (2016, Sept). Using smartwatch as telecare and fall detection device. In *2016 12th international conference on intelligent environments (ie)* (p. 242-245). doi: 10.1109/IE.2016.55

Hakim, A., Huq, M. S., Shanta, S., & Ibrahim, B. (2017). Smartphone based data mining for fall detection: Analysis and design. *Procedia Computer Science*, *105*, 46-51. Retrieved from http://www.sciencedirect.com/science/article/pii/S1877050917302065 doi: https://doi.org/10.1016/j.procs.2017.01.188

Huynh, Q. T., Nguyen, U. D., Irazabal, L. B., Ghassemian, N., & Tran, B. Q. (2015). Optimization of an accelerometer and gyroscope-based fall detection algorithm. *Journal of Sensors*, *2015*, *Article ID 452078*(452078), 8 pages. doi: http://dx.doi.org/10.1155/2015/452078

Igual, R., Medrano, C., & Plaza, I. (2013). Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine*, *12*(66). Retrieved from http://www.biomedical-engineering-online.com/content/12/1/66

Igual, R., Medrano, C., & Plaza, I. (2015). A comparison of public datasets for acceleration-based fall detection. *Medical Engineering and Physics*, *37*(9), 870-878. Retrieved from http://www.sciencedirect.com/science/article/pii/S1350453315001575 doi: https://doi.org/10.1016/j.medengphy.2015.06.009

Jatesiktat, P., & Ang, W. T. (2017, July). An elderly fall detection using a wrist-worn accelerometer and barometer. In *2017 39th annual international conference of the ieee engineering in medicine and biology society (embc)* (p. 125-130). doi: 10.1109/EMBC.2017.8036778

Kangas, M., Konttila, A., Lindgren, P., Winblad, I., & Jämsaä, T. (2008). Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait and Posture*, *28*, 285-291.

Kangas, M., Vikman, I., Nyberg, L., Korpelainen, R., Lindblom, J., & Jamsa, T. (2012). Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects. *Gait and Posture*, *35*, 500-505.

Khan, S. S., & Hoey, J. (2017). Review of fall detection techniques: A data availability perspective. *Medical Engineering and Physics*, *39*, 12-22. Retrieved from http://www.sciencedirect.com/science/article/pii/S1350453316302600 doi: https://doi.org/10.1016/j.medengphy.2016.10.014

Khojasteh, S. B., Villar, J. R., Chira, C., González, V. M., & de la Cal, E. (2018). Improving fall detection using an on-wrist wearable accelerometer. *Sensors*, *18*, 1350. doi: https://doi.org/10.3390/s18051350

Kostopoulos, P., Nunes, T., Salvi, K., Deriaz, M., & Torrent, J. (2015, Oct). F2d: A fall detection system tested with real data from daily life of elderly people. In *2015 17th international conference on e-health networking, application services (healthcom)* (p. 397-403).

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th acm sigmod workshop on research issues in data mining and knowledge discovery* (p. 2-11). ACM.

LM., B., SB., H., & et al., W. D. (2016). Learning from heterogeneous data sources: An application in spatial proteomics. *PLoS Comput Biol*, *12*, 1–26.

Medrano, C., Plaza, I., Igual, R., Sánchez, Á., & Castr0, M. (2016). The effect of personalization on smartphone-based fall detectors. *Sensors*, *16*, *Article ID 117*(1), 117. Retrieved from http://www.mdpi.com/1424-8220/16/1/117/htm doi: 10.3390/s16010117

Ngu, A., Wu, Y., Zare, H., Polican, A., Yarbrough, B., & Yao, L. (2017). Fall detection using smartwatch sensor data with accessor architecture. In H. Chen, D. Zeng, E. Karahanna, & B. I. (Eds.), *Proceedings of the international conference on smart health icsh 2017, lecture notes in computer science* (Vol. 10347, p. 81-93). Springer.

Putra, I. P. E. S., Brusey, J., Gaura, E., & Vesilo, R. (2018). An event-triggered machine learning approach for accelerometer-based fall detection. *Sensors*, *18*, 2034. doi: https://doi:10.3390/s18010020

Roberts, C. (2018, November). *How to choose a medical alert system.* Comsumer Reports, USA. Retrieved from `https://www.consumerreports.org/medical-alert-systems/how-to-choose-a-medical-alert-system/`

Sabatini, A. M., Ligorio, G., Mannini, A., Genovese, V., & Pinna, L. (2016, July). Prior-to- and post-impact fall detection using inertial and barometric altimeter measurements. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *24*(7), 774-783. Retrieved from `http://ieeexplore.ieee.org/abstract/document/7173441/` doi: 10.1109/TNSRE.2015.2460373

Senin, P., & Malinchik, S. (2013). Sax-vsm: Interpretable time series classification using sax and vector space model. In *Proceedings of the 2013 ieee 13th international conference on data mining.* IEEE. doi: 10.1109/ICDM.2013.52

Sorvala, A., Alasaarela, E., Sorvoja, H., & Myllyla, R. (2012). A two-threshold fall detection algorithm for reducing false alarms. In *Proceedings of 2012 6th international symposium on medical information and communication technology (ismict).*

Sucerquia, A., López, J. D., & Vargas-Bonilla, J. F. (2017). Sisfall: A fall and movement dataset. *Sensors*, *17*, 198. doi: http://doi.org/10.3390/s17010198

Tsinganos, P., & Skodras, A. (2017). A smartphone-based fall detection system for the elderly. In *Proceedings of the 10th international symposium on image and signal processing and analysis.*

Villar, M., & Villar, J. R. (2019). Peak detection enhancement in autonomous wearable fall detection. In *2019 the international conference on intelligent systems design and applications (isd)* (p. 3712-3715). doi: 10.1109/EMBC.2016.7591534

Willy, B., & Osterberg, C. M. (2014). Strategies for reducing falls in long-term care. *Annals of Long-Term Care: Clinical Care and Aging*, *22*, 23-32.

Wu, F., Zhao, H., Zhao, Y., & Zhong, H. (2015). Development of a wearable-sensor-based fall detection system. *International Journal of Telemedicine and Applications*, *2016*, *Article ID 576364*(1), 11 pages. Retrieved from `https://www.hindawi.com/journals/ijta/2015/576364/` doi: http://dx.doi.org/10.1155/2015/576364

Wu, P., & Dietterich, T. G. (2004). Improving svm accuracy by training on auxiliary data sources. In *In icml* (pp. 871–878).

Zhang, T., Wang, J., Xu, L., & Liu, P. (2006). Fall detection by wearable sensor and one-class svm algorithm. In I. G. Huang DS. Li K. (Ed.), *Intelligent computing in signal processing and pattern recognition* (Vol. 345, p. 858-863). Springer Berlin Heidelberg. Retrieved from `https://link.springer.com/chapter/10.1007%2F978-3-540-37258-5_104?LI=true#citeas` doi: https://doi.org/10.1007/978-3-540-37258-5-104

AUTHOR BIOGRAPHY

fanez.jpg

Mirko Fañez. Mirko Fáñez is a Higher Technician in Multi-platform Applications Development (2016). Currently working on multiple Software Development projects at Instituto Tecnológico de Castilla y León (ITCL) in Burgos, and also finishing his University Degree in IT Engineering at University of Oviedo.

villar.jpg

José Ramón Villar. Industrial Engineer in Electronics and Control (Universidad de Oviedo, 1992). Ph. D. in Computer Science (University of León, 2002). Currently, he is a Tenure with the Department of Computer Science, in the knowledge area of Computer Science and Artificial Intelligence, at University of Oviedo. His main research lines include i) Ambient Assisted Living, identification of onsets of human illnesses and ii) Soft-Computing, Hybrid Systems and Intelligent Analysis of Time Series. Main figures in the last 5 years i) JCR indexed: 17 (9 Q1), ii) in international scientific conferences: 22, iii) in international educational conferences: 3, iv) research projects with public funding: 3, v) I+D+I projects with private funding: 1, vi) innovation in education projects: 4.

delacal.jpg

Enrique de la Cal. Received the Computer Science M.Sc. (University of Oviedo, 1995). Ph.D. degree in Computer Science (University of Oviedo, 2003). Currently, he is senior lecturer with the Department of Computer Science in the knowledge area of Computer Science and Artificial Intelligence, at University of Oviedo. His research interests are in two main groups of topics: i) Ambient Intelligence and Human Activity Recognition, ii) Applied Artificial Intelligence. Main figures: 7 national publicly funded projects, 6 regional publicly funded projects, co-author of 15+ papers on impact factor journals and more than 30 contributions to international conferences.

sedano.png

Javier Sedano. Ph.D. Javier Sedano is an expert in the development of artificial intelligence, electronic systems (hardware) and industrial control systems, as well as in the design of connectionless models for the identification and modelling of dynamic systems in Big Data and Block Chain. He is the head of the Artificial Intelligence and Applied Electronics group at the Instituto Tecnológico de Castilla y León. He is also participating wit others research groups and has spent more than 25 years working on projects and publications related to artificial intelligence and system modelling. He collaborates in the organization of international scientific conferences, program committees and organizations, etc. He is a member of the IEEE Systems, Man & Cybernetics Chapter Society Spanish Chapter and has more than 100 international publications as well as records of software and industrial patents.

gonzalez.jpg

Víctor M. González. BSc, MSc, and PhD in Computer Science from the University of Oviedo, Spain, in 1993, 1997 and 2002 respectively. He is an Associate Professor with the Department of Electrical Engineering at the University of Oviedo, where he coordinates the Master on Automation Engineering and Industrial Informatics. He is the Deputy-dean for Internationalization of the Polytechnic School of Engineering of Gijón, and the coordinator of the agreement between CERN (the European Center For Nuclear Research) and the University of Oviedo for the Project Associate Program. His research interests include machine learning applied to health and well-aging. Main figures in the last 5 years i) JCR indexed 12 (8 Q1), ii) international scientific conferences: 9, iii) research projects with private funding: 11, iv) innovation in education projects: 2.

How to cite this article: Fañez M., J. R. Villar, E. de la Cal, J. Sedano, and V. M. González (2019), Transfer Learning and Information Retrieval applied to Fall Detection, *Expert Systems*, *20xx;00:1–6*.