A simple classification ensemble for ADL and Falls⋆

Enrique A. de la Cal[1], Mirko Fáñez[2], Mario Villar[3], Jose R. Villar[1], and Victor Suárez[4]

[1] University of Oviedo, Computer Science Department,
Faculty of Geology, Oviedo, Spain
{delacal,villarjose}@uniovi.es
[2] Instituto Tecnológico de Castilla y León, Pol. Ind. Villalonquejar,
09001, Burgos, Spain
mirko@mirkoo.es
[3] University of Granada, Granada, Spain mario.villarsanz@gmail.com
[4] University of Oviedo, Control and Automatica Department,
EPI, Gijón, Spain
vmsuarez@uniovi.es

Abstract. Fall Detection (FD) and ADL (Activity of Daily Living) identification is one the main challenges in a lot of real-world problems like work monitoring, healthcare systems, etc. Up to our knowledge, there are a lot of proposals in the literature for both problems separately, but few of them pose both problems at a time. A possible solution relies on on-wrist wearable devices including tri-axial accelerometers performing ADL and Fall identification autonomously. Since the dynamics of both kind of activities (FALL and ADL) are quite similar and not easy to identify, mainly in FALL and high ADLs like Running, Jogging, GoUpstairs, etc, a technique considering peaks is suitable. Thus, in this study, an ensemble between KMEANS and KNN (stands for EKMEANS) taking as input a 19 features dataset calculated from a time window whenever a peak is detected. As peak detection algorithm is used, the MAX-PEAKS algorithm presented in [15].

The proposal is evaluated using the UMA Fall, one of the publicly available simulated fall detection data sets, and compared to two classical well-known algorithms: the KNN and a Feed Forward Neural Network (NN) [15].

The results show that our proposal outperforms the NN results.

Future work includes a further analysis of the dynamics of the ensemble EKMEANS and a study of this problem using Deep-Learning.

Keywords: Human Activity Recognition, ADL identification, Fall Detection TS Clustering, TS Classification, Wearable Devices.

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

1 Introduction

Fall Detection (FD) and ADL (Activity of Daily Living) identification is one of the most important research niches in several real-world problems like healthcare, work safety, sport monitoring, etc. At present, there are a lot of proposals in the literature for both problems separately, but few of them pose both problems at a time.

Concerning FD, the solutions can be classified into three main types respect to the type of sensors used: Non-Wearable Based Systems (NWS), Wearable Based Systems (WS), and Fusion or hybrid-based Systems (FS) [17]. Current work focuses on smart-watches with built-in tri-axial accelerometers (3DACC), which is by far the most chosen option [2,5].

Regarding FD proposals, it can be stated that most of the references includes any machine learning technique applied to the focused problem. For example, [18,16] involve a feature extraction stage plus a SVM that classifies the TS windows. Likewise, [2,5,6] classify the sliding windows based on some transformations of the 3DACC magnitude and using some thresholds with very simple rules.

One of the baselines in FD is the Abbate algorithm [1], that has been extended and modified in a series of publications [7,8,14], to adapt the original location of the Abbate algorithm sensor (waist) to a sensor on the wrist. In one of our previous work [15], a new event detection mechanism to detect the high intensity fall events was presented (events that arise when the user stands up and falls either while walking, standing still, running, etc.). The idea is derived from a partial maximum peak detection method [11], where the threshold to detect the peaks is automatically determined for each user. Interestingly, this new event detection makes use of no user predefined threshold, which represents a step ahead in the event detection mechanisms in the literature. We refer to this event detection mechanism as *MAX-PEAKS*.

Since finding an appropriate value for the threshold that allows detecting all type of falls without getting confused with some ADL has proved to be a complicated problem [17], current work proposes a preliminary proposal to classify TS in a public dataset including typical ADLs as well as different kind of falls. It can be stated that most of references tackling ADL identification usually exclude FALL time series: sometimes the original dataset doesn't include FALL times series [10,9] and others the authors exclude the FALL samples [12].

Thus, this study proposes a Hybrid Artificial Intelligent System to classify datasets including ADLs and Fall TSs. The proposal includes an extension of the MAX-PEAKS algorithm presented in [15] with a more complete set of features, and the classification algorithm will be an ensemble of the well-known algorithms K-Means and KNN (stands for EKMEANS). In addition, the MAX-PEAKS features will be reduced using the PCA [13] analysis with different levels of variance.

The structure of the paper is as follows. The next section deals with the description of the proposal including the extension of the MAX-PEAKS algorithm, together with the transformations that are proposed to compute, as well as the description of the classification algorithm EKMEANS. Section 3 describes the UMA Fall dataset used, the experimental setup and shows and discusses the obtained results. Finally, conclusions are drawn.

2 The proposal

Figure 1 shows the complete procedure of the algorithm employed in this work. The general algorithm consists of four stages: first, a sliding window of $\frac{1}{4}FREQ$

Fig. 1. The general schema of the proposal

of sampling rate is considered to compute the MAX-PEAKS algorithm [15]. When a Peak is detected (FALL or NOT FALL), the transformations for two windows are calculated. After that, the feature extraction using PCA is computed and finally the classification model is carried out.

2.1 The MAX-PEAKS Peaks detection algorithm

The event detection stage For the purpose of detecting peaks in the 3DACC magnitude, the first stage is to smooth the signal using a sliding window sized $\frac{1}{4}FREQ$, with $FREQ$ being the sampling frequency. Afterwards, we apply the S_1 transformation proposed in [11]. For the current problem, the S_4 and S_5 were too complex for a smart-watch and need too wide windows of data in order to estimate the entropy. From the remaining transformations, we chose S_1 because its simplicity and similar performance among all of them. The Eq. 1 defines the calculation of S_1, where k is the predefined number of samples and t is the current sample timestamp. It is worth noticing that, although we analyze the window $[a_{t-2k-1}, a_t]$ at time t, the peak candidate is a_{t-k}, the center of the interval. The S_1 transformation represents a scaling of the TS, which makes the peak detection easier using a predefined threshold α.

$$S_1(t) = \frac{1}{2} \times \{max_{i=t-2k}^{t-k-1} a_i + max_{i=t-k+1}^{t} a_i\} \tag{1}$$

The algorithm for detecting peaks is straightforward: a peak occurs in time t if the value S_t is higher than α and is the highest in its $2k$ neighborhood. In the original report, all the parameters (k, α) were carefully determined for each problem in order to optimize the peak detection.

The new set of transformations Whenever a high intensity fall occurs there are three main parts: the activity being carried ordinarily before the fall event, the fall itself that we identify as a peak and what happens next. Because there are no public data set of real falls for healthy participants, we are not able to say accurately what happens after a fall: we can make the hypothesis that what happens after a fall is a period of relative calm, without special activity, perhaps some erratic movements of the hands. Therefore, we will divide the $[a_{t-2k-1}, a_t]$ window in three: before the peak $I_B = [a_{t-2k-1}, a_{t-k-1}]$, the peak $I_P = [a_{t-k-1}, a_{t-k+1}]$ and after the peak $I_A = [a_{t-k+1}, a_t]$. For each of these sub-intervals we propose to compute the following transformations:

AAMV Average Absolute Acceleration Magnitude Variation computed as $AAMV = \sum_{t=s}^{e-1} |a_{t+1} - a_t|/N$, with N the number of samples in the interval $[s, e]$.
E Energy of the Acceleration Magnitude $E = \sum_{t=s}^{e} a_t^2/N$
Mean Mean Activity the mean of the acceleration magnitude in the interval $[s, e]$.
SD Standard Deviation of the acceleration magnitude in the interval $[s, e]$.
AoM Amount of movement calculated as $abs(max(a_i) - min(a_i))$.
MAD Mean Absolute Difference calculated as $1/n * sum(|a_i - mean(a_i)|)$.

Therefore, we have a total of 19 transformations (6 transformations for each of the three intervals plus the new S_1 calculated for the peak interval I_P); none of which relies on thresholds of any kind. All of these transformations are well known in the context of Human Activity Recognition and Fall Detection.

Besides, and in order to analyse the importance of these features a principal components analysis has been carried out with two levels of significance of 90% and 95%.

2.2 EKMEANS: An ensemble of KMEANS and KNN

Algorithm 1 presents the complete procedure of the EKMEANS algorithm proposed in this work. This proposal consists in a user-centered ensemble of KMEANS and KNN. So, the train dataset (Stage TRAIN, Alg.1L2) for each participant p will be the fusion of data from the other participants different to p (Remaining_DATASET). Thus, since frequently in this kind of problems there are a big overlapping of the samples from different classes, our proposal consists on the execution of the well-known KMEANS[5] algorithm [4] on the TrainDataset to obtain the set of centroids K (KMEANS_RESULTS$_i/i = 1...K$). The optimal number of clusters (K) is estimated using the method "Within-cluster sum of square".

In the case of the clusters with samples belonging to more than one class (Alg.1L6), the KNN algorithm will be calculated for 1 to 15 neighbors to obtain the best number of neighbors, otherwise the predicted class for this cluster will be the one corresponding to the train samples of this cluster.

[5] The default implementation of the R platform "kmeans" function has been used

In the TEST stage (Alg.1L10), the predicted class for each test sample will be computed considering the number of classes for the cluster predicted for this sample (PK, Alg.1L12). In case, the PK cluster had more than one class for the train samples, the KNN obtained for the train samples will be used to predict the class of test SAMPLE (Alg.1L13-14), otherwise the predicted class will be the one belonging to all the train samples for this cluster.

Algorithm 1 EKMEANS(TRAINDATASET, TESTDATASET, NP: Number of Participants, K: Number of Centroids for KMEANS)

```
1: for p in 1:NP do
2:    TRAIN_STAGE ← 1
3:    Remaining_DATASET ← TRAINDATASET - {Samples of participant p}
4:    KMEANS-RESULTS ← KMEANS(Remaining_DATASET, K)
5:    for i in 1:K do
6:       if #Classes(KMEANS-RESULTS[i]) > 1 then
7:          KNN-FOR-CLUSTER[i] ← KNN(Remaining_DATASET, 1, 3, 5, 7, ..,15)
8:       end if
9:    end for
10:   TEST_STAGE ← 1
11:   for SAMPLE in DATATEST do
12:      PK ← PREDICT-KMEANS(KMEANS-RESULTS, SAMPLE)
13:      if #Classes(KMEANS-RESULTS[PK]) > 1 then
14:         PREDICTEDCLASS(SAMPLE) ← PREDICT(KNN-FOR-
            CLUSTER[PK], SAMPLE)
15:      else
16:         PREDICTEDCLASS(SAMPLE) ← Class(KMEANS-RESULTS[PK])
17:      end if
18:   end for
19: end for
```

3 Numerical results

3.1 Data set description

The publicly available simulated falls UMA Fall data set [3] is used in this study. This data set includes several activities, transitions and simulated falls regarding up to 17 participants. There is no fixed number of repetitions of each activity or simulated fall. Each participant used several 3DACC, specially one on a wrist; the sampling frequency was 20 Hz. Altogether, 208 TS are simulated falls, belonging to lateral, forward or backward falls, out of the 531 TS that are available in this data set.

3.2 Experimentation set up

The experimentation has considered just one event detection method: the MAX-PEAKS algorithm with 19 features (see section 2.1), and as classification algorithm two alternatives have been run, EKMEANS and KNN. Three datasets have been built from the MAX-PEAKS 19 features: the original MAX-PEAKS dataset (MAX-PEAKS) with the 19 features, and two PCA feature extraction with variance of 90% and 95% (MAX-PEAKS PCA90 and PCA95). Besides, two kind of labelling of the datasets have been considered: MULTICLASS (F1-F3 and A1-A5) and TWO-CLASS PROBLEM (FALL and NOT FALL) (see table 1). Besides, the results of a Feed Forward Neural Network for the MAX-PEAKS dataset using the TWO-CLASS labelling obtained in [15] will be considered. The sensitivity and specificity of the results for all the participants will be used to measure the performance of the method.

Table 1. Summary of ADL and FALL kind of activities

	FALL				
Code	F1	F2	F3		
Activity	FALL.BACKWARDS	FALL.FORWARD	FALL.LATERAL	-	-
	NOT FALL				
Code	A1	A2	A3	A4	A5
Activity	WALKING	HOPPING	CLIMBING UPSTAIRS	CLIMBING DOWNSTAIRS	BENDING

3.3 Numerical results

Regarding the three datasets, the two classification algorithms and the two labelling alternatives, we will obtain 12 tables with the numerical results for the 17 participants, but by lack of space, just the most representatives tables are included, as well as two summary tables.

Results for the TWO-CLASS labelling Typically, clustering solutions on FALL Detection presents the results as Two-class problems. Thus, we are starting this section with the results of applying the EKMEANS and KNN algorithms on the TWO-CLASS MAX-PEAKS 19 features dataset (see table 2). It can be observed at least the following issues: the Sensitivity for the EKMEANS and KNN results are quite similar (MAX-PEAKS, PCA90 and PCA95), but the Specificity for EKMEANS outperforms lightly the results for KNN for the two PCA datasets; other issue is that EKMEANS overpass clearly the Specificity of the NN results [15] with a quite similar Sensitivity.

Let's see the EKMEANS results for participant #1 (see Fig. 2). We know that three clusters (#1, #8 and #10) out of the eleven clusters obtained in the train stage contains train samples (the shaped/colored points) from the two classes (the points are not labelled by class). The remaining clusters only contains train samples belonging to one class. If the test samples are analysed (white circles with label F or NF), it can be observed that they are labelled by

Table 2. Summary of results for EKMEANS and KNN for the TWO-CLASS problem

	EKMEANS MAX-PEAKS		EKMEANS MAX-PEAKS PCA90		EKMEANS MAX-PEAKS PCA95		KNN MAX-PEAKS		KNN MAX-PEAKS PCA90		KNN MAX-PEAKS PCA95		NN[15] MAX-PEAKS	
Pid	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**
1	0.900	0.714	0.900	0.857	0.950	0.857	0.800	0.714	0.800	0.714	0.750	0.714	1.000	0.271
2	0.846	1.000	0.846	1.000	0.846	1.000	0.846	1.000	0.769	1.000	0.692	1.000	1.000	0.434
3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.889	1.000	0.944	1.000	1.000	0.357
4	0.941	1.000	0.941	1.000	0.941	1.000	0.941	1.000	0.824	1.000	0.706	1.000	1.000	0.469
5	0.500	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.625	1.000	0.500	1.000	1.000	0.439
6	0.625	1.000	0.625	1.000	0.750	1.000	0.750	1.000	0.500	1.000	0.750	1.000	1.000	0.231
7	NA	0.929	NA	NA	NA	0.786	NA	0.857	NA	0.857	NA	0.929	NA	0.439
8	NA	0.857	NA	0.857	NA	0.857	NA	0.857	NA	NA	NA	NA	NA	0.522
9	0.857	1.000	0.810	0.750	0.810	0.750	0.905	1.000	0.762	0.750	0.762	0.750	1.000	0.467
10	NA	0.929	NA	0.929	NA	0.857	NA	NA	NA	NA	NA	0.929	NA	0.351
11	0.500	1.000	0.500	0.900	1.000	0.900	1.000	0.900	0.000	0.900	0.000	1.000	1.000	0.360
12	1.000	0.692	1.000	0.615	1.000	0.692	1.000	0.692	1.000	0.923	1.000	0.923	1.000	0.417
13	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.917	1.000	1.000	0.597
14	0.875	1.000	0.750	1.000	0.875	1.000	0.750	1.000	0.875	1.000	1.000	1.000	1.000	0.125
15	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.909	1.000	0.909	1.000	1.000	0.473
16	0.947	0.826	0.947	0.870	0.930	0.913	0.947	0.870	0.930	0.957	0.930	1.000	1.000	0.389
17	0.765	1.000	0.706	1.000	0.706	1.000	0.706	1.000	0.647	1.000	0.647	1.000	1.000	0.276
μ/σ	**0.840/0.178**	**0.938/0.104**	**0.823/0.182**	**0.924/0.113**	**0.879/0.147**	**0.918/0.103**	**0.868/0.151**	**0.931/0.106**	**0.752/0.260**	**0.940/0.096**	**0.751/0.262**	**0.953/0.091**	**1.00/0.000**	**0.389/0.112**

clusters **#1(red)**, **#8(blue)**, #11(pink), #3(khaki green), **#10(purple)**, and the clusters that contains overlapped test samples (in boldface) are the same as the ones for the train samples. These test samples belonging to these three clusters are the ones affecting the Sensitivity and Specificity values.

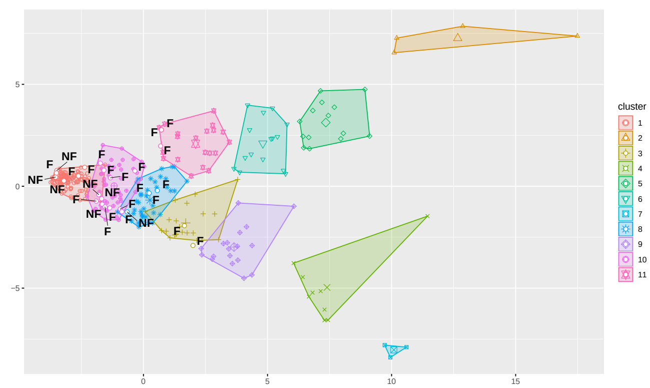

Fig. 2. The EKMEANS results for the MAX-PEAKS TWO-CLASS dataset. The train samples are plotted with different shapes and colors (11 clusters), and the test samples are white circle-shaped, labelled with F (FALL) and NF (Not FALL)

Results for the MULTICLASS labelling Table 3 includes the results of EKMEANS and KNN for all the MULTICLASS datasets considered in this section. Since the number of samples for each class are quite low, it can be stated that the Sensitivity figures are quite low for all the experiments, models and datasets. On the other side the Specificity is clearly higher than the Sensitivity in both models. In order to justify these results, let's depict the results of the EKMEANS in Fig. 3 for participant #1. In the figure, it can be seen that the test samples (white circles) belonging to class F1 are quite sparse and scattered among different clusters, that is the reason the Sensitivity was 0.0 (see table 4). Conversely, test samples from class F2 are more compact, and the results for this kind of fall confirm it (column F2 in Table 4). The remaining kinds of activities present low levels of Sensitivity for the same reason as F1. Besides in general terms, we can observe (Table 4: mean/std) that practically all the activities but F2, sport bad Sensitivity results and high Specificity.

4 Conclusions

This study proposes the EKMEANS algorithm for ADL and Falls classification, using a low computational consumption technique based on an ensemble between KMEANS and KNN.

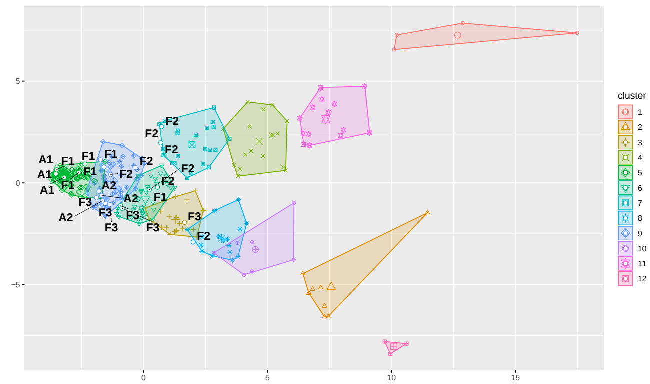

Fig. 3. The EKMEANS results for the MAX-PEAKS MULTICLASS dataset. The train samples are plotted with different shapes and colors (12 clusters), and the test samples are white circle-shaped, labelled with F# (FALL) and A# (Not FALL)

Our proposal has been compared to the classical KNN algorithm as well as the results obtained with a Feed Forward NN in [15] taking as input three variants of the public dataset UMAFALL [3]. These three datasets have been built from the UMAFALL wrist acceleration: one applying the peaks detection algorithm MAX-PEAKS [15] and the results of PCA with variance of 90% and 95%. Besides, these three datasets have been considered using two kind of labelling: on one side, considering all the different labels of activities defined in the UMAFALL dataset (MULTICLASS), on the other side the Not-FALL and FALL activities have been re-labelled just as a two-class problem.

The results show that our proposal EKMEANS outperforms clearly the Specificity of the NN results with a quite similar Sensitivity for the TWO-CLASS dataset. Besides, our proposal is a lighter computational consumption process than a NN in deployment stage. Concerning the MULTICLASS datasets, we have observed that mostly all the activities but F2 (Lateral Fall), sport bad Sensitivity results and high Specificity for all the models. Other important issue is that PCA doesn't affect positively or affect negatively in the performance of the different models used in current study.

Future work must consider a deeper analysis of the dynamics of the intra-cluster KNN for EKMEANS, as well other kind of low consumption classification meta-heuristic. Besides, in our point of view a multiclass oversampling technique must be carried out on the used datasets. Finally, the use of Deep Learning is also part of future work.

References

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

2. Bourke, A., O'Brien, J., Lyons, G.: Evaluation of a threshold-based triaxial accelerometer fall detection algorithm. Gait and Posture **26**, 194–199 (2007)
3. Casilari, E., Santoyo-Ramón, J.A., Cano-García, J.M.: Umafall: A multisensor dataset for the research on automatic fall detection. Procedia Computer Science **110**(Supplement C), 32 – 39 (2017). https://doi.org/doi:/10.1016/j.procs.2017.06.110
4. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) **28**(1), 100–108 (1979), http://www.jstor.org/stable/2346830
5. Huynh, Q.T., Nguyen, U.D., Irazabal, L.B., Ghassemian, N., Tran, B.Q.: Optimization of an acc. and gyro.-based fall det. algorithm. Journal of Sensors (2015)
6. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., Jämsaä, T.: Comparison of low-complexity fall detection algorithms for body attached accelerometers. Gait and Posture **28**, 285–291 (2008)
7. Khojasteh, S.B., Villar, J.R., de la Cal, E., González, V.M., Sedano, J., YAZĞAN, H.R.: Evaluation of a wrist-based wearable fall detection method. In: 13th International Conference on Soft Computing Models in Industrial and Environmental Applications. pp. 377–386 (2018)
8. Khojasteh, S.B., Villar, J.R., Chira, C., González, V.M., de la Cal, E.: Improving fall detection using an on-wrist wearable accelerometer. Sensors **accepted for publication**, 1–20 (2018)
9. Lin, W.Y., Verma, V.K., Lee, M.Y., Lai, C.S.: Activity monitoring with a wrist-worn, accelerometer-based device. Micromachines **9**(9), 450 (2018)
10. Lu, J., Tong, K.Y.: Robust single accelerometer-based activity recognition using modified recurrence plot. IEEE Sensors Journal **19**(15), 6317–6324 (2019)
11. Palshikar, G.K.: Simple algorithms for peak detection in time-series. Tech. rep., Tata Research Development and Design Centre (2009)
12. Sukor, A.A., Zakaria, A., Rahim, N.A.: Activity recognition using accelerometer sensor and machine learning classifiers. In: 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA). pp. 233–238. IEEE (2018)
13. Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S. Springer Verlad (2002)
14. Villar, J.R., de la Cal, E., Fañez, M., González, V.M., Sedano, J.: User-centered fall detection using supervised, on-line learning and transfer learning. Progress in Artificial Intelligence **2019**, 1–22 (2019). https://doi.org/doi:/10.1007/s13748-019-00190-2
15. Villar, M., Villar, J.: Peak detection enhancement in autonomous wearable fall detection? In: Proceedings of the 19th International Conference on Intelligent Systems Design and Applications (2019)
16. Wu, F., Zhao, H., Zhao, Y., Zhong, H.: Development of a wearable-sensor-based fall detection system. International Journal of Telemedicine and Applications **2015**, 11 (2015). https://doi.org/doi:/10.1155/2015/576364
17. Yacchirema, D., de Puga, J.S., Palau, C., Esteve, M.: Fall detection system for elderly people using iot and big data. Procedia Computer Science **130**, 603 – 610 (2018), the 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops
18. Zhang, T., Wang, J., Xu, L., Liu, P.: Fall detection by wearable sensor and one-class svm algorithm. In: Huang DS., Li K., I.G. (ed.) Intelligent Computing in Signal Processing and Pattern Recognition, Lecture Notes in Control and Information Systems, vol. 345, pp. 858–863. Springer Berlin Heidelberg (2006)

Table 3. Summary of results for EKMEANS and KNN for the MULTICLASS problem

Pid	EKMEANS MAX-PEAKS mean/std		EKMEANS MAX-PEAKS PCA90 mean/std		EKMEANS MAX-PEAKS PCA95 mean/std		KNN MAX-PEAKS mean/std		KNN MAX-PEAKS PCA90 mean/std		KNN MAX-PEAKS PCA95 mean/std	
	Sens	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**
1	0.350/0.266	0.868/0.102	0.392/0.429	0.889/0.125	0.267/0.303	0.870/0.141	0.225/0.263	0.872/0.120	0.408/0.490	0.901/0.140	0.408/0.490	0.883/0.154
2	0.333/0.471	0.870/0.082	0.488/0.342	0.887/0.073	0.571/0.332	0.880/0.064	0.369/0.442	0.833/0.157	0.369/0.442	0.868/0.095	0.286/0.481	0.860/0.075
3	0.567/0.401	0.917/0.068	0.500/0.425	0.896/0.089	0.500/0.373	0.896/0.102	0.533/0.321	0.894/0.106	0.700/0.361	0.920/0.070	0.500/0.408	0.887/0.093
4	0.493/0.384	0.912/0.099	0.627/0.394	0.947/0.071	0.553/0.398	0.928/0.084	0.593/0.414	0.922/0.115	0.593/0.414	0.949/0.033	0.527/0.392	0.937/0.035
5	0.500/0.707	0.875/0.087	0.500/0.707	0.875/0.102	0.562/0.619	0.867/0.112	0.500/0.707	0.875/0.102	0.500/0.707	0.858/0.124	0.562/0.619	0.858/0.143
6	0.700/0.298	0.905/0.147	0.433/0.435	0.885/0.173	0.367/0.415	0.862/0.163	0.433/0.435	0.899/0.115	0.400/0.435	0.884/0.112	0.267/0.435	0.797/0.196
7	0.417/0.373	0.894/0.143	0.383/0.439	0.878/0.125	0.317/0.207	0.864/0.154	0.417/0.373	0.878/0.109	0.433/0.253	0.888/0.155	0.300/0.274	0.824/0.143
8	0.375/0.479	0.915/0.070	0.375/0.479	0.925/0.070	0.250/0.500	0.901/0.077	0.333/0.471	0.913/0.075	0.333/0.471	0.893/0.067	0.417/0.500	0.910/0.083
9	0.424/0.307	0.892/0.075	0.387/0.299	0.899/0.122	0.354/0.267	0.892/0.128	0.320/0.385	0.917/0.062	0.295/0.221	0.880/0.097	0.499/0.410	0.909/0.079
10	0.217/0.331	0.843/0.189	0.267/0.435	0.881/0.173	0.200/0.447	0.889/0.127	0.200/0.447	0.843/0.225	0.400/0.548	0.816/0.304	0.400/0.548	0.851/0.237
11	0.458/0.315	0.906/0.081	0.542/0.417	0.883/0.162	0.542/0.417	0.883/0.162	0.667/0.471	0.894/0.193	0.542/0.417	0.836/0.194	0.542/0.417	0.836/0.265
12	0.452/0.369	0.910/0.090	0.405/0.407	0.902/0.098	0.357/0.390	0.895/0.110	0.333/0.385	0.887/0.111	0.357/0.339	0.895/0.118	0.357/0.390	0.895/0.132
13	0.562/0.375	0.806/0.140	0.312/0.375	0.829/0.132	0.312/0.315	0.835/0.111	0.250/0.204	0.807/0.218	0.250/0.289	0.807/0.151	0.250/0.354	0.819/0.100
14	0.333/0.471	0.723/0.119	0.417/0.500	0.754/0.175	0.396/0.427	0.765/0.121	0.333/0.471	0.704/0.247	0.250/0.500	0.749/0.203	0.333/0.471	0.715/0.216
15	0.438/0.427	0.848/0.156	0.438/0.515	0.802/0.217	0.583/0.354	0.852/0.174	0.583/0.354	0.850/0.173	0.542/0.363	0.830/0.101	0.604/0.315	0.889/0.051
16	0.329/0.333	0.891/0.122	0.384/0.333	0.902/0.142	0.413/0.322	0.905/0.115	0.424/0.323	0.907/0.108	0.364/0.279	0.905/0.093	0.371/0.314	0.899/0.100
17	0.367/0.383	0.899/0.090	0.315/0.311	0.901/0.094	0.361/0.371	0.902/0.080	0.463/0.347	0.922/0.079	0.337/0.386	0.919/0.094	0.417/0.376	0.911/0.141
μ/σ	**0.430/0.114**	**0.875/0.049**	**0.421/0.090**	**0.879/0.045**	**0.406/0.123**	**0.875/0.036**	**0.410/0.134**	**0.872/0.054**	**0.416/0.123**	**0.870/0.050**	**0.414/0.109**	**0.863/0.054**

Table 4. EKMEANS - MAX-PEAKS, Sensitivity and Specificity for the MULTI-CLASS dataset. Pid: stand for Participant ID. For the sake of space this table just includes the results for participant #1.

EKMEANS - MAX-PEAKS																		
PId	**F1**		**F2**		**F3**		**A1**		**A2**		**A3**		**A4**		**A5**		**mean/std**	
	Sens	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**	**Sens**	**Spec**
1	0.000	0.714	0.750	0.789	0.333	0.905	0.333	0.875	0.333	1.000	NA	0.926	NA	NA	NA	NA	**0.350/0.266**	**0.868/0.102**
mean	**0.337**	**0.819**	**0.523**	**0.862**	**0.238**	**0.844**	**0.868**	**0.818**	**0.217**	**0.946**	**0.111**	**0.934**	**0.306**	**0.952**	**0.267**	**0.978**	**0.430**	**0.875**
std	**0.330**	**0.120**	**0.287**	**0.107**	**0.205**	**0.064**	**0.211**	**0.142**	**0.217**	**0.050**	**0.172**	**0.079**	**0.195**	**0.060**	**0.370**	**0.035**	**0.114**	**0.049**