



Article

A Semi-Automatic Annotation Approach for Human Activity Recognition

Patrícia Bota ^{1,*}, Joana Silva ¹, Duarte Folgado ¹ and Hugo Gamboa ^{1,2}

¹ Associação Fraunhofer Portugal Research, Rua Alfredo Allen 455/461, 4200-135 Porto, Portugal; joana.silva@fraunhofer.pt (J.S.); duarte.folgado@fraunhofer.pt (D.F.); hugo.gamboa@fraunhofer.pt (H.G.)

² Laboratório de Instrumentação, Engenharia Biomédica e Física da Radiação (LIBPhys-UNL), Departamento de Física, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal

* Correspondence: p.bota@campus.fct.unl.pt

Received: 30 November 2018; Accepted: 22 January 2019; Published: 25 January 2019

Abstract: Modern smartphones and wearables often contain multiple embedded sensors which generate significant amounts of data. This information can be used for body monitoring-based areas such as healthcare, indoor location, user-adaptive recommendations and transportation. The development of Human Activity Recognition (HAR) algorithms involves the collection of a large amount of labelled data which should be annotated by an expert. However, the data annotation process on large datasets is expensive, time consuming and difficult to obtain. The development of a HAR approach which requires low annotation effort and still maintains adequate performance is a relevant challenge. We introduce a Semi-Supervised Active Learning (SSAL) based on Self-Training (ST) approach for Human Activity Recognition to partially automate the annotation process, reducing the annotation effort and the required volume of annotated data to obtain a high performance classifier. Our approach uses a criterion to select the most relevant samples for annotation by the expert and propagate their label to the most confident samples. We present a comprehensive study comparing supervised and unsupervised methods with our approach on two datasets composed of daily living activities. The results showed that it is possible to reduce the required annotated data by more than 89% while still maintaining an accurate model performance.

Keywords: human activity recognition; machine learning; active learning; semi-supervised learning; time series; self-training

1. Introduction

Over the last years, the technological advances on ubiquitous sensing mechanisms allowed the proliferation of available data, which often is unlabelled. Modern machine learning approaches require large amounts of labelled data to achieve adequate performance. This duality raises a relevant question: How can we simultaneously optimise the process of data annotation and still learn an accurate machine learning model?

Particularly, the Human Activity Recognition (HAR) field has been a source of a large quantity of available data, mostly due to its myriad of applications on real-life scenarios such as healthcare, indoor location, user-adaptive recommendations and transportation [1,2]. According to World Health Organization [3] insufficient physical activity has been identified as the fourth leading risk factor for global mortality, being one of the main causes of several health diseases and correlated with overweight and obesity. HAR research has been trying to mitigate this challenge by monitoring human movement and issuing personalised recommendations in several populations, including the elderly and patients with chronic diseases. On the other hand, the practice of physical exercise is correlated with an increase of cardio-respiratory and muscular fitness, functional health, cognitive functions and improvement

of bones and joint health. Additionally, the monitoring of the human movement can be used as a preventive and diagnosis tool for triggering and warning unusual activity such as falls, movement degeneration or cardiac abnormalities. Therefore, HAR has been the subject for numerous research studies over these contexts [2,4,5].

Most of the latest research work has focused on using machine learning pipelines to accomplish HAR. On a typical HAR framework, as seen in Figure 1 [6], two inputs are often necessary: the raw data and its respective annotation metadata in the form of labels. These labels are often provided by an expert either annotating the data during the acquisition stage or during the course of posterior data evaluation. Human motion information is mapped into raw signals using embed wearable sensors which are often located on different anatomical positions. The data preparation stage consists of reducing the noise arising from the acquisition stage and enhancing the signal characteristics using a set of pre-processing methods. The signal is divided into windows from which features are extracted. Those features, along with the provided labels, are used as the input for the machine learning classifier. Lastly, the classifier predictions are evaluated with the objective of delivering a model capable of issuing correct decisions regarding the daily activities the subject is performing.

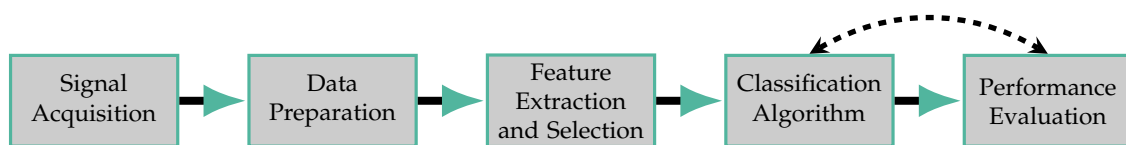


Figure 1. Schematic representation of an Human Activity Recognition (HAR) system architecture.

To build a representative machine learning model, it is often required the collection of a broad amount of labelled data, which constitutes the ground truth for Supervised Learning (SL) methods. This process aims to train a model capable of correctly generalising into new unlabelled data. The data becomes labelled during a process denoted as annotation, where each sample is mapped to its class. Most of the times, the annotation of data labels must be performed manually by the researcher, becoming a time-consuming, error-prone and expensive task [7]. For instance, let us take as an example the Cityscape dataset [8], which contains stereo video sequences with a total of 5000 high-quality annotated frames. Considering that annotating a single image can take around 1.5 h, to annotate the entire dataset in order to use it as input to a machine learning classifier, it would be required approximately 7500 h. Thus, becoming a fastidious, lengthy task whose quality will highly influence the classification output and whose time could be spent on building the classifier. Therefore, the annotation process might limit real-life application on very large datasets or complex models, where a high volume of data will increase the algorithm's performance. Under those circumstances, there is a need for the development of a method able to partly automate the data annotation process and reduce considerably its expensive cost.

In datasets with significant size, not all samples are equally informative to the classification process and an arbitrary unlabelled example may even be redundant. Active Learning (AL) provides methods to automatically identify the most relevant samples, which are posteriorly queued for expert annotation, that we denote as Oracle, without compromising the model performance. Figure 2a illustrates the behaviour of AL where samples near the decision boundary are selected for annotation since the AL system considers these as the most informative. Additionally, Figure 2b displays the Semi-Supervised Active Learning (SSAL) automatic annotation behaviour where after the most informative sample selected by AL (in yellow) is annotated by the oracle, its nearest samples are automatically annotated (as shown by the × in black).

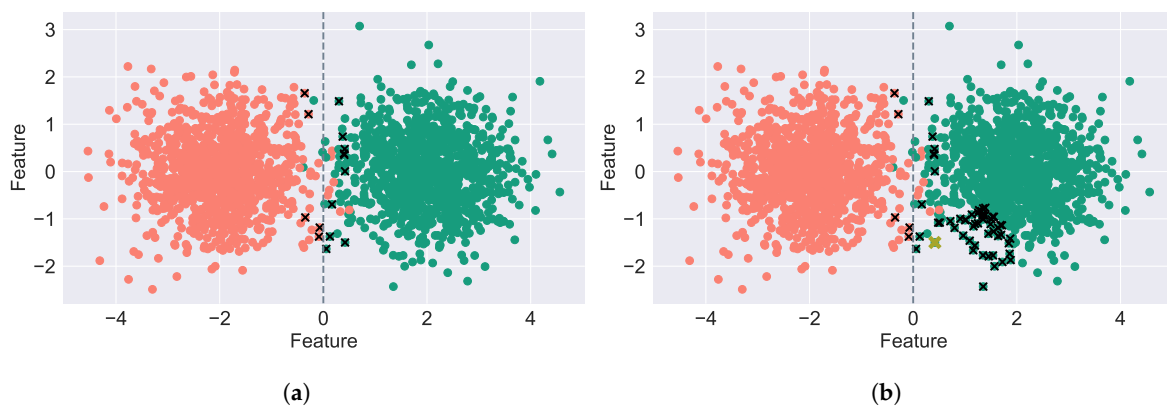


Figure 2. A dataset of 3000 samples illustrating the working principles of Active Learning (AL) and Semi-Supervised Active Learning (SSAL). The samples are illustrated with colours identifying their respective class. The samples selected by the AL for expert annotation are depicted by the \times 's. The grey vertical line denotes the decision boundary between the two classes. (a) Active Learning. (b) Semi-Supervised Active Learning.

In this work, we apply a SSAL algorithm for HAR, where we establish criteria to select the most relevant samples for annotation and propagate their label to similar samples and compare it to AL. We present two major contributions: (1) application of SSAL for HAR testing several automatic annotation methods. In literature, studies can be found applying individually both AL and Semi-Supervised Learning (SSL) to HAR. The present work extends these works, combining SSL and AL in the SSAL method and applying it in the context of HAR; (2) in order to accomplish this task with an optimal SSAL system, its detailed steps were evaluated in a comprehensive study of state-of-the-art AL (Query Strategies) Qs, (Stopping Criteria) SCs, and distance functions in the label propagation step. The SSAL method based on Self-Training (ST) applied to HAR data, starting with near zero annotated data, achieved high results on the algorithm's performance while reducing considerably the annotation effort and automatically annotating a substantial amount of the data.

The remaining of this paper is organised as follows: In Section 2 we present a brief literature review on HAR and AL. Section 3 describes the overall pipeline of the proposed methods. In Section 4, we evaluate our methodology against two datasets comprising real-world HAR data. Lastly, in Section 5, the main achievements of this work are presented along with future work directions.

2. Related Work

The discrimination of human activities is often covered either by external or wearable sensors. The former include intelligent homes, where sensors are placed in critical devices and cameras. However, these raise numerous issues, such as privacy, pervasiveness and computational complexity concerns [2,5]. This motivates the use of wearable sensors such as smartphones, since their small size, low cost and non-obtrusiveness allow to integrate them easily into the users' daily living activities, offering better management of privacy and pervasiveness by giving more control to the user.

On this account, many on-body sensor-based machine learning classification techniques have surged applied in the context of HAR with increasing improved results [2,4,5,9], namely, SL [9–13], Unsupervised Learning (UL) [9,14,15] and more recently, deep learning techniques [16–18]. SL is the most common approach for HAR, usually providing the most accurate results. However, SL techniques require high amounts of labelled data, limiting its application on very large dataset scenarios. Over the years a limited number of solutions have been proposed to reduce the amount of the necessary labelled data, namely SSL techniques [7,19–21], data automatic annotation [22] or annotation apps [23–25]. In the present work, in order to obtain accurate labelled data, an AL system identifies the most important samples to be labelled, therefore, decreasing the amount of necessary labelled data. An AL system is composed of two main parts: the Query System, that selects the most relevant samples from

a large unlabelled dataset according to a pre-defined criteria, and the Oracle, an expert annotator to label the selected samples.

In the literature, several techniques for the query system have been proposed. Shahmohammadi et al. [26] applied a Query by Committee and an uncertainty stream-based sampling strategy to a smartwatch-based approach dedicated to HAR. The AL methodology was able to achieve an accuracy of 92%, with a reduction of 46% in the amount of annotated data in comparison to SL. This study allowed to verify that through AL it is possible to create an improved classifier with a reduced number of labelled samples.

To improve the AL accuracy, one of its core points is the applied QS, which establishes a criteria to select the most relevant samples for annotation.

Alembdar et al. [27] presented three methods to measure the classifier's prediction confidence (i.e., uncertainty) in a sample's label namely: Least Confident, Margin and Entropy-based Sampling. The proposed Qs outperformed Random Sampling in the reduction of the amount of labelled data, with values from 80% to 66% data reduction.

However, experimental results show that, in some cases, uncertainty-based Qs may tend to select outliers rather than boundary samples [28]. This undesirable behaviour leads to the introduction of bias to the classifier. To overcome this issue, the authors of [28–30], used a sampling strategy combining the samples' uncertainty and the local data density, resulting in the selection of a informative sample inserted in a region of high local density. Since outliers are usually located in low density regions, this procedure minimises the selection of outliers to the data annotation.

In [30] the authors developed a SSAL framework in the context of multivariate time series, using k-Nearest Neighbour (NN) and a k-reverse Nearest Neighbour (rNN) technique to automatically label close neighbours of the newly annotated sample. In the end, for the same amount of initially labelled data, rNN method outperformed the NN method, obtaining higher accuracy, F1-score and percentage of automatically annotated samples.

Lastly, Maja Stikic et al. [31] explored SSL techniques in the context of HAR: Co-Training (two classifiers work on independent data and the most confident predictions of each classifier is used to teach the other), Self-Training (the classifier iteratively increases its training set with its most confidently predicted samples [32]) and AL. Using accelerometer data, Co-Training and ST attained very competitive results, being surpassed by AL using two Qs uncertainty-based functions. Both techniques allowed to significantly reduce the amount of necessary training labelled data and AL was able to outperform the SL technique when trained on the same amount of randomly sampled annotated data.

The literature review allowed verifying the promising results of AL and SSL for HAR in reducing considerably the data annotation effort. The present work extends the state-of-the-art of HAR, combining SSL and AL in a SSAL method applied in the context of HAR. To create an optimal SSAL system, several automatic annotation methods were tested using different distance functions and a comprehensive study regarding state-of-the-art AL Qs and SCs was performed.

3. Methods

3.1. General Active Learning Strategy

In AL, a QS function selects from a large unlabelled dataset (also referred as pool set) the samples which are more informative to be labelled by the Oracle and added to the classifier's labelled training set. Algorithm 1 [33] describes the methodology of an AL process. Following the learner's initialization on the initial training set (L), a QS selects the most informative sample (x^*) from the unlabelled data (U) for the oracle to annotate. This process is then repeated iteratively until a stopping criterion is met. Initially $L \ll U$, however in every iteration, the newly annotated sample x^* is removed from U and added to L , incrementing the labelled train set and consequently, reducing U . Hence, in every iteration the learner's training set expands with informative data and its performance improves [27,34].

The samples considered more informative are usually the samples with the highest gain for the classification process, so that, with a lower amount of labelled data and, therefore, lower data volume and manual annotation effort from the user, it is possible to reach a classification performance similar to a full labelled dataset.

Algorithm 1 General Active Learning

Input: initial train set L , unlabelled validation set U , independent test set T

Output: predicted labels for the test set

```

1:  $\theta \leftarrow clf.fit(L)$  ▷ Learns model on initial training set
2: while SC not met do
3:   selection by QS of the most informative sample:  $x^*$ 
4:   ask Oracle for  $x^*$  label
5:    $L \leftarrow L \cup x^*$  ▷ Increments the model's training set with  $x^*$ 
6:    $U \leftarrow U \setminus x^*$  ▷ Removes  $x^*$  from unlabelled samples  $U$ 
7:    $\Theta \leftarrow clf.fit(L)$  ▷ Updates model
8:   return  $clf.predict(T)$  ▷ Returns predicted labels for the test set
9: end while

```

To obtain a good accuracy in an AL system, there are three main considerations that will be addressed in the forthcoming sub-sections: the initial train set, the QS and the SC.

3.1.1. Initial Train Set

To develop a framework requiring the minimum annotation effort from the user, the initial train set was created with only one sample per class, which was randomly selected and posteriorly removed from the validation set.

3.1.2. Sample Selection Strategy

The second core element of an AL system is the QS, which must be able to select from the unlabelled dataset the sample considered as the most informative. We considered as an informative sample the one that will cause an improvement of the classification performance. Thus, through AL it is possible to optimise the trade-off between the classifier's performance and the number of labelled samples in its training set. The ability of the AL process to create a representative labelled training set, reaching a highly accurate classification with less labelled data is denoted as Selective Sampling. In contrast, in Passive Learning (PL), samples are chosen randomly from the entire dataset, resulting in a classifier requiring extra annotation effort that does not properly generalise due to its poor and non-representative training data.

Considering a probabilistic model, the classifier prediction output is a $U \times n$ matrix, where U represents the total number of the unlabelled validation set samples ($X_U = \{x_1, x_2, \dots, x_U\}$, $x_i \in \mathbb{R}^m$, $i = \{1, \dots, U\}$), and n the total number of classes existent in the validation set. Each row is a $1 \times n$ vector with the sample's predicted class probabilities with each cell value given by the prediction posterior probability - $P_\theta(y_k|x_i)$, $k \in \{0, \dots, n\}$ under the model θ .

A common metric to evaluate the sample's usefulness for the classification is to access the classifier's prediction confidence in that sample's label [21,27,31], which is given by the classifier's uncertainty in the sample's label prediction. In the present work three metrics were studied to evaluate the classifier's uncertainty, corresponding to three different uncertainty-based selective sampling functions: Least Confident Sampling, Margin Sampling and Entropy Sampling.

- **Least Confident Sampling** [33,34]: Selects the sample whose label the classifier is least certain about, according to the following equation [33–35].

$$\begin{aligned}x_{LC}^* &= \arg \max_x (1 - P_\theta(\hat{y}|x)) \\ \hat{y} &= \arg \max_y (P_\theta(y|x))\end{aligned}\quad (1)$$

where \hat{y} is the class label which the predictor considers most probable for the sample x .

- **Margin Sampling** [33,34]: Selects the sample with the minimum difference (margin) between the prediction probabilities of the first and second most likely classes, according to the following equation.

$$x_M^* = \arg \min_x (P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)) \quad (2)$$

where \hat{y}_1 and \hat{y}_2 represent the first and second class labels which the classifier considers as most probable for the sample x . Thus, the Margin Sampling QS allows to incorporate into the uncertainty calculation, the probability distribution of one more class label in comparison to Least Confident sampling.

- **Entropy Sampling** [33,34]: Selects the sample with the greatest entropy value, according to the following equation.

$$x_E^* = \arg \max_x \left(- \sum_k^n P_\theta(y_k|x_i) \log P_\theta(y_k|x_i) \right) \quad (3)$$

where \hat{y}_i represents the prediction probability of the sample x_i belonging to the class y_k . This method has the advantage of considering the prediction probability for all the class labels, in contrast to the previously mentioned QSs [33–35].

Additionally, in order to create a homogeneous initial training set, a weight $(1 - p_l)$ was introduced to the previously mentioned QSs while the training set was less than 1% of the validation set, according to the following equation [36].

$$x^* = (1 - p_l) * f \quad (4)$$

where $f = \{\text{Least Confident Sampling, Entropy Sampling, Margin Sampling}\}$ and p_l constitutes the percentage of each label in the training set.

According to the literature [28], a sample with high uncertainty will most likely be an outlier. Thus, to overcome this issue, we tested the Local Density Sampling and the Uncertainty and Local Density Sampling QSs.

- **Local Density Sampling**: Selects the sample with higher representation on the feature space, i.e., located in a high-density region, which is measured by the amount of NNs surrounding the sample, according to the following equation.

$$x_{LD}^* = \arg \max_x \left(\sum_i^U \left(\sum_j^k \frac{1}{1 + \text{dist}(NN(x_i, x_j))} \right) \right) \quad (5)$$

where x_i and x_j are two samples belonging to the unlabelled samples' dataset and dist the distance between each sample and its k -NNs. The k parameter was empirically set to 5.

- **Uncertainty and Local Density Sampling** [28,30,36]: Obtained through the linear combination between the previously mentioned QSs according to the following equation.

$$x_{UD}^* = \arg \max_x (\alpha f_0 + (1 - \alpha) f_1) \quad (6)$$

where f_0 is a density weight = {Local Density Sampling}, and $f_1 = \{\text{Least Confident Sampling, Margin Sampling, Entropy Sampling}\}$. Setting α to 1, would equal the Uncertainty and Local

Density QS to the Local Density QS, while $\alpha = 0$ to the uncertainty-based QS. The α parameter was set to 0.5 so the QS would choose the most informative sample taking into consideration equally both its local density and its prediction uncertainty.

3.1.3. Stopping Criterion

The last core point to be defined on an AL process is its SC. As it can be seen in Figure 3a, there is an instant during the AL cycle in which the classification's performance stabilises and, therefore, annotating additional samples will not improve the model's performance. Hence, the AL process should be ended at this instant, optimising the trade-off between the classifier's performance and the oracle annotation effort. Therefore, we should guarantee that the AL system must not stop too early, at the cost of resulting in a limited labelled set and under-performing classification, as well that the system does not stop too late either, at the cost of exceeding annotation work. Ideally, we would like to stop when the accuracy of the learner stabilises around its maximum value [37]. However, in a real-life application, we expect to work with unlabelled data, so its ground truth is not available and the accuracy of the classification cannot be obtained.

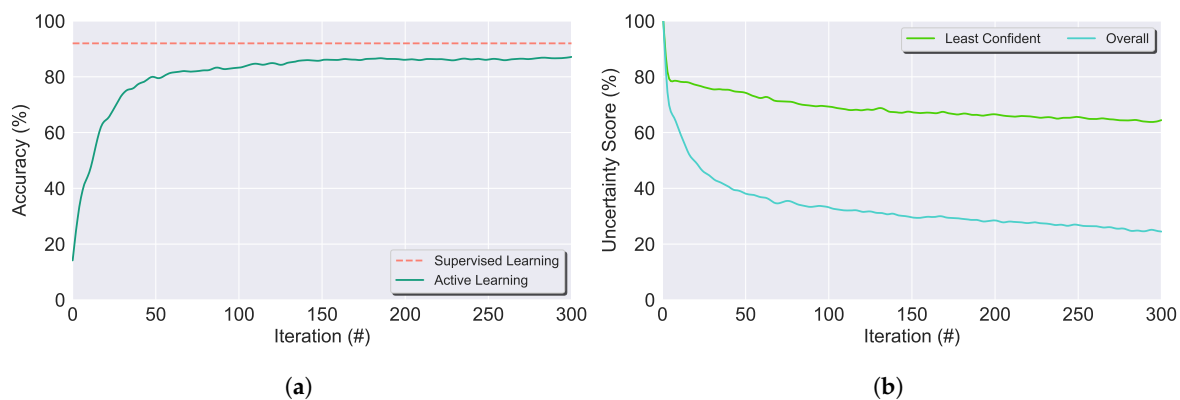


Figure 3. In (a) it is shown the AL performance of the initial 300 iterations. The horizontal red line denotes the accuracy average score of Supervised Learning (SL). In (b) it is shown the classifier Least Confidence score and Classifier Overall Uncertainty score throughout 300 iterations. (a) Least Confidence Uncertainty Score. (b) Overall Uncertainty Score.

On this account, with the goal of obtaining a SC applicable to all the methods, QSs and datasets, the following SCs were evaluated:

- **Max-Confidence SC (Max-Conf)** [38]: As previously described, in the Least Confident Sampling it is selected for the oracle to annotate the sample with the highest uncertainty, i.e., the sample that the classifier is least confident in its classification. Moreover, if the selected sample has a low uncertainty score, it is possible to presume that the classifier is able to confidently classify that sample, as well as the remaining samples. Hence, the AL process can be stopped.
- **Overall Uncertainty SC (Over-Unc)** [38]: Similar to Max-Confidence SC, but instead of stopping the AL system if the least confident score is low, it is used the average of the least confident score computed on the remaining unlabelled samples. That is, if this value, denominated overall uncertainty score, shows insignificant low values, we can assume that the classifier has sufficient confidence in the classification of the remaining unlabelled samples and, therefore, the AL cycle can stop.

Figure 3 shows that the stabilisation of the AL performance overlaps the stabilisation of both the least confident score and the overall uncertainty score. Hence, it was developed a condition to automatically detect whether the scores stabilised based on the mean and standard deviation over a given number of consecutive iterations S . The AL process stops when both the two following conditions are verified: $|\mu_k - \mu_{k-s}| \leq \Delta \mu_{SC}$ and $|\sigma_k - \sigma_{k-s}| \leq \Delta \sigma_{SC}$; $k \in \{0, 2S, \dots, N\}$, $S = 5$

and N = number of iterations. The $\Delta \mu_{SC}$ threshold was obtained through a calibration based on the stabilisation of the classifier accuracy score using the ground truth data.

- **Classification-Change SC (CC)** [37,38]: As discussed in Section 2, uncertainty-based Qs aim to select the most informative samples for the classification, which should correspond to the ones located near decision boundaries. Thus, dictating the class to which each sample is allocated to, therefore, significantly changing the classifier's performance and its prediction output. Hence, in the CC SC the AL is stopped once decision boundaries samples have been annotated and added to the classifier's training set. Under these assumptions, alterations in the classifier's prediction of the unlabelled data labels can be used to infer if the decision boundaries have been changed. Thus, if in two consecutive iterations the classifier's labels prediction has been constant, then, we can assume that the newly annotated samples are not near a decision boundary but rather inside it, hence, the AL process can be put to an end.
- **Combination Strategy SC**: Consists in a multi-criteria-based strategy that combines the prior SCs, namely Overall Uncertainty SC and Classification-Change SC (Over-CC) SC and Max-Confidence Uncertainty SC and Classification-Change SC (Max-CC) SC. The AL is stopped only if both SCs are verified. This method is justified in the cases where the uncertainty score quickly drops to insignificant low values, however, there are inconsistencies in the classifier's prediction. Thus, the annotation of new samples may result in changes on the decision boundaries and, therefore, on an improvement of the classifier's performance.

3.2. Semi-Supervised Active Learning Framework

As stated in Section 1, there is a need for an annotation technique able to partly automate the annotation process and reduce considerably the annotation cost of constructing a representative labelled dataset in the context of HAR. Thus, with the goal of significantly increasing the amount of available labelled data, we tested the SSAL framework, whose algorithm pipeline is presented in Algorithm 2 [30,39]. The SSAL model is similar to the standard AL framework, however, this method also provides the ability to automatically propagate the annotated label without requiring further inputs from the Oracle.

Algorithm 2 Semi-Supervised Active Learning

Input: initial train set L , unlabelled validation set U , independent test set T

Output: predicted labels for the test set

```

1:  $\Theta \leftarrow clf.fit(L)$  ▷ Learns model on initial training set
2: while SC not met do
3:   selection by  $Q$ , of most informative sample:  $x^*$ 
4:   ask Oracle for  $x^*$ 's label
5:    $L \leftarrow L \cup \{x^*\}$  ▷ Augments the model's training set with  $x^*$ 
6:    $U \leftarrow U \setminus \{x^*\}$  ▷ Removes  $x^*$  from unlabelled samples
7:    $\Theta \leftarrow clf.fit(L)$  ▷ Updates model
8:   automatically label confident samples  $C$  in  $U$ 
9:    $L \leftarrow L \cup \{C\}$  ▷ Augments the model's training set with  $C$ 
10:   $U \leftarrow U \setminus \{C\}$  ▷ Removes  $C$  from unlabelled samples
11:   $\Theta \leftarrow clf.fit(L)$  ▷ Updates model
12:  return  $clf.predict(T)$  ▷ Returns predicted labels for the test set
13: end while

```

Three SSAL techniques are used:

- **Self-Training Semi-Supervised Active Learning (ST-SSAL)** [31,32,40]: A classifier is trained on the available labelled data and posteriorly tested on the unlabelled data. Validation set samples having the highest prediction confidence score are added to the classifier's training set and removed from the unlabelled dataset. This process is repeated iteratively as the classifier is re-trained on an increasingly larger and larger training set. Therefore, under the assumption that

highly confident predicted labels are correct, the learner uses its own predictions to iteratively teach himself, consequently improving its performance. Hence, a sample will get annotated with \hat{y} if $P_{\theta}(\hat{y}|x) \geq \delta_{ST}$. The δ_{ST} threshold will influence the amount of propagation and its accuracy. A larger δ_{ST} will increase the automatic annotation but decrease its accuracy, since the model is less certain in the annotated sample's label. On the other hand, a smaller δ_{ST} will decrease the amount of annotation but increase its accuracy, since the few annotations are performed with high certainty. In the present work, δ_{ST} was empirically set to 0.98 in order to obtain a significant automatic annotation while maintaining a good certainty in the annotation.

- k-Nearest Neighbour Semi-Supervised Active Learning (k-NN-SSAL) [30]:** The sample selected by the QS (x^*) propagates its label to its k-NNs. The definition of k (the number of NNs to propagate x^* 's label) requires a trade-off between the amount of automatic annotation and the addition of error to the system. With a small k, few samples are automatically annotated, but, the ones annotated are done so with a good confidence, as they are close in the feature space. On the other hand, with a higher k, more samples are annotated, however, at the cost of possibly adding error to the classifier, as x^* is giving its label to samples at a further distance and therefore, may be wrongly annotated. In the present work, k empirically was set to 5, in order to obtain a significant amount of automatic annotation without compromising the classification accuracy and execution time. Figure 4 depicts the 1-NN label propagation step. Each circle represents a sample whose colours (green and red) represent two different classes. Samples in grey denote unlabelled samples. In this example, the sample x^* propagates its label to its 1-NN the sample B.
- k-rNN Semi-Supervised Active Learning (k-rNN-SSAL) [30]:** The sample selected by the QS (x^*) propagates its label to all the samples to which, regarding the labelled samples, it is their NN and it is within an empirically set distance. For the rNN method, as in [30], k was set to 1 to enhance the label propagation performance. Figure 5 illustrates 1-rNN label propagation step. Thus, in this example, the sample x^* propagates its label to the samples A, B and C.

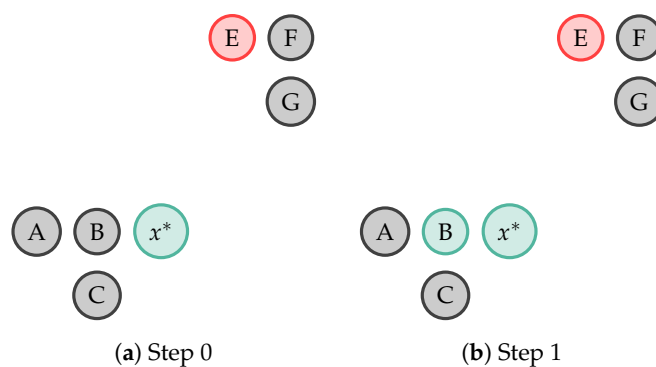


Figure 4. Example of 1-Nearest Neighbour (NN) label propagation, in which the sample x^* propagates its label (in this example represented by the colour green) to its 1-NN, the sample B. Each circle represents a sample whose colours, green and red, represent two different classes. The samples in grey denote unlabelled samples.

Distance Measures

When performing the label propagation step in the NN-SSAL and rNN-SSAL methods, there is a need for a measurement function able to obtain the distance between the different instances. Henceforth, in this section, it is provided four distance measurements. The first two applied in measuring the distance between the feature vector samples and the latter two between time series.

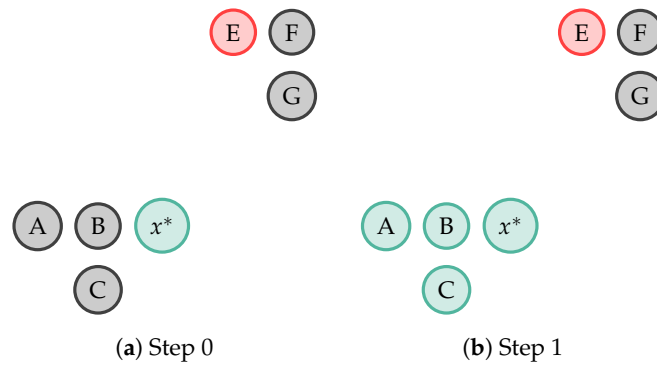


Figure 5. Example of 1-reverse Nearest Neighbour (rNN) label propagation, in which the sample x^* propagates its label (in this example represented by the colour green) to the samples to which, regarding the labelled samples, it is their NN, the samples A , B and C . Each circle represents a sample whose colours (green and red) represent two different classes. Samples in grey denote unlabelled samples.

- **Euclidean Distance:** Measures the length of the straight line distance between two samples (x_1 and x_2 , with dimension m) according to the following equation.

$$\sqrt{\sum_{i=1}^m (x_1 - x_2)^2} \quad (7)$$

- **Cosine Similarity Distance:** Measures the cosine of the angle between two samples (x_1 and x_2) according to the following equation. Cosine similarity ranges between -1 and 1, for opposite and coincident samples, respectively, with the distance value becoming larger as the samples become less similar.

$$1 - \frac{x_1 \cdot x_2}{\|x_1\| \cdot \|x_2\|} \quad (8)$$

- **Dynamic Time Warping (DTW):** Measures the similarity between two time-dependent sequences through a non-linear alignment minimising the distance between both. Moreover, the minimal distance is obtained through the computation of a local cost measure $C(S1, S2)$, where $S1 := \{s1_1, s1_2, \dots, s1_N\}$, $S2 := \{s2_1, s2_2, \dots, s2_M\}$ are two time series of length N and M ; $N, M \in \mathbb{N}$, respectively, producing a $N \times M$ cost matrix. Where each element corresponds to the Euclidean distance, between each pair of elements in the both sequences. Thus, $C(S1, S2)$, will hold a small value (low cost) if $S1$ and $S2$ are similar, or a larger value (high cost) otherwise. Hence, the DTW finds the warping path (W) yielding the minimum total cost amount all possible warping paths, by going through the low cost values in the local cost matrix [41,42].
- **Time Alignment Metric (TAM) [41]:** Uses the optimal time alignment obtained by the DTW to infer the intervals when two time series are in phase, advance or in delay in relation to each other. TAM returns a distance metric benefiting series in phase, and penalising when signals are in advance or delay with each other. Thus, resulting in an output value decreasing as the similarity between the two signals increases and increasing otherwise between 0 and 3, the former for signals constantly in phase and the latter for completely out of phase signals. Considering again, two time sequences $S1 := \{s1_1, s1_2, \dots, s1_N\}$ and $S2 := \{s2_1, s2_2, \dots, s2_M\}$ of length N and M ; $N, M \in \mathbb{N}$. Assuming $S2$ is delayed in relation to $S1$, by a total time $\overrightarrow{\theta}_{S1S2}$, advanced a total time $\overleftarrow{\theta}_{S1S2}$ and in phase by a time $\overleftrightarrow{\theta}_{S1S2}$. The TAM (Γ) is given by:

$$\Gamma = \psi_{advance} + \psi_{delay} + (1 - \psi_{phase}), \quad \Gamma \in \{R_0^+ | \Gamma \in [0 : 3]\} \quad (9)$$

$$\psi_{advance} = \frac{\overrightarrow{\theta}_{S1S2}}{N}, \quad \psi_{delay} = \frac{\overleftarrow{\theta}_{S1S2}}{M}, \quad \psi_{phase} = \frac{\overleftrightarrow{\theta}_{S1S2}}{\min(N, M)}$$

4. Results

In this section, we start by introducing the datasets used in this research, followed by an analysis of the performances of the aforementioned methods over several evaluation criteria. All the presented methods were implemented in Python using the modAL framework [43].

4.1. Datasets

The performances of the proposed frameworks were evaluated using two real-world datasets fully annotated and class balanced: the public **Human Activity Recognition Using Smartphones Dataset** from University of California Irvine (UCI) [13] and the **Continuous Activities of Daily Living (CADL)** acquired by the authors, whose information is summarised in Table 1. The CADL dataset was obtained continuously, in contrast to UCI dataset, where the activities were segmented. Thus, the CADL was used to provide a validation in a scenario more closely with the real-world requirements. To validate the proposed frameworks, a 10-fold Cross-Validation (CV) was implemented, each fold dividing the dataset into a train and test set. From each train set, one sample per class was chosen randomly to integrate the classifier's initial training set, while the rest composed the validation set, used to improve the learner in the AL process during 250 iterations. From the 10 folds, the last 5 iterations accuracy score values were averaged to compute the model's performance accuracy value and its standard deviation.

Table 1. University of California Irvine (UCI) dataset and Continuous Activities of Daily Living (CADL) dataset information on: number of users, activities performed, sensors, acquisition device, its position and dataset size.

	Datasets	
	UCI HAR Using Smartphones	Continuous Activities of Daily Living
N° of Users	30	12
Activities	Laying, sitting, standing, walking, upstairs and downstairs	Laying, sitting, standing, running, walking, upstairs and downstairs
Sensors	Accelerometer (50 Hz), gyroscope (50 Hz)	Accelerometer (100 Hz), gyroscope (100 Hz), barometer (30 Hz)
Device	Samsung Galaxy S2 (waist)	Samsung S5 (right hand) and wearable sensor (left hand, right ankle and right side of the waist)
N° of Samples	10,299	2047

4.2. Signal Processing

The data from the CADL dataset was submitted to a band-pass filter with cutoff frequencies of 0.3 Hz and 15 Hz, from which temporal, statistical and frequency domain features were extracted [11,14] from every 5 s window. The data from the UCI dataset was previously pre-processed according to [13]. A forward feature selection method was applied resulting in a feature vector composed of the features presented in Figure 6.

4.3. Model Selection

An analysis with common SL and UL techniques was performed with the purpose of finding the optimal technique to incorporate as the learner into the AL process. The respective performance results are shown in Table 2, for the SL and UL methods in accuracy and Adjusted Rand Index (ARI) score, respectively. As observed, Random Forest achieved the highest accuracy in both datasets, 91.4 (2.4)% and 89.1 (4.0)%, for the UCI and CADL dataset, respectively. For the UL methods, Spectral Clustering attained the highest ARI values for both datasets with a score of 57.8 (3.5)% and 61.9 (8.9)%, respectively. It was compared the performance results, for the public UCI dataset, to state-of-the-art researches [44],

namely, refs. [10,12,13] who have achieved accuracies of 86%, 96% and 96%, respectively. When training and evaluating the SL method on the same train and test set, it obtained an average accuracy of 89.1 (0.6)%.

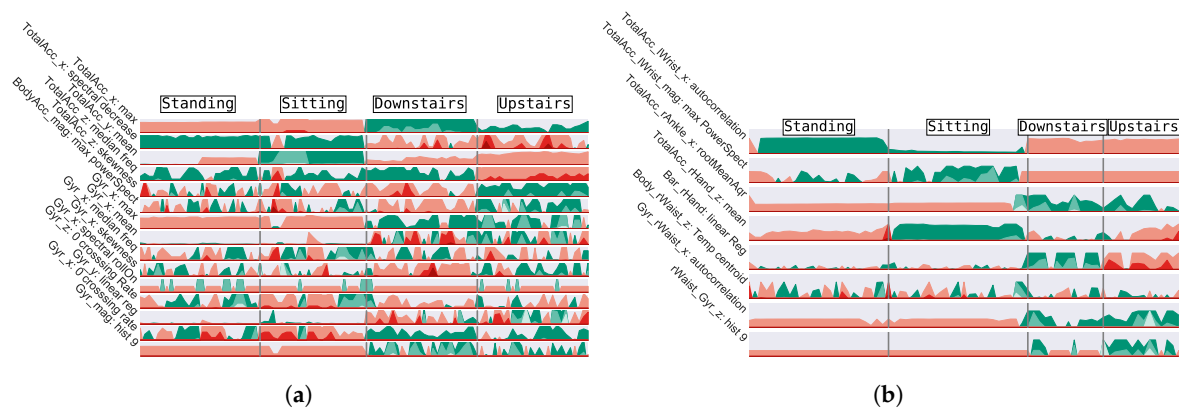


Figure 6. Horizon Plot showing the features and their behaviour along some of the dataset activities. In the y axis, it is presented the information about the sensor, its signal axis and the feature name. The green and red colours denote the signal's positive and negative values, respectively, with its intensity increasing with the feature's normalised absolute value and decreasing otherwise. (a) UCI dataset. (b) CADL dataset.

Table 2. SL and Unsupervised Learning (UL) methods classification's performance shown in accuracy and Adjusted Rand Index (ARI) score, respectively. For all listed values it is shown its 10-fold Cross-Validation (CV) average and standard deviation in percentage, the latter between parenthesis. The highest performance is shown in bold for each dataset.

(a) Supervised Learning		
Supervised Learning Method	Dataset	
	UCI	CADL
Nearest Neighbours	91.0 (1.9)	83.6 (3.4)
Decision Tree	87.4 (3.5)	83.6 (4.3)
Random Forest	91.4 (2.4)	89.1 (4.0)
SVM	90.7 (2.6)	77.6 (3.3)
AdaBoost	40.8 (6.8)	54.1 (4.6)
Naive Bayes	88.9 (2.9)	75.9 (3.1)
QDA	90.8 (2.7)	79.0 (3.7)
(b) Unsupervised Learning		
Unsupervised Learning Method	Dataset	
	UCI	CADL
K-Means	52.1 (4.3)	50.9 (6.1)
Mini Batch K-Means	50.7 (5.5)	50.5 (5.3)
Spectral Clustering	57.8 (3.5)	61.9 (8.9)
Gaussian Mixture	49.8 (2.7)	58.9 (6.6)
DBSCAN	16.4 (7.2)	13.9 (6.5)

4.4. QS Analysis

In the current sub-section, the QSs are analysed using the AL framework. This process aims to find the optimal QS, able to obtain the most representative labelled set and consequently attain the highest performance so it could be incorporated into the SSAL frameworks. In Table 3, the QSs are presented against PL (in which the QS randomly selects a sample from the unlabelled dataset), SL and UL. The comparison between the different QSs techniques is performed based on the following criteria:

1. **Accuracy:** The obtained accuracy values from the QSs are very similar and tend to the value obtained by the SL algorithm. This is supported by Figure 7, where it is presented the classifier's accuracy for the QSs throughout the AL iterations. As expected, overall, the learner becomes more reliable as its training set size increases, resulting in the continuous increase of its accuracy value throughout the iterations. Margin Sampling and Local Density * Least Confident Sampling attain the highest classification's performances, outperforming PL. However, the difference between AL and PL is low due to two reasons: (1) an initial biased prediction probability due to the classifier very small initial training set; (2) both UCI and CADL datasets are equally balanced. In this circumstance a random selection of samples is enough to create a representative dataset with a few samples from each class. Local Density and Local Density * Margin Sampling, attain the lowest score, not achieving a reasonable performance. These QSs' low performances are explained by the biased training set, non-representative of the entire dataset distribution under which the classifier operates. As observed in Figure 8a, the density weight causes the preferential selection of activities located in high-density regions, for the deterioration of the remaining as they become unknown for the classifier. Under these circumstances, the classifier does not have a homogeneous training set with sufficient amount of samples from all the class labels from which it can learn to be able to correctly predict all the samples' labels. Still, with the exception of the aforementioned QSs, the remaining results are in accordance with the literature review [29,30] with the introduction of a density weight to the uncertainty sampling functions avoiding the selection of outliers as observed in Figure 8.

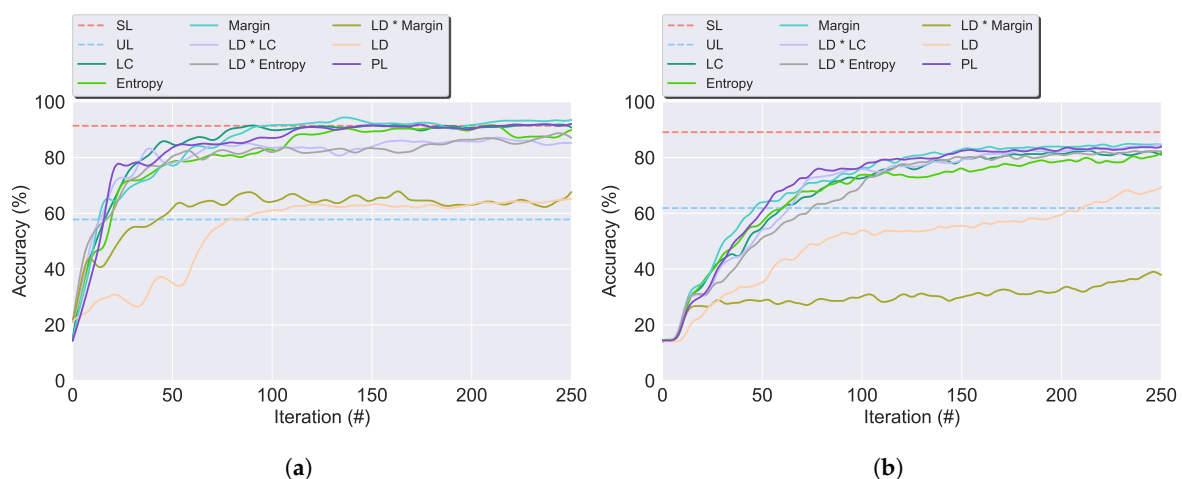


Figure 7. Average increase of the AL classifier's accuracy for the Query Strategies (QSs) throughout the cycle of iterations. The horizontal lines denote the average accuracy for SL (in red) and UL (in blue). LD denotes the Local Density Sampling and LC the Least Confident Sampling. (a) UCI dataset. (b) CADL dataset.

2. **QS Execution Time:** With the exception of the density weighted QSs, in general, the selective sampling functions hold a low execution time. For the density weighted QSs it is observed a significant increase in the QS execution time due to the calculation of the density weight which requires the calculation of each sample's NNs. This process ultimately increases the algorithm's computational complexity and execution time. The execution times were obtained using a E3-1285 v6 @ 4.10GHz CPU and 16 GB of RAM.

Due to the coherent high accuracy performance, surpassing PL, and its low execution time and computational complexity, Margin Sampling was selected as the most suitable QS to be included in both the AL and SSAL frameworks. Hence, forthcoming result presentations on this section were achieved using Margin Sampling. Besides the algorithm's performance analysis, it is also worth to mention a comparison between the amount of labelled data for SL and AL. From 100% of the validation set annotated in SL, to, approximately 2.8 (0.1)% and 13.9 (0.5)%, for the UCI

and CADL dataset, corresponding to the annotation of 250 samples and a reduction of 97.2 (0.1)% and 86.1 (0.5)% in the validation set annotation cost, respectively. These results confirm the applicability of AL in the context of HAR and its efficiency in reducing the annotation effort required to construct a highly confident classifier.



Figure 8. Principal Component Analysis (PCA) of the CADL dataset samples after performing AL for 50 iterations. The classifier's training set samples are depicted by the 'x's, whose colour identifies their respective class. The darker grey dots represent the unselected samples existent in the validation set. (a) Local Density * Margin Sampling. (b) Local Density * Least Confident Sampling.

Table 3. Experimental results for the Query Strategies in terms of: classifier's accuracy and the QS algorithm's execution time. For all listed values it is shown its 10-fold CV average and standard deviation, the latter between parenthesis. The best performing algorithm is shown in bold for each dataset.

Query Strategy	UCI Dataset		CADL Dataset	
	Accuracy in %	QS Time in s	Accuracy in %	QS Time in s
Local Density * Least Confident	87.6 (4.0)	27.2 (5.1)	83.5 (6.7)	0.9 (0.1)
Least Confident	87.9 (3.7)	0.1 (0.1)	72.0 (5.0)	0.1 (0.1)
Local Density * Entropy	85.7 (3.7)	22.5 (0.4)	80.3 (7.9)	0.9 (0.1)
Entropy	87.1 (2.5)	0.1 (0.1)	70.9 (6.0)	0.1 (0.1)
Local Density * Margin	52.5 (8.6)	22.6 (0.5)	32.8 (8.0)	0.9 (0.1)
Margin	88.4 (2.8)	0.1 (0.1)	84.8 (7.0)	0.1 (0.1)
Local Density	63.6 (5.9)	22.5 (0.4)	68.9 (8.5)	0.9 (0.1)
Passive Learning	87.0 (4.5)	0.1 (0.1)	82.0 (7.6)	0.1 (0.1)
Supervised Learning	91.4 (2.4)		89.1 (4.0)	
Unsupervised Learning	57.8 (3.5)		61.9 (8.9)	

4.5. Active Learning Semi-Supervised Analysis

This sub-section aims to present a comparison and select the optimal automatic annotation method. In Table 4 it is shown the methods presented in Section 3.2 compared against techniques previously applied in the context of HAR, described in the literature review, such as AL, PL, SL and UL, replicated in order to verify the model competitiveness.

Table 4. Experimental results for the SSAL methods: accuracy, automated annotation percentage, automated annotation accuracy and the algorithm’s execution time. For all listed values it is shown its 10-fold CV average and standard deviation, the latter between parenthesis. Following the underscore in the NN and rNN methods: Euc, Cos, DTW and TAM, denote the similarity distances used in the respective method. The best performing algorithm is shown in bold for each dataset.

Method	UCI Dataset				CADL Dataset			
	Accuracy in %	Aut Ann in %	Ann Acc in %	Time in s	Accuracy in %	Aut Ann in %	Ann Acc in %	Time in s
NN_Euc	88.1 (2.7)	13.5 (0.1)	76.8 (1.0)	92.3 (9.4)	82.5 (5.7)	68.2 (2.7)	68.2 (1.0)	44.3 (4.8)
NN_Cos	89.4 (3.0)	13.5 (0.1)	75.3 (1.4)	860.9 (7.4)	82.8 (8.2)	68.2 (2.7)	64.6 (1.7)	79.1 (4.2)
NN_DTW	70.3 (6.0)	13.5 (0.1)	39.7 (1.7)	6772.0 (4.5)	68.4 (5.4)	68.2 (2.7)	30.3 (3.1)	6735.4 (1.3)
NN_TAM	75.2 (4.8)	13.5 (0.1)	44.1 (4.0)	6771.4 (5.4)	68.8 (7.3)	68.2 (2.7)	31.6 (1.2)	6735.4 (1.9)
rNN_Euc	85.4 (2.7)	33.3 (2.0)	77.3 (3.8)	437.9 (61.9)	74.9 (7.8)	56.6 (1.3)	66.5 (2.7)	60.0 (10.8)
rNN_Cos	82.5 (3.6)	37.7 (4.0)	74.7 (5.2)	1165.3 (86.4)	71.3 (8.1)	61.7 (2.5)	59.5 (5.6)	89.7 (11.2)
rNN_DTW	65.1 (4.9)	13.7 (1.6)	41.8 (2.9)	6995.4 (74.9)	77.3 (7.1)	20.8 (0.8)	39.9 (1.9)	6739.2 (7.6)
rNN_TAM	64.5 (8.4)	11.9 (3.9)	45.7 (4.9)	6968.3 (81.2)	81.8 (8.4)	11.2 (2.0)	41.3 (4.6)	6733.8 (7.3)
ST-SSAL	84.0 (6.3)	56.7 (11.6)	86.1 (10.5)	99.3 (17.5)	84.8 (7.0)	20.9 (6.9)	92.5 (2.7)	11.3 (1.0)
AL	88.4 (2.8)			23.6 (5.0)	84.8 (7.0)			12.0 (1.0)
PL	87.0 (4.5)			23.2 (1.4)	82.0 (7.6)			10.3 (1.0)
SL	91.4 (2.4)			0.7 (0.1)	89.1 (4.0)			0.1 (0.1)
UL	57.8 (3.5)			0.3 (0.2)	61.9 (8.9)			0.1 (0.1)

1. **Accuracy:** Experimental results demonstrated that with the exception of the SSAL methods using the DTW or TAM distance, the accuracy of the proposed methods converges to the results of the SL technique. Figure 9 presents the classifier’s accuracy for the SSAL methods throughout the AL iterations. For each method, in every iteration the model training set grows, resulting in the increase of the classification’s accuracy.

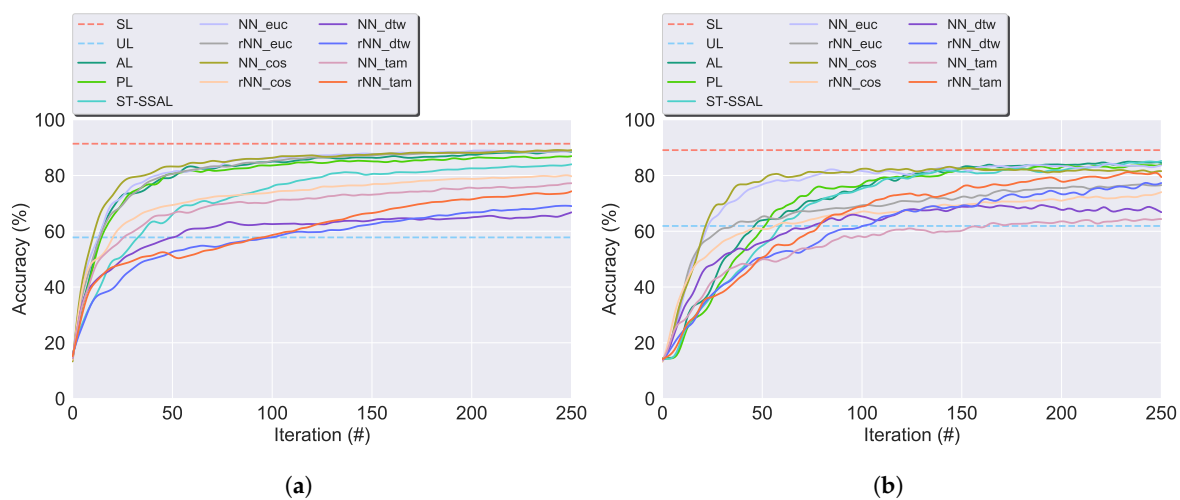


Figure 9. Classifier’s accuracy for the SSAL methods throughout the AL iterations. The horizontal lines denote the 10-CV average accuracy for SL (in red), and UL ARI score (in blue). Following the underscore in the NN and rNN methods: Euc, Cos, Dynamic Time Warping (DTW) and Time Alignment Metric (TAM), denote the distances used. (a) UCI dataset. (b) CADL dataset.

2. **Automated Annotation Percentage (Aut Ann):** Consists of the percentage of samples automatically annotated in relation to the total validation set size. Figure 10 displays the evolution on the percentage of the validation set unlabelled samples for the SSAL methods throughout the AL cycle iterations. In the AL and PL, the oracle annotates one sample per iteration, therefore, in Figure 10, both present an overlapping linear decline in the number of unlabelled samples. The NN-SSAL methods annotate six samples per iteration, one by the oracle and five by the automatic annotator, therefore, these show in Figure 10 an overlapping linear decline with higher

slope than AL and PL. On the other hand, rNN-SSAL presents a curved decline in the number of unlabelled samples, outperforming the remaining during the first iterations. ST-SSAL displays during the initial iterations an automatic annotation percentage similar to AL and PL, with only the expert annotator labelling new samples and no automatic annotation, since the 0.98 prediction confidence threshold required for automatic annotation is not reached due to the classifier small labelled training set. Once the labelled set becomes representative of the dataset, the 0.98 threshold is reached and ST-SSAL automatic annotation increases exponentially until the unlabelled dataset becomes exhausted, easily surpassing the 5 constantly automatically annotated by the NN-SSAL. On the whole, ST-SSAL attains the highest performance for the UCI dataset, and NN-SSAL for the CADL dataset, the latter closely followed by rNN-SSAL.

3. **Automated Annotation Accuracy (Ann Acc):** Consists of the percentage of correctly automatically annotated samples. Moreover, Figure 11 presents the evolution throughout the AL process of the automated annotation accuracy for the SSAL methods. As observed, ST-SSAL outperforms the remaining, attaining high results, especially for the latter iterations. ST-SSAL high annotation accuracy on the latter iterations results from the δ_{ST} threshold required for the automatic annotation to be performed. As noted, this threshold is only reached during the latter iterations when the model training set becomes representative of the dataset and predictions can be performed with high certainty. This fact contrasts with the remaining methods, where higher results are obtained during the first iterations. For the NN-SSAL methods, this is justified by the queried sample propagating its label to closer samples during the first iterations. Whereas in the latter iterations, its closest neighbours start to be already annotated so the sample's label is given to further away samples. The same is applied to rNN-SSAL, with the stabilisation of the propagation accuracy being accompanied by the stabilisation of the amount of automatic propagation (Figure 10). Additionally, this metric allows to discriminate between the performance of the different distance functions. As it can be seen, the Euclidean distance and Cosine similarity obtained similar results. In contrast to DTW and TAM, presenting a poor percentage of correctly annotated samples, explaining their low classification performance.

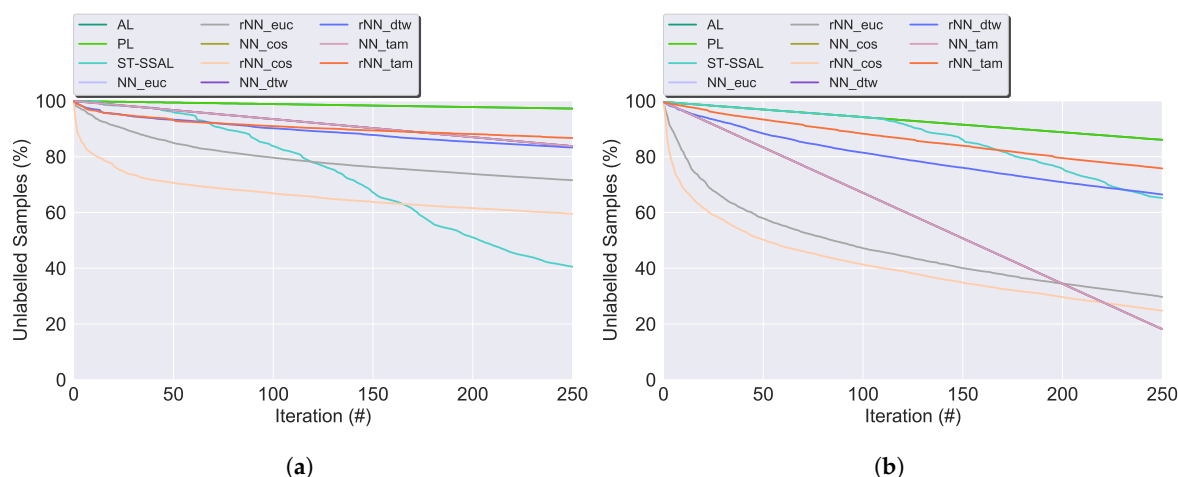


Figure 10. Evolution on the percentage of the validation set unlabelled samples for the SSAL methods throughout the AL cycle iterations. Following the underscore in the NN and rNN methods: Euc, Cos, DTW and TAM, denote the similarity distances used in the respective method. (a) UCI dataset. (b) CADL dataset.

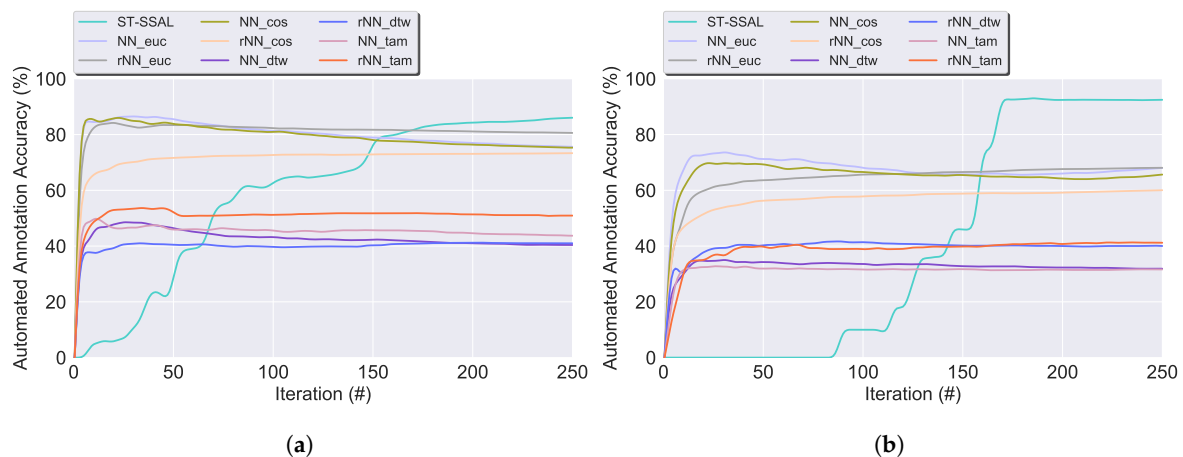


Figure 11. Percentage of correctly automatically annotated samples throughout the AL iterations, for the SSAL methods, using the UCI and the CADL datasets. Following the underscore in the NN and rNN methods: Euc, Cos, DTW and TAM, denote the similarity distances used in the respective method. (a) UCI dataset. (b) CADL dataset.

- Execution time:** AL shows the fastest execution time. The algorithm execution time, allows to favour between the different similarity measures, since the DTW and TAM expensive time and computational complexity, render those algorithms non-applicable to a viable solution. Furthermore, comparing the presented four distance metrics, Euclidean distance presents the lowest time expense and, therefore, was chosen as the most suitable distance metric.

As noted, generally speaking, ST-SSAL outperformed the remaining SSAL methods reaching a higher classification accuracy due to its good performance in the automatic annotation with high certainty. The rNN-SSAL method, although annotating a substantial percentage of the dataset, its lower automatic annotation accuracy performance resulted in the decay of its classification accuracy. To conclude, if we compare ST-SSAL and AL, both methods achieve similar classification performance. However, ST-SSAL was able to annotate a higher volume of data with similar annotation effort without compromising the classification accuracy.

4.6. Stopping Criterion Analysis

In this sub-section, we analyse the introduction of a SC to the AL process. Previously, the presented results were based on a pre-defined number of queries (250 queries). As depicted in Figure 9, depending on the dataset some models reach their highest accuracy score quicker than other, stabilising around that value for the forthcoming iterations. Thus, as explained in Section 3.1.3, in order to optimise the trade-off between the classifier's performance and the expensive training set annotation cost, the number of iterations should be minimised according to the respective algorithm and dataset.

Table 5 presents for both datasets the experimental results for the SSAL methods using the proposed SCs methods in terms of accuracy and below, total number of iterations. Moreover, in the columns SP, for both datasets it is shown the accuracy score for each method in stabilisation and the considered optimal number of iterations at the stopping point. These values were selected in order to achieve a stable accuracy performance with the minimal annotation effort and higher coherency between different folds from the 10-CV (i.e., minimal standard deviation).

Table 5. SC methods accuracy and standard deviation average, acc (std) in percetage. Accuracy score denoted as Acc, and average iterations denoted as N.it over a 10-fold CV on the UCI and CADL datasets. Moreover, for each dataset, under the Stopping Point (SP) columns, the accuracy results in stabilisation are shown and below, the considered optimal number of iterations are shown. The most suitable method is shown in bold for each dataset.

Method		UCI						CADL					
		SP	Max-Conf	Over-Unc	CC	Max-CC	Over-CC	SP	Max-Conf	Over-Unc	CC	Max-CC	Over-CC
NN_Euc	Acc	85.5 (3.3)	69.2 (12.5)	78.7 (8.7)	73.6 (11.9)	82.2 (8.2)	81.5 (9.2)	81.6 (5.1)	68.2 (8.9)	76.7 (6.3)	57.3 (14.1)	77.9 (6.0)	80.2 (6.4)
	N.it	68.5 (26.1)	20.0 (11.4)	37.5 (9.5)	28.9 (12.4)	68.0 (39.3)	82.5 (44.1)	69.6 (38.4)	44.5 (23.4)	64.5 (14.3)	21.2 (8.2)	123.0 (97.5)	103.0 (73.8)
NN_Cos	Acc	79.6 (7.3)	70.3 (11.3)	79.0 (9.6)	68.8 (13.3)	81.5 (10.4)	82.7 (10.8)	74.6 (15.6)	62.8 (16.9)	69.8 (9.9)	44.2 (20.5)	74.9 (11.0)	78.4 (10.5)
	N.it	61.1 (29.7)	22.0 (13.2)	41.5 (11.9)	16.8 (7.6)	76.5 (47.5)	67.5 (27.0)	80.5 (31.4)	28.0 (10.0)	52.5 (15.8)	13.5 (6.8)	88.5 (65.1)	88.5 (56.9)
NN_DTW	Acc	67.0 (4.3)	54.3 (6.0)	47.8 (12.4)	57.6 (244.7)	67.4 (7.4)	58.1 (17.5)	58.7 (12.5)	43.9 (7.6)	33.8 (7.3)	52.0 (17.0)	62.8 (4.0)	64.5 (7.3)
	N.it	250.0 (0.0)	26.0 (7.4)	16.0 (2.2)	244.7 (155.0)	297.7 (105.9)	220.4 (158.8)	117.0 (95.1)	22.5 (4.5)	16.0 (5.5)	215.1 (134.5)	278.3 (79.9)	305.3 (11.2)
NN_TAM	Acc	77.0 (4.9)	59.5 (9.0)	68.0 (7.9)	63.8 (20.0)	73.1 (7.0)	67.3 (12.4)	62.0 (6.0)	49.5 (8.0)	50.2 (8.4)	60.9 (12.7)	63.9 (5.2)	65.8 (5.4)
	N.it	313.2 (78.3)	37.5 (17.0)	53.5 (10.8)	244.6 (153.2)	318.1 (95.7)	260.8 (136.7)	153.0 (86.3)	47.5 (23.9)	47.5 (14.8)	274.9 (90.0)	305.3 (11.2)	305.3 (11.2)
rNN_Euc	Acc	84.2 (2.3)	72.6 (12.4)	83.7 (3.9)	61.9 (13.4)	72.6 (13.9)	84.9 (2.7)	77.2 (5.9)	59.9 (14.5)	76.9 (9.0)	45.0 (10.8)	74.9 (7.0)	78.1 (5.4)
	N.it	92.5 (23.0)	37.5 (17.6)	97.0 (37.6)	15.9 (8.7)	45.5 (27.4)	129.4 (52.1)	198.3 (133.3)	32.5 (11.8)	174.3 (97.2)	14.0 (2.6)	209.6 (125.4)	319.1 (92.7)
rNN_Cos	Acc	65.5 (9.9)	60.6 (11.6)	66.0 (8.2)	52.7 (10.4)	65.9 (13.3)	72.0 (12.1)	52.7 (16.1)	53.4 (14.7)	60.6 (11.6)	34.2 (17.5)	53.7 (9.4)	60.5 (14.4)
	N.it	37.0 (10.4)	31.0 (19.8)	37.5 (15.2)	9.7 (3.5)	63.5 (71.9)	66.5 (37.6)	43.0 (21.5)	32.0 (11.6)	72.5 (34.2)	12.3 (5.6)	87.4 (101.3)	165.7 (136.7)
rNN_DTW	Acc	38.7 (15.4)	44.6 (8.9)	43.0 (6.8)	29.2 (9.7)	56.4 (13.9)	56.1 (14.6)	41.2 (14.8)	35.4 (6.9)	35.9 (6.3)	23.4 (5.8)	56.9 (20.9)	61.2 (18.2)
	N.it	39.0 (27.2)	21.0 (5.9)	31.5 (13.2)	7.4 (0.9)	156.6 (158.0)	133.2 (142.5)	51.0 (32.9)	25.0 (7.3)	30.0 (13.0)	9.3 (4.3)	97.0 (94.2)	132.0 (87.2)
rNN_TAM	Acc	68.6 (9.4)	43.1 (10.1)	48.2 (7.0)	40.5 (10.2)	59.7 (16.7)	50.1 (15.8)	56.9 (19.6)	29.8 (7.8)	27.7 (7.4)	33.1 (22.6)	53.2 (19.0)	47.9 (20.3)
	N.it	290.7 (118.6)	17.0 (3.7)	21.0 (5.5)	12.2 (7.9)	124.3 (137.2)	128.2 (146.3)	77.0 (34.4)	18.0 (4.6)	14.5 (5.5)	22.0 (24.7)	54.0 (29.6)	128.2 (146.2)
ST-SSAL	Acc	85.2 (3.5)	48.8 (15.3)	66.3 (9.9)	49.1 (18.9)	75.3 (10.7)	84.5 (4.1)	82.8 (6.6)	33.0 (13.0)	61.9 (12.1)	15.0 (0.6)	76.2 (12.7)	84.7 (7.2)
	N.it	201.5 (83.8)	22.5 (9.5)	61.0 (29.2)	66.4 (107.1)	81.0 (41.6)	214.0 (46.5)	164.0 (70.1)	25.0 (10.2)	67.0 (21.1)	12.0 (0.0)	120.0 (28.4)	182.0 (53.9)
AL	Acc	86.1 (2.6)	60.3 (11.7)	65.7 (10.5)	37.4 (11.1)	84.0 (4.7)	86.0 (5.6)	84.6 (7.4)	51.2 (15.5)	64.0 (15.0)	15.0 (0.6)	76.8 (9.7)	76.6 (16.5)
	N.it	109.5 (24.4)	28.5 (10.1)	38.0 (16.4)	10.9 (3.9)	98.0 (26.1)	97.0 (40.8)	193.0 (51.0)	58.0 (17.9)	118.0 (59.5)	12.0 (0.0)	116.0 (42.0)	139.0 (43.1)
SL		91.4 (2.4)						89.1 (4.0)					
UL		57.8 (3.5)						61.9 (8.9)					

The most suitable SC is overall coherent between the different datasets and changes according to the SSAL algorithm. As it can be seen in Table 5, with the introduction of a SC the number of iterations, consequently, the required annotation cost was notably reduced. Therefore, optimising the computational demands of the pipeline. For the ST-SSAL (selected in the previous sub-section as the best performing SSAL method) using the Over-CC SC, an accuracy of 84.5 (4.1)%, F1 score of 82.9 (4.5)% and accuracy of 84.7 (7.2)%, F1 score of 86.3 (6.9)% was attained, with the annotation cost of 214.0 (46.5) and 182.0 (53.9) queries, for the UCI and CADL datasets, respectively, consisting of annotating 2.4 (0.5)% and 10.2 (2.8)% of the validation set. Moreover, the automated annotation along with the manually annotated samples enabled to label 55.8 (11.8)% and 19.1 (13.4)% of the validation set with an accuracy on the automated annotation of 90.5 (4.6)% and 56.7 (46.3)%. Thereupon, the ST-SSAL method allowed to reduce the manual annotation cost on 97.6 (0.6)% and 89.8 (2.8)% for both datasets.

For last, a confusion matrix for the ST-SSAL method using the Over-CC SC is presented in Figure 12, where it is possible to establish conclusions regarding the activities correctly and incorrectly predicted by the classifier. For both datasets, the misclassification was higher between Downstairs/Upstairs and Sitting/Standing. The barometer's linear regression feature, as it can be seen in Figure 6, presents high distinction between Downstairs/Upstairs, thus, allowed to improve the discrimination between these activities in the CADL dataset. Dynamic activities, due to its distinct motion characteristics and cyclic behaviour presented an overall clear discrimination against static activities.

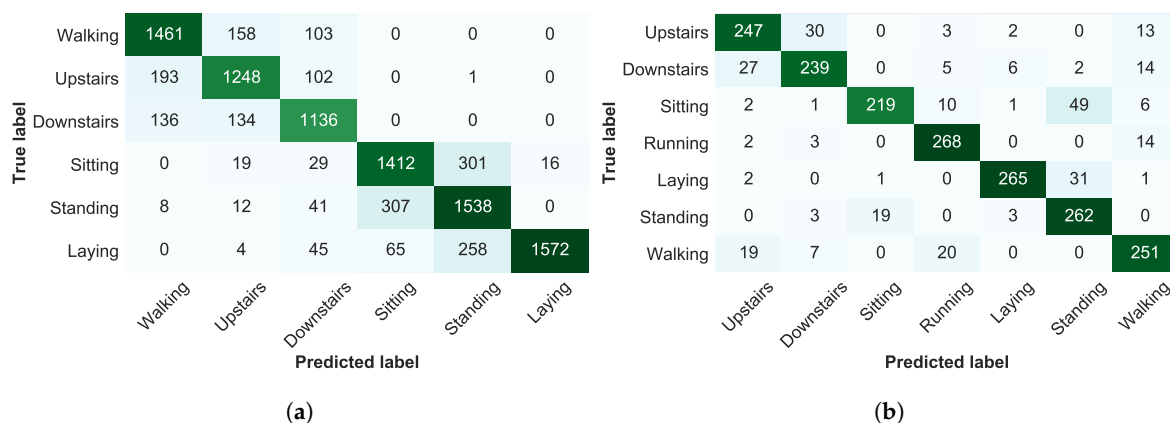


Figure 12. Confusion matrix for the Self-Training Semi-Supervised Active Learning (ST-SSAL) method using the Overall Uncertainty Classification-Change Stopping Criterion (Over-CC SC). (a) UCI dataset. (b) CADL dataset.

Lastly, comparing the performance results of the best performing method ST-SSAL method using the Over-CC SC, to state-of-the-art researches for the UCI dataset, namely, [10,12,13] who have achieved accuracies of 86%, 96% and 96%, respectively. When evaluating on the same test set, ST-SSAL obtained an accuracy of 83.2 (4.5)%, after 230.5 (21.9) queries. Therefore, although it did not outperform the aforementioned researches, satisfactory results were achieved, annotating 48.5 (18.1)% of the validation set with an accuracy of 88.0 (5.4)%, and a notable reduction of 96.8 (0.3)% in the training set annotation cost.

5. Conclusions

Over the last years, the advances on smartphone and wearable technology allowed the proliferation of their use as unobtrusive and pervasive sensors. The volume of the recorded data by these equipment is significant and poses challenges on the development of traditional machine learning approaches that rely on annotated data to guarantee accurate model performance. The process of annotating a large dataset requires a great effort by the manual annotation of an expert.

Based on the aforementioned challenges in the HAR context, this work addressed a semi-automatic data annotation approach with the goal of optimising the process of data annotation and still be able to learn an accurate machine learning model. Our method relies on two steps: (1) a QS criterion to select the most relevant samples to be labelled by an expert; (2) an automatic method to propagate the annotated sample's label over similar samples on the entire dataset.

Our main contribution consists of applying SSAL in two HAR datasets, built through a comprehensive study of state-of-the-art QSs and SCs, and the comparison to AL. These methods were evaluated over several automatic annotation strategies based on different distance functions to build an optimal SSAL system with applications for human movement.

Regarding the QS, Margin Sampling achieved the best results in the study performed with AL, reaching an accuracy of 88.4 (2.8)% and 84.8 (0.1)% for UCI and CADL, respectively, and maintaining low computational time.

If we compare ST-SSAL and AL, both methods achieve similar classification performance. However, ST-SSAL was able to annotate a higher volume of data with similar annotation effort, without compromising the classification accuracy. This paper extends the work conducted by [31] on HAR, since it applies ST on the labels previously selected by AL. The ST-SSAL using the Over-CC SC obtained an accuracy of 84.5 (4.1)% and 84.7 (7.2)% for the UCI and CADL datasets, respectively, with a reduction in the Oracle annotation effort on 97.6 (0.6)% and 89.8 (2.8)% of total number of samples for both datasets.

For future work we identified the following research lines: development of a multi-oracle system with non-expert users, allowing the evaluation of the system's response to any eventual integration of bias in the annotation process; integration of the samples' estimation annotation cost into the QS, since different samples may present different annotation costs; and the development of an annotation interface which relies on ST-SSAL to facilitate data annotation. The annotation interface can be used to support current methods based on video recordings, with each samples being correlated to a point in time and a recording is present so the user only is required to watch the selected frames given by the QS instead of dedicating extensive hours watching the entire recording.

Author Contributions: Conceptualization, H.G.; Data curation, P.B., J.S. and D.F.; Formal analysis, P.B.; Methodology, P.B. and D.F.; Project administration, H.G.; Software, J.S.; Supervision, H.G.; Writing—original draft, P.B.; Writing—review & editing, J.S., D.F. and H.G.

Funding: This work was supported by North Portugal Regional Operational Programme (NORTE 2020), Portugal 2020 and the European Regional Development Fund (ERDF) from European Union through the project Symbiotic technology for societal efficiency gains: Deus ex Machina (DEM) [NORTE-01-0145-FEDER-000026].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Patel, S.; Park, H.; Bonato, P.; Chan, L.; Rodgers, M. A review of wearable sensors and systems with application in rehabilitation. *J. NeuroEng. Rehabil.* **2012**, *9*, 21, doi:10.1186/1743-0003-9-21. [[CrossRef](#)] [[PubMed](#)]
2. Ramasamy Ramamurthy, S.; Roy, N. Recent trends in machine learning for human activity recognition—A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1254, doi:10.1002/widm.1254. [[CrossRef](#)]
3. Organisation, W.H. Global Strategy on Diet, Physical Activity and Health. Available online: <http://www.who.int/dietphysicalactivity/pa/en/> (accessed on 19 August 2018).
4. del Rosario, M.B.; Redmond, S.J.; Lovell, N.H. Tracking the Evolution of Smartphone Sensing for Monitoring Human Movement. *Sensors* **2015**, *15*, 18901–18933, doi:10.3390/s150818901. [[CrossRef](#)] [[PubMed](#)]
5. Lara, O.D.; Labrador, M.A. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1192–1209, doi:10.1109/SURV.2012.110112.00192. [[CrossRef](#)]
6. Zhang, W.; Zhang, Z.; Qi, D.; Liu, Y. Automatic Crack Detection and Classification Method for Subway Tunnel Safety Monitoring. *Sensors* **2014**, *14*, 19307–19328, doi:10.3390/s141019307. [[CrossRef](#)]

7. Bulling, A.; Blanke, U.; Schiele, B. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Comput. Surv.* **2014**, *46*, 33:1–33:33, doi:10.1145/2499621. [[CrossRef](#)]
8. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
9. Attal, F.; Mohammed, S.; Dedabrishvili, M.; Chamroukhi, F.; Oukhellou, L.; Amirat, Y. Physical Human Activity Recognition Using Wearable Sensors. *Sensors* **2015**, *15*, 31314–31338, doi:10.3390/s151229858. [[CrossRef](#)]
10. Silva, J.; Monteiro, M.; Sousa, F. Human activity classification with inertial sensors. *Stud. Health Technol. Inform.* **2014**, *200*, 101–104.
11. Figueira, C.; Matias, R.; Gamboa, H. Body Location Independent Activity Monitoring. In Proceedings of the BIOSIGNALS, Rome, Italy, 21–23 February 2016.
12. Romera-Paredes, B.; Aung, M.; Bianchi-Berthouze, N. A One-vs-One Classifier Ensemble with Majority Voting for Activity Recognition. In Proceedings of the ESANN 2013 proceedings 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 24–26 April 2013; pp. 443–448.
13. Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; Reyes-Ortiz, J.L. A Public Domain Dataset for Human Activity Recognition using Smartphones. In Proceedings of the ESANN 2013 proceedings 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 24–26 April 2013.
14. Machado, I.P.; Gomes, A.L.; Gamboa, H.; Paixão, V.; Costa, R.M. Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization. *Inf. Process. Manag.* **2015**, *51*, 204–214, doi:10.1016/j.ipm.2014.07.008. [[CrossRef](#)]
15. Luísa Gomes, A.; Paixão, V.; Gamboa, H. Human Activity Recognition Based on Novel Accelerometry Features and Hidden Markov Models Application. In Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2015), Lisbon, Portugal, 12–15 January 2015; SCITEPRESS—Science and Technology Publications, Lda: Setúbal, Portugal, 2015; Volume 4, pp. 76–85, doi:10.5220/0005215800760085. [[CrossRef](#)]
16. Murad, A.; Pyun, J.Y. Deep Recurrent Neural Networks for Human Activity Recognition. *Sensors* **2017**, *17*, 2556, doi:10.3390/s17112556. [[CrossRef](#)]
17. Ordóñez, F.J.; Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* **2016**, *16*, 115, doi:10.3390/s16010115. [[CrossRef](#)] [[PubMed](#)]
18. Li, F.; Shirahama, K.; Nisar, M.A.; Köping, L.; Grzegorzec, M. Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors. *Sensors* **2018**, *18*, 679, doi:10.3390/s18020679. [[CrossRef](#)] [[PubMed](#)]
19. Stikic, M.; Larlus, D.; Ebert, S.; Schiele, B. Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2521–2537, doi:10.1109/TPAMI.2011.36. [[CrossRef](#)] [[PubMed](#)]
20. Hossain, H.M.; Khan, M.A.A.H.; Roy, N. Active learning enabled activity recognition. *Pervasive Mob. Comput.* **2017**, *38*, 312–330. [[CrossRef](#)]
21. Liu, R.; Chen, T.; Huang, L. Research on human activity recognition based on active learning. In Proceedings of the 2010 International Conference on Machine Learning and Cybernetics, Qingdao, China, 11–14 July 2010; Volume 1, pp. 285–290. doi:10.1109/ICMLC.2010.5581050. [[CrossRef](#)]
22. Cruciani, F.; Cleland, I.; Nugent, C.; McCullagh, P.; Synnes, K.; Hallberg, J. Automatic Annotation for Human Activity Recognition in Free Living Using a Smartphone. *Sensors* **2018**, *18*, 2203, doi:10.3390/s18072203. [[CrossRef](#)] [[PubMed](#)]
23. Tonkin, E.L.; Burrows, A.; Woznowski, P.; Laskowski, P.; Yordanova, K.Y.; Twomey, N.; Craddock, I.J. Talk, Text, Tag? Understanding Self-Annotation of Smart Home Data from a User’s Perspective. *Sensors* **2018**, *18*, 2365, doi:10.3390/s18072365. [[CrossRef](#)] [[PubMed](#)]
24. Schröder, M.; Yordanova, K.; Bader, S.; Kirste, T. Tool Support for the Online Annotation of Sensor Data. In Proceedings of the 3rd International Workshop on Sensor-Based Activity Recognition and Interaction (iWOAR ’16), Rostock, Germany, 23–24 June 2016; ACM: New York, NY, USA, 2016; pp. 9:1–9:7, doi:10.1145/2948963.2948972. [[CrossRef](#)]

25. Cleland, I.; Han, M.; Nugent, C.; Lee, H.; McClean, S.; Zhang, S.; Lee, S. Evaluation of Prompted Annotation of Activity Data Recorded from a Smart Phone. *Sensors* **2014**, *14*, 15861–15879, doi:10.3390/s140915861. [[CrossRef](#)]
26. Shahmohammadi, F.; Hosseini, A.; King, C.E.; Sarrafzadeh, M. Smartwatch Based Activity Recognition Using Active Learning. In Proceedings of the Second IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE '17), Philadelphia, PA, USA, 17–19 July 2017; IEEE Press: Piscataway, NJ, USA, 2017; pp. 321–329, doi:10.1109/CHASE.2017.115. [[CrossRef](#)]
27. Alemdar, H.; van Kasteren, T.L.M.; Ersoy, C. Using Active Learning to Allow Activity Recognition on a Large Scale. In Proceedings of the International Joint Conference on Ambient Intelligence, Amsterdam, The Netherlands, 16–18 November 2011; Keyson, D.V., Maher, M.L., Streitz, N., Cheok, A., Augusto, J.C., Wichert, R., Englebienne, G., Aghajan, H.; Kröse, B.J.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 105–114.
28. Zhu, J.; Wang, H.; Tsou, B.K.; Ma, M. Active Learning With Sampling by Uncertainty and Density for Data Annotations. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *18*, 1323–1331, doi:10.1109/TASL.2009.2033421. [[CrossRef](#)]
29. Zhu, J.; Wang, H.; Yao, T.; Tsou, B.K. Active Learning with Sampling by Uncertainty and Density for Word Sense Disambiguation and Text Classification. In Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08), Manchester, UK, 18–22 August 2008; Association for Computational Linguistics: Stroudsburg, PA, USA, 2008; Volume 1, pp. 1137–1144.
30. He, G.; Li, Y.; Zhao, W. An uncertainty and density based active semi-supervised learning scheme for positive unlabeled multivariate time series classification. *Knowl.-Based Syst.* **2017**, *124*, 80–92, doi:10.1016/j.knosys.2017.03.004. [[CrossRef](#)]
31. Stikic, M.; Van Laerhoven, K.; Schiele, B. Exploring semi-supervised and active learning for activity recognition. In Proceedings of the 12th IEEE International Symposium on Wearable Computers, Pittsburgh, PA, USA, 28 September–1 October 2008; Volume 1, pp. 81–88, doi:10.1109/ISWC.2008.4911590. [[CrossRef](#)]
32. Zhu, X.; Goldberg, A.B. Introduction to semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2009**, *3*, 1–130. [[CrossRef](#)]
33. Fu, Y.; Zhu, X.; Li, B. A survey on instance selection for active learning. *Knowl. Inf. Syst.* **2013**, *35*, 249–283, doi:10.1007/s10115-012-0507-8. [[CrossRef](#)]
34. Settles, B. *Active Learning Literature Survey*; Computer Sciences Technical Report 1648; University of Wisconsin–Madison: Madison, WI, USA, 2009.
35. Aggarwal, C.C.; Kong, X.; Gu, Q.; Han, J.; Yu, P.S. *Active Learning: A Survey*. Technical Report; CRC Press: Boca Raton, FL, USA, 2014.
36. Chen, Y.; Mani, S.; Xu, H. Applying active learning to assertion classification of concepts in clinical text. *J. Biomed. Inform.* **2012**, *45*, 265–272, doi:10.1016/j.jbi.2011.11.003. [[CrossRef](#)] [[PubMed](#)]
37. Bloodgood, M.; Vijay-Shanker, K. A Method for Stopping Active Learning Based on Stabilizing Predictions and the Need for User-adjustable Stopping. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL '09), Boulder, CO, USA, 4–5 June 2009; Association for Computational Linguistics: Stroudsburg, PA, USA, 2009; pp. 39–47.
38. Zhu, J.; Wang, H.; Hovy, E. Multi-criteria-based Strategy to Stop Active Learning for Data Annotation. In Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08), Manchester, UK, 18–22 August 2008; Association for Computational Linguistics: Stroudsburg, PA, USA, 2008; Volume 1, pp. 1129–1136.
39. He, G.; Duan, Y.; Li, Y.; Qian, T.; He, J.; Jia, X. Active Learning for Multivariate Time Series Classification with Positive Unlabeled Data. In Proceedings of the 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), Vietri sul Mare, Italy, 9–11 November 2015; pp. 178–185, doi:10.1109/ICTAI.2015.38. [[CrossRef](#)]
40. Wei, L.; Keogh, E. Semi-supervised Time Series Classification. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06), Philadelphia, PA, USA, 20–23 August 2006; ACM: New York, NY, USA, 2006; pp. 748–753, doi:10.1145/1150402.1150498. [[CrossRef](#)]
41. Folgado, D.; Barandas, M.; Matias, R.; Martins, R.; Carvalho, M.; Gamboa, H. Time Alignment Measurement for Time Series. *Pattern Recognit.* **2018**, *81*, 268–279. [[CrossRef](#)]

42. Müller, M. *Dynamic Time Warping*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 69–84, doi:10.1007/978-3-540-74048-3_4.
43. Danka, T.; Horvath, P. modAL: A Modular Active Learning Framework for Python. Available online: <https://github.com/cosmic-cortex/modAL> (accessed on 26 November 2018).
44. Reyes-Ortiz, J.L.; Ghio, A.; Parra, X.; Anguita, D.; Cabestany, J.; Català, A. Human Activity and Motion Disorder Recognition: Towards smarter Interactive Cognitive Environments. In Proceedings of the ESANN 2013 proceedings 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 24–26 April 2013.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).