Human Activity Recognition From Sensorised Patient´s Data in Healthcare: A Streaming Deep Learning-Based Approach

Sandro Hurtado[1]*, José García-Nieto[1], Anton Popov[2], Ismael Navas-Delgado[1]

[1] Institute for Software Technologies and Software Engineering (ITIS), Biomedical Research Institute of Málaga (IBIMA), Department of Computer Languages and Computing Sciences, University of Málaga ETSI Informática, Campus de Teatinos, Málaga, 29071 (Spain)
[2] Department of Electronic Engineering, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv (Ukraine)

Received 4 June 2021 | Accepted 25 March 2022 | Early Access 6 May 2022

Abstract

Physical inactivity is one of the main risk factors for mortality, and its relationship with the main chronic diseases has experienced intensive medical research. A well-known method for assessing people's activity is the use of accelerometers implanted in wearables and mobile phones. However, a series of main critical issues arise in the healthcare context related to the limited amount of available labelled data to build a classification model. Moreover, the discrimination ability of activities is often challenging to capture since the variety of movement patterns in a particular group of patients (e.g. obesity or geriatric patients) is limited over time. Consequently, the proposed work presents a novel approach for Human Activity Recognition (HAR) in healthcare to avoid this problem. This proposal is based on semi-supervised classification with Encoder-Decoder Convolutional Neural Networks (CNNs) using a combination strategy of public labelled and private unlabelled raw sensor data. In this sense, the model will be able to take advantage of the large amount of unlabelled data available by extracting relevant characteristics in these data, which will increase the knowledge in the innermost layers. Hence, the trained model can generalize well when used in real-world use cases. Additionally, real-time patient monitoring is provided by Apache Spark streaming processing with sliding windows. For testing purposes, a real-world case study is conducted with a group of overweight patients in the healthcare system of Andalusia (Spain), classifying close to 30 TBs of accelerometer sensor-based data. The proposed HAR streaming deep-learning approach properly classifies movement patterns in real-time conditions, crucial for long-term daily patient monitoring.

Keywords

Deep Learning, Healthcare, Human Activity Recognition, Semisupervised Learning, Spark Streaming Processing.

DOI: 10.9781/ijimai.2022.05.004

I. Introduction

Physical inactivity is one of the main risk factors for chronic diseases such as cardiovascular, cancer and diabetes [1], [2]. Knowing the habits and types of activity carried out by people and their relationship with these diseases is a key task to design treatment strategies and prevention recommendations. Numerous advances in Human Activity Recognition (HAR) has been crucial to deepen in high-level knowledge about people's daily life [3]. One of the main objectives of HAR is to provide long-term monitoring of people's daily activities to allow medical doctors to get additional information of their patients to design care plans that may prevent or help against chronic diseases.

HAR has gained much attention in healthcare due to its wide range of applications, such as monitoring of geriatric patients specially focused on fall detection [4]–[6], as well as many other studies related to chronic diseases such as Parkinson, obesity, cardiovascular and neurodegenerative diseases [7]–[10]. These research activities have shown that HAR can effectively improve the quality of health care for some groups of people suffering from some pathologies or chronic diseases.

HAR mainly focus on two types of methods: video-based and sensor-based. Video-based methods provide a dense feature space to allow fine-grained analysis in HAR. However, it is exposed with a high complex background of images, since an environment with very strict conditions, such as well-positioned cameras and individuals, is required for data collection process with a high cost at the level of computing resources, energy consumption and price. Therefore, video-based methods remain limited in epidemiological studies where the evaluation of daily physical activity requires a reliable, accurate, and low-cost methodology. Sensor-based methods are widely used in scientific physical activity studies since they provide better adaptability in variable environments, high recognition accuracy and low power consumption. Furthermore, in [3] the use of accelerometers is exposed as the most used sensor in the literature since most wearable devices are equipped with them and have easy access. Additionally, the use of accelerometer is considered a reasonably competent sensor for

* Corresponding author.
E-mail address: sandrohr@uma.es

Please cite this article in press as:
S. Hurtado, J. García-Nieto, A. Popov, I. Navas-Delgado. Human Activity Recognition from Sensorised Patient´s Data in Healthcare: A Streaming Deep Learning-Based Approach, International Journal of Interactive Multimedia and Artificial Intelligence, (2022), http://dx.doi.org/10.9781/ijimai.2022.05.004

recognising of many types of activities since most of them are simple body movements. This work is motivated by an ongoing collaboration project in the context of a real-world healthcare system (in Andalusia, Spain). We focus on a sensor-based approach, with the main propose of discriminating basic posture change movements or activities of a group of patients with obesity and cardiovascular problems. The goal of the project is to provide tools to practitioners to follow the daily routine of their patients and thus prevent sedentary lifestyle. In this sense, many related studies in the literature have reported high classification accuracy [11]–[14]. However, most of them have been tested in academic datasets on young, healthy subjects, that can hardly resemble the conditions of a real patient's environment. Besides, most of these experiments have been carried out under controlled environments, where activity conditions are restricted.

However, as observed in actual healthcare scenarios, a series of critical issues arise related to the limited amount of available labelled data to build a classification model regarding to the total volume and velocity of sensorised data. In addition, the discrimination ability of features is often difficult to capture for different classes, since the variety of movement patterns in certain group of patients, e.g. obesity and/or geriatric patients, is bounded and maintained over time. Another issue is the usual class imbalance of data registered in this kind of sensor data streams. Due to samples representing specific constant postures, such as sleeping, sitting, active, inactive, etc., are perceptually abundant, compared other ones (running, up-stairs, etc.). Therefore, these challenges demand the design and development of hybrid data-driven approaches, where semi-supervised models can act at the core of data processing workflows, usually involving modern Big Data technologies.

In this study, a streaming classification model for Human Activity Recognition in healthcare systems, is proposed for patient monitoring in real-time. This proposal is based on a combination strategy of public labelled/private unlabelled raw data integration, semi-supervised classification with Convolutional Neural Networks (CNNs) and Spark streaming processing.

Guided by practical requirements, accelerometer sensor-based data have been considered in this work since low power consumption and use of resources are mandatory through long-term daily patient monitoring in uncontrolled environments. In this sense, as sensorised samples are mostly unlabelled, a data fusion task is conducted with commonly used datasets in the literature (WISDM [15], PAMAP2 [16], HUGADB [17] and USC-HAD [18]). These datasets have been previously labelled according to systematic procedures and share common attributes. This way, labelled and unlabelled samples are integrated for feeding the semi-supervised models to classify new incoming flows of data, through Spark streaming processing engine, by following a sliding window strategy.

In this approach, semi-supervised models are generated with Encoder-Decoder CNNs [19], which allows data augmentation by considering unlabelled samples and statistical features, hence embracing the global properties of the accelerometer time series. For testing purposes, a real-world case study is conducted with a group of more than 300 overweight patients in the healthcare system of Andalusia (Spain), classifying close to 30 TBs of accelerometer sensor-based data.

The main contributions of this study are summarised as follows:

- A streaming semi-supervised HAR strategy is proposed for monitoring overweight patients in the context of a real-world healthcare system, involving a data fusion task of accelerometer sensorised data from labelled/unlabelled samples.
- Thorough experimentation is conducted for model selection and validation, where a semi-supervised CNN-Encoder-Decoder is evaluated with varying amounts of unlabelled data.
- The resulting analysis workflow is deployed on a cluster of Spark nodes, so the continuous classification of 30 TBs sensor data is predicted for a group of patients. The proposed HAR streaming deep-learning approach properly classifies movement patterns in real-time conditions, which is crucial for long-term daily patient monitoring.

The remainder of this paper is structured as follows. Section II presents a review of related studies in the current state of the art. In Section III, methodology and approach are described. The experimental procedure is explained and results are analysed in Section IV. Finally, Section V contains concluding remarks and future work.

II. Related Work

The discovery of patterns of human activity has led to several studies on how to analyze the data collected through activity bracelets, smartwatches and smartphones [20]. Many classification methods have been used in previous studies, especially conventional approaches using Machine Learning algorithms [21] such as Extra Trees, AdaBoost, Random Forest (RF), Naive Bayes, k-nearest Neighbours (kNN) or Support Vector Machines (SVM). To name some representative studies of them, in [22] SVM was used to carry out the classification problem of HAR, collecting inertial sensor data through a smartphone mounted in the waist of the individuals. C4.5 Decision Tree and Naive Bayes classifiers were used to recognize 20 daily activities in [23]. In [24] kNN was declared the best classifier in comparison with C4.5 (J48) Decision Tree, Multilayer Perceptron Neural Network, Naive Bayes, logistic regression, and ensembles based on boosting and bagging. However, they still showed classification failures in similar activities.

Even when conventional approaches have obtained promising results with high-level classification accuracies in different controlled environments, these methods rely on feature-based classification guided by human domain knowledge, which supposes a heavily effort in the pre-processing data stage. Besides, the discrimination of very similar activities for these methods is still a difficult task. Deep Learning (DL) algorithms seem to be a good solution to overcome these problems since they conduct layer-by-layer structural modelling for specific feature extraction and allow the classification process after the segment pre-processing of raw data. One of the first approaches can be found in [25], where HAR classification is carried out with CNNs by extracting features without any domain-specific knowledge about raw-data. Also in [11], Convnets is proposed to perform efficient and effective HAR using smartphone sensors by exploiting the inherent characteristics of activities and 1D time-series signals, at the same time providing a way to automatically and adaptively extract robust features from raw data. Various state-of-the-art classification techniques under different scenarios are compared in [12], showing how deep neural networks perform with the best accuracy when the training data volume is drastically reduced.

Many other HAR studies have been implemented with deep learning methods, such as convolutional and recurrent approaches [9], [13], [14], [26]. In this sense, a thorough survey is reported in [3] where new challenges and trends are identified for this area. In concrete, two of these main challenges are related to the online/streaming processing or sensorised data, and the requirement of dealing with unlabelled data. These are, in fact, the direct consequence of working in real-world environments, requiring the management of high volumes of continuously sensorised data. Recent proposals [19], [27] are based on suitable semi-supervised frameworks to cope with these issues, although they are still limited when tackling with scalable data processing.

Moreover, in order to alleviate some of the drawbacks encountered in the literature, we have made an exhaustive study of general features

in the existing methods, as exposed in [3], [20], [28]–[31]. We have distinguished four main challenges pertaining to human activity recognition. These features are presented below:

- *Design issues*:
  1. *Cost*: Cost is a key factor for any technique. If accuracy of a solution is good but cost is too high, then it is of no practical use. Accelerometers are inexpensive, require relatively low power, and are embedded in most of today's cellular phones [32].
  2. *Obtrusiveness*: To be successful in practice, HAR systems should not require the user to wear many sensors nor interact too often with the application. There are systems which require the user to wear four or more accelerometers or carry a heavy rucksack with recording devices. These configurations may be uncomfortable, invasive, expensive, and hence not suitable for activity recognition.
  3. *Energy consumption*: extending the battery life is a desirable feature, especially for medical applications that are compelled to deliver critical information (Long term monitoring).
  4. *Sampling rate (frequency)*: low sampling frequencies tend to lose information in specific movements.
- *Data collection protocol drawbacks:*
  5. *Real world environments (No controlled environment)*: The procedure followed by the individuals while collecting data is critical in any HAR. In [33] demonstrated 95.6% of accuracy for ambulation activities in a controlled data collection experiment, but in a natural environment (i.e., outside of the laboratory), the accuracy dropped to 66%!
  6. *Large volume of data*: A comprehensive study should consider a large number of individuals.
  7. *Long term patient monitoring*: most studies do not offer patient monitoring over time, which is essential to improve the problem of HAR.
  8. *Data collection Flexibility*: people perform activities in a different manner which means that an acceptable number of subjects is needed for the study so that the trained model is flexible enough to work with other subjects.
- *Model selection drawbacks:*
  9. *Semi-supervised learning*: Typically, HAR systems rely on large amount of labelled training data. However, annotating data can be challenging in some situations, especially when the granularity of the activities is great or the user is unwilling to help with the gathering process. Using semi-supervised learning, these unlabelled data can still be used to train a recognition model.
  10. *Deep Learning*: Deep learning algorithms attempt to learn high-level features from data in an incremental manner. Nevertheless, in classical machine learning, domain experts must extract features from raw sensor data in order to make the patterns more visible for the learning algorithm.
- *Model evaluation drawbacks:*
  11. *Model generalisation*: People certainly perform activities in a different manner due to particular physical characteristics. We have proposed to evaluate activity recognition systems based on the subjects rather than of the segmented windows. This prevents over-fitting on the subjects and helps to achieve better generalisation results.
  12. *Latency*: Latency is a critical factor. If a solution is accurate but takes long time to provide the results, it is not practical.
  13. *Real time classification/real-time decision making*: This is important for human activity recognition because getting the results in real time is a compulsion in many situations.

Table I shows a comparison between our approach and a set of related works found in the literature of HAR in this section, according to the list criteria exposed above. As can be observed, desirable features related to real-world environments as real-time processing of the sensorised data, dealing with unlabelled data and managing of high volumes of continuously sensorised data are covered by our approach, which represent an advantage with regards to these compared works.

The proposed approach is conceived to cope with these limitations by combining semi-supervised Encoder-Decoder CNN dynamic models with Spark streaming processing in the context of real-world healthcare environments.

TABLE I. Comparison of Related Works Found in the Literature in Human Activity Recognition. The Comparison Has Been Made According to Four Main Challenges Encountered in the State of the Art Pertaining to Human Activity Recognition. Additionally, Our Streaming *Semi-Supervised Deep-Learning Approach* (*SSSDA*) Is Presented in This Table as *SSSDA*. It Worth to Note That Our Approach Represent an Advantage With Regards to these Compared Works in Terms of Real-Time Classification in Real-World Environments.

Features/HAR refs	[22]	[23]	[24]	[25]	[11]	[9]	[13]	[14]	[26]	[19]	[27]	*SSSDA*
1. *Cost*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. *Obtrusiveness*	✓	×	✓	×	✓	✓	✓	✓	✓	×	✓	✓
3. *Energy*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4. *Sampling-rate*	×	×	-	✓	×	×	×	×	✓	✓	≈✓	✓
5. *Real-environment*	×	≈✓	×	✓	-	≈✓	×	✓	×	×	✓	✓
6. *Large data-volume*	×	×	×	✓	×	×	×	-	×	×	×	✓
7. *Long-monitoring*	×	×	×	≈✓	×	×	×	×	×	×	×	✓
8. *Data-flexibility*	×	×	×	✓	×	×	×	≈✓	×	-	×	✓
9. *Semi-supervised*	×	×	×	×	×	≈✓	×	×	×	✓	✓	✓
10. *Deep-Learning*	×	×	×	✓	✓	✓	×	✓	✓	✓	✓	✓
11. *Model-generalisation*	×	✓	×	×	✓	✓	×	×	×	×	×	✓
12. *Latency*	-	✓	-	-	×	≈✓	×	×	×	-	×	✓
13. *Real-time classify*	-	✓	×	≈×	×	≈✓	×	×	×	×	×	✓

III. Methodology and Approach

This section is devoted to present the data acquisition strategy and data pre-processing tasks conducted for data consolidation. The semi-supervised deep learning model used is also described. After this, the overall approach is detailed to illustrate how all the elements are integrated.

A. Data Acquisition Strategy

In this work, we have followed a combined strategy for data acquisition that consists in merging private patient's sensor data and academic datasets. The former source comprises data streams of unlabelled attributes (patients' movements) that have to be classified. The latter one considers a series of labelled datasets from related studies of human activity recognition time-series in the literature. The main purpose of this strategy is to generate an enriched dataset that, after a feature engineering process for data fusion, is suitable for feeding semi-supervised models, avoiding bias and overfitting problems as much as possible.

Sensor data are generated using GENEActiv[1] wearable devices, which incorporate a MEMS triaxial accelerometer placed on the non-dominant wrist of the study subjects.

Each measurement of this bracelet contains three real values on each of the sensor axes ('x-y-z') with a sampling rate at 100Hz, range of +/- 8g and resolution of 12 bits. In this way, after a weekly observation period, a total amount of 30 TBs of raw movement data was collected from 300 patients' daily activities. This final time series dataset is a set of observations $X = (x_t^1, x_t^2 \ldots x_t^L)$ where each one is recorded at a specific time T and L as a length of time-step.

Nevertheless, as commented before, sensorised data still lacks class labelled features, which are required for model training. Therefore, a series of widely used datasets in the literature have been considered in the proposed approach, each one of them contributing with labelled samples for different, sometimes overlapping, activities. These datasets are: WISDM (Actitracker) [34], PAMAP2 [35], USC-HAD [36] and HuGaDB [37]. These datasets were previously labelled according to systematic procedures and sharing common attributes. The time-series recorded in these datasets have been collected from heterogeneous devices (smartphones and bracelets) located in different parts of the body, considering a different number of individuals and with a different sampling frequency (e.g. WISDM at 20Hz, HUGADB at 50Hz, USC-HAD and PAMAP2 at 100Hz) in the study. Moreover, they have been modelled to consider different sets of daily activities, which are recorded through different time intervals.

Therefore, a thorough pre-processing phase has been carried out to homogenise all these data sources, including those commonly detected activities among all the individuals in observation. In concrete, these shared activities are: running, walking, sitting, standing, up stairs and down stairs, which are used as labelled categories for the semi-supervised models worked in this proposal.

B. Data Pre-Processing

Besides, raw data have been normalised through Z-score Normalisation. Feature standardisation makes the values of each feature in the raw data have zero-mean and unit-variance. This normalisation is formulated in (1), where x is the original feature vector, x' is the normalised value, $\tilde{x} = average(x)$ is the mean of that feature vector, and σ is its standard deviation.

$$x' = \frac{x - \tilde{x}}{\sigma} \quad (1)$$

Also, linear interpolation have been conducted to tackle with missing values and to fill gaps in raw data time series. This method searches for a straight line that passes through the end points x_A and x_B, as formulated in (2), where xi are observed data, X_i are the interpolated value(s) of missing data and α is the interpolation factor that varies from 0 to 1.

$$X_i = (1 - \alpha)x_B + \alpha x_A \quad (2)$$

However, the most relevant task in this regard has been re-sampling data. In particular, down-sampling and up-sampling are performed on data, since when dealing with "waves" in time-series, it is observed that low sampling frequencies tend to lose information in specific movements, where a high frequency is required to identify them correctly. For this reason, we must determine the wave frequency according to the type of recognition faced. Fig. 1 shows an example where raw data of a patient's activities ("walking" and "cycling") are collected by an accelerometer sensor on a wrist. After re-sampling, data are transformed for each activity at frequencies of 100Hz (top), 50Hz (middle) and 20Hz (bottom). The effect of this re-sampling is illustrated and it is possible to identify some losses in the data information as long as the frequency is decreasing. It can be observed in Fig. 1 a), where different waves peaks "disappear" provoking inconsistent data representations at different sampling frequencies. Therefore, a high re-sampling (100Hz) is performed to keep informative level in samples, while making data homogeneous for all the sources.

Another quite common, yet important, issue registered in HAR datasets is the class imbalance. Even more in real-world sensor data from the particular case of obesity patients, where the balance between classes is not guaranteed and biased to sedentary activities. For example, the "sitting" activity is more frequent in the case of overweight patients than the "running" activity, producing an important class imbalance that could lead learning models to behave with a bias towards majority classes. As a consequence, algorithms will fail in the classification of the underrepresented minority classes, which provokes a severe decreasing in the overall accuracy of the results [38].

In order to cope with class imbalance, several approaches have been used such as oversampling and under-sampling methods at the data level and many other solutions at the algorithmic level trying to trade-off the class imbalance in modelling time . In the context of HAR, Synthetic Minority Oversampling Technique (SMOTE) is a common over-sampling method used to generate new synthetic data of the minority classes. It has shown a great deal of success in several applications where SMOTE helps to enhance the classification accuracy for imbalanced datasets. For example, in data balancing was used through SMOTE oversampling approach, leading the worked model to reach high accuracy results.

By default, SMOTE re-samples all classes excepting the majority class, that is, the minority classes are increased to reach the total number of the majority class. However, the study in [39] suggested combining SMOTE with random under-sampling of the majority class, since a high over-sampling could provoke model over-fitting. For this propose, our methodology addresses class imbalance at training stage by balancing classes in two separate steps: firstly, SMOTE oversampling technique is used to over-sample those minority classes to have 50% of the number of examples of the majority class. Then, under-sampling using random elimination is performed on the majority classes, to have 20% more than the minority class. Then a difference of 20% between classes of samples is obtained, which helps the model to avoid problematic class imbalance, preventing the generation of synthetic data in a high percentage.

[1] https://www.activinsights.com/products/geneactiv/

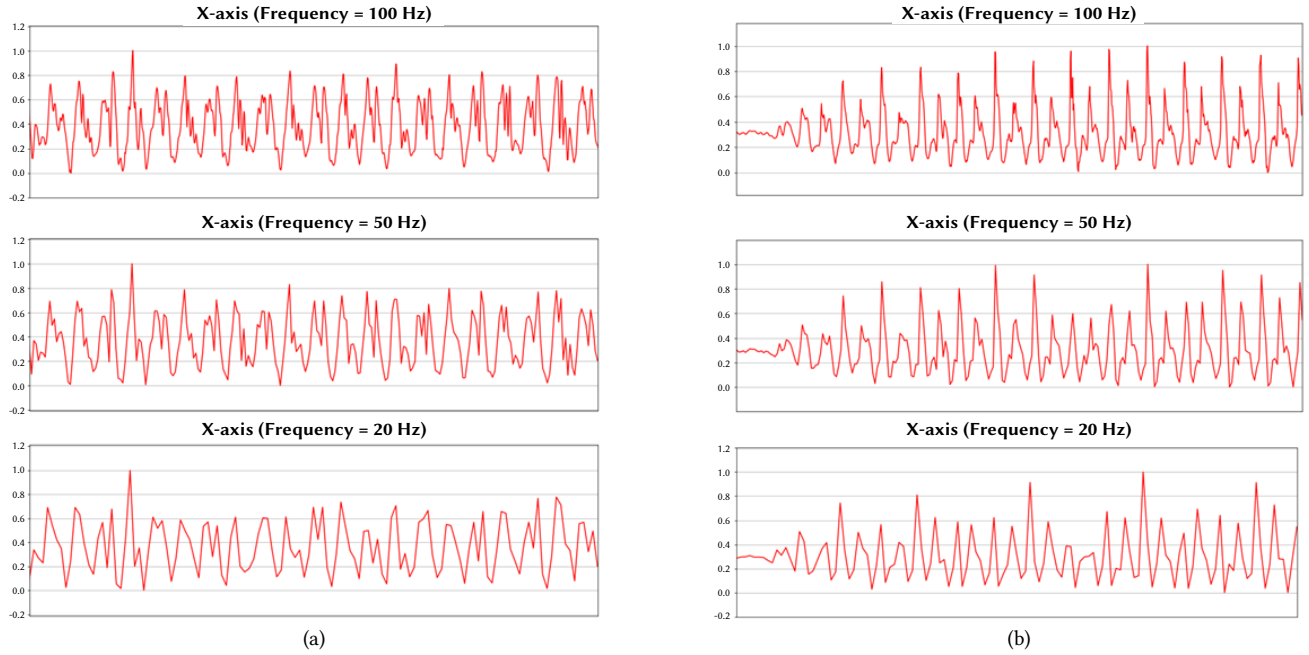


Fig. 1. Raw data from accelerometer sensor of different activities: Walking (a) and cycling (b) at 100 Hz (top) and resampled data at 50 Hz (middle) and 20 Hz (bottom). It can be noticed that as the sampling rate decreases, aspects at high frequency are removed from the wave.

C. Semi-Supervised Modelling

One of the main challenges arising in this study is the possibility of taking advantage of dealing with labelled and unlabelled data. In this sense, the use of semi-supervised learning techniques constitutes a suitable option to perform predictive analysis, since they allow to train models with, labelled and unlabelled samples, which mainly improve generalisation and avoid over fitting [19].

In particular, the use of CNN based approaches has been shown to perform successfully for HAR, since they provide hidden representations of data and to identify patterns in activity time-series [25], [27]. Therefore, considering a dataset with N pairs $(x_1, t_1), (x_2, t_2), ..., (x_N, t_N)$, being x_i a sliding window input with length T and t_i the label representing a given activity, we adopt a similar semi-supervised strategy to a CNN Encoder-Decoder [27] in our approach. In this, labelled samples $\{(x_i, t_i) | \ 1 \le i \le N\}$ are used together with unlabelled ones $\{x_i \,|\, N+1 \le i \le N+M\}$ in training, to fit the model with both data sources (sensorised and academic).

In general, the encoder network maps a given input signal $x \in X \subset \mathbb{R}^{d_0}$ to a feature space $z \in Z \subset \mathbb{R}^{d_k}$, whereas the decoder takes this feature map as an input, process it and produce an output $y \in Y \subset \mathbb{R}^{d_L}$

The rationale behind the CNN Encoder-Decoder for semi-supervised classification is to include noise into all the layers of the network, so it works to minimise the distance between the clean input and the reconstructed decoder one. In this way, the learning procedure can be summarised in the following steps:

1. Labelled and unlabelled data are processed by the clean encoder to compute hidden variables in the middle layers z_i^k;
2. Both labelled and unlabelled data are corrupted with Gaussian noise and transformed to an abstract representation $\tilde{z}_i^k$, by the noisy encoder;
3. Labelled data $(\tilde{x}_i, 1 \le i \le N)$ are used to perform the prediction task on a softmax based on cross entropy cost. The predicted classes are denoted with $\tilde{y}_i$;
4. The decoder works to reconstruct unlabelled samples $(\tilde{x}_i, N+1 \le i \le N+M)$ which are denoted with $\hat{x}_i$, so they should be as close as possible to the corresponding input (x_i). To measure this similarity, square error is computed.

The cost function is formulated in (3) as an aggregation of the supervised cross entropy of the noisy output $\tilde{y}_i$ predicting the class activity t_i for the input x_i (first term in this equation), whereas the unsupervised cost (second term in this equation) is the denoising square error between clean input x_i and their noisy reconstruction output $\hat{x}_i$.

$$Cost = -\frac{1}{N}\sum_{i=1}^{N} \log P(\tilde{y}_i = t_i | x_i) + \frac{\lambda}{N}\sum_{i=N+1}^{N+M} || \hat{x}_i - x_i ||_2^2 \tag{3}$$

Therefore, the semi-supervised CNN Encoder-decoder allows unlabelled samples from sensor streaming sources to take part in the learning model in training time, so it will avoid bias to certain classes and promote generality.

D. Overall Approach

A general overview of the proposed approach is illustrated in Fig. 2, where all the elements are organised, from data acquisition to model evaluation and human activity prediction. It partially follows the so-called activity recognition chain (ARC), extensively studied in [44] as a general-purpose framework for processing time-series sensorised data, training and evaluating HAR workflows. These steps are thoroughly described next:

1. *Data acquisition.* As commented before, we have followed a combined strategy of self data collection from sensors together with public datasets, with the aim of feeding a semi-supervised model with unlabelled and labelled samples, respectively. Nevertheless, public datasets have been generated with different devices and human conditions, sometimes far from the habits observed in our patients (with obesity), so a preliminary exploration phase has been conducted to select that public dataset containing distributions more similar to our self-collected (private) data. In this regard, Fig. 3 shows the boxplot distributions of the three accelerometer axis (x,y,z) for each of the four considered public datasets (WISDM, PAMAP2, USC-HAD and HuGaDB), taking into account the 6 activities which have in common these datasets (walking, running, sitting, standing, downstairs, upstairs), as well as for our private data. After this process, the WISDM dataset is selected to provide our model with labelled samples, since it contains in overall the

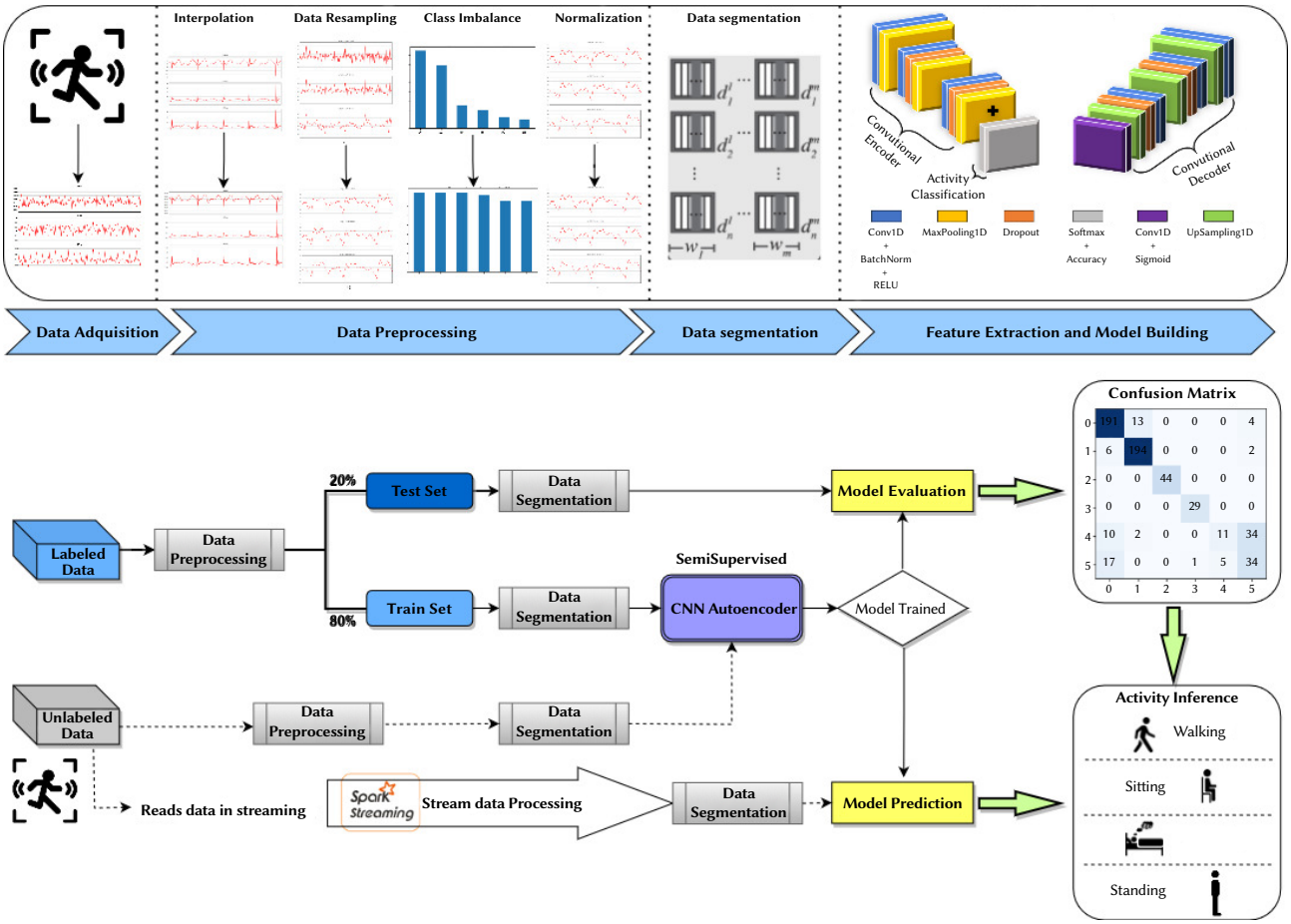


Fig. 2. General overview of the proposed approach that is presented as a HAR workflow. This workflow is composed of several steps: (1) **Data acquisition**: the data is acquire combining unlabelled data sensors (private dataset) and from public datasets. (2) **Data pre-processing**: these data is pre-process, which involves interpolation for missing data imputation, re- sampling, class imbalance processing and normalisation. Also labelled dataset is then split into two subsets with 80% of selected samples for training and 20% of remaining ones for testing. (3) **Data segmentation**: a temporal sliding window with size of 400, corresponding to roughly 4 seconds of physical activity data, and overlap of 100 (1 second) is performed to labelled and unlabelled data. (4) **Feature extraction and model training**: a CNN Encoder-Decoder model is trained with labelled and unlabelled, capturing the most relevant characteristics of the training data in order to provide activity inference of the 30TB of unlabelled data. (5) **Model evaluation**: the model is evaluated with the test sets where confusion matrix and deviated metrics are obtained (Precision, Recall, F1-score) (6) **Streaming processing and activity recognition**: once the model is evaluated and provide us promising results an Spark Streaming classification process is carry out.

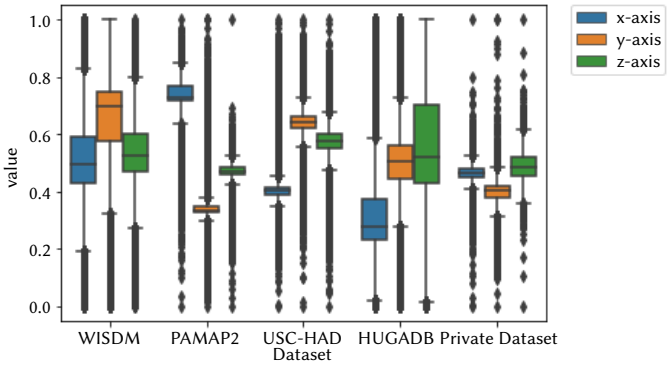


Fig. 3. Boxplot distributions of the three accelerometer axis corresponding to WISDM, PAMAP2, USC-HAD and HUGADB, taking into account the 6 activities which have in common these datasets (walking, running, sitting, standing, downstairs, upstairs). Also our private dataset was included in the bloxplot distribution.

closest axis distributions to the sensorised data of our patients. Therefore, we avoid the model to underfit with excessive data variation. When the instances are augmented using the WISDM dataset the model became more stable with smaller standard deviation. On the contrary, using all datasets together to train the model add additional variation and it deteriorates the model too much. In concrete, WISDM (Actitracker) dataset considers 6 activities registered in a controlled environment: jogging, walking, ascending stairs, descending stairs, sitting and standing. A number of 36 individuals have taken part in these measures.

2. *Data pre-processing*. A second step of data processing is performed (as explained before) on labelled and unlabelled data, which involves interpolation for missing data imputation, re-sampling, class imbalance processing and normalisation. It is worth to note, we re-sampling WISDM dataset from 20Hz to 100Hz (same frequency of our private dataset) in order to keep data information as commented before in Fig. 1. The labelled dataset is then split into two subsets with 80% of selected samples for training and 20% of remaining ones for testing.
3. *Segmentation*. At this step, data samples are still structured in the time domain, since all the axis points are collected at a certain time instant from sensors. Therefore, a segmentation stage is required to transform these input data into the frequency domain, more suitable for training deep learning models as signal processing prediction tasks. To do so, for each axis attribute in the dataset, a temporal sliding window with size of 400, corresponding to

roughly 4 seconds of physical activity data, and overlap of 100 (1 second), is performed. This overlapping among windows guarantees high numerosity of training and testing samples to train the model. To match the input shape of the CNN-Encoder-Decoder, it is necessary to reshape the sample obtained in the previous step. Therefore, each window comes in the form of a matrix of values, of shape N x 400 x 3, where N is the number of samples resulting of the segmentation, 400 is the time window and 3 is the number of features to train the model (x-axis, y-axis, z-axis). In this segmentation, sliding windows are checked to contain samples from just one human activity.

4. *Feature extraction and model training.* This step entails the semi-supervised learning task, which merges the labelled segments in the training set with those unlabelled from sensors. The CNN Encoder-Decoder involves up-sampling for maxpooling decoding, as well as convolutional operation for deconvolution [27]. As argued in [27], using this semi-supervised CNN Encoder-Decoder, it is possible to learn the network and features simultaneously from the data.
5. *Model evaluation.* Once the model is built, an evaluation step is carried out with regards to the test set, where confusion matrix and deviated metrics are obtained (Precision, Recall, F1-score, etc). It is worth noting that this test set is completely obtained from the public dataset (in this case WISDM), although the model has been trained with both, public and private data, so final predictions are expected to show certain model generalisation with moderate accuracies. The final goal is to get a prediction model suitable for a very dynamic data flow environment, but not for a specific dataset in a certain time period.
6. *Streaming processing and activity recognition.* Finally, a streaming processing task is deployed through an Apache Spark environment, in which new sensorised data are pre-processed to be predicted according to the model previously built. An internal segmentation step is carried out with streaming data by using a similar sliding window size as used in model training phase. This is then a continuous process of human activity label assignation of new samples regarding patient's movements, which can be now monitored by practitioners.

The whole process is repeated with a certain frequency to rebuild models with updated data. Therefore, the framework to monitor patient's movements will consider new individuals in a transparent way to the learning model, since new sensor data will be in the same Spark streaming source.

IV. Experimental Results and Analysis

In this section, we investigate the effects of training a semi-supervised CNN Encoder-Decoder using labelled data from one public dataset (WISDM) and unlabelled data from our private dataset.

The goal is to be able to classify the 30 TB of unlabelled data. The Convolutional Encoder will compress the input signal x into a space of latent variables ($h = f(x)$) , then learning how to reconstruct the data back from the reduced encoded representation. Meanwhile, the Convolutional Decoder works to reconstruct the input signal based on the information previously collected ($r = g(h)$), as observed in Fig. 4. Therefore, the latent variable space h will capture the most relevant characteristics of the training data.

In this regard, the algorithm learns how to reconstruct the input by using the Adam optimiser [45] and using the mean square error as a loss function. Therefore, the model will be able to extract more significant characteristics from the unlabelled data that will help us to make predictions.

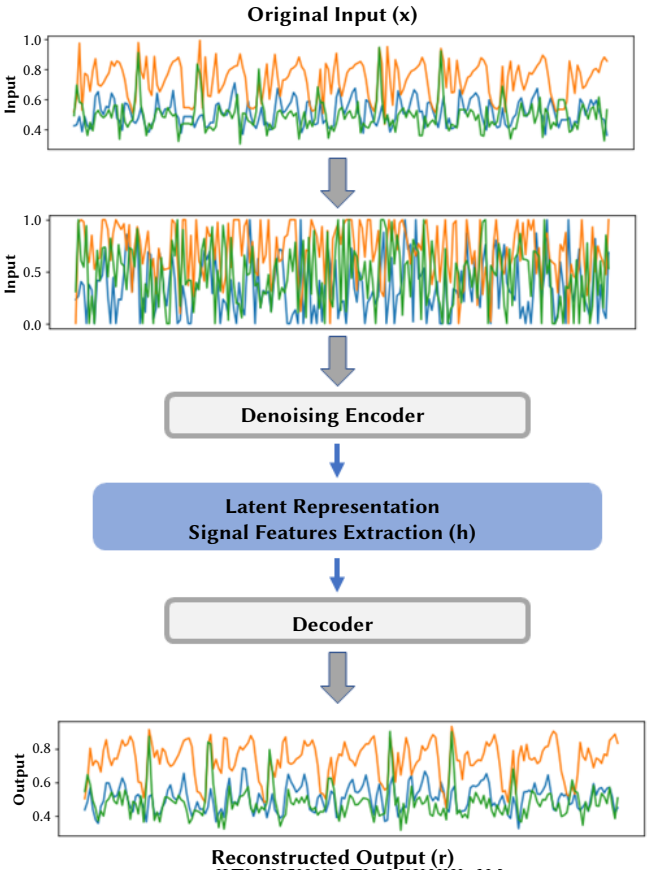


Fig. 4. General structure of the CNN Encoder-Decoder, contains a clean convolutional Encoder, noisy convolutional encoder, and a convolutional decoder. Labelled and unlabelled data are processed by clean convolutional encoder and then corrupted with Gaussian noise. Then the convolutional decoder works to reconstruct the clean inputpxq from high-level representation $r = g(h)$.

A. Model Selection

The full structure of our CNN-Encoder-Decoder model is shown in Fig. 2.

Encoder: The encoder network consists of three down-sampling blocks. Each down sampling block is composed of 1D convolutional layers with kernel size of 3, followed by a max-pooling layer. Additionally, for each block a batch normalisation is added to reduce internal co-variate shift [46], accelerating the training process of the model, and a dropout layer was added to improve generalisation performance and avoid over fitting. It then follows an structure [Conv1D + BatchNorm + MaxPooling1D + Dropout]

Decoder: Each encoder layer has a corresponding decoder layer. Thus, the decoder network consists of three up-sampling blocks composed of 1D convolutional layers with a kernel size of 3, followed by an up-sampling layer. As for the encoder, for each up-sampling block, batch normalisation and dropout layers were added, with a structure [Conv1D + BatchNorm + UpSampling1D + Dropout].

Bayesian optimisation has been used for efficient hyper-parameter tuning [47]. The hyper-parameters were tuned by performing 10-fold Stratified Shuffle Split cross-validation on the training set using Bayesian optimisation, obtaining a filter size of 64 for each of the 1D convolutional layers, which is activated by the Restricted Linear Unit (ReLU) function. Moreover, each of the max-pooling and up-sampling layers contains a pooling size of 2 and the dropout was set to 0.1 for each one. The Bayesian optimisation was executed with a batch size of 50, 500 and 1000, obtaining the best results with 50.

In order to assess the performance of our classification methodology system, we split the available dataset into 80% train data and 20% test data. This was done based on the subjects rather than of the segmented windows. In this regard, train data contain from subjects 1 to 32 of WISDM dataset and test data include the rest of the subjects (32 to 36). Thus, for each experiment four subjects out of 36 are always kept isolated to evaluate the model. This prevents over-fitting on the subjects and helps to achieve better generalisation results.

To comprehensively evaluate the model, we used several evaluation metrics to evaluate the classification results: accuracy, precision, recall, F1-score, loss function, receiver operating characteristic (ROC) and normalised discounted cumulative gain (NDCG), as shown in Table 2. It should be noted, we opted to estimate the mean F1-score (Fm-score), that is the mean F1-score across all the classes. It's shown in (4) and (5), where TP is the number of true positives in prediction, FP are the false positives and FN are the number of false negatives.

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN} \tag{4}$$

$$Fm - score = \frac{2 * precision * recall}{precision + recall} \tag{5}$$

The CNN Encoder-Decoder has been implemented in TensorFlow using Keras. The experiments to evaluate the model have been executed on a machine with 16 CPUs (Intel(R) Xeon ® Gold 6130 CPU 2.10GHz). After each epoch of training, we evaluate the performance of the model on the validation set. Each model is trained for at least 50 epochs. Training stop condition is configured if there is no increase in validation performance for 10 subsequent epochs. We select the epoch that showed the best validation-set performance and apply the corresponding model to the test set.

B. Sensitivity to Unlabelled Sample Size

In this section, we study the performance of our semi-supervised CNN Encoder-Decoder model trained with varying amounts of unlabelled data. The amount of the unlabelled data will be proportional to a percentage of samples of the labelled data used for training. Therefore, we evaluate the metrics of our model trained using unlabelled data of 10%, 20%, 30%, 50%, 80%, 100%, 150% proportion of labelled data used for training, as shown in Table II. The number of unlabelled samples varies from 97,814 (10% of train labelled data) to 1,467,222 (150% of train labelled data).

Fig. 5 shows how the Fm-score evolves when varying the number of unlabelled examples in the experimental results. Fm-score generally decreases when there are more unlabelled samples as expected. This is explained by the fact that unlabelled data comes from a different dataset then including variation. However, it can be observed in Fig. 5 that for percentages of unlabelled data less than 100%, we obtain a high Fm-score in the result.

TABLE II. Metrics Obtained With Varying Number of Unlabelled Examples in Training Set. The Amount of Unlabelled Data Is Taken As A percentage of the Training Set of the Labelled Data (WISDM Dataset). The Number of Unlabelled Samples Varies From 97,814 (10% of Train Data) to 1,467,222 (150% of Train Data)

Metrics: Public data (labelled) + Private data (Unlabelled)						
%	acc	loss	recall	Fm-score	roc	ndcg
0	0.981	0.069	0.981	0.981	0.998	0.998
10	0.976	0.075	0.977	0.967	0.995	0.997
20	0.971	0.076	0.949	0.949	0.992	0.993
30	0.951	0.148	0.940	0.938	0.991	0.990
50	0.947	0.151	0.925	0.926	0.990	0.988
80	0.905	0.292	0.905	0.903	0.987	0.985
100	0.875	0.319	0.872	0.871	0.983	0.984
150	0.685	0.601	0.685	0.655	0.941	0.981

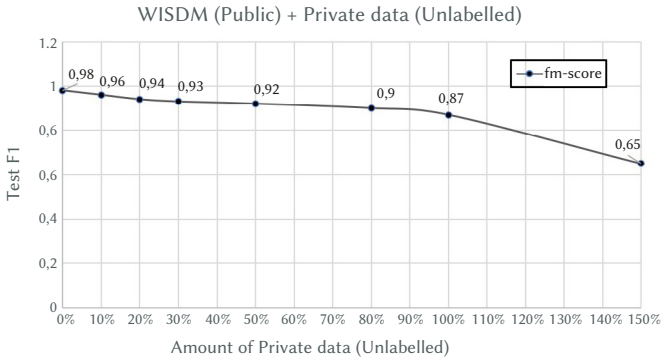
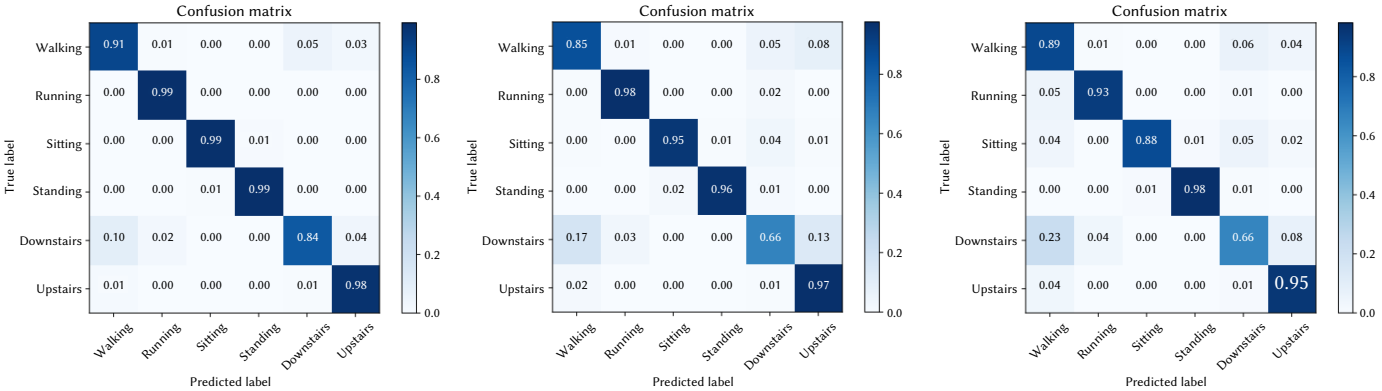


Fig. 5. Fm-scores obtained with varying number of unlabelled examples in training set.

Thus, our approach can potentially learn the network and features simultaneously from the data using unlabelled data in our CNN Encoder-Decoder model. Therefore, it is possible to use this model as core predictor. To do so, we have chosen the amount of 80% of unlabelled data to classify the 30 TB from sensors, since at this point, the model is still getting good results (Fm-score = 0.90).

More in depth, Fig. 6 shows the resulting confusion matrices when varying the amount of unlabelled data with 10%, 50% and 80% in the model training. It can be observed that the model achieves promising



(a) 10% of unlabelled data (b) 50% of unlabelled data (c) 80% of unlabelled data

Fig. 6. Illustration of confusion matrices showing the sensitivity of the networks for each individual class when varying 10%, 50% and 80% of unlabelled data when training the semi-supervised CNN-Encoder-Decoder.

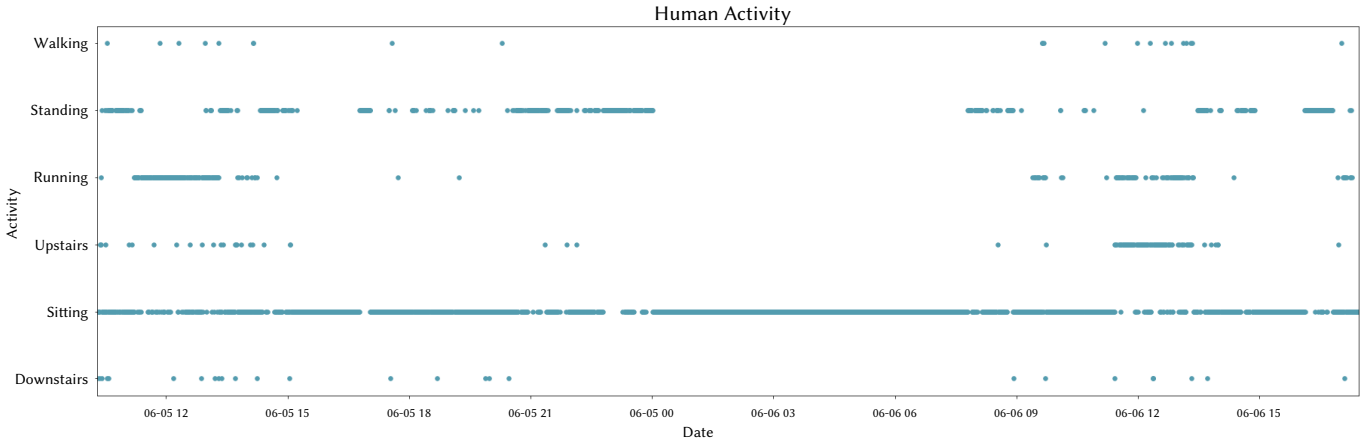


Fig. 7. Snapshot of the Human Activity Recognition for a randomly anonymous patient. It is shown how during the night sitting (resting) is the main activity, later around 8:30, the patient starts to be more active and does short movements. Then, at 12:00 the patient seems to start some moderate activity and finally, after 00:00 resting is the main activity.

predictions for activities walking, running, sitting, standing and upstairs even when increasing the number of unlabelled samples. In contrast, the model start to show limited predictions in detecting downstairs, since, if we see the patterns between walking and downstairs, they are characterised with very close signal shapes in movements, as mentioned in [15]. This is general an acceptable precision, since even for 80% unlabelled data it still gets good predictions for all classes.

As we know, it is hard to assess performance in unlabelled data, but we still need to know if it passes *"the eye test"*. For this propose, we classify a randomly chosen sample of unlabelled data in order to demonstrate that the distributions of the predictions are reasonable. It is shown in Fig. 7 (format date is month-day hour) how the main activity is resting (sitting and standing) as we expected. It's normal since this unlabelled data correspond to one of the 300 overweight patients in the healthcare system of Andalusia. In the same way, during the night (from 00:00 to 08:30 approximately) the patient is totally resting (sitting) . Later, the patient is standing and starts to be more active. Then around 12:00, the patient seems to starts to do moderate physical activity (running and upstairs). It can be seen that on both days at 12:00 (06-05 12:00 and 06-06 12:00) the patient carries out physical activity. This could be explained by the fact that patients follow the doctors' instructions doing daily exercise to avoid sedentary life. Afterwards, the patient does some short movements and finally, after 00:00 resting is the main activity.

It should be note, the classification has been carried out according to the labels that we have from the WISDM dataset, however our private dataset provide us a long-term monitoring of patient's daily activities where we can find more activities and transitions between activities. Even so, the results obtained in Fig. 7 seem quite reasonable to us for this first approach in which we try to address the problem of HAR in a real world case without previously labelled activities in our dataset.

C. Additional Experiments

Additional experiments have been implemented to demonstrate the feasibility of the proposed semi-supervised methodology. A first experiment was carried out to see whether the model was able to pass *"the eye test"* without taking into account the semi-supervised approach. In consequence, the model was trained only with raw data from WISDM dataset. After that, a classification task was performed from a randomly chosen sample from our 30TB private unlabelled dataset. As expected, the model didn't pass *"the eye test"* without using unlabelled private data in the training phase (Fig. 11).

Moreover, the proposed methodology has been synthetically evaluated by using another public dataset as a simulation of the unsupervised portion. In this sense, HUGADB dataset has been considered as *"unlabelled dataset"* and WISDM as labelled dataset. HUGADB dataset was classified with and without considering our proposed semi-supervised methodology. Finally, the model was evaluated if it can predict the activities in HUGADB dataset. In this experiment, we concluded that using the semi-supervised approach give us better predictions as observed in Table IV in Appendix. The same experiment was carried out with PAMAP2 as *"unlabelled dataset"*. See Appendix for more details in the experiments.

D. Computational Performance

To carry out the streaming classification process, a deployment of the complete approach has been conducted on a virtualisation environment operating on an on-premise high-performance cluster computing platform. This infrastructure is located at the Ada Byron Research Center of the University of Malaga (Spain). It comprises several units of virtualisation that allows to visualise the performance of the cluster. Concretely, this platform has 10 virtual machines, each one with 16 cores (CPU 16 x 2.10 GHz), 128 GB RAM and 1 TB of virtual storage (adding up to 176 cores, 1408 GBs of memory and 10 TB HD storage). These virtual machines have been used with the role of Worker node (Apache Spark) to make the activity predictions. The Master node, which runs the Keras CNN Encoder-Decoder, is hosted in a different machine with 16 cores at 2.10 GHz, 128 GB RAM and 5,000 TB of virtual storage space. All these nodes use Linux 4.15.0-118-generic 64-bit distribution. The whole cluster uses Spark 3.0.1.

Additionally, an NFS distributed file system has been configured to be able to access the sensorised data from all the machines. The Master node will physically store the data (server), while the Worker nodes will behave as clients to access the data remotely. In this way, it is possible to perform the activity prediction in parallel from the different machines connected to the same network to access remote files as if they were local ones.

For the parallelisation of Spark streaming processes the classification of activities accessing a directory at the NFS distributed system. The data is passed in streaming from the repository. Each of the CSV files that are included in the directory will behave as a Spark streaming batch that will go through a segmentation process by time windows (400 rows corresponding to 4 seconds of monitoring activity) as observed in Fig. 2. Finally, the CNN Encoder-Decoder model trained will predict the activity of each batch in streaming. The results are saved in text files using the same name as the original CSV files (See Code Snippet 1).

TABLE III. Experimental Results Spark Streaming Computational Performance

	Running Time (seconds)				Speedup			Efficiency		
Batch Size	T_1	T_{40}	T_{80}	T_{160}	S_{40}	S_{80}	S_{160}	E_{40}	E_{80}	E_{160}
64 MB	28.10	6.29	7.15	7.08	4.46	3.93	3.96	11.16%	4.91%	2.47%
128 MB	69.17	4.71	4.03	4.22	14.68	17.16	16.39	36.71%	21.45%	10.24%
256 MB	124.65	5.74	10.44	10.94	21.72	11.92	11.39	54.29%	14.92%	7.12%
512 MB	244.28	5.85	34.34	34.05	41.76	7.11	7.17	104.39%	8.89%	4.48%
1 GB	462.75	8.18	124.56	115.21	56.57	3.72	4.02	141.48%	4.64%	2.51%

Code Snippet 1: Spark streaming segmentation and classification by batch

```
//Read csv in Streaming with Spark from directory
df = spark.readStream(directory)
//Load the CNN-Encoder-Decoder model
model = keras.load(model)

classify(batch, batch_id, model):
     // we set time window to 400 (4 seconds of activity)

     time_window = 400
     // raw data segmentation by time Window
     batch.map(lambda x,y: [raw_data],time_window)
     // group by time_window
     batch.reducebyKey(lambda x,y: x+y)
     // activity prediction of raw data
     batch.map(lambda r: model.predict(r))
     // save the result
     batch.saveAsTextFile(batch_id+ .txt)

// Streaming classification for each batch
df.foreachBatch(classify(batch, batch_id, model))
```

The performance of the proposed streaming solution has been evaluated through a series of experiments to measure the performance in terms of *Speedup (SN)* and the *Efficiency (EN)*. Thus we analyse the computational effort and the data management process. The standard formula of the *Speedup* calculates the ratio of *T1* over *TN*, where *T1* is the running time of the analysed algorithm in 1 processor and *TN* is the running time of the parallelised algorithm on *N* processing units (processors or cores), while the *Efficiency (EN)* is calculated as shown in (6).

$$SN = \frac{T1}{TN} \; EN = \frac{SN}{TN} * 100 \tag{6}$$

Table III shows the running time in seconds used by the Spark streaming classification approach running on 40, 80 and 160 cores with different batch sizes of raw data. This way, we have centred on file sizes of 64 MB, 128 MB, 256 MB, 512 MB and 1 GB, since they are the average size of CSV files that are in the 30 TB of data. In this sense, we measure the computational influence of using different number of cores with different batch size. This table also contains the corresponding Speedup and Efficiency values to the resulting times. As mentioned, the running time is reduced in relation to the increase in the number of cores used in the parallel model. The highest reduction in time is obtained when our approach is configured with 40 cores in parallel, for which the running time is reduced from 28.10 s to 6.29 s in the case of the smallest batch size (64 MB), and from 462.75 s to 8.18 s with the biggest batch size (1 GB) used in the experiments. Also, in terms of efficiency, the highest percentage, 141.48%, is reached with 40 cores with a batch size of 1 GB reaching the best efficiency. In contrast, it decreases as the number of resources gets larger. This behaviour was somewhat expected as the particular cluster configuration involves computing overheads due to virtualisation and network communications, so a trade-off setting is reached with less nodes, but stabilising from 80 nodes in advance. Considering the results, it is worth mentioning that both cluster configurations (80 and 160 cores) yield similar speedup and efficiency values, which indicates that the bottleneck is due to the parallel infrastructure, so increasing the number of cores do not compensate the synchronisation and communication costs.

Therefore, according to the results the best configuration to obtain the maximum performance in the streaming classification process with Spark, are observed when using the cluster resources with 40 cores and a batch size of 1 GB (Fig. 8). In this regard, we can consider our Spark streaming classification methodology as a real-time classification since we can classify 1 GB in 8.18s, that is approximately 12,000,000 of samples rows, what is equivalent to almost one week of daily patient activities monitoring (30 TBs in 2 days and 8 hours).

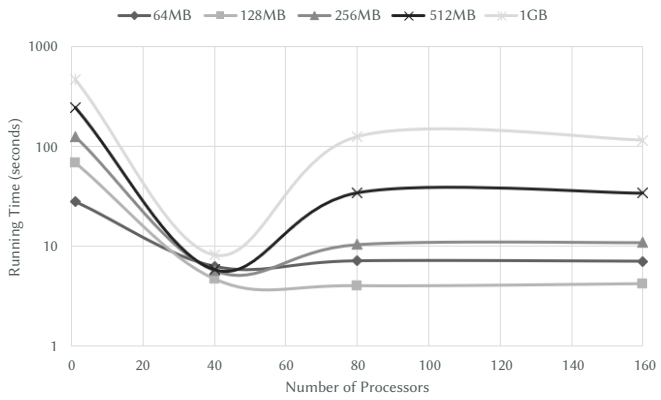


Fig. 8. Running time in seconds (logarithmic scale) of the Spark Streaming process classification executed on 40, 80 and 160 cores in the cluster computing platform.

In terms of computational effort, we have plotted the *Load one* measure of the entire cluster while running experiments with 40 and 160 cores with a batch size of 1 GB in Fig. 9 and Fig. 10 respectively, to check the overall CPU load. In particular, the *Load one* computes the number of threads at kernel level that are running and being queued while waiting for CPU resources, averaged over the last minute. We could interpret this number in relation with the number of hardware threads available on the machine and the time it takes to drain the run queue. Fig. 9 captures a short time (close to minute 8:00) in which the master node (Spark driver) delivers tasks to the worker nodes and they start to undertake data processing jobs when we run the experiment with 40 cores and 1 GB of batch size. The *Load one* measure in Fig. 10 shows an increasing activity in minute 9:20 approximately, even more than in the previous experiment when increasing the number of cores to 160.

V. Conclusions

This article presents a novel approach for Human Activity Recognition in healthcare systems for obesity patient monitoring. It comprises a combination of public (labelled) and private (unlabelled) raw data integration, semi-supervised classification with CNN Encoder-Decoder and Spark streaming processing with sliding

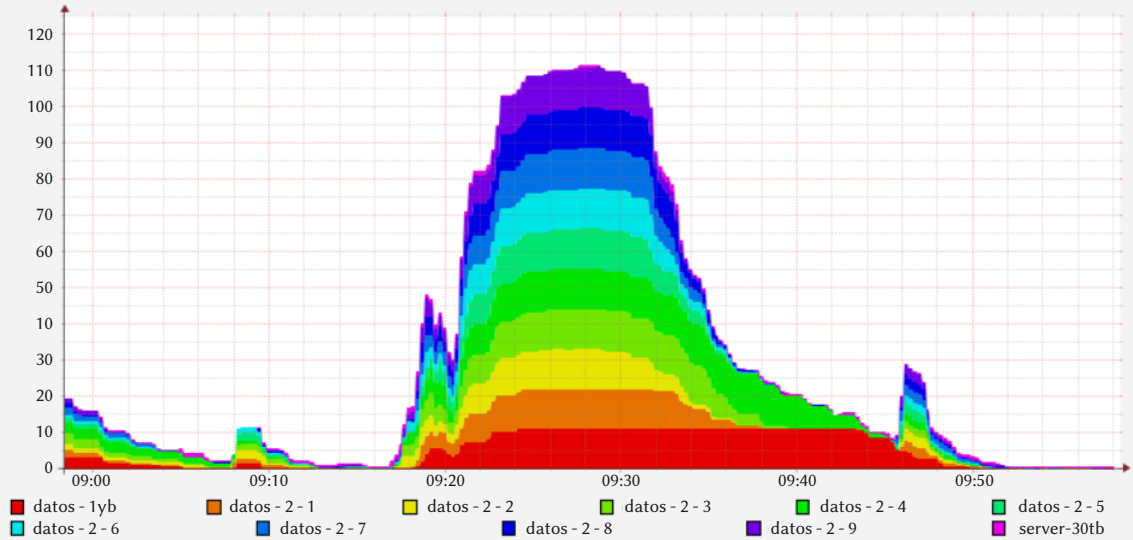


Fig. 9. Load_one. Number of threads per node (40 cores configuration). Plot captured from Ganglia cluster monitoring system for the master node (server-30tb) and 10 worker nodes.

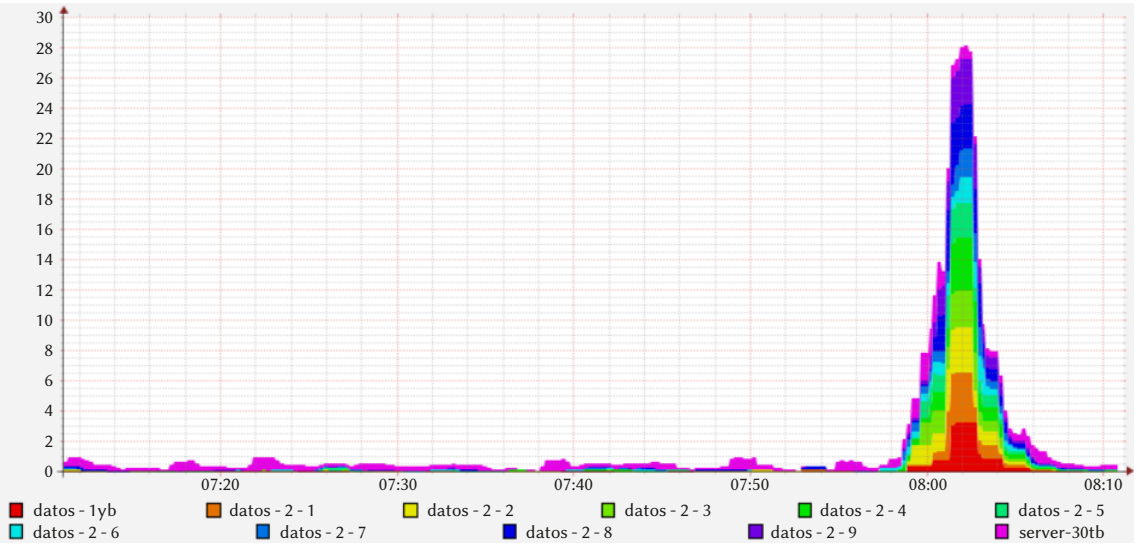


Fig. 10. Load_one. Number of threads per node (160 cores configuration). Plot captured from Ganglia cluster monitoring system for the master node (server-30tb) and 10 worker nodes.

window, to allow continuous activity recognition. The proposal has been validated in the context of a real-world case study with a group of 300 overweight patients in the healthcare system of Andalusia (Spain), classifying close to 30 TBs of accelerometer sensor-based data in real-time conditions, which is crucial for long-term daily patient monitoring.

The experimental results demonstrate that our proposed method can achieve significant Fm-scores training the model even with 100% of unlabelled data (proportion of data labelled used for train), since from this point the results decrease below to 0.8 of Fm-score. Finally, we choose the amount of 80% of unlabelled data, since at this percentage, the model reach a trade-off result (Fm-score = 0.90) between Fm-score and amount of unlabelled data added to the model. Moreover, in order to demonstrate the performance of our model we observe that the distributions of the predictions in unlabelled data are reasonable, as shown in Fig. 7.

In addition, an Spark streaming process for the activity classification was implemented in a cluster computing platform to be able to classify the raw data sensor in real-time. For this propose, we found out the best configuration to minimise the running computation time of the streaming classification, using the cluster with 40 cores and predicting with streaming batch size of 1 GB, being able to classify one week of daily patient monitoring in approximately 8 seconds.

The proposed approach represents a step forward to meet the challenges identified in a recent survey [3], which mainly consist in the generation of real-time activity recognition platforms and the development of more accurate unsupervised modelling for this problem. As argued by authors of this survey, the performance of deep learning still relies on labelled samples to a large extent, which added to the fact that acquiring sufficient activity labels is expensive and time-consuming, makes unsupervised activity recognition an urgent task. Our semi-supervised deep learning on Spark streaming processing is a solution in this direction.

Future lines of research include the generation of advanced visualisations and alarms system to support practitioners in healthcare in patient monitoring. From the perspective of prediction models, the development and use of new ensemble semi-supervised methods will enhance the precision in this kind of environments, where unlabelled data continuously flow in streams and should be properly processed as fast as they are captured.

Appendix

In the following we present the complete list of *Additional Experiments* presented in section IV subsection C. These experiments have been carried out to study the impact of specific design decisions in the context of the downstream task.

A. First Experiment

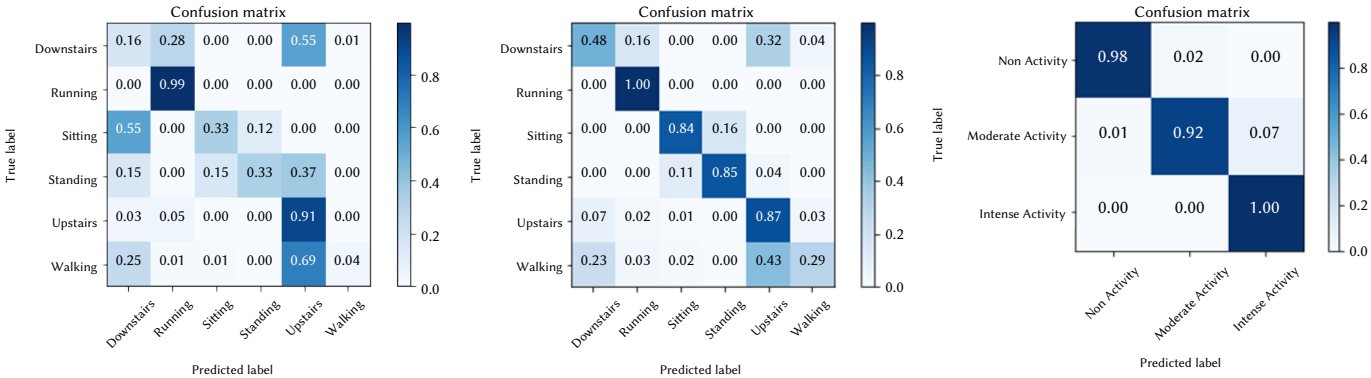
In this first experiment, we wanted to see whether the model was able to pass *"the eye test"* without taking into account the semi-supervised approach. For this propose, the model was trained only with labelled data from WISDM dataset without considering our private unlabelled data in the training phase. Afterwards, the prediction of a randomly chosen sample (five days prediction) from our 30TB private unlabelled data set was performed, as shown in Fig. 11. Can be observed that the model predicts running and walking downstairs as the main activities of the patient even during the nights and rarely predicts the activities of standing and sitting, despite the fact that these are the most prevalent behaviours among obese patients. Overall, it may be said the model is not able to make reasonable predictions if the unsupervised task is not used in the training regime.

B. Second Experiment

In a second experiment, the proposed semi-supervised methodology has been synthetically evaluated by using another public dataset as a simulation of the unsupervised portion. In this sense, HUGADB dataset has been considered as *"unlabelled dataset"* since it contains in overall the closest axis distributions to the sensorised data of WISDM dataset and the lowest standard deviation in the data as shown in Fig. 3. Hence, we study the performance of our semi-supervised CNN Encoder-Decoder model trained with a combination of WISDM as public annotated data WISDM and 70% of HUGADB dataset as a simulation of the unsupervised portion to classify the activities in HUGADB, as observed in Fig. 12. First, the model has been trained only with labelled data from WISDM without considering unlabelled data in the training phase. Afterwards, the model has been validated in the remaining 30% of HUGADB dataset, as shown in Fig. 12a. Subsequently, to demonstrate the feasibility of our semi-supervised approach the model has been trained again but this time 70% of HUGADB has been taken into account as a simulation of the unsupervised portion in the training phase. As previously, the model has been validated in the remaining 30% of HUGADB dataset, as shown in Fig. 12b. It can be appreciated that our semi-supervised approach improves the predictions results from 0.414 to 0.704 in terms of Fm-score, as shown in Table IV.

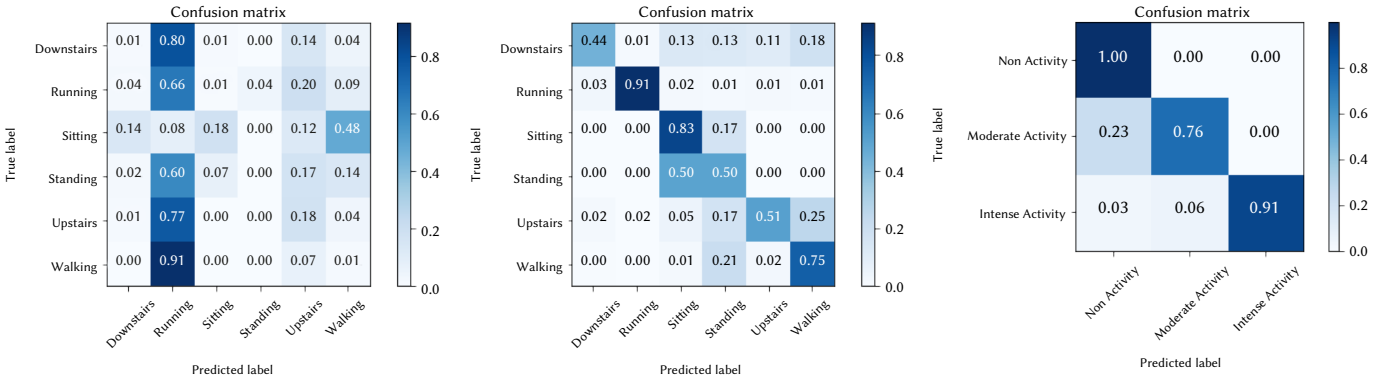
This second experiment has been repeated with another public dataset as a simulation of the unsupervised portion to verify the quality of the semi-supervised approach. In this case PAMAP2 has been selected since it contains different axis distributions to the sensorised data of WISDM dataset and the highest standard deviation

Fig. 11. Activity classification of a randomly chosen sample (five days prediction) from our 30TB private unlabelled data set. For these predictions, the model has been trained only with labelled data from WISDM dataset without considering our semi-supervised strategy with private unlabelled data in the training phase



(a) 0% of unlabelled data (b) 70% of unlabelled data (c) 70% of unlabelled data

Fig. 12. Illustration of confusion matrices showing the sensitivity of the networks for each individual when varying the percentage of unlabelled data in the training regime from 0% to 70% (HUGADB as a simulation of the unsupervised portion). In addition, the dimensionality of HAR classification problem has been reduced into three basic classes in Fig.(c): Non Activity (sitting and standing), Moderate Activity (walking, walking downstairs and walking upstairs) and Intense Activity (running).



(a) 0% of unlabelled data (b) 70% of unlabelled data (c) 70% of unlabelled data

Fig. 13. Illustration of confusion matrices showing the sensitivity of the networks for each individual when varying the percentage of unlabelled data in the training regime from 0% to 70% (PAMAP2 as a simulation of the unsupervised portion). In addition, the dimensionality of HAR classification problem has been reduced into three basic classes in Figure (c): Non Activity (sitting and standing), Moderate Activity (walking, walking downstairs and walking upstairs) and Intense Activity (running).

in the data as shown in Fig. 3. It's shown in Table 4, how the semi-supervised methodology increases the predictions results from 0.129 to 0.667 in terms of Fm-score. Also, in Fig. 13 the semi-supervised strategy increases the accuracy in all the classes.

TABLE IV. Metrics Evaluation With Varying Number of Unlabelled Examples in Training Set. HUGADB and PAMAP2 Datasets Have Been Taken As A simulation of the Unsupervised Portion to Synthetically Evaluate the Proposed Semi-supervised Methodology

Metrics: Public data (labelled) + Public data (Unlabelled)				
Labelled/Unlabelled	%	acc	recall	Fm-score
WISDM/HUGADB	0%	0.461	0.461	0.414
WISDM/HUGADB	70%	**0.722**	**0.722**	**0.704**
WISDM/PAMAP2	0%	0.173	0.173	0.129
WISDM/PAMAP2	70%	**0.667**	**0.667**	**0.667**

In spite of improving the quality of results with our semi-supervised approach, the model starts to show limited predictions in detecting some activities. For example, for the model it's difficult to predict downstairs and walking, since, if we see the patterns between walking and downstairs, they are characterised with very close signal shapes in movements, as commented before in the paper. Furthermore, static activities can be recognised easily than periodic activities (running, walking, etc.). However, highly similar postures (sitting and standing) create great complexities in case of separation due to notable overlapping in feature space as observed in Fig. 13b. In general, the dimensionality of HAR classification problem can be reduced by classifying into three basic types: Non Activity (sitting and standing), Moderate Activity (walking, walking downstairs and walking upstairs) and Intense Activity (running) as shown in Fig. 12c and Fig. 13c. It's worth to note that we can obtain promising results that will allow us to provide patient activity information to doctors which is essential to prevent obesity. In conclusion, it can be said that the semi-supervised approach achieve improvements in the results, when trying to predict activities from a dataset that the model has never seen before. With the semi-supervised strategy the model can extract important features from the unlabelled data that help us to make better predictions.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Science and Innovation via Grant TIN2017-86049-R (AEI/FEDER, UE) and Andalusian PAIDI program with grant P18-RT-2799.

References

[1] K. González, J. Fuentes, J. L. Márquez, "Physical inactivity, sedentary behavior and chronic diseases," *Korean journal of family medicine*, vol. 38, no. 3, p. 111, 2017.

[2] W. L. Haskell, S. N. Blair, J. O. Hill, "Physical activity: health outcomes and importance for public health policy," *Preventive medicine*, vol. 49, no. 4, pp. 280–282, 2009.

[3] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[4] P. Bet, P. C. Castro, M. A. Ponti, "Fall detection and fall risk assessment in older person using wearable sensors: a systematic review," *International journal of medical informatics*, 2019.

[5] A. Bourke, J. O'brien, G. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & posture*, vol. 26, no. 2, pp. 194–199, 2007.

[6] F. Bagala, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, J. Klenk, "Evaluation of accelerometer-based fall detection algorithms on real- world falls," *PloS one*, vol. 7, no. 5, 2012.

[7] F. M. Palechor, A. De la Hoz Manotas, P. A. Colpas, J. S. Ojeda, R. M. Ortega, M. P. Melo, "Cardiovascular disease analysis using supervised and unsupervised data mining techniques.," *JSW*, vol. 12, no. 2, pp. 81– 90, 2017.

[8] D. Arifoglu, A. Bouchachia, "Activity recognition and abnormal behaviour detection with recurrent neural networks," *Procedia Computer Science*, vol. 110, pp. 86– 93, 2017.

[9] G. Kalouris, E. I. Zacharaki, V. Megalooikonomou, "Improving cnn-based activity recognition by data augmentation and transfer learning," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2019, pp. 1387–1394, IEEE.

[10] A. Papagiannaki, E. I. Zacharaki, K. Deltouzos, R. Orselli, A. Freminet, S. Cela, E. Aristodemou, M. Polycarpou, M. Kotsani, A. Benetos, *et al.*, "Meeting challenges of activity recognition for ageing population in real life settings," in *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2018, pp. 1–6, IEEE.

[11] C. A. Ronao, S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[12] Y. Saez, A. Baldominos, P. Isasi, "A comparison study of classifier algorithms for cross-person physical activity recognition," *Sensors*, vol. 17, no. 1, p. 66, 2017.

[13] T. Lv, X. Wang, L. Jin, Y. Xiao, M. Song, "Margin- based deep learning networks for human activity recognition," *Sensors*, vol. 20, no. 7, p. 1871, 2020.

[14] F. Cruciani, A. Vafeiadis, C. Nugent, I. Cleland, P. McCullagh, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, R. Hamzaoui, "Feature learning for human activity recognition using convolutional neural networks," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 18–32, 2020.

[15] J. R. Kwapisz, G. M. Weiss, S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74– 82, 2011.

[16] A. Reiss, D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*, 2012, pp. 108–109, IEEE.

[17] R. Chereshnev, A. Kertész-Farkas, "Hugadb: Human gait database for activity recognition from wearable inertial sensor networks," in *International Conference on Analysis of Images, Social Networks and Texts*, 2017, pp. 131–141, Springer.

[18] M. Zhang, A. A. Sawchuk, "Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1036– 1043.

[19] D. Balabka, "Semi-supervised learning for human activity recognition using adversarial autoencoders," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, UbiComp/ISWC '19 Adjunct, New York, NY, USA, 2019, p. 685–688, Association for Computing Machinery.

[20] O. D. Lara, M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[21] A. Subasi, K. Khateeb, T. Brahimi, A. Sarirete, "Human activity recognition using machine learning methods in a smart healthcare environment," in *Innovation in Health Informatics*, M. D. Lytras, A. Sarirete Eds., Next Gen Tech Driven Personalized MedSmart Healthcare, Academic Press, 2020, pp. 123 – 144, doi: https://doi.org/10.1016/B978-0-12-819043-2.00005-8.

[22] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes- Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*, 2012, pp. 216–223, Springer.

[23] L. Bao, S. S. Intille, "Activity recognition from user- annotated acceleration data," in *International conference on pervasive computing*, 2004, pp. 1–17, Springer.

[24] W. Wu, S. Dasgupta, E. E. Ramirez, C. Peterson, G. J. Norman, "Classification accuracies of physical activities using smartphone motion sensors," *Journal of medical Internet research*, vol. 14, no. 5, p. e130, 2012.

[25] Y. Chen, Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 1488–1492, IEEE.

[26] N. Y. Hammerla, S. Halloran, T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[27] M. Zeng, T. Yu, X. Wang, L. T. Nguyen, O. J. Mengshoel, I. Lane, "Semi-supervised convolutional neural networks for human activity recognition," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 522–529.

[28] A. D. Antar, M. Ahmed, M. A. R. Ahad, "Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: a review," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, 2019, pp. 134–139, IEEE.

[29] S. Slim, A. Atia, M. Elfattah, M. Mostafa, "Survey on human activity recognition based on acceleration data," *International Journal of Advanced Computer Science and Applications*, vol. 10, pp. 84–98, 2019.

[30] Z. Hussain, M. Sheng, W. E. Zhang, "Different approaches for human activity recognition: A survey," *arXiv preprint arXiv:1906.05074*, 2019.

[31] A. Gupta, K. Gupta, K. Gupta, K. Gupta, "A survey on human activity recognition and classification," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 0915–0919, IEEE.

[32] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 1–27, 2010.

[33] F. Foerster, M. Smeja, J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in human behavior*, vol. 15, no. 5, pp. 571–583, 1999.

[34] J. R. Kwapisz, G. M. Weiss, S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, p. 74–82, Mar. 2011, doi: 10.1145/1964897.1964918.

[35] A. Reiss, M. Weber, D. Stricker, "Exploring and extending the boundaries of physical activity recognition," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, 2011, pp. 46–50.

[36] M. Zhang, A. A. Sawchuk, "Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *ACM International Conference on Ubiquitous Computing (Ubicomp) Workshop on Situation, Activity and Goal Awareness (SAGAware)*, Pittsburgh, Pennsylvania, USA, September 2012.

[37] R. Chereshnev, A. Kertész-Farkas, "Hugadb: Human gait database for activity recognition from wearable inertial sensor networks," in *Analysis of Images, Social Networks and Texts*, Cham, 2018, pp. 131–141, Springer International Publishing.

[38] H. He, E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[39] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "Smote: synthetic minority over- sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[40] K. T. Nguyen, F. Portet, C. Garbay, "Dealing with imbalanced data sets for human activity recognition using mobile phone sensors," 2018.

[41] S. Ertekin, J. Huang, L. Bottou, L. Giles, "Learning on the border: active learning in imbalanced data classification," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 127–136.

[42] D. A. Cieslak, T. R. Hoens, N. V. Chawla, W. P. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Mining and Knowledge Discovery*, vol. 24, no. 1, pp. 136–158, 2012.

[43] L. G. Fahad, S. F. Tahir, M. Rajarajan, "Activity recognition in smart homes using clustering based classification," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 1348–1353, IEEE.

[44] A. Bulling, U. Blanke, B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," vol. 46, no. 3, 2014, doi: 10.1145/2499621.

[45] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1–2, IEEE.

[46] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

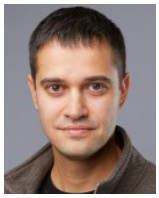
[47] J. Snoek, H. Larochelle, R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

Sandro Hurtado

Sandro Hurtado PHD student in Bioinformatics applications, with a Degree in Health Engineering with a mention in Biomedicine (2017) and a Master's Degree in Software Engineering and Artificial Intelligence (2019), from the University of Málaga. His main lines of research are the development of ontologies in the domain of Gene Regulation Networks and Software applications for the collection, consolidation and analysis of clinical data, thus providing medical and biological information to researchers and doctors in this field.

Prof. José García-Nieto

He received his Ph.D. degree in computer science with honors from the University of Málaga (Spain) in 2013, and his degree in engineering with honors in 2006, also from the University of Málaga. His current research interests include optimisation and machine learning algorithms, Big Data processing, Web Semantics and their application to real-world problems in interdisciplinary domains of Precision Agriculture, Bioinformatics and Smart Cities. His research activity has resulted in scientific publications consisting of 40 journal articles, 4 book chapters and more than 40 papers in referred international and national conferences.



Prof. Anton Popov

Anton Popov is the Artificial Intelligence/Deep Learning technical lead at Ciklum with 15+ years of experience in the development and implementation of bio-signal analysis and classification algorithms. He is affiliated with Igor Sikorsky Kyiv Polytechnic Institute as an Associate Professor. Since 2002, Anton has been a member of the IEEE Engineering in Medicine and Biology Society, has published 100+ papers.

Prof. Ismael Navas-Delgado

Computer Engineer (2002), Doctor by the University of Málaga (2009) and Master in Cell Biology and Molecular Biology (2008). His research is developed within the KHAOS Research group participating in multiple research projects (15), being the second principal investigator of the projects TIN2014-58304-R and TIN2017-86049-R, and technology transfer (2). His research activity focuses on the integration of data through the use of semantic technologies and their application to Life Sciences.