Generalizable Features for Anonymizing Motion Signals Based on the Zeros of the Short-Time Fourier Transform

Pierre Rougé, Ali Moukadem, Alain Dieterlen, Antoine Boutet, Carole Frindel

Generalizable features for anonymizing motion signals based on the zeros of the Short-Time Fourier Transform

Pierre Rougé
Univ Lyon, INSA Lyon, CREATIS
Lyon
pierre.rouge@creatis.insa-lyon.fr

Ali Moukadem
Université Haute-Alsace, IRIMAS
Mulhouse
ali.moukadem@uha.fr

Alain Dieterlen
Université Haute-Alsace, IRIMAS
Mulhouse
alain.dieterlen@uha.fr

Antoine Boutet
Univ Lyon, INSA Lyon, Inria, CITI
Lyon
antoine.boutet@insa-lyon.fr

Carole Frindel
Univ Lyon, INSA Lyon, CREATIS
Lyon
carole.frindel@creatis.insa-lyon.fr

Abstract

Thanks to the recent development of sensors and Internet of Things (IoT), it is now common to use mobile application to monitor health status. These applications rely on sensors embedded in the smartphones that measure several physical quantities such as acceleration or angular velocity. However, these data are private information that can be used to infer sensitive attributes. This paper presents a new approach to anonymize the motion sensor data, preventing the re-identification of the user based on a selection of handcrafted features extracted from the distribution of zeros of the Shot-Time Fourier Transform (STFT). This work is motivated by recent works which highlight the importance of the zeros of the STFT [1] and link them in the case of white noise to Gaussian Analytical Functions (GAF) [2] where the distribution of their zeros is formally described. The proposed approach is compared with an extension of an earlier work based on filtering in the time-frequency plane and doing the classification task based on convolutional neural networks, for which we improved the evaluation method and investigated the benefits of gyroscopic sensor's data. An extensive comparison is performed on a first public dataset to assess the accuracy of activity recognition and user re-identification. We showed not only that the proposed method gives better results in term of activity/identity recognition trade-off compared with the state of the art but also that it can be generalized to other datasets.

***K*eywords** Activity, Privacy, Time-Frequency, Gaussian Analytic Functions, Classification, Machine Learning, Random Forest

1 Introduction

The wide adoption of Internet of Things (IoT) devices have democratized quantified self applications and revolutionized patient monitoring in the healthcare domain [3, 4]. This monitoring relies on sensors that measure motion signals (e.g., accelerometer, gyroscope and magnetometer). These signals are further sent to a cloud server to be analyzed and processed through advanced signal processing and machine learning pipeline [5] to compute and present multiple estimators to users or practitioners (such as the number of steps or the activity performed during the day, the quality of the sleep, or the burned calories). Although this information is very useful for self-assessment or remote monitoring, it is closely related to the health status of the associated user and consequently sensitive. Sharing this sensitive information to third party applications exposes users to privacy threats (e.g., attribute inference or re-identification) and discrimination [6]. Moreover, health related data attract much attention nowadays. For instance, an increasing number of health insurers are seeking access to this data to better predict rates and encourage their members to wear fitness trackers [7].

To mitigate the risks of privacy leakage, several approaches have been proposed providing different privacy and utility trade-off. While some of them rely on collaborative learning (federated learning) to avoid sharing data with the server [8], others sanitize raw data to avoid unwanted inferences or re-identification [9]. Other approaches try to minimize the data sent to the server. For instance, [10] extracts locally on the device temporal and frequency features, and sends only the features the most important for the activity detection task while normalizing features leading to re-identification. Instead of processing features from the temporal and frequency domain separately, another approach [11] transforms the signal to a time-frequency representation before filtering high coefficients to limit re-identification. The resulting time-frequency representation is then directly processed by a convolutional neural network (CNN) to predict activity recognition. While this data minimization process (i.e., directly based on classifying time-frequency representation) is simple and attractive, the identification by the CNN of useful information in this representation is complex. In addition, the filtering scheme can be improved to provide a better utility and privacy trade-off (i.e., maintaining an accurate activity detection while preventing re-identification).

In this article, we propose a new approach motivated by the recent link made between Gaussian analytic functions (GAF) and time-frequency transforms of Gaussian white noise [2], which share the same distribution of their zeros. We apply this theoretical work on motion signals in order to extract features based on the zeros of the Short-Time Fourier Transform (STFT). These zeros in the time-frequency domain are represented in the form of graphs based on the Delaunay triangulation [1] and the most important features for the two targeted classification tasks (i.e. activity detection and re-identification) are evaluated using Random Forest (RF). Furthermore, we propose a feature selection scheme retaining the most compact and efficient feature set in terms of utility privacy trade-off. This method is first applied and tested on the public MotionSense [12] dataset based on two different classifiers. Then, to guarantee the good use of our approach in production environments, we ensure that there is a small difference between the accuracy rates obtained on the MotionSense dataset and the rates obtained on the independent MobiAct [13] dataset.

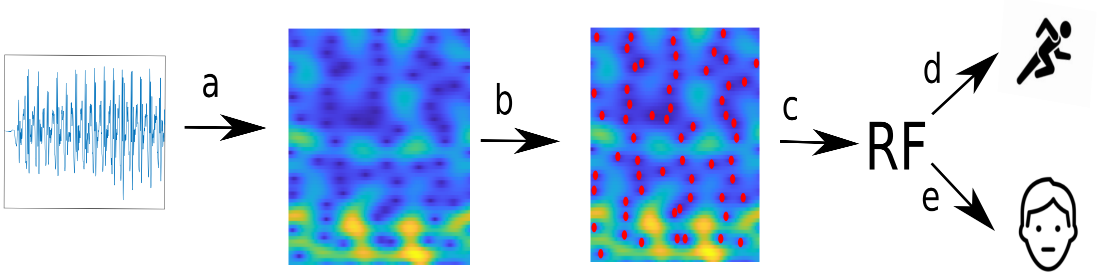


Figure 1: Outline of the approach: **a**. Transformation from time to time-frequency domain, **b**. Detection of STFT zeros (shown in red), **c**. Random Forest classifier, **d**. Activity Recognition, **e**. Identity Recognition

The pipeline of the proposed method is illustrated in Figure 1. We evaluate the utility and privacy trade-off provided by our approach based on the zeros of the STFT against an optimized version of the state-of-the-art approach [11] that we use as a baseline. Results on the reference dataset MotionSense show that the STFT's zeros approach offers a better utility-privacy trade-off (accuracy of respectively 0.80/0.30 in activity/identity recognition with RF and 0.81/0.32 with DNN) than the baseline (accuracy of respectively 0.73/0.30 in activity/identity recognition). We also demonstrate that our zero of the STFT based approach is applicable to other datasets. Indeed, by using all the features selected by our method on the MotionSense dataset, we obtain on the MobiAct dataset an accuracy of respectively 0.83/0.12 in activity/identity recognition with RF and 0.87/0.14 with DNN. Finally, our approach requires less training time than the baseline for the activity recognition classifier. Thus, in a centralized framework where the model must be updated with new data to improve its performance, our approach is more efficient in terms of computation time.

This paper is an improved and extended version of the conference paper published in [14]. More precisely, we demonstrate that our conclusions are robust to the choice of the classifier by comparing our results obtained for RF with those obtained for a Dense Neural Network (DNN) classifier. We also discuss in Section 4.6 the computation time and utility of our method in a real centralized architecture where the model must be updated regularly and where our method has the advantage of requiring less training time. And more importantly, the results of the transfer of our method on the MobiAct dataset in Section 4.5 demonstrate its generability. The rest of the paper is organized as follows. Section 2 gives more details on background and the optimized version of the state-of-the-art approach [11]. Section 3 details our STFT's zeros approach while Section 4 presents the results of the evaluation. Finally, Section 5 discusses conclusions and future work.

2 Background and state-of-the-art

2.1 STFT and Bargmann connection

Time-frequency domain allows studying the frequency evolution of a signal during time; it is particularly useful to analyze non-stationary signals. The most common transform from the time domain to the time-frequency domain is the Short Time Fourier Transform (STFT). The STFT for a given signal $x(t)$ and a window function $w(t)$ is given by:

$$S_x^{\omega}(t,f) = \int\limits_{-\infty}^{+\infty} x(\tau)w^*(\tau - t)e^{-2j\pi f\tau}d\tau, \tag{1}$$

In the case of a Gaussian window $\omega(t) = g(t) = \frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-t^2}{2\sigma^2}}$, the STFT can be written through Bargmann transform [15] as follows [2]:

$$S_x^g(t,-f) \propto e^{-i\pi tf}e^{-\frac{\pi}{2}z^2}B_x(z), \tag{2}$$

where $z = t + if \in \mathbb{C}$ and $B_x(z)$ is the Bargmann transform defined as follows :

$$B_x(z) = 2^{1/4}\int_{-\infty}^{+\infty} x(t)e^{2\pi tz - \pi t^2 - \frac{\pi}{2}z^2}dt. \tag{3}$$

This means that the STFT can be completely characterized by its zeros. Equation 2 shows that the zeros of the STFT are the zeros of the Bargmann transform which are also the zeros of Gaussian analytic functions (GAFs). It ensures some regularity for the zeros distribution of the STFT for white Gaussian noise. The mathematical details and properties of these links are detailed in [2].

2.2 CNN-based filtering approach

In this section, we present the state-of-the-art approach proposed in [11] and the optimization of the CNN we applied.

Filtering: It was observed in [11] that in the time-frequency representation the difference between activities is encoded through texture and that on the other hand, the difference between subjects is encoded by contrast.

According to these observations, it was proposed in [11] to filter high coefficients of the spectrograms to remove user's information to prevent re-identification. In this study, we used this filtering method with different percentages of high coefficient removed going from 0% to 90% with steps of 10%.

CNN classifier: In this study, we used six different CNN classifiers that can be broken down into 3 categories: those taking as input accelerometer data only, those taking as input gyroscope data only and those using both accelerometer and gyroscope data as input. In each of these categories, a CNN has been constructed to classify activities into 4 classes and another to classify the user's identity into 24 classes. The results of these models were compared to assess the utility of adding the gyroscope data into the framework.

For all CNN, we considered a model with four convolutional layers followed each by one max pooling layer and at the end a final softmax dense layer for classification. The number of filters on the first layer was set to a power of two and for each layer the number of filters was set to the next power of two.

Optimization of CNN using accelerometer or gyroscopic data only: The two tasks (activity recognition and identification) are different by nature, it is therefore necessary to optimize the architecture and the hyperparameters of the CNN independently for these 2 tasks. Fine tuning requires testing a lot of hyperparameter combinations. In this work, the optimization process focused on a certain amount of hyperparameters and possible ranges of values. These hyperparameters were the number of filters on the first convolutional layer, the batch size and the learning rate. Also, during the optimization process, if a hyperparameter value was obviously not adapted we chose to withdraw this value from the process, so the test of the combinations is not exhaustive.

Two different fusion schemes for the three axes of the sensors were also evaluated and compared: late and early fusion. The early fusion strategy consists in combining images from the 3 axes at the entry of the network. The late fusion strategy consists in using three independent convolutional branches to process each input independently then combining the features map from the three branches just before the dense layer.

The optimization was made using the accelerometer data without any filtering and the selected model was later also used on gyroscopic data only.

Model using both accelerometer and gyroscope data: Few tests were done to evaluate the values of the hyperparameters around those found for the model on the accelerometer data to ensure its transferability. Three fusion strategies were also assessed: (i) early fusion on axes, (ii) early fusion on sensors and (iii) late fusion. The early fusion on axes consists in merging the images of the two sensors (accelerometer and gyroscope) corresponding to the same axis at the input of the CNN, then three independent convolutional branches process the three axes and finally the characteristic maps of the three branches are merged just before the final dense layer. Early fusion on sensors is the same idea where images of the three axes (x, y, z) are merged at the input of the CNN corresponding to the same sensor. And the late fusion strategy is the same idea as previously, except that in this case we have six different convolutional branches each corresponding to a distinct sensor and axis. From our comparisons, we chose the late fusion strategy with a batch size of 512 and a learning rate of 0.0025, which was the set of parameters giving the best results in terms of accuracy on the validation set for both activity and identity recognition tasks.

Training implementation: The dataset was split into training and test sets according to the trials created during the acquisition phase: thus trials 1 to 9 were used for the training phase and trials 11 to 16 for the test phase. More precisely, during the training phase, 90% of the whole set was used for training and 10% for validation. We used categorical cross entropy as the loss function and Adam as the optimizer. The maximum number of epochs was set to 200 and regulated with an early stopping criterion.

3 STFT zeros approach

The link established between the STFT and the GAFs guarantees a regular and well known distribution of the STFT zeros in case of white Gaussian noise [2, 1]. In this sense, the distribution of zeros can provide information on the presence of noise or signal for the development of filtering schemes. In this work, we exploit the distribution of zeros to extract handcrafted features to classify activities while preserving privacy. The intuition behind this idea is that the presence of a signal will modify the distribution of zeros in the time-frequency domain and mark this distribution by the signal signature. The first step in this process is to detect the zeros from STFT representation. We use a Gaussian window for the STFT to ensure the link between STFT and GAFs. The value of σ for the Gaussian window is set empirically to 0.05. Since we are working on discrete STFT, the zeros are not perfect and the energy spreading around instantaneous frequencies of the signal's components will affect the intensity of the zeros. To detect them, we used a 3x3 mask sliding through STFT: if the value in the center of the mask is the minimum and the maximum value covered by the mask exceeds a certain threshold, we consider a zero in the center of the mask. Here the threshold was set to $\max(S_x^g)/10^4$, with S_x^g the modulus of the STFT representation (see Equation 1).

3.1 Features associated with STFT zeros

Once the zeros are detected, features associated with their distribution must be extracted for use in a standard classifier. To this purpose, we connect them with a Delaunay triangulation and create a graph. Examples of obtained graphs for different subjects and activities are given in Figure 2.

Since numerical zeros are not perfect zeros and their intensity is influenced by the energy in their area, we chose to order them according to their intensity. The advantage of ordering the zeros is that we can easily construct features attached to a zero (for example its intensity) and above all compare them between two different graphs. It suffices then to compare the features associated with the zeros of the same rank in the ordering. Two types of features can be constructed: those which characterize the global graph of zeros and those which are attached to a particular zero. In our dataset, the minimum number of zeros detected was 48, so for each STFT we extracted 48 zeros: the 24 zeros of minimum intensity and the 24 zeros of maximum intensity.

Global Features: To analyze the distribution of the zeros, we studied the distance between them. We constructed three distributions: euclidean distance, distance on time-axis and distance on frequency-axis between each zero. For each of these distributions, four statistical moments were used as global features: mean, standard deviation, skewness and kurtosis. From the graph, mean and max edge length were also extracted as features.

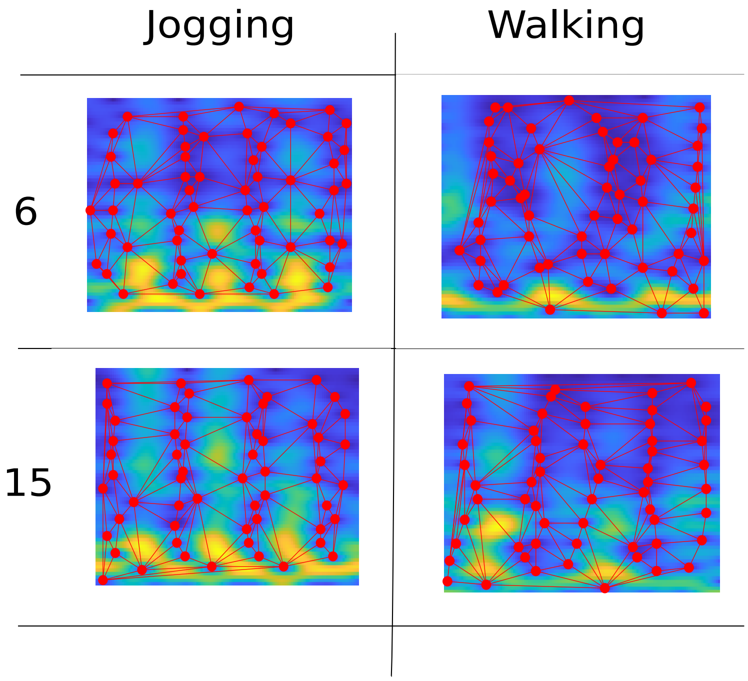


Figure 2: Examples of STFT representations superposed with the associated graph formed from STFT zeros for different subjects in lines and different activities in columns

.

Local Features: To characterize the zero itself, we used as a feature its intensity and its coordinates in the time-frequency plane. From the graph of zeros, we also considered the zeros belonging to a neighborhood of order 1 and used their mean intensity as a feature. We also computed the mean energy crossed by each edge to reach the neighbors and used the average of these energies as a feature. Finally, to characterize the edges of the graph, we also used the average angle of the edge with respect to the x-axis and the area of the triangles connected to the zero.

To investigate patterns in the region surrounding the zeros, we used Haralick features [16]. Fourteen features were calculated from the gray level co-occurence matrix (GLCM). We considered two GLCM with respectively an offset of (0,1) and (1,0). The idea is to consider the co-occurence along the time and the frequency direction independently. We used a window of size 30x30 to capture the region surrounding the zero. In the end, a given STFT image results in a total of 1694 features summed up in Table 1. In this table we associate each category of features with a color for future figures : in blue the global features linked with the distribution of zeros in the time-frequency plane, in red the features extracted from the graph representation, in yellow the local features associated with each zero and their neighborhood and in green the Haralick features associated with each zero.

3.2 Random Forest Classifier

To identify both activity and subjects from the features extracted from the STFT's zeros, we used a Random Forest (RF) classifier. Two parameters were more particularly studied: the number of trees used in the forest and the maximum depth of trees. The values tested were respectively in the following ranges $[400, 500, 600]$ and $[25, 50, 75, 100]$. To investigate these values, we made a 5-fold cross-validation over each of the possible combinations and selected the one which gave the best result in accuracy.

As with the CNN-based approach, at a given time there are 6 STFT images from two sensors each with 3 axes. For each of the STFT images, a feature matrix – as described in Section 3.1 – has been calculated. To build the RF model we choose to concatenate all the feature matrices associated with the 6 STFTs (see Model A in Figure 3).

3.3 Feature selection

Once the RF models have been constructed, it is possible to observe the importance of each of the features in the two classification tasks. Our goal is to remove features useful for identity recognition but not for activity recognition. Then

Features	Number	Color
Moments of the Euclidean distance distribution between zeros	4	🟦
Moments of the Euclidean distance distribution between zeros on time-axis	4	🟦
Moments of the Euclidean distance distribution between zeros on frequency-axis	4	🟦
Maximum edge length	1	🟥
Mean edge length	1	🟥
Zeros intensity	48	🟨
Mean intensity of the neighborhood of each zero	48	🟨
Mean energy crossed to reach each neighbor for each zero	48	🟨
Haralick features (in time) for each zeros	672	🟩
Haralick features (in frequency) for each zeros	672	🟩
Mean angle of edges linked to each zero	48	🟥
Mean are of triangle linked to each zero	48	🟥
Coordinate in time of each zero	48	🟨
Coordinate in frequency of each zero	48	🟨
Total	1694	

Table 1: Summary of features extracted from images. The colors represent the types of features (blue for global features, red for features related to the graph, yellow for features related to zeros and green for features related with the neighborhood of the zeros).

we propose a feature selection scheme, illustrated in Figure 3. The first step to select the most appropriate features is to compute the average importance of the features resulting from each sensor/axis pair. The sensor/axis pairs contributing most to the activity recognition task have been retained (see Model B in Figure 3). After that, the correlation between each feature was calculated and if this correlation between two features exceeded a certain threshold, we removed from the model the less important feature for the activity recognition task (see Model C in Figure 3). These operations make it possible to improve the utility-privacy trade-off. In our experiments, the correlation threshold was set at 0.5.

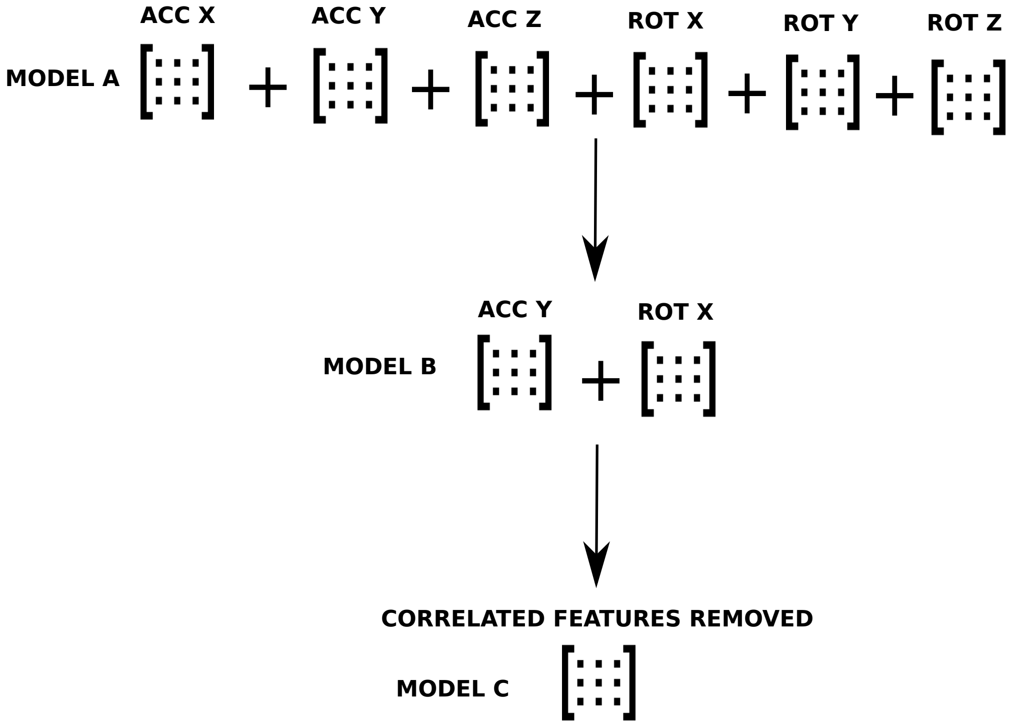


Figure 3: Illustration of feature selection method

.

3.4 Dense neural network

an RF classifier allow us to compute the importance of features in the two tasks, but this method might overfit. Therefore to be sure that the results in classification are not impacted by the choice of the classifier, we also used a DNN classifier. This network is five units combining a dense layer and a dropout layer (set to 20%) plus a final dense layer for classification. The results of this network are given on the basis of a 5-fold cross-validation (just as for the RF).

4 Evaluation and discussion

4.1 Experimental Settings

Datasets: In this study, we used the public dataset MotionSense [12] to develop our dual classifier. This dataset was collected with an iPhone 6S placed in the participant's trousers front pocket at a frequency rate of 50 Hz. The dataset provides time-series data from 3-axis accelerometer and gyroscope and includes recordings (15 trials) of 24 different participants for six activities: downstairs, upstairs, walking, jogging, standing and sitting. The time-series are split in sliding windows such that each window corresponds to an activity and a participant. The window length was fixed to 2.56 seconds with an overlap of 50%, the average cadence range of walking is about 1.5 steps per second so about 3 walking steps were captured by the window. In this study, only the four dynamic activities were considered: downstairs, upstairs, walking and jogging.

To measure the generalization of our approach, we further test it on another public dataset MobiAct [13]. This dataset was collected with a Samsung Galaxy S3 placed freely by the participants in their front pocket. Initially the frequency rate was 200 Hz but we downsampled the time series to have a frequency rate equivalent to MotionSense (namely 50 Hz). The dataset is composed of time-series data from 3-axis accelerometer and gyroscope for 61 patients. The time-series are split in the same fashion that for the MotionSense dataset. Finally, we considered the same four dynamic activities as in MotionSense : downstairs, upstairs, walking and jogging.

Time-Frequency images: The images formed by the STFT module are considered to train the CNN or to extract features from the zeros of the STFT. The size of the generated images is 65 × 128 which corresponds to 25 Hz × 2.56 sec.

Accuracy: To assess the performance of the CNNs and the RF classifiers, we computed an accuracy score defined as:

$$Accuracy = \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} 1(y_i, \widehat{y}_i), \tag{4}$$

where n_{samples} is the number of samples and $1(x, y)$ the indicator function which gives 1 if $x = y$ and 0 otherwise. For the CNNs classifiers the given accuracies were averaged over ten experiments and for the RF over a 5-fold cross validation.

Privacy measure: Considering the x-y plane where the x and y axes respectively represent accuracy in identity and activity recognition tasks, we measure the privacy of our approach by calculating the distance between a point (x, y), defining the performance of our method, and the point $(0, 1)$. The $(0, 1)$ point characterizes the desired optimal where the recognition in identity is null and that in activity perfect (equal to 1). Thus any method is characterized by its point in this plane and uses this distance to evaluate privacy. The goal is to minimize this distance. In parallel, to compare the different CNNs, the Area under the utility-privacy Curve (AUC) for different levels of filtering is studied.

Gini importance: To compute features importance in the RF model we use the Gini impurity define for an ensemble of samples S by :

$$Gini(S) = 1 - \sum_{i=1}^{N} p(i)^2, \tag{5}$$

with N the numbers of classes in the task and $p(i)$ the percentage of class i in the ensemble. The Gini impurity is computed for a node in a tree by considering S as the ensemble of samples reaching the node. The node is said to be pure if the Gini impurity is equal to 0, namely if every sample in the S ensemble belongs to the same class. When we want to know the quality of a split S we compute the Gini impurity of the successors nodes S_1 and S_2 and we compute the Gini gain define by :

$$Gain_{Gini}(S) = Gini(S) - (p_1 * Gini(S_1) + p_2 * Gini(S_2)), \tag{6}$$

with p_1 and p_2 the rate of samples reaching respectively the nodes S_1 and S_2. To compute the importance of a feature, the average Gini gain of nodes using the feature for splitting is calculated, and this gain is weighted by the probability of reaching the node. After that, all the features' importance are normalized so that the sum of all features' importance equals 1. Finally, we divide the importance by the mean importance in order to avoid being sensitive to the number of features. Thereby, a feature is considered significant if its importance is superior to 1.

Transfer test: Our method consists in a smart selection of features based on the zeros of the time-frequency representation preserving the identity of the user. But this selection was made on a specific dataset, so the selection of features offered may not generalize well to other data. To evaluate the effectiveness of our method in a real use case, we need to test it on another dataset. The test consists of calculating on the MobiAct database the features suggested by the analysis of the MotionSense database and training a classifier for identity recognition and a classifier for activity recognition on the MobiAct dataset.

4.2 Optimization of the CNN

This section present the results related to the optimization of the CNN. Tables 2 and 3 present the results of the four more efficient architectures in respectively the identity and activity recognition tasks. In these tables, column "Filter" refers to the number of filters in CNN first layer, "Lr" to the learning rate and "Acc Val" to the average accuracy on the validation set over ten experiments.

Model	Filter	Batch size	Lr	Acc Val
Late fusion	**16**	**256**	**0.005**	**0.77**
Late fusion	8	256	0.005	0.75
Early Fusion	16	256	0.005	0.76
Early fusion	8	256	0.01	0.69

Table 2: Accuracy on validation set for the four most efficient architectures in the identification task. The selected model is shown in bold.

Model	Filter	Batch size	Lr	Acc Val
Late fusion	**16**	**256**	**0.005**	**0.93**
Early Fusion	16	256	0.005	0.91
Late fusion	8	256	0.005	0.92
Early Fusion	8	256	0.005	0.91

Table 3: Accuracy on validation set for the four most efficient architectures in the activity recognition task. The selected model is shown in bold.

For the identification task, we selected the model with late Fusion and batch size=256, number of filters=16 and learning rate=0.005, because it was the combination showing the best accuracy on validation set (Table 2). For the activity recognition task the results gave similar performances for the different models. We first chose the model with early Fusion, batch size=256, number of filters=16 and learning rate=0.005 because it has fewer parameters than the late Fusion one (103 588 versus 309 892 weights). But during the experiments presented in Section 4.3, the early fusion model turned out to be difficult to transfer in the case of filtering, so we decided to return to the model using late fusion but with the same parameters.

4.3 Role of gyroscopic sensors data

In this section we investigate the benefit of learning from gyroscopic sensors' data in terms of utility-privacy trade-off. Figures 5 and 6 respectively represent the activity and identity accuracy results for different filtering levels for three different CNNs. On the one hand, Figure 6 shows that the CNNs learned from accelerometer or gyroscope data result in degraded accuracy at the same pace for the identity recognition tasks. On the other hand, Figure 5 indicates that, for the activity recognition task, the filtering affects less the performance of the CNN learned from accelerometer data than that learned from gyroscope data. Furthermore, it exhibits that a CNN combining accelerometer and gyroscope data allows to boost the performance in identity recognition but not in activity recognition (to be compared with Figure 6). This observation is confirmed by the Table 4 presenting the distance to the optimal point in the identity-activity plane, considering the level of filtering and for the three CNNs. In terms of normalized AUC, the CNN learned from accelerometer data also outperforms the one learned from gyroscope data as shown in Table 5. Even though the CNN combining accelerometer and gyroscope data presents a slightly better AUC than the one based on accelerometer data, the results based on the distance to the optimal point (see Table 4) demonstrate that learning only from the accelerometer data offers a better utility-privacy trade-off. Indeed, for each level of filtering, the method learning only from accelerometer data has the lowest distance and also has the overall minimum distance (0.41 obtained for 90% filtering). In the rest of this study the result obtained at 90% of filtering with the model learning only for accelerometer data will be our reference to compare our new proposed approach.

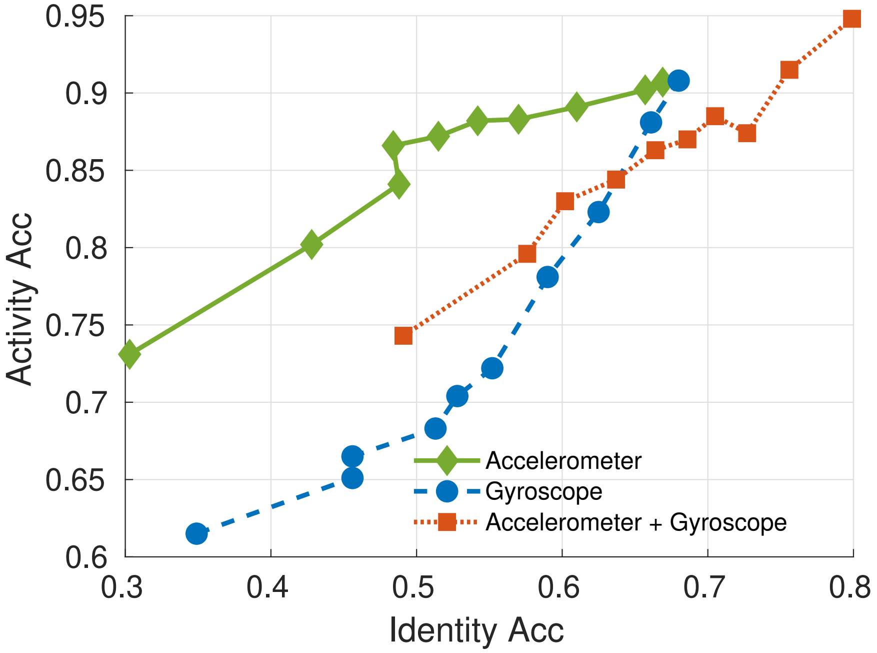


Figure 4: Activity versus identity accuracy for CNN using respectively accelerometer data, gyroscope data and both.

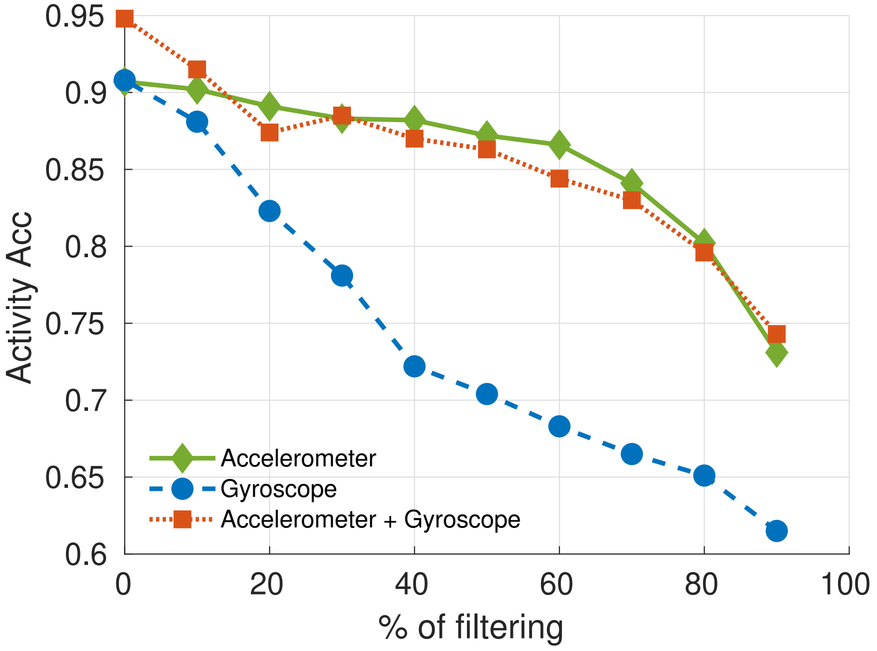


Figure 5: Activity accuracy versus % of filtering for CNN using respectively accelerometer data, gyroscope data and both

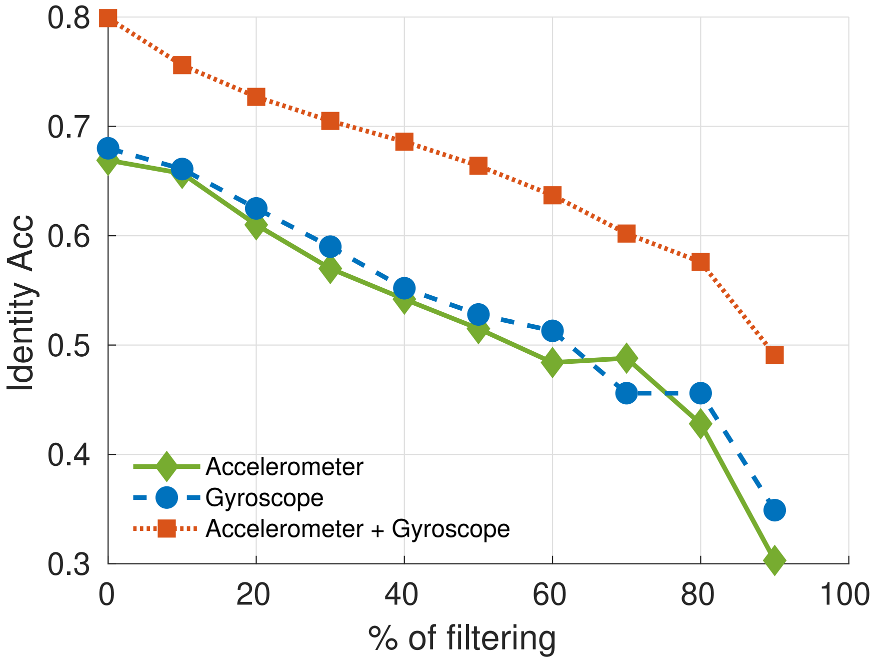


Figure 6: Identity accuracy versus % of filtering for CNN using respectively accelerometer data, gyroscope data and both

% filtering	Gyro	Acc + Gyro	Acc
0	0.80	0.69	0.68
20	0.74	0.65	0.62
40	0.70	0.62	0.55
60	0.66	0.60	0.50
80	0.61	0.57	0.47
90	**0.55**	**0.52**	**0.41**

Table 4: Distance to the top corner in identity-activity plane for different levels of filtering and for the three CNNs. The filtering level providing the best utility-privacy trade-off is depicted in bold.

Gyro	Acc + Gyro	Acc
0.719	0.844	0.835

Table 5: Normalized AUC for the CNNs using respectively only accelerometer, gyroscope and both data.

4.4 The zeros of the STFT

In this section we present the results of our proposed approach based on the zeros of the STFT on the MotionSense dataset. With the RF model detailed in 3.2 we first obtained an accuracy score of 85% for the activity recognition task and 72% for the identity recognition task as shown in Table 7-model A. Table 6 gives the mean importance of each pair of sensors and axes computed as it's explained in Section 4.1. It shows that for the identity recognition task no pair was preponderant in the decision because all the importance are close to 1, whereas for the activity the acceleration on y-axis and the rotation speed around x-axis were decisive. We have therefore decided to limit ourselves to these two channels allowing a first improvement in the measure of privacy as shown in the Table 7-model B.

Next, we applied our method to remove the correlated features. We managed to reduce the number of features to 415 and significantly improve the utility-privacy trade-off, as shown in the Table 7-model C. Switching from model A to C makes it possible to reduce the accuracy in the identity recognition task (from 0.72 to 0.30 for the RF model and from 0.82 to 0.32 for the DNN model) without reducing the accuracy in the activity in the activity recognition task too much (from 0.85 to 0.80 for the RF model and from 0.92 to 0.81 for the DNN model). Maintaining a high final accuracy in the activity recognition task is essential to ensure the effectiveness of the approach for clinical application. Finally, we

Sensor/Axis	Activity	Identity
Acceleration on x-axis	0.6	0.9
Acceleration on y-axis	**1.7**	**0.9**
Acceleration on z-axis	0.6	0.9
Rotation speed around x-axis	**1.4**	**1.0**
Rotation speed around y-axis	0.7	1.0
Rotation speed around z-axis	0.9	1.1

Table 6: Ratio of the mean importance of descriptors for each pair of sensor and axe. The selected pairs are represented in bold.

manage to improve privacy with regard to the state of the art, which is confirmed by the distance to the optimal point in the identity-activity plane which is 0.36 with the RF model and 0.37 with the DNN model versus 0.41 for the baseline.

Table 7 also presents the results of the DNN classifier for the different models proposed. The results show that independently of the classifier, our feature selection improves the utility-privacy trade-off. Indeed, switching from model A to C, the distance to the optimal point is improved from 0.74 to 0.36 for the RF model and from 0.82 to 0.37 for the DNN model. It should also be noted that overall the DNN classifier performs better than the RF classifier in the activity recognition task.

Model	Activity Acc.		Identity Acc.		Distance to opt.		Features
	RF	DNN	RF	DNN	RF	DNN	
A	0.85	0.92	0.72	0.82	0.74	0.82	10164
B	0.81	0.85	0.52	0.60	0.55	0.62	3388
C	**0.80**	**0.81**	**0.30**	**0.32**	**0.36**	**0.37**	**415**

Table 7: Results on the MotionSense dataset for both activity and identity recognition tasks depending on the features used; Model A : all features, Model B: features from acceleration on y-axis and rotation speed on x-axis, Model C : same features as in B with deletion of correlated features. The model providing the best utility-privacy trade-off is depicted in bold.

Our method is based on a smart selection of features that benefit more to the activity recognition task than to the identity recognition task. Indeed, in Figure 7 we can see the distribution of importance for both task. It shows that for the identity recognition task the distribution is concentrated around 1 which correspond to the average importance. On the other hand for the activity recognition task the importance are more spread towards more significant importance score. This show that for the activity recognition task there are significant features whereas there is no useful information for identity recognition task in the remaining features. This explains why our selection of features allows preventing identity recognition while maintaining good accuracy in activity recognition.

4.5 Transfer on MobiAct

In this section we study the transfer of our method on the MobiAct dataset. Table 8 gives the results of the transfer test explained in 4.1. The results are similar to the ones obtained on MotionSense dataset, indeed with our method we observe from Model A to Model C a small decrease in activity accuracy (-6% for the RF model and -14% for the DNN model) but also a major decrease in the identity accuracy (-58% for the RF model and -61% for the DNN model). The results seem even more significant than with the previous dataset (final accuracy in identity recognition of 0.12 with the RF model and 0.14 with the DNN model) but this is mainly due to the fact that there are more users in this dataset making the re-identification task more difficult. Moreover, we can see that we succeeded to diminish the distance from the optimal point from 0.38 to 0.21 for the RF model and from 0.65 to 0.19 for the DNN model. The important thing is that our method succeeded in improving the utility-privacy trade-off between model A and model C showing that it generalized well to other datasets.

To understand why the generalization works so well, we also performed the feature selection on the MobiAct dataset and we compared the selected features to the one found with MotionSense. With the MobiAct dataset we have 457 final features of which 217 are shared with MotionSense. Furthermore, the other features kept belong to the same categories of features as it can be seen in Figure 8. This bar graph represents for the two datasets the features kept among the 3388 features in the model B to give the Model C using our selection based on correlation and feature importance explained in Section 3.3. It highlights again the important role of Haralick features in the model.

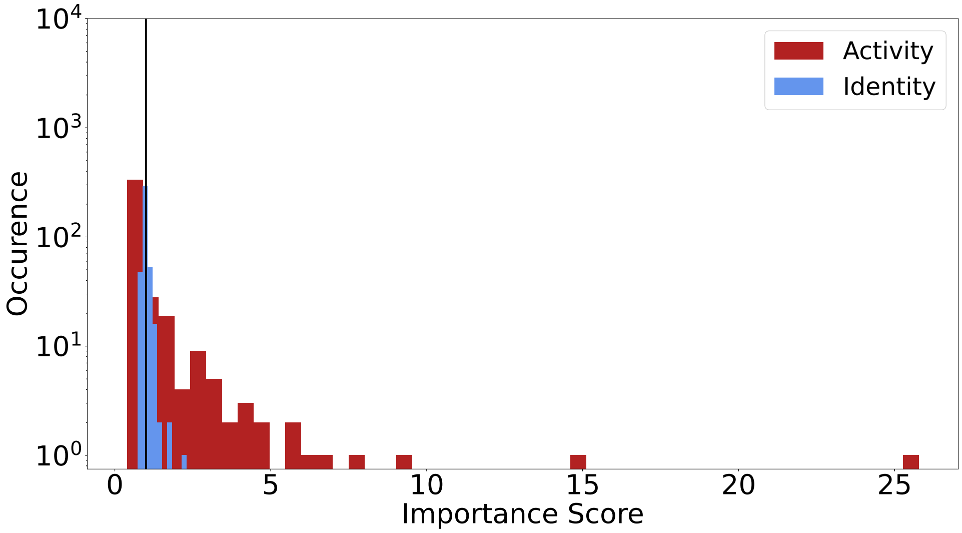


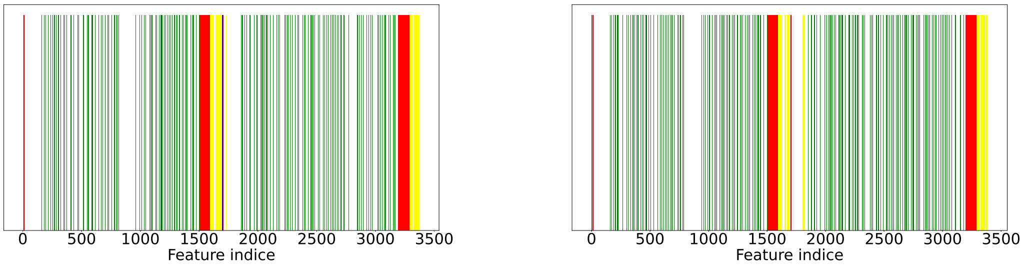
Figure 7: Histogram of features importance score for both task for Model C. In red the activity recognition task and in blue for the identity recognition task. The black line represents the importance score equal to 1 separating significant and non-signification features.

Model	Activity Acc.		Identity Acc.		Distance		Features
	RF	DNN	RF	DNN	RF	DNN	
A	0.86	0.96	0.35	0.65	0.38	0.65	10164
B	0.85	0.94	0.26	0.47	0.30	0.47	3388
C	**0.83**	**0.87**	**0.12**	**0.14**	**0.21**	**0.19**	**415**

Table 8: Result of classification in both tasks on the MobiAct dataset depending on the features used; Model A : all features, Model B: features from acceleration on y-axis and rotation speed on x-axis, Model C : features selected by our method using the MotionSense dataset. The model providing the best utility-privacy trade-off is depicted in bold.

4.6 Computational Time

Lastly, our approach can be leveraged to minimize the personal data sent to a third party server providing an application using motion sensors. Indeed, the preprocessing and the feature extraction from the zeros of the STFT can be done locally on the user smartphone, and only the most interesting features (in terms of utility and privacy trade off) can be then sent to the server. This data minimization scheme has an impact on the computational cost, and more precisely who bears this cost. Usually in a fully centralized architecture, all the data from users are sent to the server which



(a) MotionSense (b) MobiAct

Figure 8: Features retained for MotionSense and MobiAct datasets. The colors refer to those used in Table 1 and are explained in its legend.

performs all the processing. This processing can include the preparation of the data (i.e., the preprocessing) and the learning tasks on a regular basis to update the models. In this case, although the computational cost is only supported by the third party server, nothing prevents it (or an adversary in case of data leak) from inferring private and sensitive information about users such as health or demographic attributes. With data minimization instead, the users support the cost of their privacy by processing on their device (their smartphone) the data to extract the most important features. The third party server, in turn, only supports the training tasks with a limited number of features.

Table 9 shows the average computational time for the preprocessing and training phase for each method. The preprocessing includes the computation of the STFT and the filtering scheme for the state-of-the-art method and the computation of the STFT and the extraction of the handcrafted features for our proposed zeros-based method. The training phase corresponds to the training of the associated classifier for the activity recognition task (utility task). For the pre-processing step, we calculated the time required on 57,279 spectrograms for the zero-based methods and 38,186 for the filter-based method. The training time was averaged on 5 training for the DNN and 10 training for the CNN and the RF. These results show that the preprocessing for our method requires a time lower than 100 ms, which is compatible with a real-time processing knowing that all the methods operate time windows of 2.56 s. Additionally, our method operates on a small amount of handcrafted features used by simple machine learning algorithms to predict activity. These machine learning algorithms require less training time and less computing power than the CNN required by the state-of-the-art method to process full STFT images (see Table 9). This makes our approach more suitable for a centralized architecture where the model is updated regularly.

Model	Preprocessing	Training
Features Zeros DNN	93.2 ms	17.9 ± 0.8 s
Features Zeros RF	93.2 ms	21.4 ± 0.1 s
Filtered spectrogram CNN	3.4 ms	83.8 ± 3.9 s

Table 9: Computation time for preprocessing and training phase for each method

5 Conclusion

In this paper, we presented a new privacy-preserving approach based on the zero's distribution in the time-frequency domain. We extracted new features based on graphs formed by the zeros in the time-frequency plane and we proposed a feature selection method to boost the utility-privacy trade-off. The proposed approach showed better performance compared to an extended version of a state-of-the-art reference method based on time-frequency image filtering and a CNN classifier. Moreover, we showed that the zero's time-frequency features are generalizable to other datasets, and can be used with simple machine learning approaches requiring a short training time making it simple to deploy on a centralized framework. To extend this work, it may be interesting to test other optimized time-frequency transforms and explore their zero's distribution as features. Also, it's possible to study the graph of local maxima and the phase information of the time-frequency representation to enrich the set of the extracted features. In addition, the proposed approach can be extended to other types of signals and applications and compared with fashionable approaches such as deep learning classification approaches. Finally, in this study, we dealt with only the re-identification threat, however it could be interesting to extend this work to other privacy threats such as gender inference or age inference.

Statements and Declarations

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Patrick Flandrin. Time–frequency filtering based on spectrogram zeros. *IEEE Signal Processing Letters*, 22(11):2137–2141, 2015.

[2] Rémi Bardenet, Julien Flamant, and Pierre Chainais. On the zeros of the spectrogram of white noise. *Applied and Computational Harmonic Analysis*, 48(2):682–705, 2020.

[3] Btihaj Ajana. Digital health and the biopolitics of the quantified self. *Digital Health*, 3:2055207616689509, 2017.

[4] Pete B Shull, Wisit Jirattigalachote, Michael A Hunt, Mark R Cutkosky, and Scott L Delp. Quantified self and human movement: a review on the clinical impact of wearable sensing and feedback for gait analysis and intervention. *Gait & posture*, 40(1):11–19, 2014.

[5] Mark Hoogendoorn and Burkhardt Funk. Machine learning for the quantified self. *On the art of learning from sensory data*, 2018.

[6] Dominik Leibenger, Frederik Möllers, Anna Petrlic, Ronald Petrlic, and Christoph Sorge. Privacy challenges in the quantified self movement–an eu perspective. *Proceedings on privacy enhancing technologies*, 2016(4):315–334, 2016.

[7] P. Olson. Wearable tech is plugging into health insurance. *Forbes*, 2014.

[8] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

[9] Antoine Boutet, Carole Frindel, Sébastien Gambs, Théo Jourdan, and Rosin Claude Ngueveu. Dysan: Dynamically sanitizing motion sensor data against sensitive inferences through adversarial networks. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 672–686, 2021.

[10] Théo Jourdan, Antoine Boutet, and Carole Frindel. Toward privacy in iot mobile devices for activity recognition. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 155–165, 2018.

[11] Noëlie Debs, Théo Jourdan, Ali Moukadem, Antoine Boutet, and Carole Frindel. Motion sensor data anonymization by time-frequency filtering. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1707–1711. IEEE, 2021.

[12] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In *Proceedings of the international conference on internet of things design and implementation*, pages 49–58, 2019.

[13] George Vavoulas, Charikleia Chatzaki, Thodoris Malliotakis, Matthew Pediaditis, and Manolis Tsiknakis. The mobiact dataset: Recognition of activities of daily living using smartphones. In *International Conference on Information and Communication Technologies for Ageing Well and e-Health*, volume 2, pages 143–151. SciTePress, 2016.

[14] Pierre Rougé, Ali Moukadem, Alain Dieterlen, Antoine Boutet, and Carole Frindel. Anonymizing motion sensor data through time-frequency domain. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2021.

[15] Valentine Bargmann. On a hilbert space of analytic functions and an associated integral transform part i. *Communications on pure and applied mathematics*, 14(3):187–214, 1961.

[16] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.