Souza Leite, Clayton; Xiao, Yu

Improving Cross-Subject Activity Recognition via Adversarial Learning

Published in:
IEEE Access

DOI:
10.1109/ACCESS.2020.2993818

Published: 11/05/2020

Document Version
Publisher's PDF, also known as Version of record

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 19, 2020, accepted May 5, 2020, date of publication May 11, 2020, date of current version May 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993818

Improving Cross-Subject Activity Recognition via Adversarial Learning

CLAYTON FREDERICK SOUZA LEITE AND YU XIAO
Department of Communications and Networking, Aalto University, 02150 Espoo, Finland
Corresponding author: Yu Xiao (yu.xiao@aalto.fi)

This work was supported in part by the Business Finland under Grant 1660/31/2018, and in part by the European Union's Horizon 2020 Research and Innovation Programme under Grant 777222.

ABSTRACT Deep learning has been widely used for implementing human activity recognition from wearable sensors like inertial measurement units. The performance of deep activity recognition is heavily affected by the amount and variability of the labeled data available for training the deep learning models. On the other hand, it is costly and time-consuming to collect and label data. Given limited training data, it is hard to maintain high performance across a wide range of subjects, due to the differences in the underlying data distribution of the training and the testing sets. In this work, we develop a novel solution that applies adversarial learning to improve cross-subject performance by generating training data that mimic artificial subjects - i.e. through data augmentation - and enforcing the activity classifier to ignore subject-dependent information. Contrary to domain adaptation methods, our solution does not utilize any data from subjects of the test set (or target domain). Furthermore, our solution is versatile as it can be utilized together with any deep neural network as the classifier. Considering the open dataset PAMAP2, nearly 10% higher cross-subject performance in terms of F1-score can be achieved when training a CNN-LSTM-based classifier with our solution. A performance gain of 5% is also observed when our solution is applied to a state-of-the-art HAR classifier composed of a combination of inception neural network and recurrent neural network. We also investigate different influencing factors of classification performance (i.e. selection of sensor modalities, sampling rates and the number of subjects in the training data), and summarize a practical guideline for implementing deep learning solutions for sensor-based human activity recognition.

INDEX TERMS Human activity recognition, deep learning, adversarial learning, data augmentation, cross-subject performance.

I. INTRODUCTION

Deep learning techniques such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) have recently been applied to implement human activity recognition (HAR) using wearable sensors, and have proved to outperform shallow learning techniques like Support Vector Machine (SVM). Examples, as listed in Table 1, include recognition of hand gestures (e.g. raise, lower) and body movements (e.g. walking, sitting) from readings of inertial measurement unit (IMU). Among the activities which have not been well studied, the ones involving hand-object interaction are essential for implementing emerging augmented/mixed reality applications, such as cognitive assembly and maintenance assistance. In this work, we will take activities involved in the process of elevator panel maintenance as an example, and investigate the challenges and practical solutions of deep learning-based hand-object interaction recognition.

One key challenge in applying deep learning for HAR is the **cross-subject performance degradation**. As different subjects conduct the same activities in different ways, the gap in data distribution between the training and testing sets often causes significant performance degradation when testing the trained deep learning models on subjects not included in the training set. Ideally, this issue could be addressed by having a training set composed of data recorded with tens, or possibly hundreds, of different subjects. However, data collection and labeling is a laborious and time-consuming task. As a matter of fact, existing open datasets, such as PAMAP2 [29],

The associate editor coordinating the review of this manuscript and approving it for publication was Xian Sun.

TABLE 1. Recent works on sensor-based human activity recognition.

Reference	Methods	Type of sensors	Datasets
Xu et al. [36], 2019	Inception network architecture with CNN and Gated Recurrent Unit (GRU)	IMU	Opportunity [3], PAMAP2 [29] and smartphones dataset (walking, sitting, standing, lying, etc.)
Long et al. [17], 2019	Residual neural network with CNN and LSTM layers	IMU	Opportunity and UniMiB SHAR (standing up, lying down, running, sitting down, walking, falling, etc.)
Li et al. [15], 2018	LSTM, Bidirectional LSTM (Bi-LSTM), GRU, Bi-GRU, and Hidden Markov Models (HMM)	IMU	12 dynamic hand gestures (drawing in the air a few English letters and numerals)
Peng et al. [27], 2018	CNN-LSTM	IMU	Opportunity [3]
Münzner et al. [22] 2017	CNN	Accelerometers and gyroscopes	Activities as walking, ascending and descending stairs, standing and sitting.
Mohammad et al. [19], 2017	CNN	Accelerometers	74 cooking activities
Guan et al. [7], 2017	Ensembles of LSTM	IMU	Opportunity [3], PAMAP2 [29] and Skoda [39]

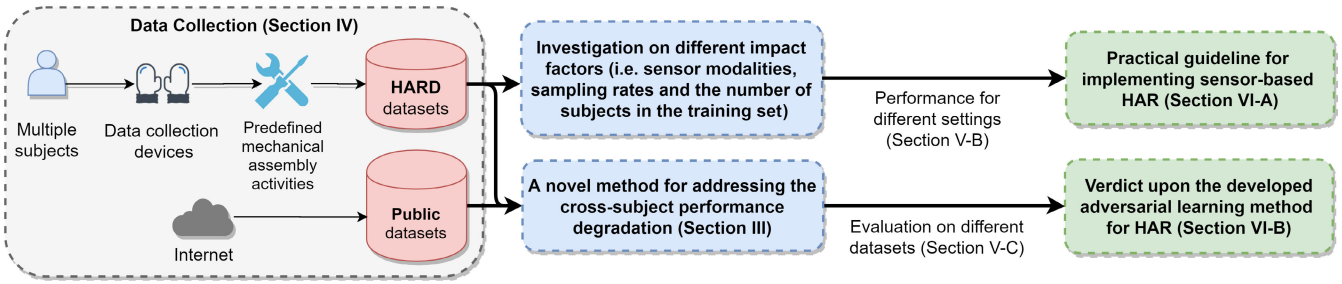


FIGURE 1. Overview of our work.

Opportunity [3] and Daphnet [1], typically contain no more than 10 subjects. Therefore, the question is how to improve cross-subject performance having limited subject-variability in the training set. To the best of our knowledge, the works of Jiang *et al.* [10] and Khan *et al.* [14] are the only ones that propose to address this problem. However, as we will discuss in Section 2, their solutions present limitations that we aim to overcome with our method: 1) the need for training with data from subjects pertaining to the test set and 2) a different model must be trained for every different single or group of test subjects.

Implementing HAR is also a choice of different sensor modalities, sampling rates, and the number of subjects to collect data with. It might sound plausible that maximizing these three factors would also result in the maximization of the classification performance at the end. However, above a certain point, further maximizing them provides negligible or nonexistent performance gains. Instead, it may cause practical issues as increased resource consumption and processing delay. Even though this is a key trade-off in implementing HAR systems, there is a lack of practical guidelines on the selection of a minimal sensible setting for the aforementioned factors, which characterizes another challenge to be addressed in HAR.

This paper aims to solve both challenges. Our key contributions in this paper are summarized as follows. 1) We develop a novel deep learning solution for bridging the gap in performance across different subjects in HAR with wearable sensors. To achieve this goal, while training the activity classifier, our solution generates additional training data that mimic artificial subjects - with the purpose of increasing subject variability - and instructs the activity classifier to ignore subject-dependent information in the data. Our solution is versatile since there isn't any restriction on which activity classifier to use. Taking a CNN-LSTM baseline as the classifier and PAMAP2 as the dataset, our solution provides a gain of nearly 10% in cross-subject performance (in terms of mean F1-score) compared to the sole use of the CNN-LSTM baseline. Applied to the state-of-the-art Inno-HAR [36] classifier, the leap in performance reaches almost 5%, also for the PAMAP2 dataset. These improvements correspond to a decreased need for variability of subject behavior in the training set, which can be translated into fewer subjects with which to collect and label data. 2) We provide deep insights into the impact of different influencing factors on classification performance and summarize a practical guideline based on our findings from experiments.

Figure 1 illustrates the blocks that form the structure of this work, as well as their corresponding sections. The rest of this paper is organized as follows. Section II introduces the background. Section III presents the method proposed in this work. Section IV describes the datasets, with the

experimental results presented in Section V. Section VI summarizes the practical guideline and further discusses our method and the remained issues. Section VII presents the related work before we conclude this work in Section VIII.

II. BACKGROUND

A. HUMAN ACTIVITY RECOGNITION (HAR)

HAR refers to the class of methods used for automatically understanding what task humans are performing by analyzing video, readings of wearable sensors, or wireless signals reflected by the human body [35]. The algorithms for HAR can be classified into shallow and deep learning methods. Common shallow methods in HAR include SVM [13], [20], [23], k-nearest neighbors (kNN) [16], [24], linear discriminant analysis (LDA) [9], and random forest (RF) [21]. Deep learning approaches, such as LSTM [7], [15], CNN-LSTM [25], [27], CNN [22], and convLSTM [26], have shown impressive leaps in performance compared to their shallow counterparts by learning to automatically extract features from raw sensor data, thus dropping the need for having human experts to provide hand-engineered features. A summary of recent works is listed in Table 1. Regarding the activities to be recognized, our work also serves to reinforce the scarce attention that is being given to activities involving hand-object interactions.

B. DOMAIN SHIFT

In computer vision, one often faces the problem of performance degradation when the training and the test sets present differences in terms of illumination, pose and image quality [34]. Such differences in the underlying data distribution of the training set (i.e. source domain) and the test set (i.e. target domain) are named **domain shift** (or domain gap) and may bring huge discrepancies in performance when testing the model.

In HAR, when training a deep learning model on the labeled source domain data, since the distribution of the raw data depends on the subject, it is expected that the part of the network responsible for the feature extraction process outputs subject-dependent information (features) to the classification layers. That is, the extracted features depend on the behavioral style of the subjects of the training set. Hence, the domain shift problem is also present in HAR as a result of the difference between the behavioral styles of the subjects in the training set and those in the test set. There are a few factors that determine the behavioral style of a subject.

- Different subjects might perform the same activity in significantly distinctive ways. The activity of walking, for instance, presents enough differences across subjects such that it is possible to identify people by their gait.
- The level of dexterity and speed of performing the activities also differ from subject to subject. For instance, one can observe clear differences (e.g. in speed) in the behavior of a maintenance engineer when disassembling elevator buttons in comparison with a non-technician.
- In energy-demanding activities, different subjects may experience varying levels of tiredness that change in distinctive ways how they perform the activity.
- By performing the activities, the subjects can involuntarily shift the placement of the sensors.

C. DOMAIN ADAPTATION AND DATA AUGMENTATION

We envision that the cross-subject performance degradation can be minimized by reducing the domain shift through domain adaptation or by augmenting the subject variability in the training set through data augmentation.

Domain adaptation (DA) techniques aim at reducing this performance degradation by bridging the domain gap. While there exists a handful of DA methods in the literature [4], the so-called adversarial DA methods (a subset of DA methods) have recently shown impressive results [42] and increasingly attracted the interest of many researchers [31]. In adversarial DA, a network - the discriminator - is trained to distinguish data between domains, while another network - the generator - learns to generate domain-indistinguishable data, thus confusing the discriminator. These two networks pit against each other - hence the term ''adversarial''. Following this concept, the generator could be the feature extraction layers of the activity classifier trying to learn subject-indistinguishable features, whereas the discriminator could be a network that tries to predict the subject given the extracted features. A limitation of DA methods is that they require the use of labeled or unlabeled target domain data during the training phase. In this work, we follow the concept of adversarial learning to address the cross-subject performance degradation, however, we drop the need for utilizing any data pertaining to the target domain during the training of the classifier.

Data augmentation techniques generate artificial data that are combined with the real data during the training of the classifier. Simple techniques include adding noise to the sensor readings or increasing/decreasing their magnitude. Adversarial learning have also shown impressive results in generating artificial data [12]. Again, two networks - a discriminator and a generator - are pitted against each other. The generator receives random noise as input and is required to learn how to transform such noise into an output that resembles the real data. The generator's output is fed into the discriminator, which is trained to distinguish between real and artificial data. This adversarial learning method does not guarantee that the training set - formed by artificial and real data - contains a higher subject variability, since the artificial data created by the generator should exhibit the same data distribution as the real training data. In our work, we utilize adversarial learning to generate artificial data. However, in our method, the artificial data are generated to present a different distribution from the real training data such that they mimic synthetic subjects.

III. ADVERSARIAL LEARNING

We divide this section into two parts. First, we explain the architecture of the method that generates artificial data with

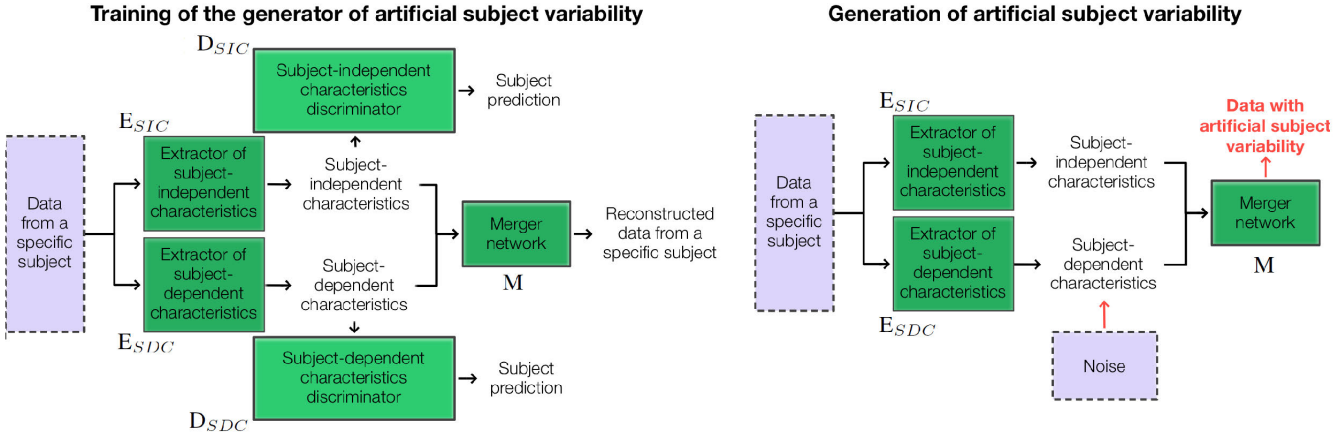


FIGURE 2. The architecture of the subject variability-oriented data augmentation. During training, this architecture is required to extract subject-dependent and subject-independent characteristics from the original data and merge these characteristics back together to reconstruct the data. During the artificial data generation, the architecture is fed with original data from which subject-dependent characteristics are extracted and altered, thus generating data from an artificial subject performing the same activity as in its original counterpart.

rich forms of subject variability. Such artificial data are combined with the original training data and used to train the activity classifier, explained in the second part of this section.

A. GENERATING ARTIFICIAL SUBJECT VARIABILITY

Denoting x as a sequence of time-series data, our goal is to find which activity (among a set of predefined activities) is performed in x. We start from the premise that x contains subject-dependent characteristics, i.e. information that can be used to classify which subject (among the set of subjects in the training data) generated x, and subject-independent characteristics. Also, let us presume that there exist functions $\mathrm{E}_{SDC}(\cdot)$ and $\mathrm{E}_{SIC}(\cdot)$ that can extract from x subject-dependent and independent characteristics,respectively. Moreover, let $\mathrm{M}(\mathrm{E}_{SDC}(x), \mathrm{E}_{SIC}(x))$ be a so-called merger function whose goal is to reconstruct the original data x from its constituents $\mathrm{E}_{SDC}(x)$ and $\mathrm{E}_{SIC}(x)$.

If we perturb $\mathrm{E}_{SDC}(x)$, we can theoretically create data that can represent artificial subjects. We refer to as $\mathrm{A}(x, \eta)$ (given in Eq. 1) the data created from x representing an artificial subject given a disturbance η to $\mathrm{E}_{SDC}(x)$.

$$A(x, \eta) = \mathrm{M}(\mathrm{E}_{SIC}(x), \mathrm{E}_{SDC}(x) \odot \eta) \tag{1}$$

where $\odot$ represents the Hadamard product - also known as the element-wise product - and η is an injected noise.

The functions $\mathrm{E}_{SDC}(\cdot)$, $\mathrm{E}_{SIC}(\cdot)$ and $\mathrm{M}(\cdot)$ are characterized by neural networks. Given that the goal of M is to reconstruct the split-data, we define the reconstruction loss function as in Eq. 2.

$$\mathcal{L}_{\mathrm{RECONS}} = \mathop{\mathbb{E}}_{x \sim p_x} [\|\mathrm{M}(\mathrm{E}_{SIC}(x), \mathrm{E}_{SDC}(x)) - x\|_2] \tag{2}$$

To split x into its two constituents, we utilize two discriminator networks denoted as $\mathrm{D}_{SDC}(\cdot)$ and $\mathrm{D}_{SIC}(\cdot)$. We require both discriminators to learn to predict, in a supervised way, the subject to whom the input is related and to achieve maximum certainty about the prediction. Hence, the classes predicted by the discriminators are subject IDs. However, differently from $\mathrm{D}_{SDC}(\cdot)$, $\mathrm{D}_{SIC}(\cdot)$ establishes a mini-max game with $\mathrm{E}_{SIC}(\cdot)$. That is, $\mathrm{E}_{SIC}(\cdot)$ tries to confuse $\mathrm{D}_{SIC}(\cdot)$ by outputting information such that it is impossible for $\mathrm{D}_{SIC}(\cdot)$ to predict which subject the information is related to, while $\mathrm{D}_{SIC}(\cdot)$ does its best to learn to distinguish between subjects in its input. This mini-max game is employed with adversarial training. Before detailing how the weights of the networks are learned with adversarial training, let us define the cross-entropy and entropy loss functions for subject classification, respectively $\mathcal{L}_{\mathrm{CES}}$ and $\mathcal{L}_{\mathrm{H}}$.

$$\mathcal{L}_{\mathrm{CES}}(O, P) = \mathop{\mathbb{E}}_{x \sim p_x} [\sum_{i=1}^{N} u_i \log(\mathrm{O}_i(\mathrm{P}(x)))] \tag{3}$$

where N is the number of subjects in the training data, u_i and $\mathrm{O}_i(\mathrm{P}(x))$ are the label and the probability prediction for subject i given by a function $\mathrm{O}(\cdot)$ to a transformation $P(x)$ of x, respectively.

$$\mathcal{L}_{\mathrm{H}}(\mathrm{O}, \mathrm{P}) = \mathop{\mathbb{E}}_{x \sim p_x} [H(\mathrm{O}(\mathrm{P}(x)))] \tag{4}$$

where $H(\cdot)$ is the Shannon entropy function.

From these functions, the weights of $\mathrm{E}_{SIC}(\cdot)$ and $\mathrm{D}_{SIC}(\cdot)$ (Eq. 5 and Eq. 6) are learned in an adversarial approach as in the vanilla GANs [6], with the exception that the concept of source and target domains is not valid here. Instead, each subject represents a domain and $\mathrm{E}_{SIC}(\cdot)$ learns to map different domains (subjects) into a common domain as in categorical GANs [30]. Note that the $\mathrm{E}_{SIC}(\cdot)$ and $\mathrm{D}_{SIC}(\cdot)$ networks represent respectively the generator and the discriminator in the common GANs scheme. In our notation, the weights of a network O are expressed as θ_O and the asterisk as in θ_O^* expresses the optimal values for θ_O.

$$\theta^*_{\mathrm{E}_{SIC}} = \arg\min_{\theta_{\mathrm{E}_{SIC}}} \lambda_{RE}\mathcal{L}_{\mathrm{RECONS}} - \lambda_{HE}\mathcal{L}_{\mathrm{H}}(\mathrm{D}_{SIC}, \mathrm{E}_{SIC}) \tag{5}$$

$$\theta^*_{\mathrm{D}_{SIC}} = \arg\min_{\theta_{\mathrm{D}_{SIC}}} \lambda_{CE}\mathcal{L}_{\mathrm{CES}}(\mathrm{D}_{SIC}, \mathrm{E}_{SIC}) + \lambda_{HD}\mathcal{L}_{\mathrm{H}}(\mathrm{D}_{SIC}, \mathrm{E}_{SIC}) \quad (6)$$

where λ_{RE}, λ_{HE}, λ_{CE} and λ_{HD} are positive real-valued constants.

The weights of $\mathrm{E}_{SDC}(\cdot)$ and $\mathrm{D}_{SDC}(\cdot)$ are given similarly (Eq. 7 and Eq. 8), however there isn't a mini-max game between these two networks - which is seen by the positive sign before $\mathcal{L}_{\mathrm{AH}}$ in Eq. 7.

$$\theta^*_{\mathrm{E}_{SDC}} = \arg\min_{\theta_{\mathrm{E}_{SDC}}} \lambda_{RE}\mathcal{L}_{\mathrm{RECONS}} + \lambda_{HE}\mathcal{L}_{\mathrm{H}}(\mathrm{D}_{SDC}, \mathrm{E}_{SDC}) \quad (7)$$

$$\theta^*_{\mathrm{D}_{SDC}} = \arg\min_{\theta_{\mathrm{D}_{SDC}}} \lambda_{CE}\mathcal{L}_{\mathrm{CES}}(\mathrm{D}_{SDC}, \mathrm{E}_{SDC}) + \lambda_{HD}\mathcal{L}_{\mathrm{H}}(\mathrm{D}_{SDC}, \mathrm{E}_{SDC}) \quad (8)$$

Finally, the optimal weights of the merger network (Eq. 9) are simply given as the result of the minimization of the reconstruction loss function. Figure 2 illustrates the scheme of the generation of artificial subject variability.

$$\theta^*_M = \arg\min_{\theta_M} \mathcal{L}_{\mathrm{RECONS}} \quad (9)$$

B. THE CLASSIFIER

We indicate as $F(\cdot)$ and $C(\cdot)$ as the feature extraction and the classification layers, respectively, of the activity classifier. First, let us the define the cross-entropy function of the activity classification $\mathcal{L}_{\mathrm{CL}}$.

$$\mathcal{L}_{\mathrm{CL}} = \mathop{\mathbb{E}}_{x\sim p_x}[\sum_{i=1}^{K} y_i(\lambda_T \log(\mathrm{C}_i(x)) + \lambda_A \log(\mathrm{C}_i(A(x,\eta))))] \quad (10)$$

where K is the number of activity classes, y_i is the label for class i of the labeled sample x, and λ_A and λ_T are positive real-valued constants that weigh the importance of correctly classifying the original and the artificial data, respectively. Notice that the loss function includes both real data x and artificial data $A(x, \eta)$.

The optimal weights of the classification layers (Eq. 11) can be promptly defined as those which minimize $\mathcal{L}_{\mathrm{CL}}$. Since the feature extraction layers $F(\cdot)$ are required to learn subject-independent features, we employ a third discriminator $\mathrm{D}_{SIF}(\cdot)$ whose goal is to play a mini-max game with the feature extraction layers similar to the case of $\mathrm{E}_{SIC}(\cdot)$ and $\mathrm{D}_{SIC}(\cdot)$. Hence, the optimal weights of $F(\cdot)$ and $\mathrm{D}_{SIF}(\cdot)$ are expressed in Eq. 12 and Eq. 13, respectively. Figure 3 illustrates the scheme involving the activity classifier.

$$\theta^*_{\mathrm{C}} = \arg\min_{\theta_{\mathrm{C}}} \mathcal{L}_{\mathrm{CL}} \quad (11)$$

$$\theta^*_{\mathrm{D}_{SIF}} = \arg\min_{\theta_{\mathrm{D}_{SIF}}} \lambda_{CE}\mathcal{L}_{\mathrm{ACE}}(\mathrm{D}_{SIF}, \mathrm{F}) + \lambda_{HD}\mathcal{L}_{\mathrm{AH}}(\mathrm{D}_{SIF}, \mathrm{F}) \quad (12)$$

$$\theta^*_{\mathrm{F}} = \arg\min_{\theta_{\mathrm{F}}} \lambda_{CL}\mathcal{L}_{\mathrm{CL}} - \lambda_{HF}\mathcal{L}_{\mathrm{AH}}(\mathrm{D}_{SIF}, \mathrm{F}) \quad (13)$$

where λ_{CL} and λ_{HF} are positive real-valued constants.

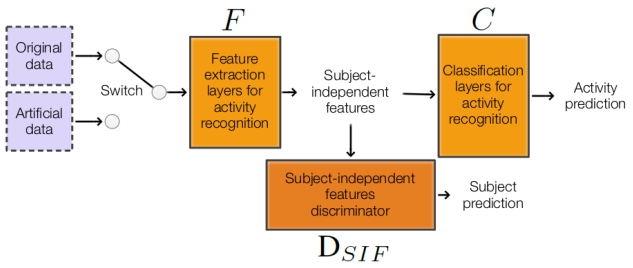


FIGURE 3. The activity classifier.

It should be noted that once the training has been completed, all networks, except for the feature extraction and classification layers, can be discarded as they only served the purpose of assisting the activity classifier in obtaining robustness against cross-subject performance degradation. Furthermore, we explicitly differentiate between the extracted subject-independent characteristics (the output of $\mathrm{E}_{SIC}(\cdot)$) and the subject-independent features (the output of $\mathrm{F}(\cdot)$). The reason for this is that the loss functions for learning $\mathrm{E}_{SIC}(\cdot)$ and $\mathrm{F}(\cdot)$ differ, hence it is clearly not expected that $\mathrm{E}_{SIC}(\cdot) = \mathrm{F}(\cdot)$. The overall step-by-step algorithm is detailed in Algorithm 1. We used a fixed number of epochs as the convergence criteria for the training of the networks. To improve the stability of the adversarial training, we have forced Lipschitz continuity through spectral normalization [18] on all networks except for the activity classifier.

Note that there isn't any restriction concerning the structure of the activity classifier. This is the versatility of our method. As a matter of fact, in Section V, we apply our method with two different activity classifiers: a CNN-LSTM baseline and InnoHAR [36].

IV. DATASETS

A. THE USED DATASETS

1) HARD

We collected three different datasets - namely HARD, HARD2 and HARD3 - of hand activities reproducing an elevator maintenance process. We asked participants to perform 7 different activities: 1) press buttons, 2) unplug the elevator cables, 3) plug the elevator cables back in, 4) remove the panel's button, 5) insert the buttons on the panel, 6) use a screwdriver to loosen and tighten screws in the panel and 7) use a hammer with the purpose of only mimicking the movement of hitting an object. Moreover, the null class is also considered, resulting in 8 different classes. Each participant took roughly 15 minutes to perform all the requested activities. The first setup (i.e. HARD) uses flex sensors on all fingers of both hands, thumb pressure sensors on both hands, and accelerometers on the back on each hand. The data were recorded at 25Hz with 19 different subjects. In the second setup (HARD2), gyroscopes on the back of each hand were used in addition to those sensors of the first setup, however, now the data were recorded at a sampling rate of 16.67Hz

Algorithm 1 Training of Our Method

Load training data D_{TRAIN}
Create networks $F, C, E_{SIC}, E_{SDC}, D_{SIC}, D_{SDC}, M$, and D_{SIF}
Define $\lambda_T, \lambda_A, \lambda_{CL}, \lambda_{CE}, \lambda_{HD}, \lambda_{HF}, \lambda_{HE}$, and λ_{RE}
while *not converged* **do**
  for *all* x_{train} *in* D_{TRAIN} **do**
    Compute gradients of $\mathcal{L}_{\text{RECONS}}, \mathcal{L}_{\text{H}}(D_{SIC}, E_{SIC}), \mathcal{L}_{\text{CES}}(D_{SIC}, E_{SIC}), \mathcal{L}_{\text{H}}(D_{SDC}, E_{SDC}), \mathcal{L}_{\text{CES}}(D_{SDC}, E_{SDC})$
    Perform optimization step on $\theta_{E_{SIC}}, \theta_{D_{SIC}}, \theta_{E_{SDC}}, \theta_{D_{SDC}}, \theta_M$
  end
end
while *not converged* **do**
  for *all* x_{train} *in* D_{TRAIN} **do**
    Sample random noise η
    Compute $A(x_{train}, \eta)$
    Compute gradients of $\mathcal{L}_{\text{CL}}, \mathcal{L}_{\text{ACE}}(D_{SIF}, F), \mathcal{L}_{\text{AH}}(D_{SIF}, F)$
    Perform optimization step on θ_F, θ_C, and $\theta_{D_{SIF}}$
  end
end

with 9 subjects. The last setup (HARD3) was only recorded with 4 subjects using accelerometers on each hand at a rate of 104Hz.

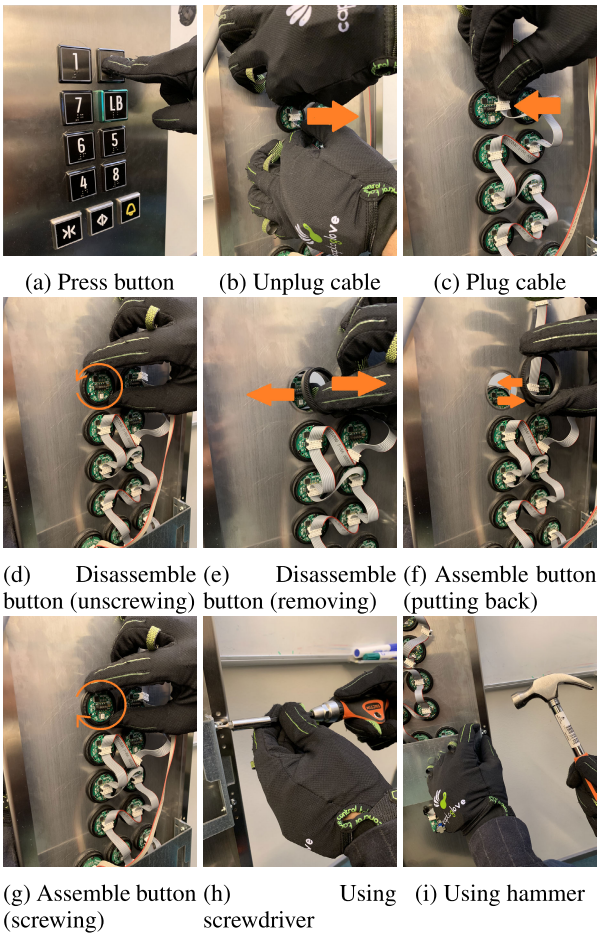
2) PAMAP2

The PAMAP2 dataset [29] includes data collected from a heart monitor and three IMUs attached to the chest, hand, and ankle of the subject, respectively. There are in total 18 different physical activities performed by 9 different participants, as well as transient activities labeled as the null class.

Out of the 18 activities, 6 are rarely present in the data. To avoid having a heavily imbalanced dataset and following previous works [7], only the remaining 12 activities are considered in our experiments: lying quietly, sitting, standing, ironing, vacuum cleaning, ascending stairs, descending stairs, walking, Nordic walking, bicycling, running, and rope jumping. Furthermore, the sampling rate is reduced from 100Hz to 33.3Hz (higher sampling rates than 33.3Hz do not show any improvement in the performance, but add further computational cost and memory footprint), and the missing values present in the raw data (reported as NaN values), as well as data originated by transient activities, are discarded.

3) OPPORTUNITY

This dataset was recorded from 4 participants with 23 body-worn sensors. It incorporates 18 domestic activities: cleaning a table, opening/closing the fridge, opening/closing the dishwasher, opening/closing 3 different drawers, opening/closing 2 different doors, toggling lights on and off, and drinking from a standing and sitting position. The sampling frequency was set to 30Hz. When running the experiments, we considered sensory readings (as in [7]) from the upper limbs, the back, and both feet.



(a) Press button (b) Unplug cable (c) Plug cable
(d) Disassemble button (unscrewing) (e) Disassemble button (removing) (f) Assemble button (putting back)
(g) Assemble button (screwing) (h) Using screwdriver (i) Using hammer

FIGURE 4. Tasks used in the HARD datasets.

4) DAPHNET

The Daphnet [1] dataset uses three wearable accelerometers placed on the ankle, thigh, and trunk of eight Parkinson's disease patients to detect freezing of gait (FOG). FOG is a condition that causes sudden impediments of walking elevating the risk of falls. The Daphnet data was recorded during various walking tasks of 10 different participants and have three different annotations: 1) transient activities (which are discarded here), 2) freezing of gait, and 3) normal movements. The data were recorded with a sampling rate of 64Hz, however, we downsample it to 32Hz by decimation and discard the transient activities, following [40].

B. DATA PRE-PROCESSING

As a pre-processing step, all the data is normalized to zero mean and unit variance. We choose a sliding window of approximately 2.5 seconds for the HARD, HARD2 and HARD3 datasets, with 50% of overlapping. Following other works [8], [26], [40], the PAMAP2 and Daphnet datasets have, respectively, a window size of approximately

TABLE 2. Network architectures. The Rectified Linear Unit (ReLU) was used as activation function after the CNN layers and the first fully-connected (FC) layer in the D_{SIC}, D_{SDC}, D_{SFI}, and *C* networks. In these networks, the second FC layer contains the same number of neurons as the number of classes and is followed by the *softmax* function. The convolutional kernel for all CNN layers was set to 3 × 3, whereas the max-pooling kernel size was 2 × 2. The number of filters in the three CNN layers of F, E_{SIC}, E_{SDC} networks are, respectively, 8, 16, and 32. In the three CNN layers of D_{SIC}, D_{SDC}, D_{SFI}, the number of filters are 8, 4, and 2, respectively. 16, 8 and 1 are the number of filters in the transposed CNN layers of the *M* network.

Network	Structure
F, E_{SIC}, E_{SDC}	Three CNN layers with max-pooling operation after each layer
D_{SIC}, D_{SDC}, D_{SFI}	Three CNN layers with max-pooling operation after each layer accompanied by two FC layers.
M	Three transposed CNN layers
C	An LSTM layer with 64 hidden units followed by two FC layers.

5.12 seconds and one second with 78% and 50% of overlap. Following [8], the sliding window size for the Opportunity dataset was set to 1 second with 50% of overlap. The label for each sliding window corresponds to the activity whose duration occupies the largest percentage of the window.

V. EVALUATION

We implemented the workflow illustrated in Figure 1, and will present the experimental setup and results of each step in this section. The experiment contains two parts. The first part focuses on the evaluation of classification performance and its influencing factors without applying our proposed method - i.e. without using artificially generated training data and without requiring the feature extraction layers to learn subject-independent features. The second part applies our novel method and evaluates its effectiveness in improving cross-subject performance. In both parts, we choose the mean (over all classes) F1-score as a performance metric and calculate it following Eq. 14. In cases of imbalanced class distribution, a common case in HAR, the mean F1-score can prove particularly more meaningful than the accuracy metric [26].

$$\overline{F}_1 = \frac{1}{K}\sum_{i=1}^{K}\frac{2TP_i}{2TP_i + FP_i + FN_i}, \tag{14}$$

where TP_i, FP_i and FN_i represent the number of true positives, false positives and false negatives of a class i, respectively. The number of classes is given by K.

A. EXPERIMENTAL SETUP

The network architectures are described in Table 2. Note that the activity classifier (composed of the feature extraction layers F and the classification layers C) follows a CNN-LSTM architecture. This choice was influenced by its superior performance compared to other basic networks [27]. The hyper-parameters were chosen by trial and error instead of using any automatic hyper-parameter tuning methods such as grid or random search for the following reasons. Firstly, the process of hyper-parameters search requires heavy computation. Due to limited computational resources, it is expected to reduce the number of searches during model training. Secondly, since the methods proposed here can easily lead to an imbalanced competition between the networks trained in an adversarial way, we need a human in the loop to understand the effects of each hyper-parameter and propose meaningful values for them. In Section VI, based on our experience in fine-tuning by trial and error, we provide a brief guideline on how to choose sensible values for the hyper-parameters.

All the aforementioned networks were coded in Python 3.7.4 using the TensorFlow 2.0 framework. We used an NVIDIA Tesla V100 to run the code. To guarantee the reproducibility of results, the initial seed for all random operations was chosen to be zero. We used Adam as the optimization algorithm with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. As the convergence criteria, we used a fixed number of epochs - 50 epochs and 150 epochs for the experiments of Section V-B and Section V-C, respectively.

B. CLASSIFICATION WITHOUT ADVERSARIAL LEARNING

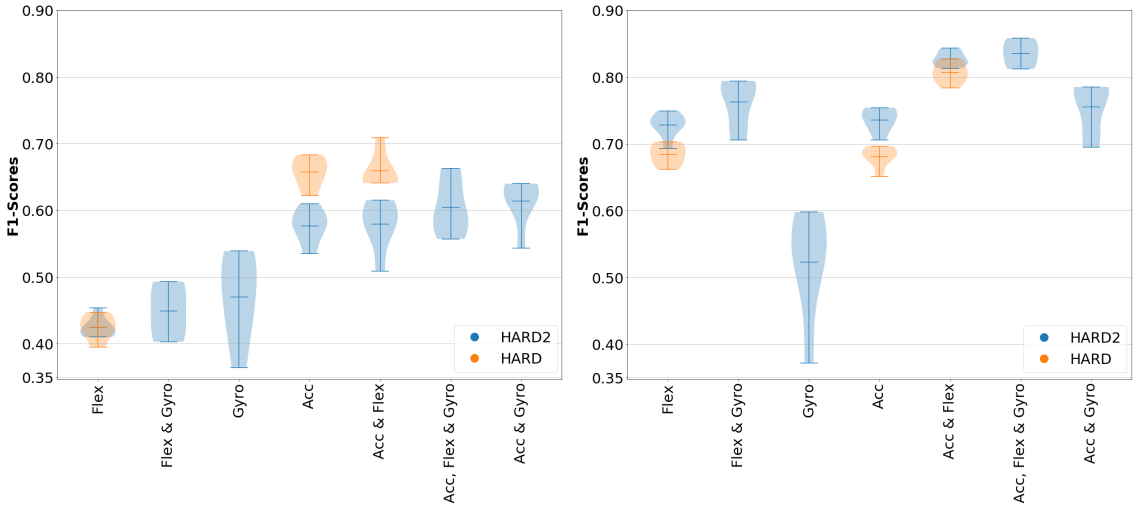
To evaluate the impact of different factors on the classification performance, we compare the performance between different combinations of sensor modalities, sampling rates and numbers of subjects in the training set, respectively. The CNN-LSTM architecture used for the tests in this section is formed by the F and C networks shown in Table 2.

1) SELECTION OF SENSOR MODALITIES

HARD and HARD2 were collected using the smart gloves equipped with flex sensors, accelerometer, and gyroscope. For comparison, we tested the data collected with 7 different configurations: 1) flex sensors only, 2) accelerometer only, 3) gyroscope only, 4) flex sensors and accelerometer, 5) flex sensors and gyroscope, 6) accelerometer and gyroscope, and 7) flex sensors, accelerometer and gyroscope.

The effect of each sensor is measured in both cross-subject and same-subject scenarios. In the cross-subject scenario, one random subject was chosen to compose the validation set and another one for the test set. The data from the remaining subjects formed the training set - that is, 17 and 6 subjects, respectively, for the training set of the HARD and the HARD2. In the same-subject scenario, the entire dataset was randomly divided into training (60%), validation (20%) and test (20%) sets. For both scenarios and for each different configuration of sensor modality, we performed six different experiments with varying subjects in the training, validation and test sets. The results are averaged over these six runs of tests.

In the cross-subject scenario, as shown in Figure 5, the accelerometer readings are more informative than those of the flex sensor or the gyroscope. The significantly low



(a) cross-subject scenario (b) same-subject scenario

FIGURE 5. Violin plots of the effects of accelerometers, gyroscopes and flex sensors in the performance of classification models of hand activities.

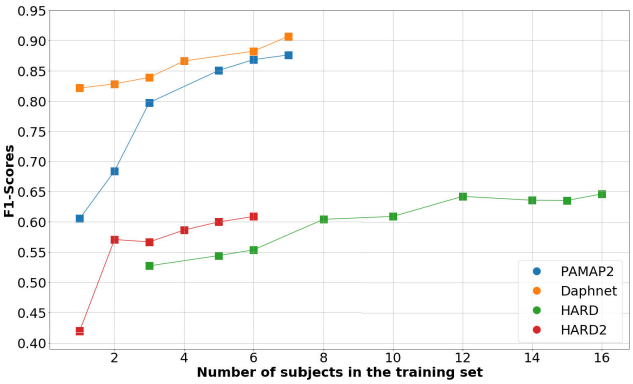


FIGURE 6. The cross-subject performance increase with the number of subjects in the training set .

performance of the flex sensors can be attributed to the fact that there exists a huge variability in the ways the subjects move their fingers to perform a certain activity. Also, we have noticed that during the data collection sessions, the flex sensors inside the gloves can slide along the finger, thus constantly changing its position. However, in the same-subject scenario, the flex sensors can be as informative as the accelerometer.

Between Figure 5a and Figure 5b, there is an indication of a trade-off between the cross-subject and same-subject performance. In the cross-subject scenario, with more subjects in the training set, the classification model becomes more generalized to maintain high performance across subjects. However, in the same-subject scenario, training on a large number of subjects harms the performance. As the number of subjects in the training set increases, the extracted features of the deep learning model become more subject-independent. This is desired when our goal is to have a model that generalizes better when fed with data from a new subject. However, the loss of subject-specific features makes it more difficult for the classification layers to make correct predictions on unseen data of the subjects present on the training set.

2) THE EFFECT OF THE NUMBER OF SUBJECTS IN TRAINING SET

Using the CNN-LSTM classifier architecture, we varied the number of subjects in the training set of the HARD, HARD2, PAMAP2 and Daphnet datasets, while keeping one subject in the validation set and a different one in the test set. For each dataset and for each number of subjects in the training set, we performed 5 runs. Therefore, 5 different subjects were present in the validation and test sets considering all the runs. In each run, the training, validation and test sets were randomly generated. Figure 6 shows the evolution of the mean F1-scores as the number of subjects in the training set grows.

As we increase the number of activities, the addition of a subject in the training set is likely to impact more the performance. This can be explained as follows. The more activities we desire to classify, the higher the chances of having activities that can be performed in rather different ways by different subjects. Therefore, to learn features helpful in classifying activities irrespective of the subject, the deep learning classifier needs to be trained on subject-rich data. As an example, when we vary the number of subjects from 1 to 5 in the training set, for each additional subject included, the Daphnet dataset (solely 2 activities included) reports an average F1-score increase of 1.3%, whereas the PAMAP2 (including 12 activities) shows growth of 6.1%. In all cases, it is noticed an ever slower growth in performance - i.e. saturation - as the number of subjects is increased.

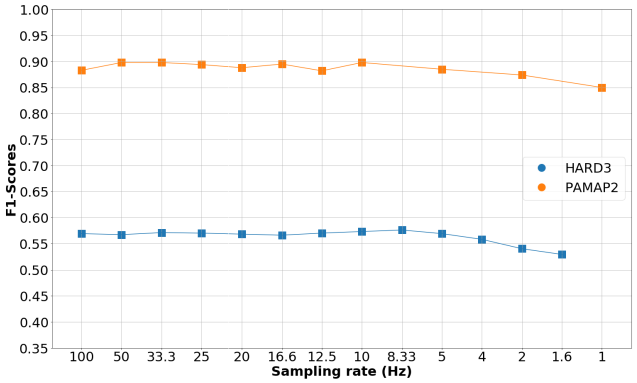


FIGURE 7. The impact of the sampling rate on the F1-scores.

3) THE EFFECT OF THE SAMPLING RATE

To evaluate the effect of the sampling rates of sensor readings on the classification performance, we downsampled the data from the PAMAP2 and HARD3 - both recorded originally at 100Hz - to various sampling rates while maintaining the same window size (5.12 and 2.5 seconds, respectively, for the PAMAP2 and HARD3 datasets) and overlapping percentage (78% and 50% for, respectively, the PAMAP2 and HARD3 datasets). Subjects 5 and 6 form the validation and test sets, respectively, for the PAMAP2 dataset. This is a common choice of subjects in the literature [7], [8], [26], [37], [40]. Decimation was used as the downsampling method. The CNN-LSTM classifier is also used here.

From Figure 7, in the range of 100-10Hz, we only observe very small and random variations in the performance of the classifier. This indicates that, for the considered activities, it is unnecessary to sample data at rates above 10Hz (when the maximum sampling rate available is 100Hz). While there isn't any appreciable performance variation in the 100-10Hz range, the elapsed time for performing a forward and a backward pass on the network is, respectively, roughly 7x and 3.5x longer at 100Hz than at 10Hz. It is also highly dubious that, for these activities, a sampling rate in the range of 100-1000Hz would provide any benefit in terms of prediction performance.

C. CROSS-SUBJECT PERFORMANCE WITH ADVERSARIAL LEARNING

In the evaluation of our method for cross-subject performance improvement, we utilized four datasets: Opportunity, HARD, HARD2, and PAMAP2. The Daphnet dataset was discarded. As discussed earlier, the Daphnet dataset includes only 2 very simple activities and, according to the experiments of Section V-B2, it did not exhibit appreciable cross-subject performance degradation. Additionally to having a CNN-LSTM network as the activity classifier, we also performed tests having InnoHAR [36] - composed of inception layers followed by GRU layers - as the activity classifier since it has exhibited state-of-the-art performance in HAR. The goal is to compare the performance of each of these two activity classifiers with and without our adversarial learning method.

TABLE 3. Number of subjects in each dataset.

	Datasets			
	OPP.	HARD	HARD2	PAMAP2
Number of subjects	4	19	8	9

Denoting n as the number of subjects in a particular dataset, we have performed n different experiments for the dataset. Each experiment contains a different subject in the test set. The same is valid for the validation set. Therefore, the training set is always composed of $n - 2$ subjects. Table 3 lists the number of subjects for each dataset. The performance of all experiments for a particular dataset is then averaged. We remind that our method does not utilize any data from the validation or test set for training. Table 4 presents the results of all the experiments for all the considered datasets. The performances for DeepConvLSTM [26] and the LSTM with Uniqueness Attention [41] are also reported. However, these classifiers have not been used with our method since overall they do not perform as good as InnoHAR, which is already being combined with our method.

We were able to obtain an average improvement of 3.41% when utilizing our method combined with either the CNN-LSTM or the InnoHAR classifier. We estimate that such an improvement may be equivalent to adding 2-6 subjects to the training set. We emphasize that this performance gain was achieved without utilizing any data from subjects belonging to the test set. Domain adaptation techniques may achieve higher improvements in terms of performance. However, they require unlabeled or partially labeled data from subjects of the test set, which signifies additional burden in collecting and partially labeling data. Quantifying the difference in improvement between our method and domain adaptation methods is a topic for future research.

Our CNN-LSTM classifier has 2 orders of magnitude fewer parameters and is 1 order of magnitude computationally lighter compared to InnoHAR. Therefore, our method is able to provide gains in performance comparable to having a more complex network architecture. As an example, for the PAMAP2 case, our CNN-LSTM classifier achieved even significantly higher performance than InnoHAR when used with our adversarial learning method.

We hypothesize that 4 different factors can determine the performance gain for a certain dataset:

1) The nature of the activities in the dataset. Different activities present distinct levels of variability across subjects. In general, activities of higher complexity - e.g. preparing a sandwich - allow for a greater level of variation across subjects than simpler activities as pressing a button. The PAMAP2 dataset includes activities of higher complexity compared to the other datasets used in this work. As a matter of fact, we believe this is the main reason for the significantly higher performance gain observed with respect to the other datasets.

TABLE 4. Results on the classification performance. Abbreviations: OPP - Opportunity, PA. - PAMAP2.

Methods	Datasets			
	OPP.	HARD	HARD2	PAMAP2
CNN-LSTM	0.6031	0.5816	0.5000	0.7259
CNN-LSTM + Our Method	0.6381	0.6005	0.5301	**0.8156**
InnoHAR [36]	0.6550	0.6113	0.5889	0.7663
InnoHAR [36] + Our Method	**0.6765**	**0.6266**	**0.6049**	0.8130
DeepConvLSTM [26]	0.5950	0.5450	0.4978	0.7150
LSTM with Uniqueness Attention [41]	0.6292	0.5316	0.4960	0.7854

2) The number of sensors. Decoupling subject-dependent and subject-independent characteristics becomes harder when the number of sensors increases, since the data becomes more complex to learn from.
3) The amount of data per subject. Higher amounts of data per subject help in the learning process of subject-dependent and subject-independent characteristics.
4) The number of subjects in the training set. The aforementioned learning process is negatively affected when the number of subjects is scarce. On the other hand, the purpose of our method is to help the activity classifier in learning subject-independent features without resorting to having an exceedingly high number of subjects. For the HARD dataset - with 19 subjects - the performance gain is slightly smaller than for the HARD2 dataset - with 8 subjects - even though both datasets have the same activities and a similar number of sensors and amount of data per subject.

With respect to the classifier, the CNN-LSTM classifier showed a slightly higher performance gain (4.34%) compared to the InnoHAR classifier (2.48%). We speculate that the higher dimensionality of the features in the activity classifier leads to a harder adversarial learning process between the feature extraction layers of the classifier and the subject-independent features discriminator. In computer vision, this is equivalent to the limitation of GANs in generating high-resolution images and is a well-known open problem [11].

VI. DISCUSSION

Based on the experiments carried out in this work, we summarize the practical guideline for sensor-based HAR. Our adversarial learning method, along with its limitations and possible areas for future work, is also discussed.

A. PRACTICAL GUIDELINE

We have seen that among flex sensors, gyroscopes, and accelerometers, the latter ones are more recommended for implementing sensor-based HAR since they provide more helpful and less subject-dependent information that led to considerably higher classification scores in cross-subject scenarios. Gyroscopes, when used with accelerometers, can lead to significantly better results in both cross and same-subject cases, however, using gyroscopes by themselves is not recommended. The use of the flex sensors is not appropriate for cross-subject scenarios, as these sensors extract quite subject-variant data. We only advise using flex sensors in combination with accelerometers and gyroscopes in a same-subject case.

We have seen that the number of subjects to include in a training dataset depends on how much the activities we want to classify can differ from one subject to the other. We recommend using approximately 5 different subjects to compose the training dataset in sensor-based HAR, as we have seen that empirically this is a number that balances the time-consuming task of data collection and labeling and the performance scores in cross-subject scenarios.

Finally, when the available sampling rate does not exceed 100Hz, a choice of 15Hz keeps both data transmission and processing times at more suitable values for real-time implementation of HAR without any degradation in the classification performance. We have not studied the effects of a sampling rate above 100Hz. It is possible that, for instance, in case one desires to recognize activities directly related to the use of highly vibrating machines (e.g. hairdryer or electric screwdriver), sampling at lower than 100Hz may not be enough to correctly distinguish between activities. On the other hand, the trade-off between transmission delays and sampling rate also needs to be taken into account in case of real-time HAR. Future work could revolve around the inclusion of other modalities of sensors as sEMG, as well as the effect of the sampling rate in activities related to machine operation.

B. THE ADVERSARIAL LEARNING METHOD FOR HAR

Regardless of which sensor modalities are present in the data, our adversarial learning method was able to provide performance improvements in all cases, especially for the PAMAP2 dataset. The duration of one epoch of training using our method is approximately twice as much as solely training using the activity classifier. Considering the PAMAP2, the training duration utilizing 150 epochs and run on an NVIDIA Tesla V100 lasted for approximately 4.2 hours. Training only the activity classifier for 50 epochs lasted for roughly 42 minutes on the same GPU. This is acceptable since the duration of the training represents only a small fraction of the total time taken to collect, label, and prepare the data for training. Most importantly, the inference time is never affected since all other neural networks, except for the activity classifier (networks F and C), are discarded. Also, there isn't any reason to suspect that the practical guideline detailed previously doesn't hold true when applying our method.

Our method utilizes 8 networks and 8 real-valued constants. To reduce time and resource-consuming efforts associated with the search for optimal hyper-parameters, we have compiled a guide (Table 5) based on all our experiments. It should be noted that the hyper-parameters that led to

TABLE 5. Recommended values for the hyper-parameters of our proposed method.

Networks	Learning rate interval	Constants	Architecture
C	[1e-4, 1e-3]	$\lambda_T = 1$, $\lambda_{CL} = 1$ $\lambda_A = [0.25, 0.75]$, Noise $\sim N(0,\sigma^2)$, with σ in [1, 10], $\lambda_{HF} = [0.1, 0.6]$	LSTM and FC Classifier
F			CNN
D_{SIF}	[2e-5, 2e-4]	$\lambda_{CE}, \lambda_{HD} = [0.5, 1]$	CNN
E_{SIC}	[2.5e-4, 2.5e-3]	$\lambda_{RE} = [1, 2]$, $\lambda_{HE} = [0.1, 0.6]$	CNN
E_{SDC}			CNN
M			Transposed CNN
D_{SDC}		$\lambda_{CE}, \lambda_{HD} = [0.5, 1]$	CNN Classifier
D_{SIC}	[5e-5, 5e-4]		

the best validation performance in one dataset might not serve to a different dataset. Therefore, even though this guide is based on experiments with diverse datasets, its only purpose is to serve as a starting point.

Our fine-tuning by trial and error followed the principle of maintaining a balanced adversarial competition between networks and we always used the performance on the validation set to make comparisons between choices of hyper-parameters. We observed that setting the learning rates of the networks D_{SIC} and D_{SIF} to lower values compared to those of the F and E_{SIC} networks resulted in better performance. This is due to the fact that the first group of networks has a simpler task than the latter group. Starting with a lower value for the magnitude of the noise and gradually increasing it - until a performance drop became evident - also proved to be a good practice. It was also noticed that assigning slightly lower values for λ_{CE} and λ_{HD} compared to λ_{HE} and λ_{HF} produced better results. However, we are unsure about the reasons for this. These relations between the mentioned parameters showed to be consistent across the datasets used. Concerning the parameters λ_{RE} and λ_A, we did not observe consistent relations. Nevertheless, we were able to determine an appropriate interval for each of them.

As a way to generate artificial subject variability, we have injected noise into the subject-dependent characteristics before the reconstruction. It is reasonable to conjecture that, in some cases, this can result in synthetic data with unrealistically fabricated subject variability. As future work, we would like to unravel, at least to some extent, the black-box nature of this process in order to obtain artificial data that more faithfully represent the reality. Furthermore, the performance obtained in the test set is sensitive to the hyper-parameters used during the training phase. As future work, an automatic search for sensible hyper-parameters - can be researched.

VII. RELATED WORK

A. DOMAIN ADAPTATION

In [2], the authors used shallow DA approaches to bridge the domain gaps across people's age, sensor placement and the environment in HAR. In some cases, they achieved a significant increase in performance ranging between 8% and 12%. In other cases, however, the DA approaches reduced the performance. Their experiments were conducted with public datasets as PAMAP2.

In the work of Wang *et al.* [33], the authors developed a DA method for different scenarios: adaptation between similar body parts on the same person, different body parts on the same person, and similar body parts on different people. Their method was evaluated with public datasets, as PAMAP2 and OPPORTUNITY, against six common alternatives performing on average better.

Ye [38], to address the scarcity of labeled data in a certain dataset, proposed a method to leverages labeled data from different domains (in this case, datasets), providing a significant improvement on the performance of activity recognition models even when only a small fraction of annotated data of the target domain is available.

In [14], Khan *et al.* performed DA in HAR in cross-device (smartphone to smartwatch and vice-versa) and cross-subject scenarios. Their method - named HDCNN - consists of first training a deep learning model on the labeled source domain data and then adapting it to the unlabeled target domain. In the adaptation, the authors proposed to minimize the Kullback-Leibler divergence between the weights of the source domain model and those of the target domain.

In [10], the authors proposed a device-free HAR system that uses adversarial DA to bridge the gap between different domains, each of which representing different physical environments and different groups of subjects. Their solution consists of feature extraction layers that are trained to output environment and subject-independent information.

The aforementioned works present two unaddressed issues: 1) during training, they need to utilize data from subjects on which the HAR algorithm will be tested (i.e. target domain data), which results in an additional burden even if their methods do not require such data to be labeled; and 2) a different model must be trained for every single or group of test subjects. In [10], even if the feature extraction layers are trained to remove subject-specific information, in practice, they are still limited to cut out only subject-specific characteristics seen in the training data. Therefore, it doesn't completely solve the problem and still leaves room for improvement.

In our work, we design a HAR scheme that 1) removes the need for utilizing any data from the target domain during training and 2) aims at training the activity classifier to ignore subject variability present not only in the training data but also artificially generated subject variability that is never seen in the training data.

B. DATA AUGMENTATION

Wang *et al.* [32] used vanilla GANs to artificially generate data as a data augmentation framework for HAR. The use of simple vanilla GANs present difficulties in learning to generate data with rich subject variability since the training

is performed in such a way that the artificial data exhibit the same data distribution as the real data from the training set. Therefore, we don't find the data generation method of [32] appropriate for creating artificial subject variability.

Erol *et al.* [5] also utilized GANs to generate synthetic data. However, instead of the vanilla GAN approach, they conditioned the generator to class labels and train the discriminator to predict the class of the synthetic data given by the generator. With their own dataset, the authors achieved an improvement of approximately 3% when training the activity classifier with both real and synthetic data. Their approach is not compared with the one by Wang *et al.* [32]. For the same reason as the previously mentioned work, this one cannot be used to generate data from synthetic subjects.

Rashid and Louis [28] proposed four distinct data augmentation methods for time-series data: scaling, rotation, time-warping, and jittering. These methods are limited to IMU sensors. In scaling, the magnitude of the raw data is changed while preserving the label. Rotation applies artificial changes in the data that mimic different orientations of the sensors, considering that the labels should be invariant to such transformations. Time-warping alters how fast or slow an activity is performed. Finally, jittering simulates random additive sensor noise to increase the robustness of the classifier to small variations. Applied to their own dataset, the authors were able to achieve an accuracy improvement of at least 10%. While these techniques may help in reducing cross-subject performance degradation, it was not designed for such purpose. The authors did not claim their methods address cross-subject performance degradation nor did they perform experiments to evaluate their potential in addressing this issue.

To the best of our knowledge, our work is the first 1) to utilize data augmentation to explicitly increase subject variability in the training and 2) to perform experiments to evaluate this data augmentation scheme in addressing the cross-subject performance degradation.

VIII. CONCLUSIONS

In this paper, we have drawn the attention to an understudied yet a crucial challenge in HAR: cross-subject performance degradation. We have then proposed a novel method for addressing this adverse variance of classification performance seen across different subjects in HAR. As a result of various experiments, we have demonstrated its potential in providing appreciable performance gains that reduce the need for larger data collection and annotation procedures with various subjects, as it is common in HAR. With additional experiments related to sensor modalities, sampling rates and the number of subjects in the training set, we have proposed a practical guideline for implementing more efficient and better-performing sensor-based HAR solutions.

REFERENCES

[1] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Troster, "Wearable assistant for Parkinson's disease patients with the freezing of gait symptom," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 436–446, Mar. 2010.

[2] P. Barbosa, K. Dearo Garcia, and J. Mendes-Moreira, *Unsupervised Domain Adaptation for Human Activity Recognition* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2018, pp. 623–630.

[3] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, Nov. 2013.

[4] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *CoRR*, abs/1702.05374, pp. 1–46, Oct. 2017.

[5] B. Erol, S. Z. Gurbuz, and M. G. Amin, "GAN-based synthetic radar micro-Doppler augmentations for improved human activity recognition," in *Proc. IEEE Radar Conf. (RadarConf)*, Apr. 2019, pp. 1–5.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[7] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," in *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, Jun. 2017, vol. 1, no. 2, pp. 1–28.

[8] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 1533–1540.

[9] S. Jiang, B. Lv, W. Guo, C. Zhang, H. Wang, X. Sheng, and P. B. Shull, "Feasibility of wrist-worn, real-time hand, and surface gesture recognition via sEMG and IMU sensing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3376–3385, Aug. 2018.

[10] W. Jiang, D. Koutsonikolas, W. Xu, L. Su, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, and X. Ma, "Towards environment independent device free human activity recognition," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, 2018, pp. 289–304.

[11] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of Gans for improved quality, stability, and variation," *CoRR*, abs/1710.10196, pp. 1–26, Feb. 2017.

[12] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *CoRR*, abs/1812.04948, pp. 1–12, 2018.

[13] V. Kartsch, S. Benatti, M. Mancini, M. Magno, and L. Benini, "Smart wearable wristband for EMG based gesture recognition powered by solar energy harvester," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[14] M. A. A. H. Khan, N. Roy, and A. Misra, "Scaling human activity recognition via deep learning-based domain adaptation," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–9.

[15] C. Li, C. Xie, B. Zhang, C. Chen, and J. Han, "Deep Fisher discriminant learning for mobile hand gesture recognition," *Pattern Recognit.*, vol. 77, pp. 276–288, May 2018.

[16] K.-Y. Lian, C.-C. Chiu, Y.-J. Hong, and W.-T. Sung, "Wearable armband for real time hand gesture recognition," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 2992–2995.

[17] J. Long, W. Sun, Z. Yang, O. Ian Raymond, and B. Li, "Dual residual network for accurate human activity recognition," *CoRR*, abs/1903.05359, pp. 1–18, Mar. 2019.

[18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *CoRR*, abs/1802.05957, pp. 1–26, Feb. 2018.

[19] Y. Mohammad, K. Matsumoto, and K. Hoashi, "A dataset for activity recognition in an unmodified kitchen using smart-watch accelerometers," in *Proc. 16th Int. Conf. Mobile Ubiquitous Multimedia*, New York, NY, USA, 2017, pp. 63–68.

[20] A. Moin, A. Zhou, A. Rahimi, S. Benatti, A. Menon, S. Tamakloe, J. Ting, N. Yamamoto, Y. Khan, F. Burghardt, L. Benini, A. C. Arias, and J. M. Rabaey, "An EMG gesture recognition system with flexible high-density sensors and brain-inspired high-dimensional classifier," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[21] C. K. Mummadi, F. P. P. Leo, K. D. Verma, S. Kasireddy, P. M. Scholl, and K. Van Laerhoven, "Real-time embedded recognition of sign language alphabet fingerspelling in an IMU-based glove," in *Proc. 4th Int. Workshop Sensor-Based Activity Recognit. Interact.*, New York, NY, USA, 2017, p. 11.

[22] S. Münzner, P. Schmidt, A. Reiss, M. Hanselmann, R. Stiefelhagen, and R. Därichen, "CNN-based sensor fusion techniques for multimodal human activity recognition," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, New York, NY, USA, 2017, pp. 158–165.

[23] K. Murao and T. Terada, "A recognition method for combined activities with accelerometers," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput.*, New York, NY, USA, 2014, pp. 787–796.

[24] N. Normani, A. Urru, L. Abraham, M. Walsh, S. Tedesco, A. Cenedese, G. A. Susto, and B. O'Flynn, "A machine learning approach for gesture recognition with a lensless smart sensor system," in *Proc. IEEE 15th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Mar. 2018, pp. 136–139.
[25] T. Okita and S. Inoue, "Recognition of multiple overlapping activities using compositional CNN-LSTM model," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput.*, New York, NY, USA, Sep. 2017, pp. 165–168.
[26] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
[27] L. Peng, L. Chen, Z. Ye, and Y. Zhang, "AROMA: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors," in *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, Jul. 2018, vol. 2, no. 2, pp. 1–16.
[28] K. M. Rashid and J. Louis, "Times-series data augmentation and deep learning for construction equipment activity recognition," *Adv. Eng. Informat.*, vol. 42, Oct. 2019, Art. no. 100944.
[29] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.
[30] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *CoRR*, abs/1511.06390, pp. 1–20, Dec. 2015.
[31] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2962–2971.
[32] J. Wang, Y. Chen, Y. Gu, Y. Xiao, and H. Pan, "SensoryGANs: An effective generative adversarial framework for sensor-based human activity recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[33] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, "Deep transfer learning for cross-domain activity recognition," in *Proc. 3rd Int. Conf. Crowd Sci. Eng. (ICCSE)*, New York, NY, USA, 2018, p. 16.
[34] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.
[35] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial WiFi devices," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1118–1131, May 2017.
[36] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019.
[37] Z. Yang, O. I. Raymond, C. Zhang, Y. Wan, and J. Long, "DFTerNet: Towards 2-bit dynamic fusion networks for accurate human activity recognition," *IEEE Access*, vol. 6, pp. 56750–56764, 2018.
[38] J. Ye, "SLearn: Shared learning human activity labels across multiple datasets," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–10.
[39] P. Zappi, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, "Activity recognition from on-body sensors by classifier fusion: Sensor scalability and robustness," in *Proc. 3rd Int. Conf. Intell. Sensors, Sensor Netw. Inf.*, 2007, pp. 281–286.
[40] M. Zeng, H. Gao, T. Yu, O. J. Mengshoel, H. Langseth, I. Lane, and X. Liu, "Understanding and improving recurrent networks for human activity recognition by continuous attention," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, New York, NY, USA, 2018, pp. 56–63.
[41] Z. Zheng, L. Shi, C. Wang, L. Sun, and G. Pan, "LSTM with uniqueness attention for human activity recognition," in *Artificial Neural Networks and Machine Learning*. Cham, Switzerland: Springer, 2019, pp. 498–509.
[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

CLAYTON FREDERICK SOUZA LEITE received the B.S. degree in mechanical engineering from the Federal University of Pernambuco, Brazil, in 2015, and the M.S. degree in robotics engineering from the University of Genoa, Genoa, Italy, and the Warsaw University of Technology, Warsaw, Poland, in 2018. He is currently pursuing the Ph.D. degree with Aalto University. His current research interests include deep learning-based human activity recognition and deep model compression.

YU XIAO received the Ph.D. degree in computer science from Aalto University, in 2012. She is currently an Assistant Professor with the Department of Communications and Networking, Aalto University. Her current research interests include mobile crowdsensing, augmented reality, and edge computing.