Journal of Ambient Intelligence and Smart Environments 1 (2021) 1–15
IOS Press

Collaborative Activity Recognition with Heterogeneous Activity Sets and Privacy Preferences

Gabriele Civitarese [a,*], Juan Ye [b], Matteo Zampatti [a] and Claudio Bettini [a]
[a] *Department of Computer Science, University of Milan, Italy*
E-mails: gabriele.civitarese@unimi.it, matteo.zampatti@studenti.unimi.it, claudio.bettini@unimi.it
[b] *School of Computer Science, University of St. Andrews, UK*
E-mail: juan.ye@st-andrews.ac.uk

Abstract. One of the major challenges in Human Activity Recognition (HAR) based on machine learning is the scarcity of labeled data. Indeed, collecting a sufficient amount of training data to build a reliable recognition problem is often prohibitive. Among the many solutions in the literature to mitigate this issue, collaborative learning is emerging as a promising direction to distribute the annotation burden over multiple users that cooperate to build a shared recognition model. One of the major issues of existing methods is that they assume a static activity model with a fixed set of target activities. In this paper, we propose a novel approach that is based on Growing When Required (GWR) neural networks. A GWR network continuously adapts itself according to the input training data, and hence it is particularly suited when the users share heterogeneous sets of activities. Like in federated learning, for the sake of privacy preservation, each user contributes to the global activity classifier by sharing personal model parameters, and not by directly sharing data. In order to further mitigate privacy threats, we implement a strategy to avoid releasing model parameters that may indirectly reveal information about activities that the user specifically marked as private. Our results on two well-known publicly available datasets show the effectiveness and the flexibility of our approach.

Keywords: activity recognition, collaborative learning, semi-supervised learning, privacy

1. Introduction

Human activity recognition (HAR) is a well-established research topic [1]. The objective of HAR is to infer human daily activities ranging from low-level activities like walking or sitting, to high-level activities like cooking or watering flowers. HAR has been a key enabler for many real-world applications, including healthcare and well-being.

Sensor-based HAR performs the recognition task by analyzing data generated by wearable sensors and/or environmental sensors [1]. Recently, some efforts also focused on activity recognition based on radio frequency signals[2, 3]. The majority of sensor-based approaches in the literature are based on supervised machine learning [4]. Those methods require a large amount of training data to build a robust activity classifier. Unfortunately, the acquisition of labeled sensor data is costly, time-consuming and intrusive [5]. Indeed, one of the major challenges in HAR is the scarcity of labeled data for realistic scenarios [6].

In order to mitigate this problem, solutions based on collaborative learning have been proposed [7, 8]. Those approaches assume that each participating user is able to collect a few labeled data that is willing to share. The data collected from the participating subjects is then aggregated to build a shared activity classifier. The results of those works are promising, since the shared activity model can reliably recognize activities while requiring limited annotated data from each user. The major drawback of those approaches is that

*Corresponding author. E-mail: gabriele.civitarese@unimi.it.

data about human activities can be considered as sensitive by many individuals. Hence, relevant privacy concerns arise when activity data is shared to untrusted third parties [9].

Recently, the federated learning paradigm has been proposed to mitigate the privacy problem in general collaborative machine learning [10]. Indeed, federated learning is a particular collaborative learning approach, where each participant only shares the parameters of a locally trained model instead of sharing data. While federated learning significantly reduces the amount of outsourced sensitive information, it relies on the assumption that the model architecture is static, and hence that the set of output classes (i.e., activities in our domain) is fixed. Moreover, it has been proven that it is still possible to infer sensitive information even from model parameters [11]. Despite some privacy preserving techniques have been proposed to mitigate the indirect leak of private information, it is not trivial to consider personal privacy preferences in this framework. In the specific case of activity recognition, users may have different preferences on which activities they wish to keep private.

In this paper, we present a novel collaborative learning approach, called **CollAR**. Like existing collaborative learning methods, **CollAR** is based on a shared activity model that is collaboratively built thanks to the personal models of the participating users. Each user can share a portion of his/her model parameters to complement the lack of labeled data. In **CollAR**, different users can contribute with different sets of activities, depending on privacy preferences and/or availability of labeled data.

In **CollAR**, the shared activity model is deployed on an untrusted cloud service provider, and it is initialized thanks to a small set of volunteers that are willing to collect a limited starting training set. Then, the shared model can be directly downloaded and used by new users. Each user can further personalize its local model using an active learning mechanism. Periodically, the shared model is updated thanks to updates coming from the users. Each time an update of the shared model is available, it can be further integrated in the local models of participating users.

CollAR is based on the Growing When Required (GWR) neural network [12, 13], which takes inspiration from self-organizing map (SOM) [14, 15], their extension self-organizing neural network (SOINN) [16, 17] and growing neural gas (GNG) [18–20]. Differently from solutions based on neural networks with a fixed architecture (i.e., where the number of layers and neurons is defined before training and it does not change), GWR is an incremental neural network that is capable of automatically and dynamically adapting its architecture. Indeed, GWR learns the topology of the network by adding and removing neurons (and the links between them) when needed in order to accurately approximate the training data. This dynamic evolution capability makes it possible to continuously adapt the activity model with the changes in sensor data and activity patterns.

In order to mitigate privacy risks, the participating users only share a portion of their local model parameters (i.e., GWR neurons). Moreover, **CollAR** allows each user to define personal privacy preferences about the activities that she wants to keep private. Hence, the shared parameters will not include information related to activities that the user considered sensitive in her preferences. The flexibility of GWR makes it possible to adapt the activity model even in the presence of heterogeneous privacy preferences among the participating users.

The key contributions of the paper are summarized as follows:

- We propose a novel distributed collaborative learning approach for human activity recognition. Thanks to the underlying machine learning algorithm, **CollAR** provides a flexible framework for users that share heterogeneous sets of activities.
- In order to mitigate privacy issues, in **CollAR** each participating user only shares a portion of his/her local model parameters according to personal privacy preferences.
- We performed an extensive empirical evaluation considering two publicly available datasets. The results show the effectiveness and flexibility of **CollAR**.

The rest of the paper is organized as follows. Section 2 reviews the relevant work in HAR with focus on semi-supervised learning, transfer learning and collaborative learning. Section 3 describes **CollAR** in details. Section 4 illustrates our experiment setup and evaluation methodology, and presents and analyses the results. Section 5 discusses the strengths and limitations of our approach. Finally, Section 6 concludes the paper and points to future research directions.

2. Related Work

Human activity recognition (HAR) has attracted a lot of attention in the last decades and various tech-

niques have been proposed [4, 21, 22]. In this section, we review the most relevant techniques in the literature to tackle the labeled data scarcity problem for HAR, including knowledge-based reasoning, semi-supervised learning, transfer learning, and collaborative learning. We also review relevant works that tackled the privacy problems that arise in collaborative learning settings.

2.1. Knowledge-based Reasoning

In order to completely avoid the acquisition of labeled data, many works in the literature proposed knowledge-based approaches for activity recognition, especially considering smart-home settings [23, 24]. Those techniques rely on logic formalism (e.g., ontologies) that explicitly model the complex relationships between sensor events and activities based on common-sense knowledge. In particular, complex activities are defined in terms of their simpler components. The sequences of simple actions, recognized based on firing of specific sensor events, are then matched in real-time to activity definitions in order to infer the current activity performed by the monitored subject. Knowledge-based reasoning has been also proposed in combination with probabilistic reasoning [25, 26]. Semantic reasoning can also be used for the dynamic segmentation of sensor events [27, 28]. The main issue of knowledge-based approaches is their rigidity. Moreover, domain modeling is a time consuming and manual effort, and there is no guarantee that the resulting semantic model can comprehensively cover all the possible context conditions. Indeed, domain experts and knowledge-engineers are required to create the formal model. Finally, knowledge-based approaches can not be directly applied on inertial sensors data obtained from mobile and wearable devices.

2.2. Semi-supervised Learning

A few works proposed alternative machine learning approaches for activity recognition to overcome the issues of supervised learning. For instance, there exists some research efforts on unsupervised learning techniques [29–31]. However, those approaches require a large pool of data to discover significant patterns. Moreover, a certain amount of labeled data is still required to reliably associate each discovered cluster with its corresponding activity class.

In order to combine the strengths of supervised and unsupervised approaches, semi-supervised learning methods for activity recognition have been proposed [32]. These approaches only rely on a small labeled training sets to initialize the recognition model, that is continuously improved using unlabeled data. In the literature, the most common semi-supervised strategies used for activity recognition are self-learning [33], co-learning [34, 35], and active learning [36–39].

In particular, active learning proved to be particularly effective for activity recognition. This approach requires an explicit feedback from the users to obtain labels for the most informative unlabeled data points: when the classifier is uncertain about the current prediction, a query is triggered to the user in order to obtain the actual activity that he/she was performing. The feedback is used to update the recognition model with new labeled examples. **CollAR** is based on active learning to support personalization: new users initialize their personal model using the shared model, then the personal model is refined through active learning.

2.3. Transfer Learning

Transfer learning is another approach that has been explored to tackle the data scarcity challenge for activity recognition [5]. The objective of transfer learning is to transfer knowledge learned from a *source* domain (with labeled data) to a target domain (with unlabeled data). Transfer learning in activity recognition has been mainly applied to take advantage of labeled data acquired in a specific sensing setting to recognize activities in different sensing settings.

Wang et al. [40] proposed a stratified transfer learning to improve the recognition rate on cross-domain activity recognition. In this work cross-domain means different body positions where wearable sensors are worn. Recently, Change et al. [41] designed several domain adaptation techniques and performed more systematic evaluation on sensor wearing diversity; that is, transferring the activity model from one wearing position (e.g., chest) to another (e.g., thigh).

CollAR is not a transfer learning technique, since we assume that each participating user has the same sensing setup. Nonetheless, we believe that **CollAR** addresses an orthogonal problem, and hence it could be extended with transfer learning methods to adapt the shared model to the specific sensing setup of each user.

2.4. Collaborative Learning

Activity recognition based on collaborative learning is becoming increasingly popular. Collaborative learn-

ing aims at creating a shared model by aggregating smaller models that are built by periodically updates from multiple participants [42]. Moreover, the participants also receive periodic updates from the shared model to further improve their local model thanks to collaboration.

Civitarese et al. [7] proposed a knowledge-based approach based on active learning to collaboratively refine a probabilistic semantic model in charge of recognizing activities in real-time. While this method does not require training data, the human engineering effort that is required to build a comprehensive semantic model is significant. Ye et al. [8] proposed a cross learning technique to train each personal model on its own data and complement each other's labeling by querying the uncertain data to other personal models. This technique works on heterogeneous environments that have different sensor deployments with diverse sets of activities. While the above mentioned methods mitigate the data labeling problem thanks to collaboration, they did not take into account the privacy issues that emerge when sharing personal data.

On the other hand, federated learning is a more generic collaborative learning framework that explicitly considers privacy aspects [10]. In particular, only the parameters of local models are shared and aggregated in a privacy-preserving manner to build a collaborative model, thus avoiding the sharing of private data. Federated learning has been recently applied to activity recognition [43–45]. The major drawback of federated learning is that the collaborative model takes into consideration a predetermined set of activities that is fixed and does not change with time.

CollAR is a distributed collaborative learning model, sharing the same goals of the above works. Differently from existing works, **CollAR** is flexible in accommodating new activities classes and, at the same time, it is designed to mitigate privacy risks.

2.5. Privacy preservation in collaborative learning

Several works have considered the privacy issues arising by sharing data or model parameters as part of collaborative learning. Among them, Liu et al. propose a collaborative learning system that train a cloud model by sharing users' data in a privacy-preserving manner [46]. The system perturbs the training data on the mobile device using lightweight transformations to preserve the privacy of the individual training samples. The idea is to approximately reconstruct the association between encrypted feature vectors and labels using regression, without compromising the privacy of the original feature vectors.

Osia et al. have proposed a deep private feature extraction approach that extract features that maximize the mutual data with the primary information and minimize the correlation with the sensitive information [47]. In this way, the features extracted will not disclose private or identifiable information but still result in high accuracy in classification tasks. They apply the technique over face image datasets and remove identity information, but extract other features that would be effective for classification such as age, gender and facial traits.

Shokri et al. have designed a model sharing technique based on deep learning [11]. The idea is to build a personal model using each user's own data and then, during this personal network training, the system performs a selective sharing of model parameters (*gradient*). This parameter sharing, interleaved with local parameter updates during stochastic gradient descent, allows users to benefit from other users' models without explicit sharing of training data inputs. They also extend their system with a differential privacy technique. The overall objective is to collaboratively train a neural network that can be used privately and independently by each participant. Each user then downloads a subset of the parameters from the server and uses them to update the local model.

Our collaborative learning approach takes inspiration from the above model sharing approach [11], with three main differences: a) by using GWR as a learning model, what our users share are neurons from their private GWR, instead of gradient based parameters, b) the choice of neurons to be shared is influenced both by the user's privacy preferences, and by a randomized process, c) there is no assumption on the complete homogeneity of the set of activities considered by the participating users.

3. CollAR

3.1. Overall architecture

The main objective of **CollAR** is to address the labelled data scarcity problem thanks to a novel semi-supervised collaborative learning approach.

In **CollAR**, each subject can contribute to building and improving an activity model that is shared among the participating users. The shared model is deployed as a cloud service whose provider is an untrusted third

party. Since the information about performed activities is sensitive, in **CollAR** each user contributes to the collaborative model only by sharing a portion of his/her personal model. Moreover, each user can choose to keep private the information related to a subset of the activities.

The shared model in the cloud needs an initialization phase. During this phase, a limited set of users is required to acquire a small labelled dataset to create personal activity models. Once the personal models are built, these users share some of the personal parameters to initialize the collaborative model.

Once the shared activity model is ready, new users can start using **CollAR** without the need of acquiring labelled data. Indeed, a new user simply initializes his/her personal model by fetching the shared model from the cloud service. Then, in order to personalize the recognition model on each user, **CollAR** adopts an active learning strategy: the system asks a feedback to the user (i.e., the activity label) when it is not confident about the current prediction. The feedback is then used to update the user's personal model.

Since personal models incrementally evolve over time thanks to active learning, each user can periodically contribute in updating the collaborative activity model by sharing small portions of the updated personal model.

The combination of collaborative and active learning allows our system to significantly reduce the reliance on training data and, at the same time, to reach satisfactory recognition rates.

The data workflow of **CollAR** is depicted in Figure 1. Note that the *'Just Share'* users are the ones that contribute to the shared model initialization, while all the three types of users can coexist once the shared model has been initialised.

CollAR is flexible in accommodating different types of activities from the participating users. For example, suppose that some users share 'walking' and 'running' activities while other users share 'walking' and 'sitting' activities. The system is capable of integrating the personal models with these different output classes together, even during the initialization phase. Moreover, the incremental nature of **CollAR** allows the system to incorporate new activities. For instance, suppose that the current shared model has 3 activities: 'walking', 'running', and 'sitting'. When a user shares a new activity 'climbing stairs', then the system is able to include this activity in the integrated model.

CollAR uses a customized version of Growing When Required (GWR) neural network [12] as the backbone of the system. As we will explain later in this section, GWR makes it possible to implement the above envisioned system. Since GWR is an incremental machine learning algorithm, it is suitable to collaboratively and continuously build the shared model and, at the same time, to personalize users' activity models with active learning. Moreover, GWR makes it possible to incrementally introduce new activities. Finally, the GWR model allows the participating users to share only a small portion of the parameters from the personal activity models. In particular, **CollAR** implements a privacy-preserving strategy to only share a small amount of parameters, also taking into account users privacy preferences. Indeed, some users may be willing to hide information related to some activities.

In the following, we describe each component of the system in details. We introduce the background of GWR neural network in Section 3.2. We show how we use GWR to train personal models in Section 3.3. Section 3.4 describes how to integrate personal models to train the shared model. Section 3.5 shows our semi-supervised approach to continuously update personal models. Finally, Section 3.6 presents our privacy-preserving parameter sharing strategy.

3.2. Background on GWR

A Growing When Required (GWR) network is an incremental neural network capable of dynamically adding and removing neurons whenever the current state of the model does not sufficiently match the input data [12]. A GWR network is represented by a collection of neurons and edges. Similarly to self-organising maps [14], each neuron j is associated with a weight vector $\mathbf{w}_j \in \mathbb{R}^n$ in the $n-$dimensional feature space. Moreover, each neuron j is also associated with a habituation level $h_j \in [0,1]$ that is initialized to $h_j = 1$. Intuitively, habituation resembles the decrease in efficiency over time of habituated synapses of biological brains.

During the training phase, when a new feature vector is available, the GWR algorithm analyzes if it is sufficiently matched by the current network. First, given the current input $\mathbf{x}(t) \in \mathbb{R}^n$ and the weight vector of each existing neuron, the algorithm computes a best-matching neuron b, randomly selected from the

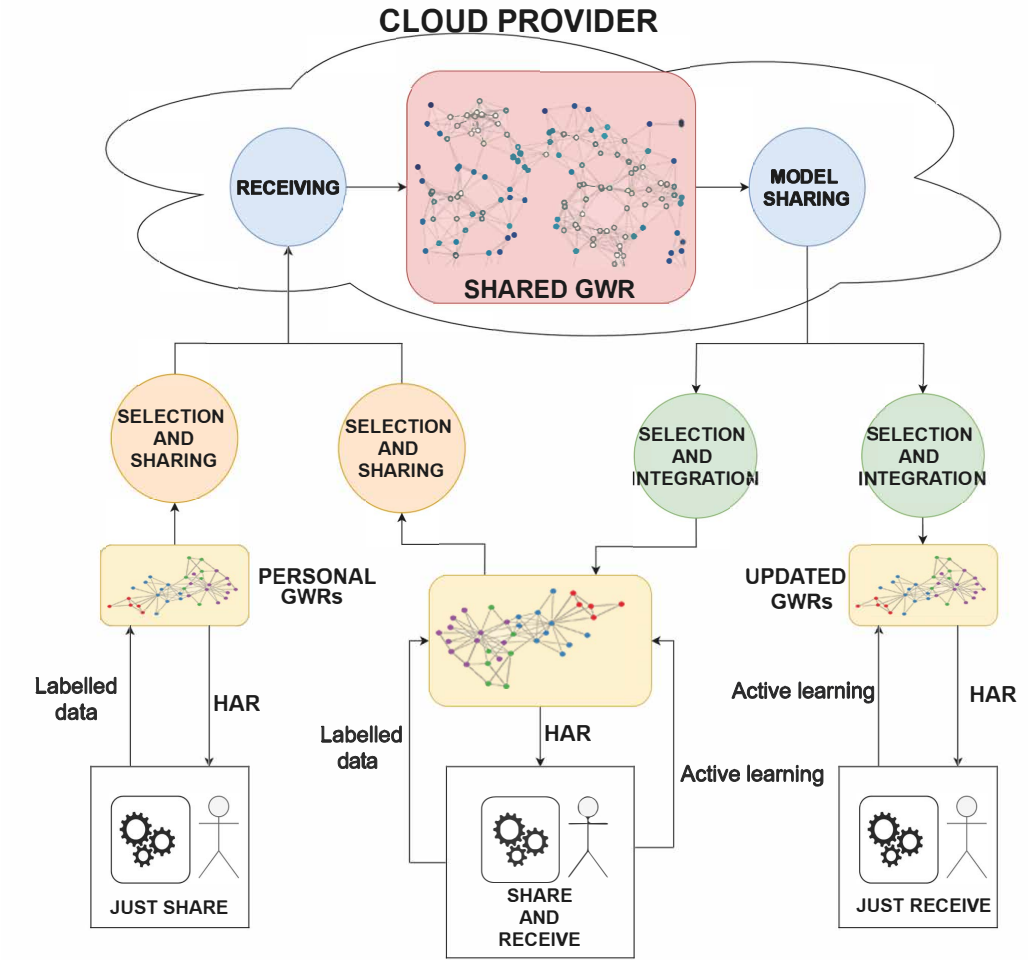


Fig. 1. Workflow of **CollAR**.

set[1] resulting from the following formula:

$$\arg\min_j \|\mathbf{x}(t) - \mathbf{w}_j\| \tag{1}$$

The network activation at the current time t is computed as the distance d between the current input and the weight vector of the best matching neuron:

$$d = \|\mathbf{x}(t) - \mathbf{w}_b\| \tag{2}$$

If d exceeds the activation threshold ACT_T, a new neuron is added to the network. The new neuron is associated with a weight vector that is the average of the input feature vector and the weight vector $\mathbf{w}_b$ associated to the best-matching neuron.

On the other hand, if the distance does not exceed the threshold, the weight vector $\mathbf{w}_b$ of the best-matching neuron is updated as follows:

$$\Delta\mathbf{w}_b = \epsilon_b \cdot h_b \cdot (\mathbf{x}(t) - \mathbf{w}_b) \tag{3}$$

where ϵ_b is a constant learning rate. Moreover, in this case, the habituation level h_b is updated as follows:

$$\Delta h_b = \tau \cdot \kappa \cdot (1 - h_b) - \tau \tag{4}$$

where τ and κ are constants that control the monotonically decrease of the habituation level.

The GWR algorithm also creates an edge between the best and the second best matching neuron based on input data. The second best is computed as shown

[1] In the literature [48] this set is often considered a singleton since it is very unlikely in practice that multiple neurons with the exactly the same distance exist.

before but excluding b from the considered neurons. Each edge is associated with an *age* factor. Whenever the algorithm considers a new input, the age of all edges is incremented; the exception is that the age of both the edge between the best and second best neurons and of the edge between the new neuron (if created) and the best is assigned to zero. Depending on their *age*, neurons and edges can be dynamically added and/or removed, thus reflecting the changes of input data. Hence, the network is growing as required. More details about the GWR algorithm can be found in [12].

3.3. Training personal GWR models

The algorithm that we propose to train and update personal GWR models from labelled data in **CollAR** is derived by a recently proposed extension of GWR for supervised learning in human action recognition [49]. Each neuron stores, for each label, the number of times it has been the best-matching neuron for that label during the training phase. Frequencies are used for classification: the predicted label for an input data is the most frequent label of the corresponding best-matching neuron. Frequencies are represented as an associative matrix H. The value $H(j,l) \in \mathbb{N}$ is the number of times that a neuron j was the best matching neuron for an input data labelled as l.

As we explained in Section 3.2, a new neuron n is added to the network when a new input data labelled as l is far away from the best matching neuron. In this case, the network will store $H(n,l) = 1$, and $H(n,A) = 0$ for the activities $A \neq l$.

The frequencies of each neuron are continuously updated with incoming labelled input data. Suppose that a new input data labelled as l' is provided to the network. If the input data is sufficiently close to its best matching neuron b, we increment $H(b,l')$ by one. The high-level pseudo-code for GWR training is shown in Algorithm 1.

Once the network is trained, it can be used to classify unlabelled data points. For each unlabelled input data we derive a posterior probability distribution over the possible activities. The most likely activity in the distribution is the output of the classification.

The standard strategy proposed in the literature is to compute the probability distribution from the frequencies of the best-matching neuron [49]. However, during preliminary experiments we observed that the best matching neuron alone is not completely representative for the input data. Hence, differently from [49], we average the probability distribution obtained from the frequencies of the best matching neuron with two other probability distributions:

- *top-k*: we select the top-k best matching neurons to the input. For each one of these neurons, we obtain a probability distribution from the frequencies. We compute the average of the probability distributions weighted by the distance.
- *best matching and its neighbours*: we compute the average of the probability distributions of the best-matching neuron and its neighbours (i.e., the neurons directly linked to the best-matching neuron).

Algorithm 1 Updating the personal GWR network with labelled data points

Input: A labeled dataset D, A personal GWR
Output: A trained GWR network

if Personal GWR is empty **then**
  Initialize the network with K random samples from D
end if
for each epoch **do**
  for each feature vector $fv \in D$ labeled as l **do**
    Compute the best matching neuron b (Equation 1)
    $d \leftarrow$ distance between $\mathbf{w_b}$ and fv (Equation 2)
    if $d \leqslant$ ACT_T **then**
      $H(b,l) = H(b,l) + 1$
      Update the weight vector of b (Equation 3)
      Update the habituation level h_b (Equation 4)
      Set *age* on the link between b and the second best matching neuron to 0.
      Increment age by 1 for the remaining links
    else
      Add a new neuron j to the network
      $H(j,l) = 1$
      $H(j,A) = 0$ for each activity $A \neq l$
      Create a link between j and b with *age*$= 0$
      Initialize habituation level $h_j = 1$
      Initialize weight vector $\mathbf{w}_j$ as the average between fv and $\mathbf{w}_b$
    end if
  end for
  Remove links where *age*>AGE_T
end for
Remove unlinked neurons

3.4. The shared GWR model

The participating users in **CollAR** collaborate in building a shared GWR model that is deployed in the cloud.

The shared GWR model requires a bootstrap phase in order to be initalized. A restricted number of volunteers acquire a small amount of labelled data to initialize their personal GWR model using Algorithm 1. Subsequently, these users share a portion of their neurons to the cloud model using a sharing approach that will be described in Section 3.6.

New users of **CollAR** simply fetch the shared model to initialize their personal model, thus avoiding the acquisition of additional labelled data.

Periodically, the participating user may contribute with a personal update to further improve the shared model by sharing a portion of the neurons from their personal GWRs. The sharing mechanism is the same one used for model initialization and it will be presented in Section 3.6.

The shared model is initialized/updated with a subset of neurons from the personal GWRs of multiple users. Each shared neuron is associated with its weight vector and frequencies. The shared GWR model is initialized/updated using Algorithm 2. This algorithm is very similar to Algorithm 1. The main difference is that the input is a set of shared neurons, instead of labelled data. For each shared neuron j, we compute the distance between j and its best matching neuron b. If the distance is below the activation threshold, we combine the frequencies of b with the frequencies of j mitigated by the distance between b and j. Otherwise, we directly include j in the GWR network and we link it to b. In this latter case, the corresponding weight vector is the average between w_j and w_b.

Note that, to reliably update the shared model, the cloud service triggers an update process only when a sufficient number of shared neurons are available. At the end of the training process, the cloud service notifies the participating users that a new version of the shared model is available. When participating users are notified about a new available update of the shared model, they may decide to download it to improve their personal model. The local integration of the shared model update is performed using Algorithm 2.

3.5. Semi-supervised updates of personal GWRs

Each user can further personalize the personal GWR model adopting a semi-supervised strategy based on active learning. For each unlabeled feature vector, the current personal GWR model is used to obtain a probability distribution over the possible activities. If the probability of the most likely activity is lower than a threshold *ACTIVE_LEARNING_T*, we assume that there is uncertainty in the prediction and hence we ask the user to provide a feedback about the activity that he/she is actually performing. The feedback is associated with the feature vector. When a sufficient number of new labeled feature vectors is collected through active learning (according to the *ACTIVE_LEARNING_BATCH_SIZE* parameter), they are provided as new labeled examples to the personal GWR (using Algorithm 1).

Algorithm 2 Update GWR by neuron sharing

Input: A GWR, a set of shared neurons N
Output: An updated GWR network

if GWR is empty **then**
  Initialize the network with K random samples from N
end if
for each epoch **do**
  for each neuron $j \in N$ **do**
    Compute the best matching neuron b (Equation 1)
    $d \leftarrow$ distance between b and j (Equation 2)
    if $d \leqslant$ ACT_T **then**
      for each activity A **do**
        $H(b,A) = H(b,A) + (e^{-\sqrt{d}} \cdot H(j,A))$
      end for
      Update the weight vector of b (Equation 3)
      Update habituation level h_b (Equation 4)
      Set *age* on the link between b and the second best matching neuron to 0.
      Increment age by 1 for the remaining links
    else
      Add j to the network
      Create a link between j and b with $age = 0$
      Initialize habituation level $h_j = 1$
      Initialize weight vector as the average between $\mathbf{w}_j$ and $\mathbf{w}_b$
    end if
  end for
  Remove links where *age*>AGE_T
end for
Remove unlinked neurons

Since personal models evolve over time, the participating users may periodically share with the cloud a portion of their neurons to collaboratively improve the shared GWR model, using the strategy presented in Section 3.6. The update can be performed periodically; e.g., daily, weekly, monthly, etc.

3.6. Privacy-preserving neuron sharing

Users may have privacy concerns in releasing to untrusted parties information about certain activities that they perform. In our architecture, the cloud service provider, as well as the other participating users, are considered *honest but curious* adversaries: they follow the protocol but they may try to infer from the data that they observe information about the activities performed by a given user (e.g., type of activity, when it was performed, and possibly even how it was performed).

As opposed to a data sharing model that would release directly data or feature vectors from which the cloud provider may infer the activity and possibly even finer grain information, **CollAR** only shares system parameters, in the form of a fraction of the neurons in the personal GWR networks. Indeed, **CollAR** users may share part of their neurons to the cloud model in two cases: 1) when they are included in the set of users in charge of collecting labeled data to initialize the cloud model, 2) when they participate in the continuous collaborative update of the shared model.

Despite neurons are only model parameters, they have associated data on the position in the feature space (i.e., the weight vector) and on the frequency of *activation* for the different activities. Hence, an adversary may indeed exploit this data to infer sensitive information.

Several privacy protection techniques may be applied to mitigate this privacy violation risk, including for example, perturbing weight vectors and frequencies in order to achieve differential privacy following an approach similar to the one proposed in [11]. In practice, this means that for hiding activity A_1, the system should insert noise in the released information so that an adversary would have a very low probability of distinguishing the shared neurons from the ones obtained in the case that A_1 was not performed. However, this approach may lead to a significant drop in performance depending on the size and distribution of data.

We take a simpler approach in favor of utility, exploiting the observed robustness of the system even when sharing only a small portion of the neurons, and the property of **CollAR** of allowing different users to have personal models built on different sets of activities.

Based on user privacy preferences, we select the neurons that should be shared from a copy of the personal GWR network that has never seen any data about private activities. We call this network the *private twin* of the personal GWR. More precisely, **CollAR** operates as follows:

- Each user provides a set $\mathbf{A}_p$ of *'private activities'* that is not willing to share.
- When a personal GWR is created, its private twin GWR_p is also created.
- During the supervised initialization (Algorithm 1) the two networks are trained in parallel except that feature vectors labeled with $A \in \mathbf{A}_p$ are ignored by GWR_p, i.e., they do not change the network. Similarly, when active learning is performed on the user, the newly labeled samples will refine both networks except for the ones labeled in $\mathbf{A}_p$ that will not modify GWR_p.
- When the user shares model parameters with the cloud, the neurons are taken from the private twin network that has never been exposed to private data. Sharing all the neurons of GWR_p is a possible choice, but experimental results indicate that this strategy may negatively impact the recognition rate of the resulting shared model. Hence, we share only a fraction of them, selected as follows: for each neuron j, we say that j is primarily associated with an activity A if A is the activity with the highest frequency for j. For each non-private activity A, we randomly pick $p\%$ of the neurons that are primarily associated with A in GWR_p and send them to the cloud with their associated parameters. Note that the same strategy is adopted if the user is selected among the ones used to initialize the model.
- Each shared neuron is only associated with its original weight vector and frequencies.
- When a user receives an update from the cloud, the neurons are integrated both in the personal GWR and in its private twin in the same way (using Algorithm 2).

While this technique avoids sharing of information directly connected with a private activity, some attacks based on background knowledge on the correlation between private and non-private activities may still be performed. These attacks are naturally mitigated by the absence of precise temporal information at the cloud

side (the sharing is performed in batches) and may also be contrasted by using the same knowledge at the user side, considering as private also the correlated activities.

Clearly, if all the participating users consider activity A as private the globally initialized network will not be able to recognize A and the parameters integrated by the cloud at each update step will not help anyone in improving their ability to recognize A. This is anyway expected if nobody wants to share information about that activity, but privacy preferences often differ among users and this problem may be mitigated if the system has many participants. The effect of this privacy-preserving strategy on performance will be illustrated in Section 4.5.

4. Experiments and Evaluation

In this section, we evaluate the effectiveness of **CollAR** through extensive experiments on two real-world, third-party datasets. Our evaluation process was inspired by the following key questions:

1. *online collaborative learning* – How effective is **CollAR** in integrating shared neurons to recognise activities on new users that download the shared model?
2. *heterogeneous sets of activities* – How much utility (i.e., recognition rate) **CollAR** drops when shared neurons are obtained from personal models with heterogeneous sets of activities?

4.1. Datasets

In order to validate the effectiveness of **CollAR** , we considered two well-known HAR datasets: PAMAP2 [50] and DSADS [51]. Both datasets consider physical activities that are monitored using inertial sensors (i.e., accelerometer, magnetometer and gyroscope) from wearable devices.

PAMAP2 contains data collected from 3 Colibri wireless inertial measurement units (IMU) from 9 users performing 12 activities. Each user wore the sensors on their *wrist* of the dominant arm, *chest*, and *ankle* on the dominant side. From this dataset we excluded a user (identified as 9 in the dataset) that only performed one activity. Also, we excluded 4 activities that were insufficiently represented in the dataset. Hence, in this work we consider the following 8 activities from PAMAP2: *lying*, *sitting*, *standing*, *walking*, *ascending stairs*, *descending stairs*, *vacuum clearning*, and *ironing*. Each of those activities was performed approximately for 3 minutes by each user, except for ascending/descending stairs due to the limitations of the building where the activities were carried out. More information about this dataset can be found in [50].

DSADS contains data collected from 8 users wearing accelerometer units in 5 positions: torso, right arm, left arm, right leg, and left leg. This dataset includes the following 19 activities: *sitting*, *standing*, *lying on the back*, *lying on the right side*, *ascending stairs*, *descending stairs*, *standing in an elevator still*, *moving around in an elevator*, *walking*, *walking on a treadmill*, *walking on a treadmill in inclined position*, *running on a treadmill*, *exercising on a stepper*, *exercising on a cross trainer*, *using an exercise bike (horizontal)*, *using an exercise bike (vertical)*, *rowing*, *jumping*, and *playing basketball*. Each user in the dataset performed each activity for 5 minutes. Hence, we did not exclude any user or activity in our experiments when using DSADS. More information about this dataset can be found in [51].

4.2. Implementation Details and Hyperparameter Tuning

We implemented **CollAR** starting from the publicly available Python implementation of the work presented in [52].

Table 1 summarizes the many hyperparameters of **CollAR**, providing a short description, and indicating the specific values that we chose in our experimental evaluation. This table does not cover hyperparameters related to privacy preferences, that will be discussed later in this section. Note that we used the values suggested in [52] for the constants κ and τ of Equation 4, and constant K of Algorithms 1 and 2. We chose the values of the remaining hyperparameters by performing a grid search, and selecting the combination that achieved the best recognition rate in terms of F1-score.

4.3. Comparing GWR with state-of-the-art classifiers

In order to evaluate the effectiveness of GWR as activity classifier, we compared it with other classifiers. In particular, we evaluated if the supervised approach proposed in Algorithm 1 reaches recognition rates that are comparable with other classifiers commonly used for activity recognition. We also evaluate if GWR reaches acceptable results even if trained with a limited amount of data. Note that in this experiment

Table 1
Hyperparameters in **CollAR**

Hyperparameter	Descrption	Chosen value
Number of epochs	This parameter determines the number of times that the training process is repeated on the training set for model convergence.	50 for personal training, 40 for shared model updating and 3 for active learning updates
ϵ_b	Learning rate in Equation (3)	0.26
τ	Constant in Equation (4). Together with κ, it decides how quickly the habituation level h_n of a neuron n rate decreases, indirectly controlling the convergence speed of the corresponding weight vector w_n during training.	0.3
κ	Constant in Equation (4). See description of τ.	1.05
K	The number of labeled examples randomly sampled from the dataset to initialize the GWR network	2
k	The number of best-matching neurons considered for classification using the top-k best neurons strategy (see Section 3.3)	6 for DSADS and 2 for PAMAP2
ACT_T	Activation threshold in Equation (2). It decides when a new neuron should be added in the network during training. This hyperparameter is sensitive to the number of training samples in the dataset and it determines the density of the network.	0.5
AGE_T	Age threshold. It determines how frequently to remove links between neurons that are not representatives of training data. Since unlinked neurons are eventually removed from the model, this parameter is particularly important when adapting the model to the end users.	1300 for DSADS and 700 for PAMAP2
ACTIVE_LEARNING_T	Confidence threshold for active learning. This threshold is used to balance the number of queries and the recognition rate.	0.53 for DSADS and 0.58 for PAMAP2
ACTIVE_LEARNING_BATCH_SIZE	The number of active learning feedback used to periodically update the personal model.	2

we evaluated GWR as a supervised classifier for activity recognition, while the experiments on collaborative learning are presented in the following subsections.

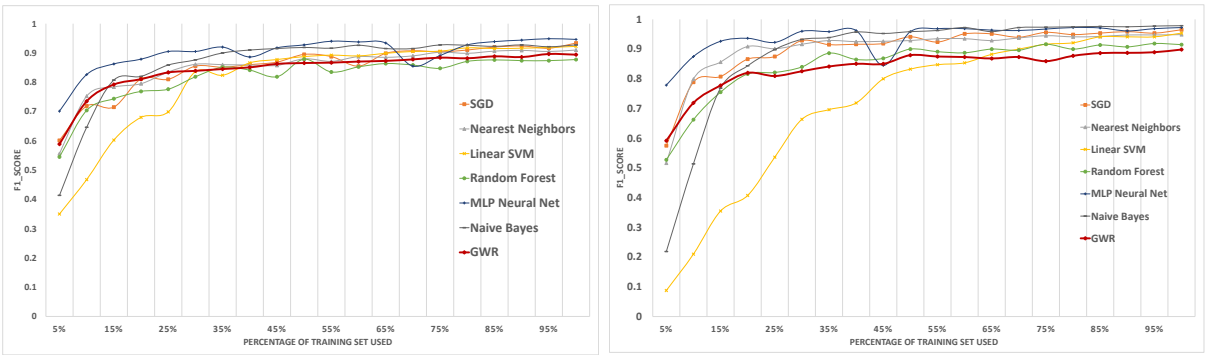
For each user, we split data into training (80%) and testing (20%). Then we use p% of training data to train the classifier, varying p from 10 to 100 with step size 10. We iterated this process for each user and averaged the resulting macro F1-scores. We selected a large number of supervised learning techniques for comparison, including SVM (with RBF kernel), Random Forest, and Multi-Layer Perceptron (MLP)[2].

Figure 2 shows the results on both datasets. GWR achieves comparable results to the state-of-the-art techniques, leading to F1_Scores slightly lower than the best performing techniques. However, with respect to the compared classifiers, the advantage of GWR is the ability to adapt to new samples and learn new activities, which is desirable for long-term sustainable collaborative activity learning. Hence, the small drop in F1 score is acceptable considering the advantages of GWR.

[2]We use the implementation of these techniques from the Python Scikit-learn library [53].

4.4. Effectiveness of Online Collaborative Learning

The key contribution of **CollAR** is to enable online collaborative learning, with the objective of reducing the reliance on training data while achieving a satisfying recognition rate. We adopted the leave-one-subject-out evaluation methodology to evaluate the effectiveness of **CollAR**. In this experiment, we assume that there are no users with private activities. First, each of the N-1 users trains a personal GWR model using Algorithm 1 and then shares a portion of its neurons according to the strategy proposed in Section 3.6. Shared neurons are used to build the collaborative GWR model using Algorithm 2. The shared model is hence fetched by the left out user. For this user, we used 80% of data for active learning and 20%



(a) PAMAP2 (b) DSADS

Fig. 2. F1-scores of **CollAR** and GWR on PAMAP2 and DSADS

for testing the personal model. We iterated this process for each user, computing the average F1 score.

We adopted the same evaluation methodology for all the following experiments. We have experimented with different neuron sharing percentages on p% (see Section 3.6) ranging from 10% and 90% to obtain the lowest value that provided robust results. Indeed, it is preferable that each user only shares as little neurons as possible while achieving satisfying recognition rates. In the end, we selected $p = 35\%$.

In order to demonstrate the advantage of **CollAR**, we compared our technique with the following two baselines:

- *personal model* – This baseline is useful to compare the recognition rate of a personal GWR model (i.e., only trained with data from the left out user) with the **CollAR** collaborative model (i.e., trained with data from other users). We split each user's data into training (80%) and testing (20%). We train a personal GWR model with q% of each user's training data using Algorithm 1 and evaluate it on test data. Since we consider the setting where only a small amount of labeled training data is available, we experimented with different values of q% ranging from 1% to 5%. We selected 3% since it is the lowest value leading to a satisfying F-1 score.
- *data sharing model* – This baseline is useful to compare the recognition rate of a collaborative system that is based on sharing data instead of model parameters with the recognition rate of **CollAR**(based on neuron sharing). In particular, each one of the $N-1$ users share p'% of their feature vectors to build a collaborative GWR model using Algorithm 1. Then, the shared model is deployed on the remaining left out user. For the sake of fairness, we selected the value of p'% as 35% (i.e., the same as $p\%$).

Table 2 compares the recognition accuracy of **CollAR** with the considered baselines. For each approach, we report the results with and without active learning and the query rate (i.e., the average ratio of active learning questions). Considering the PAMAP2 dataset, **CollAR** significantly outperforms personal training and reaches similar results with respect to data sharing. On the other hand, DSADS exhibits similar recognition rates for all the settings. These results indicate that sharing neurons instead of sharing data has almost no impact on activity recognition, with the advantage of mitigating privacy risks. Moreover, we can also observe that the collaborative model of **CollAR** leads to equal or better results than personal models, hence confirming the effectiveness of our collaborative framework.

While the recognition rates reported in Table 2 may seem not completely satisfactory, it is important to note that the considered datasets only include a rather low number of users (both PAMAP2 and DSADS involve data from 8 users). We expect that our collaborative approach would show its benefits (in terms of recognition rate) by considering a significantly larger number of participating users.

In the following, we provide insights about the recognition rate of **CollAR** at the activity granularity. Figure 3 and Figure 4 show the confusion matrices obtained by **CollAR** on PAMAP2 and DSADS datasets, respectively. From these results it emerges that **CollAR** often confuses static activities that are characterized by similar patterns, like *sitting*, *standing* and *lying*. This occurs on both datasets. On PAMAP2

Table 2
Comparison of F1-scores between **CollAR**, personal training and sharing data. AL (Active learning), Q (Query rate)

Datasets	Personal Training			**CollAR**			Sharing Data		
	NO AL	AL	Q	NO AL	AL	Q	NO AL	AL	Q
PAMAP2	62.82%	63.95%	1.60%	72.65%	78.41%	6.08%	73%	78.33%	6.18%
DSADS	68.93%	76.93%	4.34%	68.98%	74.1%	4.05%	68.73%	73.18%	1.84%

dataset, *sitting* and *standing* are also sometimes confused with *ironing*, that is a static activity as well. Similarly, on both datasets it emerges that *ascending stairs* and *descending stairs* are difficult to discriminate. Considering DSADS, *CollAR* only confuses very similar activities: *standing in an elevator* vs *moving in an elevator*, *walking on horizontal treadmill* vs *walking on an inclined treadmill*, and *using horizontal exercise bike* vs *using vertical exercise bike*. On the other hand, **CollAR** performs well on those activities that have distinctive movement patterns on both datasets. We believe that these results are encouraging, since **CollAR** only confuses activities that are known to be particularly difficult to discriminate due to their very similar motion patterns.

4.5. *Robustness of **CollAR** in presence of heterogeneous sets of activities and privacy preferences*

One of the key reasons that we advocate GWR in **CollAR** is its flexibility for integrating personal models that are built on different sets of activities. In contrast, most of the collaborative and federated learning techniques mandate that the individual models have the same model architecture, the same parameter spaces, and the same output class set.

This flexibility is useful both in the case of heterogeneous sets of activities due to different user routines and in the case of heterogeneous sets of activities due to privacy preservation following the strategy described in detail in Section 3.6. In both cases the neurons shared by each user may not contain any information about some activities.

In order to estimate the drop in performance observed in these cases, we perform a leave-one-subject-out cross-validation and, at each fold, we randomly assign to each user a subset of hidden activities (either private and hence not considered in the private twin network, or not even performed and hence not considered in both of the personal GWR and its twin). For each of the N-1 users, we train a personal model with all of his/her available data. Note that in both cases of activities hidden because private or hidden because not performed, the resulting private twin network will be the same, hence the same neurons and associated data are shared to the cloud model. Based on these sets of shared neurons, **CollAR** using Algorithm 2 builds a collaborative model, which will be deployed and integrated as personal model of the left out user. Data of the left out user is randomly divided in 80% used for active learning and 20% for testing. The performance is first tested on the 20% without active learning and then on the same data but after active learning. We iterate the process and average the results according to the leave-one-out methodology.

In order to understand the impact, we compare the results with the ideal condition in which each user contributes to the whole set of activities **A** (no hidden activities). In Figure 5 we show the results obtained by comparing three settings considering 1, 2, and $|\mathbf{A}|/2$ hidden activities with the objective of considering the more realistic cases of a small percentage of activities and a quite extreme case in which half of the activities are hidden.

As we can see, there is no significant drop in F1-scores on **CollAR** even when each user only contributes with half of the activities. **CollAR** can integrate the personal models, complement their activity sets, and build a cloud model to cover all the activities. Also when considering active learning, the query rates do not increase and are stable, which again demonstrates that the cloud model has formed a well-covered model on all the activities. There is a slight fluctuation between sharing $|A|-1$ and $|A|-2$ activities; for example on PAMAP2, F1-score of sharing $|A|-2$ activities without active learning is 1.2% higher than F1-score of sharing $|A|-1$ activities, and on DSADS, F1-score of sharing $|A|-2$ activities with active learning is 0.33% higher than F1-score of sharing $|A|-1$ activities. This fluctuation is mainly due to the randomness when we sample activities.

In Figure 5b, we witness a performance drop on DSADS dataset, when only half of the activities are being used. After inspecting the results, we find that the drop may be due to the fact that DSADS has more classes to learn; i.e., 19 activities vs 8 in PAMAP2. In principle, the less classes each user contributes, the

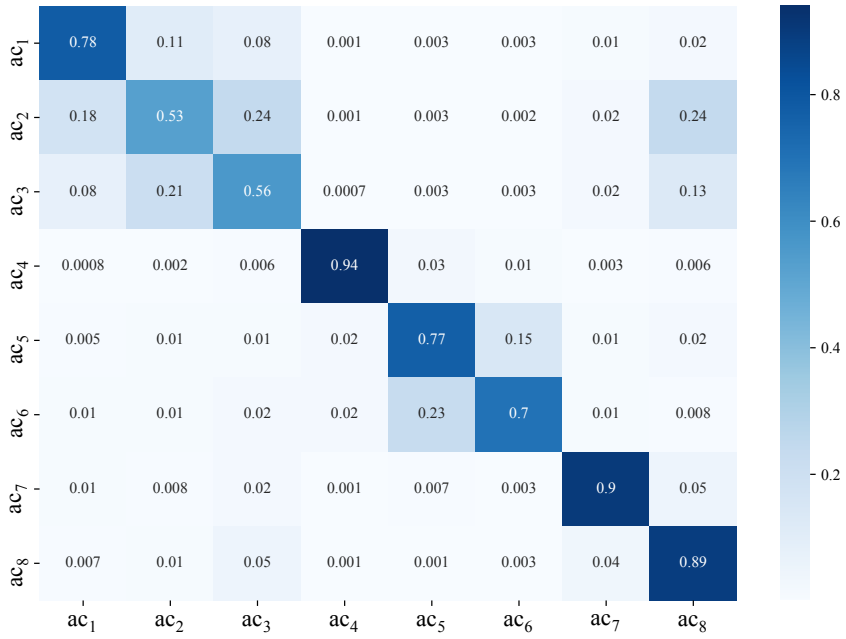


Fig. 3. PAMAP: confusion matrix. ac1=*lying*, ac2=*sitting*, ac3=*standing*, ac4=*walking*, ac5=*ascending stairs*, ac6=*descending stairs*, ac7=*vacum cleaning*, ac8=*ironing*.

more difficult the learning task; that is, we need to complement with a large number of classes from the other users. So when each user only contributes half of the activities, the cloud model needs to find another half from the other users. In DSADS, we need to find 10 new activities while in PAMAP2 we only need to find other 4. Considering the limited number of users, and consequently the limited number of personal models covering the missing activities, the above observation explains the drop of performance on the dataset with the larger number of activities.

5. Discussion

In the following, we discuss the strengths and the limitations of **CollAR**.

5.1. Flexibility

The results presented in Section 4.5, and especially the ones reported in Figure 5, indicate that **CollAR** is a promising framework to handle heterogeneous sets of activities in collaborative learning settings. This flexibility is particularly required in real-world deployments, since users may have different habits, or they may specify different privacy preferences about the types of activities they are willing/not willing to share. Our results show that, even when participating users only share data from half of the overall types of activities, **CollAR** reaches recognition rates that are only slightly worse than the ones obtained when users share all of their data.

While these results are encouraging, they still need to be confirmed on datasets with a larger number of users. Another limitation of our experiments is that we randomly sampled the shared activities for each user based on a uniform distribution, while in real-world scenarios the actual distribution may be influenced by user profiles and personal habits.

*5.2. Privacy properties of **CollAR***

CollAR offers important privacy properties that are required to reduce the gap towards real-world deployments. First, instead of directly sharing personal sensor data, each user only shares parameters (i.e., neurons) from the local model. Indeed, sensor data may reveal specific activity patterns that may be considered as sensitive (e.g., walking patterns may reveal health issues). Besides, **CollAR**'s privacy preferences give control to the end-users about the activities they prefer not to share according to their perception about personal sensitive information. As explained in more detail in

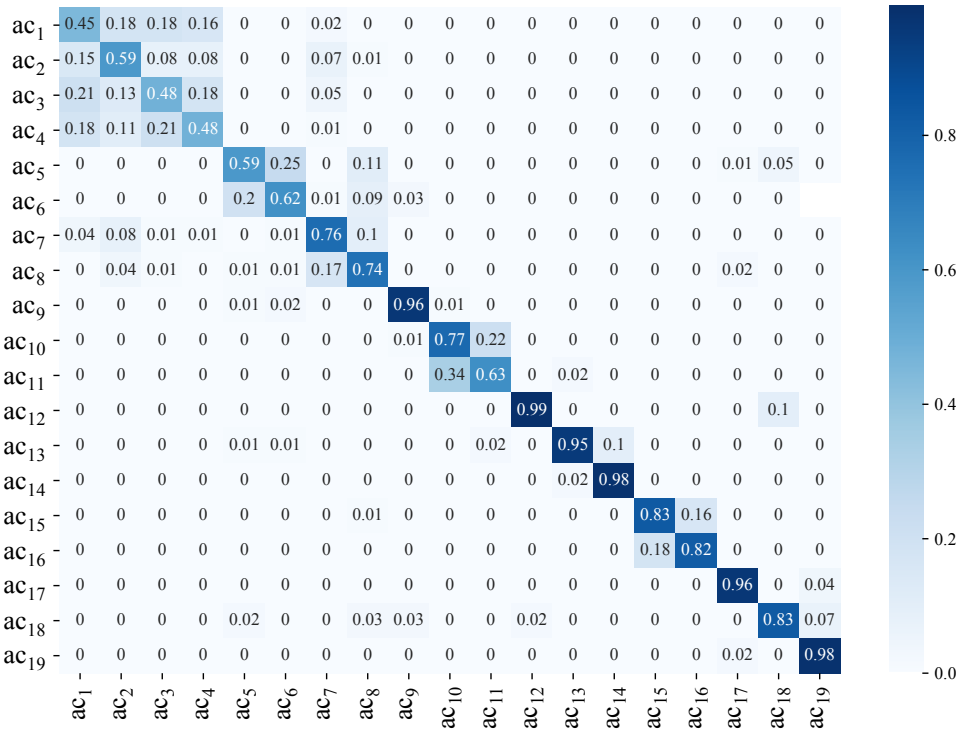
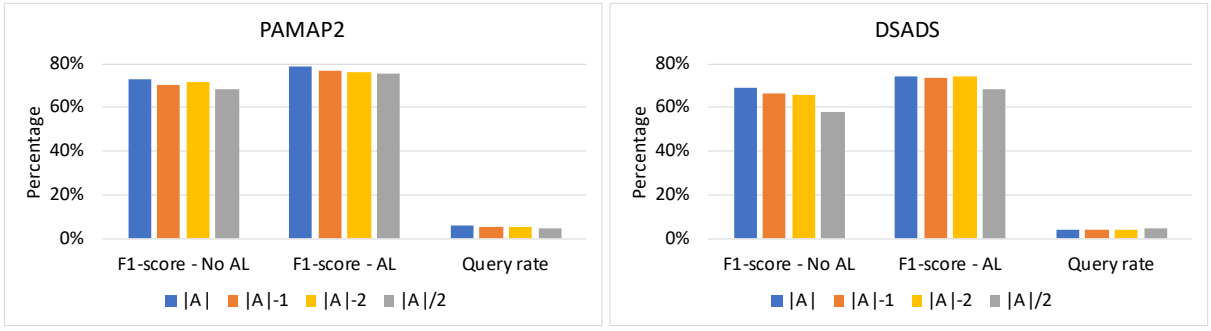


Fig. 4. DSADS: confusion matrix. ac1=*sitting*, ac2=*standing*, ac3=*lying on the back*, ac4=*lying on the right side*, ac5=*ascending stairs*, ac6=*descending stairs*, ac7=*standing in an elevator still*, ac8=*moving around in an elevator*, ac9=*walking*, ac10=*walking on a treadmill*, ac11=*walking on a treadmill in inclined position*, ac12=*running on a treadmill*, ac13=*exercising on a stepper*, ac14=*exercising on a cross trainer*, ac15=*exercise bike (horizontal)*, ac16=*exercise bike (vertical)*, ac17=*rowing*, ac18=*jumping*, ac19=*playing basketball*.



(a) PAMAP2 (b) DSADS

Fig. 5. Comparison of performance with different numbers of hidden activities.

Section 3.6, in order to prevent an adversary from using the shared model parameters to infer the execution of private activities, **CollAR** does not share any neuron that is detected as associated with them.

However, we did not evaluate if the collaborative model can still reveal some sensitive information about participating users. For instance, an adversary may use domain specific background knowledge on the correlation between private and non-private activities to infer hidden activities given the ones that are shared. An adversary could also try to analyze the properties of shared neurons (e.g., the weight vectors) to reconstruct users' raw sensor data. In order to mitigate these potential problems, **CollAR** does not share all the *non-private* neurons, but only a randomized subsample of them (35% in our experiments). In future work, we will study whether some realistic privacy attacks are still possible in **CollAR** and how to mitigate them.

We also want to mention that the privacy properties of **CollAR** do not significantly impact its recognition capabilities. As Table 2 shows, the recognition rate of **CollAR** is similar to a collaborative approach that shares data instead of model parameters. As we also previously discussed in Section 5.1, *CollAR* is robust and flexible when considering heterogeneous privacy preferences, that are indeed one of the major strengths of our method.

5.3. Impact of active learning

In order to fine-tune the shared model on each user, **CollAR** takes advantage of active learning. When the system is uncertain, we directly ask the user the label about the activity that she was performing. Table 2 clearly shows that active learning improves the overall recognition rate by 5% on both datasets with a low number of triggered queries. We also expect that the number of active learning questions will significantly decrease over time. In future work, we will explore additional semi-supervised techniques that do not require user interaction, like label propagation [54].

5.4. Heterogeneous sensing setups

In this work, we make the assumption that participating users that contribute to the shared model are actually monitored with the homogeneous sensing setups. However, different users may use different types of wearable sensors positioned in different parts of the body (e.g., a user may use his/her smartphone in the pocket while another one the smart-watch on the wrist). In order to make another example, when considering smart-home activity recognition, different homes may have very different configurations.

As we mentioned in Section 2.3, transfer learning tackles the orthogonal problem of adapting a recognition model trained on a specific sensor settings to recognize activities monitored with different sensing solutions. In future work we will investigate how to include transfer learning in **CollAR**, with the objective of collaboratively learning a shared model including users with heterogeneous sensing infrastructures.

5.5. Need for evaluation on a larger scale

We evaluated **CollAR** considering two widely adopted HAR datasets: PAMAP2 and DSADS. Those datasets have been often used as benchmark in the literature, and we chose them since they have a wide number of activities (especially DSADS). However, the major limitation of those datasets is that they only include a relatively low number of users. Unfortunately, this is a typical characteristic of public HAR datasets. Hence, while our results are encouraging, we need to confirm them using larger datasets with a higher number of users. Besides considering larger datasets, when available, we also plan to evaluate **CollAR** in a real-world scenario. This evaluation has the potential of better understanding how subjects would actually choose their privacy preferences and its impact on collaborative learning.

6. Conclusion and Future Work

We presented **CollAR**, a novel semi-supervised collaborative learning approach based on Growing When Required (GWR) neural networks. In **CollAR**, each participating user contributes to the collaborative model by sharing only a small portion of parameters of the personal model. Our framework also allows each user to define which activities to keep private. Thanks to active learning, each user can continuously improve the personal model.

Our results indicate that **CollAR** reaches comparable recognition rates with respect to solutions based on data sharing. By sharing only model parameters, **CollAR** mitigates privacy risks related to outsourcing activity data. The results also show that **CollAR** is flexible with respect to users that contribute with heterogeneous sets of activities. This takes into account privacy

preferences but also real-world situations where every users actually perform different sets of activities.

Besides the many possible future research directions that we previously presented in Section 5, we will also explore if and how alternative self-organizing neural networks (as e.g., [55]) can be effectively adopted in our collaborative framework.

References

[1] L. Chen, J. Hoey, C.D. Nugent, D.J. Cook and Z. Yu, Sensor-based activity recognition, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(6) (2012), 790–808.

[2] Z. Chen, C. Cai, T. Zheng, J. Luo, J. Xiong and X. Wang, RF-Based Human Activity Recognition Using Signal Adapted Convolutional Neural Network, *IEEE Transactions on Mobile Computing* (2021).

[3] S. Ding, Z. Chen, T. Zheng and J. Luo, RF-Net: A Unified Meta-Learning Framework for RF-Enabled One-Shot Human Activity Recognition, in: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, SenSys '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 517–530–. ISBN 9781450375900. doi:10.1145/3384419.3430735.

[4] O.D. Lara and M.A. Labrador, A survey on human activity recognition using wearable sensors, *IEEE communications surveys & tutorials* **15**(3) (2012), 1192–1209.

[5] D. Cook, K.D. Feuz and N.C. Krishnan, Transfer learning for activity recognition: A survey, *Knowledge and information systems* **36**(3) (2013), 537–556.

[6] C. Bettini, G. Civitarese and R. Presotto, CAVIAR: Context-driven Active and Incremental Activity Recognition, *Knowledge-Based Systems* (2020), 105816.

[7] G. Civitarese, C. Bettini, T. Sztyler, D. Riboni and H. Stuckenschmidt, newNECTAR: Collaborative active learning for knowledge-based probabilistic activity recognition, *Pervasive and Mobile Computing* **56** (2019), 88–105.

[8] J. Ye, S. Dobson and F. Zambonelli, XLearn: Learning Activity Labels across Heterogeneous Datasets, *ACM Trans. Intell. Syst. Technol.* **11**(2) (2020). doi:10.1145/3368272.

[9] C. Bettini and D. Riboni, Privacy protection in pervasive systems: State of the art and technical challenges, *Pervasive and Mobile Computing* **17** (2015), 159–174.

[10] Q. Yang, Y. Liu, T. Chen and Y. Tong, Federated machine learning: Concept and applications, *ACM Transactions on Intelligent Systems and Technology (TIST)* **10**(2) (2019), 1–19.

[11] R. Shokri and V. Shmatikov, Privacy-preserving deep learning, in: *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

[12] S. Marsland, J. Shapiro and U. Nehmzow, A self-organising network that grows when required, *Neural networks* **15**(8–9) (2002), 1041–1058.

[13] G.I. Parisi, J. Tani, C. Weber and S. Wermter, Lifelong learning of human actions with deep neural network self-organization, *Neural Networks* **96** (2017), 137–149.

[14] T. Kohonen, The self-organizing map, *Proceedings of the IEEE* **78**(9) (1990), 1464–1480.

[15] A. Forti and G.L. Foresti, Growing Hierarchical Tree SOM: An unsupervised neural network with dynamic topology, *Neural networks* **19**(10) (2006), 1568–1580.

[16] F. Shen, H. Yu, K. Sakurai and O. Hasegawa, An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network, *Neural Computing and Applications* **20**(7) (2011), 1061–1074.

[17] S. Furao and O. Hasegawa, An incremental network for on-line unsupervised classification and topology learning, *Neural networks* **19**(1) (2006), 90–106.

[18] E.J. Palomo and E. López-Rubio, The growing hierarchical neural gas self-organizing neural network, *IEEE transactions on neural networks and learning systems* **28**(9) (2016), 2000–2009.

[19] A. Andreakis, N.v. Hoyningen-Huene and M. Beetz, Incremental unsupervised time series analysis using merge growing neural gas, in: *International Workshop on Self-Organizing Maps*, Springer, 2009, pp. 10–18.

[20] E.J. Palomo, M.A. Molina-Cabello, E. López-Rubio and R.M. Luque-Baena, A New Self-Organizing Neural Gas Model based on Bregman Divergences, in: *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, pp. 1–8.

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

[22] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu and P. Havinga, Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey, in: *23th International conference on architecture of computing systems 2010*, VDE, 2010, pp. 1–10.

[23] L. Chen and C. Nugent, Ontology-based activity recognition in intelligent pervasive environments, *International Journal of Web Information Systems* (2009).

[24] G. Meditskos and I. Kompatsiaris, iKnow: Ontology-driven situational awareness for the recognition of activities of daily living, *Pervasive and Mobile Computing* **40** (2017), 17–41.

[25] K. Gayathri, K. Easwarakumar and S. Elias, Probabilistic ontology based activity recognition in smart homes using Markov Logic Network, *Knowledge-Based Systems* **121** (2017), 173–184.

[26] G. Civitarese, T. Sztyler, D. Riboni, C. Bettini and H. Stuckenschmidt, POLARIS: Probabilistic and ontological activity recognition in smart-homes, *IEEE Transactions on Knowledge and Data Engineering* (2019).

[27] J. Ye, G. Stevenson and S. Dobson, USMART: An unsupervised semantic mining activity recognition technique, *ACM Transactions on Interactive Intelligent Systems (TiiS)* **4**(4) (2014), 1–27.

[28] G. Okeyo, L. Chen, H. Wang and R. Sterritt, Dynamic sensor data segmentation for real-time knowledge-driven activity recognition, *Pervasive and Mobile Computing* **10** (2014), 155–172.

[29] Y. Kwon, K. Kang and C. Bae, Unsupervised learning for human activity recognition using smartphone sensors, *Expert Systems with Applications* **41**(14) (2014), 6067–6074.

[30] D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou and Y. Amirat, An unsupervised approach for automatic activity recognition based on hidden Markov model regression, *IEEE Transactions on Automation Science and Engineering* **10**(3) (2013), 829–835.

[31] M.-S. Lee, J.-G. Lim, K.-R. Park and D.-S. Kwon, Unsupervised clustering for abnormality detection based on the tri-axial accelerometer, *ICCAS-SICE* **2009** (2009), 134–137.

[32] Z.S. Abdallah, M.M. Gaber, B. Srinivasan and S. Krishnaswamy, Activity recognition with evolving data streams: A review, *ACM Computing Surveys (CSUR)* **51**(4) (2018), 71.

[33] B. Longstaff, S. Reddy and D. Estrin, Improving activity classification for health applications on mobile devices using active and semi-supervised learning, in: *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, IEEE, 2010, pp. 1–7.

[34] Y.-S. Lee and S.-B. Cho, Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data, *Neurocomputing* **126** (2014), 106–115.

[35] D. Guan, W. Yuan, Y.-K. Lee, A. Gavrilov and S. Lee, Activity recognition based on semi-supervised learning, in: *Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007. 13th IEEE International Conference on*, IEEE, 2007, pp. 469–475.

[36] E. Hoque and J. Stankovic, AALO: Activity recognition in smart homes using Active Learning in the presence of Overlapped activities, in: *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, IEEE, 2012, pp. 139–146.

[37] T. Miu, P. Missier and T. Plötz, Bootstrapping personalised human activity recognition models using online active learning, in: *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, IEEE, 2015, pp. 1138–1147.

[38] Z.S. Abdallah, M.M. Gaber, B. Srinivasan and S. Krishnaswamy, Adaptive mobile activity recognition system with evolving data streams, *Neurocomputing* **150** (2015), 304–317.

[39] H.S. Hossain, M.A.A.H. Khan and N. Roy, Active learning enabled activity recognition, *Pervasive and Mobile Computing* **38** (2017), 312–330.

[40] Y. Chen, J. Wang, M. Huang and H. Yu, Cross-position activity recognition with stratified transfer learning, *Pervasive and Mobile Computing* **57** (2019), 1–13. doi:https://doi.org/10.1016/j.pmcj.2019.04.004. http://www.sciencedirect.com/science/article/pii/S1574119218303432.

[41] Y. Chang, A. Mathur, A. Isopoussu, J. Song and F. Kawsar, A Systematic Study of Unsupervised Domain Adaptation for Robust Human-Activity Recognition, *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **4**(1) (2020). doi:10.1145/3380985.

[42] Q. Wu, K. He and X. Chen, Personalized federated learning for intelligent iot applications: A cloud-edge based framework, *IEEE Computer Graphics and Applications* (2020).

[43] Y. Chen, X. Qin, J. Wang, C. Yu and W. Gao, FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare, *IEEE Intelligent Systems* (2020), 1–1.

[44] K. Sozinov, V. Vlassov and S. Girdzijauskas, Human Activity Recognition Using Federated Learning, in: *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, IEEE, 2018, pp. 1103–1111.

[45] Y. Zhao, H. Liu, H. Li, P. Barnaghi and H. Haddadi, Semi-supervised Federated Learning for Activity Recognition, *arXiv preprint arXiv:2011.00851* (2020).

[46] B. Liu, Y. Jiang, F. Sha and R. Govindan, Cloud-enabled privacy-preserving collaborative learning for mobile sensing, in: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012, pp. 57–70.

[47] S.A. Osia, A. Taheri, A.S. Shamsabadi, K. Katevas, H. Haddadi and H.R. Rabiee, Deep private-feature extraction, *IEEE Transactions on Knowledge and Data Engineering* **32**(1) (2018), 54–66.

[48] Y. Cheng, Convergence and Ordering of Kohonen's Batch Map, *Neural Comput.* **9**(8) (1997), 1667–1676–.

[49] G.I. Parisi, J. Tani, C. Weber and S. Wermter, Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization, *Frontiers in neurorobotics* **12** (2018), 78.

[50] A. Reiss and D. Stricker, Pamap2 physical activity monitoring data set, *Retrieved April* **30** (2012), 2019.

[51] B. Barshan and M.C. Yüksek, Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units, *The Computer Journal* **57**(11) (2014), 1649–1667.

[52] G.I. Parisi, C. Weber and S. Wermter, Self-organizing neural integration of pose-motion features for human action recognition, *Frontiers in Neurorobotics* **9** (2015).

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research* **12** (2011), 2825–2830.

[54] Y. Hu, B. Wang, Y. Sun, J. An and Z. Wang, Graph-Based Semi-Supervised Learning for Activity Labeling in Health Smart Home, *IEEE Access* **8** (2020), 193655–193664.

[55] C. Wiwatcharakoses and D. Berrar, SOINN+, a Self-Organizing Incremental Neural Network for Unsupervised Learning from Noisy Data Streams, *Expert Systems with Applications* **143** (2020), 113069. doi:https://doi.org/10.1016/j.eswa.2019.113069. http://www.sciencedirect.com/science/article/pii/S0957417419307869.