Pervasive and Mobile Computing 104 (2024) 101972

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Edge human activity recognition using federated learning on constrained devices ☆

Angelo Trotta [a,*], Federico Montori [a,b], Leonardo Ciabattini [a], Giulio Billi [a], Luciano Bononi [a], Marco Di Felice [a,b]

[a] *University of Bologna, Italy*
[b] *Advanced Research Center for Electronic Systems, Bologna, Italy*

ARTICLE INFO

Keywords:
Human Activity Recognition (HAR)
Internet of Things (IoT)
Edge computing
Federated Learning

ABSTRACT

Human Activity Recognition (HAR) using wearable Internet of Things (IoT) devices represents a well investigated researched field encompassing various application domains. Many current approaches rely on cloud-based methodologies for gathering data from diverse users, resulting in the creation of extensive training datasets. Although this strategy facilitates the application of powerful Machine Learning (ML) techniques, it raises significant privacy concerns, which can become particularly severe given the sensitivity of HAR data. Moreover, the labeling process can be extremely time-consuming and even more challenging for IoT wearable devices due to the absence of efficient input systems. In this paper, we address both aforementioned challenges by designing, implementing, and validating edge-based Human Activity Recognition (HAR) systems that operate on resource-constrained IoT devices, which relies on the utilization of Self-Organizing Maps (SOM) for activity detection. We incorporate a feature selection process before training to reduce data dimensionality and, consequently, the SOM size, aligning with the resource limitations of wearable IoT devices. Additionally, we explore the application of Federated Learning (FL) techniques for HAR tasks, enabling new users to leverage SOM models trained by others on their respective datasets. Our federated Extreme Edge (EE)-aware HAR system is implemented on a wearable IoT device and rigorously tested against state-of-the-art and experimental datasets. The results demonstrate that our C++-based SOM implementation achieves a consistent reduction in model size compared to state-of-the-art approaches. Furthermore, our findings highlight the effectiveness of the FL-based approach in overcoming personalized training challenges, particularly in onboarding scenarios.

1. Introduction

The widespread adoption of wearable Internet of Things (IoT) devices, equipped with Inertial Measurement Units (IMU) sensors, has fueled the development of Human Activity Recognition (HAR) systems across diverse application domains, spanning healthcare, fitness, and smart mobility. In telerehabilitation systems, HAR solutions enable fine-grained monitoring of motor activities performed by remotely connected patients [2,3]. Within the fitness domain, they support the synthesis of advanced statistics by combining sports performance metrics with physiological responses [4,5]. Likewise, numerous smart city mobile applications leverage HAR techniques for tasks such as mobility trace collections and parking spot recommendations [6,7]. Despite the varied techniques

☆ A preliminary version of this paper has appeared in *IEEE SMARTCOMP 2023* [1].

* Corresponding author.
E-mail address: angelo.trotta5@unibo.it (A. Trotta).

https://doi.org/10.1016/j.pmcj.2024.101972
Received 22 December 2023; Received in revised form 29 May 2024; Accepted 3 August 2024
Available online 8 August 2024

employed for activity recognition, most of the existing works follow a common, cloud-centric architecture [8]: data gathered from different users are centralized, and a robust Machine Learning (ML) model is trained on-situ [5]. While this methodology leads to the creation of extensive datasets [9] and enhances the accuracy of ML models, it introduces potential concerns related to privacy and scalability. Indeed, HAR raw data, being inherently sensitive, may expose users to vulnerabilities even with anonymization techniques due to the possibility of re-identification attacks [10]. Furthermore, supervised ML techniques require manual labeling of raw IMU data, which can be particularly cumbersome for wearable IoT devices that lack efficient input systems.

To address privacy concerns, the adoption of edge computing has become a prevalent technique, enabling data processing near IoT devices [11]. However, leveraging edge computing for Human Activity Recognition (HAR) systems introduces novel challenges that must be considered. Indeed, wearable IoT devices, commonly utilized for HAR data collection, are inherently constrained in terms of computational and storage resources [12]. Consequently, many complex, supervised Deep Learning (DL) techniques proposed for HAR may be impractical to implement. Moreover, the restriction on data sharing poses a unique challenge, requiring users to train on their own data for all the activities they aim to monitor. While this approach may not always lead to a performance drop [13], it may significantly affect the application of HAR systems, given the cost and time challenges associated with the training phase, and specifically the aforementioned constraints of the labeling process.

In this paper, we address the design, implementation, and validation of an effective edge-friendly solution for HAR tasks. Our innovative methodology integrates established and novel data processing techniques to address the privacy and scalability challenges while enhancing activity detection accuracy. Specifically, our solution leverages unsupervised Self-Organizing Maps (SOM) to minimize manual data labeling [14]. Its implementation on resource-constrained IoT devices is further facilitated by a feature selection phase aimed at balancing the SOM size with HAR accuracy. Additionally, we tackle the onboarding challenge, occurring when new users join the HAR task without prior training. In this regard, we explore the application of Federated Learning (FL) techniques on top of SOM models [15,16], in order to allow users to benefit from HAR knowledge shared by other peers without accessing raw data, hence preserving their privacy. Our HAR technique is implemented on a real IoT wearable prototype (ESP32 M5Stack), encompassing the full architecture with edge data management functionalities. The C++ implementation of SOM ensures a model size reduction of up to 8 times compared to state-of-the-art solutions [17]. We validate our approach using the popular UCI HAR dataset and a newly collected dataset through our prototype when worn by different users. In both cases, our approach demonstrates comparable performance to state-of-the-art DL architectures, with only an 8% difference. The FL solution enhances personalized learning for onboarding scenarios and remains comparable to centralized models while avoiding the need to share raw user data.

Specifically, four main contributions are provided in this paper:

- We design and develop an Extreme Edge (EE) Human Activity Recognition (HAR) system based on the Self Organizing Map (SOM) technique, complemented by a feature selection phase. We investigate the trade-off considerations involving SOM size, input feature dimensions, and HAR accuracy.
- We extend the application of the solo ML technique to a distributed scenario, accommodating multiple users eager to participate in the same HAR task without sharing their raw sensor data. To achieve this, we explore the integration of Federated Learning (FL) techniques with the SOM model, incorporating weight exchange and updates.
- We implement the proposed solution on a prototype IoT device, integrated into an edge–cloud framework; the prototype has been used to build a small-case HAR training set which is made available online for further exploration.
- We compare the performance of the EE-HAR system against state-of-the-art DL models based on centralized learning, This evaluation encompasses both our dataset and the UCI dataset. Furthermore, we assess the enhancements brought about by the FL approach in our proposed solution.

Compared to our previous work [1], this extended version includes several significant enhancements and novel contributions. We have designed a FL approach specifically for HAR systems in extreme edge scenarios that builds on SOMs, handling the limited computational resources and data privacy concerns inherent in these environments. Additionally, we have implemented a three-layer architecture comprising extreme edge devices, edge nodes, and a cloud node, facilitating improved data processing, model training, and system scalability. Our performance analysis includes a detailed examination of the onboarding scenarios where new, untrained users are introduced into the system. This is extremely important, as in real HAR scenarios it is unlikely that everyone participate in the FL training phase at the same time, as in most of the FL benchmarks in literature. Moreover, in this paper we have published a new HAR dataset featuring multiple subjects, enhancing the reproducibility of our experiments and providing a valuable resource for the research community.

The rest of the paper is structured as follows. Section 2 reviews the state of the art of HAR techniques focusing on edge-computing solutions. Section 3 presents the proposed HAR architecture and the interactions among components. Section 4 focuses on the learning techniques. Section 5 introduces the extension to distributed environments based on FL. Section 6 details the prototype implementation on the target IoT device. Section 7 describes the performance evaluation utilizing both the UCI and our dataset. Section 8 concludes the work.

2. Related works

HAR emerged in the academic domain during the 1980s, gaining prominence in the past two decades, notably within sectors such as elder care, healthcare, and ambient intelligence scenarios [18]. HAR approaches broadly fall into two categories: knowledge-based and data-driven. The former explores semantic correlations among event types and activity classes [19], while the latter entails the collection and subsequent analysis of sensory data. Activity recognition can be viewed as a pattern-matching problem, and various DL techniques have been suggested to address it [20]. Despite the availability of various public HAR datasets [21,22], the proliferation of novel applications underscores the need for finely tuned and personalized datasets. The challenge is accentuated in supervised DL models, which necessitate labeled data for dataset creation. In order to alleviate such challenge with HAR systems, researchers are delving into both unsupervised and semi-supervised classification methods [23].

In the domain of unsupervised approaches, the work presented in [24] introduces a cluster-based framework designed to label multidimensional inertial signals. This approach includes through two distinct phases: firstly, a recurrent auto-encoder extracts meaningful spatiotemporal features from the inertial data. Subsequently, in the second phase, these extracted features seamlessly integrate into an unsupervised clustering algorithm, facilitating the prediction of unlabeled signals. A comprehensive survey on the application of clustering techniques to inertial signals can be found in [25]. Among the semi-supervised approaches, Unsupervised Domain Adaptation (UDA) techniques have emerged as potent tools for distinguishing new users from existing ones and aligning their unlabeled data with features present in labeled datasets. In this landscape, the SALIENCE architecture, described in [23], employs discriminators at both the feature and sample label levels. Additionally, an Attention-based Neural Network is employed to identify sensors housing strongly discriminating features, prioritizing their significance in the learning process. Another study, presented in [26], introduces an architecture comprising a bidirectional Generative Adversarial Neural Network (Bi-GAN) coupled with Kernel Mean Matching (KMM), in order to facilitate the transfer of activity detection knowledge across datasets characterized by heterogeneous feature spaces.

FL represents a compelling approach for HAR, providing a dual advantage of effective learning and privacy preservation. In [16], the authors introduce a Federated Learning System for HAR (HARFLS), relying on federated averaging and incorporating a novel Perceptive Extraction Network (PEN). The study illustrates that even the least performing DL model within this framework outperforms the best performing ML technique in terms of average F1 score. In other solutions reported in [13,27,28], authors describe the integration of FL with semi-supervised learning and clustering methodologies, addressing challenges associated with unlabeled data and non-independently and identically distributed data.

In the following, we focus on the research studies that are most related to the contributions of our study. In [15], the authors introduce two federated models – a Deep Neural Network (DNN) and a softmax regression – customized to operate on vitalized edge devices with constrained resources. Moving on to [29], the GWEP method is presented as an FL approach rooted in model compression. This strategy employs joint quantization and model pruning, addressing the balance between efficiency and computational constraints to align with the capabilities of resource-constrained devices. In our previous study [12], we investigated the application of dynamic range quantization. This technique is implemented to reduce the size of the Convolutional Neural Network (CNN) loaded onto a microcontroller unit. [30] delves into the efficiency of various binarized neural networks (BNN) for classifying human activities, which are extreme examples of quantization: they constraint weights, and activation functions to binary values. Exploring the classification of daily gesture samples collected through an inertial ring and bracelet, [31], employs a spectrum of supervised and unsupervised ML methods. The study investigates the efficacy of unsupervised techniques, such as the K-Means and Gaussian Mixture Model, alongside their supervised counterparts. In [17] a SOM and a Convolutional Neural Network (CNN) collaborate on a microcontroller unit, proving the viability of feature extraction methodologies in resource-constrained environments.

Our paper differs from previous works in the following aspects: (*i*) we showcase the implementation of a novel SOM library for the Arduino Software Development Kit (SDK), resulting in a substantial reduction of the model size compared to [17]; (*ii*) we introduce a pre-processing mechanism, consisting of ANOVA-F feature selection, enabling further reduction in the amount of data processed on the EE; (*iii*) we meticulously assess the trade-off between SOM accuracy and model size across various configurations of both techniques; (*iv*) we implement FL as a privacy-aware method for onboarding new users into the HAR system.

3. System architecture for cooperative HAR

In this Section, we present the architecture of our HAR system, specifically designed for wearable, resource-constrained IoT devices. The system architecture includes a cooperative module to address the challenges of efficient and privacy-aware computing in environments with limited computational power, storage, and data processing capabilities. The architecture has been designed to fulfill the following requirements:

- *Flexibility*: users can define and add new activities dynamically.
- *Customizability*: the system adapts to different user needs, allowing for personalized training phases.
- *Ease of Use*: the system minimizes manual data labeling, simplifying the user experience.
- *Cooperation*: enables multiple users to contribute to the model training process, enhancing the overall system performance and accuracy.

To achieve these objectives, our architecture incorporates three types of computational nodes: the IoT wearable device (the EE device), the user's edge node, and an external server which includes a cooperative functionality for enhanced model training. The HAR process is divided into three phases: data gathering, model training, and HAR inference. These computational phases are allocated statically to a device: data gathering and HAR inference take place on the wearable device, while the training phase is conducted betweeen the central server and the user's edge node, depending on the type of training (single, federated or centralized, more on this in Section 5). The architecture is depicted in Fig. 1. Below, we provide detailed descriptions for each phase.

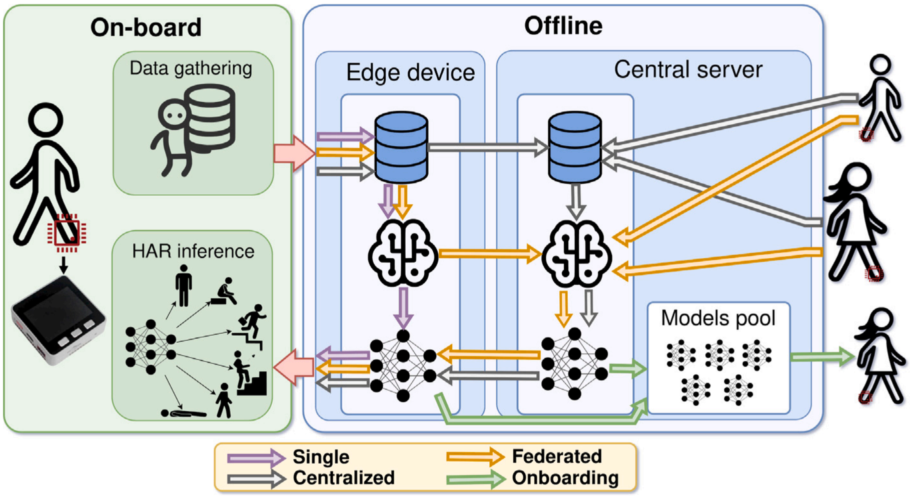


Fig. 1. The proposed HAR architecture. The purple path indicates the *single* approach; the gray path specify the *centralized* approach; the yellow path indicates the *federated* system. With the green arrows we specify the *on-boarding* process where new users use pre-trained model.

- *Data gathering*: in this initial phase, the IoT device actively samples raw sensor data while the user performs various activities. The system utilizes IMU sensors, including accelerometers, gyroscopes, and magnetometers, that operate at a predetermined sampling rate. During this stage, the system is designed to operate autonomously, eliminating the need for user intervention for data labeling. The raw data collected from the IMU sensors is then wirelessly transmitted to an edge node. Here, it is collected into unlabeled HAR datasets for subsequent model training phases.
- *Model training*: this phase is pivotal in developing the ML model using the HAR dataset gathered from the IoT devices. Initially, the IMU data undergoes a preprocessing stage where outliers and missing values are identified and removed. To enhance data quality, noise filtering techniques, particularly the Butterworth low-pass filter, are applied. Subsequently, feature extraction is carried out on the sensor data, focusing on both frequency and temporal domains. This process is elaborated in Section 7.1. The training of ML models is conducted using the preprocessed datasets. Our approach utilizes the SOM technique, detailed in Section 4.1, which organizes the data into K distinct classes. The number of classes, K, is either predetermined or dynamically calculated from the data using methods like the elbow method [32]. At this stage, we assign semantic labels to each class, correlating them with specific human activities. Notably, the labeling occurs post the application of the unsupervised SOM method, differing from the example-based labeling in supervised learning. To enhance efficiency, we have incorporated a feature selection phase using the ANOVA-F technique (see Section 4.2). This step aims to minimize the number of input features for the SOM, thereby reducing the processing load.
  The model training phase offers two distinct methodologies: single and cooperative training:
  - *Single Training*: in this approach, individual users train the SOM model solely with their data, ensuring privacy as the data does not leave the user's edge device. This flow is depicted in Fig. 1 with purple arrows.
  - *Cooperative Training*: this method can be executed with two different methods:
    - *Data-Level Cooperation (Centralized Model)*: data from various users are collected on the external server to train a comprehensive SOM model. This centralized model benefits from the diverse data pool, potentially enhancing accuracy and generalizability. This is shown in Fig. 1 with gray arrows. We can see here that the data are collected in the central server and the ML model is trained using the aggregated dataset from all the cooperating users.
    - *Training-Level Cooperation (Federated System)*: here, each device independently trains a local model. Instead of raw data, only SOM model parameters are shared with the server, which then aggregates these to refine the global model. This federated approach maintains data privacy by keeping raw data localized. In Fig. 1 this flow is indicated with yellow arrows. Notably, here the raw data are not shared.

  The system can generate multiple SOM models for varying network sizes, with the optimal model being one that balances high accuracy with the storage constraints of the IoT device. Upon completion of the training phase, these trained SOM models are stored in a *models pool* on the central server, which acts as a central repository (green arrows in Fig. 1). This setup ensures that the models are readily available for deployment to users' IoT devices, facilitating the practical application of the HAR system in diverse real-world environments.
- *HAR inference*: post-training, the trained SOM model (either from local or cooperative training), along with the selected feature set, is selected from the *model pool* and it is automatically transferred back to the IoT device. This pool enables also the onboarding feature (see Section 5) where new users utilize a pre-trained model, avoiding the data gathering and model training

phases It is important to note that while this process ensures the efficient deployment of the model, the specific aspects of protection and security related to model transfer and deployment are not covered in this work, while our focus is primarily on the functionality and performance of the HAR system. The device processes real-time IMU data to extract features, which are then fed into the SOM for activity classification. Once the activities are classified, the HAR system communicates the results to the user. This communication can be facilitated through various output peripherals, such as a display screen, or transmitted to other devices for further use or analysis.

4. Learning models for the EE

In this Section, we present the ML and pre-processing techniques used in the model training module of Fig. 1. We rely on Self Organizing Maps (SOMs) for human activity classification and the ANOVA-F technique for feature reduction.

4.1. Self-organizing map

The self-organizing map (SOM) [14] is a type of unsupervised learning algorithm that can be used for dimensionality reduction and clustering. SOMs are a form of artificial neural network that can map high-dimensional input data onto a lower-dimensional output space while preserving the topological properties of the input data. In our HAR system, we used SOMs as ML techniques to address the HAR task for the reasons below:

- They are lightweight and computationally efficient, making them well-suited for deployment on constrained devices with limited computational power and storage capacity [17]. This is in contrast to other ML/DL techniques which can have high storage requirements and long computational times. KNN, for example, requires storing all training data; FNN and SVM, on the other hand, have a complex implementation of transfer and kernel functions, respectively.
- They are well-suited for unsupervised learning, which is particularly useful for HAR systems as pointed out in Section 2. Indeed, SOMs can discover patterns or structures in the data without any labeled information, which can be useful for identifying features or patterns in the sensor data that are associated with different activities.
- They can be easily customized for specific sensor data formats or feature reduction methods, which is essential for deployment on constrained devices. By optimizing the SOM model size for the specific constraints of the device, we have a strategy to address the trade-off between performance, energy consumption and storage.

More in detail, a SOM is a distinct type of artificial neural network that employs competitive learning for training. In this approach, nodes compete for the opportunity to respond to specific subsets of input data, as opposed to error-correction learning methods (e.g., backpropagation with gradient descent) utilized by other artificial neural networks. Like most artificial neural networks, SOMs support two primary modes: training and mapping. The training phase uses an input data set, or "input space", to create a lower-dimensional representation known as the "map space". The mapping phase then employs this generated map to classify additional input data. Typically, the training process aims to represent an input space with p dimensions as a two-dimensional map space. An input space consists of p features, each representing a dimension. The map space is composed of *nodes* or *neurons*, which are organized in a two-dimensional rectangular grid of size $N \times N$. Each node n_i in the map space in the grid can be hence identified via its coordinates $\langle x_i, y_i \rangle$, with $0 < i < N$. The node n_i is linked to a weight vector w_i of p dimensions, that signifies the node's position in the input space. Although nodes in the map space remain stationary, the training process involves adjusting weight vectors towards the input data (by minimizing a distance metric such as *Euclidean* distance) without disrupting the topology derived from the map space. Once trained, the map can classify additional observations in the input space by identifying the node with the nearest weight vector (i.e., the smallest distance metric) to the input space.

During the initialization of the SOM algorithm, the neurons' weights are assigned random values. Then, iterating over the input data, for each training example a, the Best Matching Unit (BMU) is calculated:

$$\mathrm{bmu} = \arg\min_i \{ \|a - w_i\| \}$$

The weight of the BMU node, and its neighborhood, is then updated as follows:

$$w_i = w_i + \eta(t) \cdot h_{i,\mathrm{bmu}}(t) \cdot (a - w_i)$$

Here, $\eta(t)$ is the learning rate at learning iteration t defined with a learning rate decay rule $\eta(t) = \eta_0 \cdot e^{(-t/\hat{\eta})}$, with η_0 and $\hat{\eta}$ as custom parameters. The neighborhood kernel function $h_{i,\mathrm{bmu}}(t)$ defines the neighborhood size and is defined by:

$$h_{i,\mathrm{bmu}}(t) = e^{-\frac{d^2_{i,\mathrm{bmu}}}{2\cdot\theta^2(t)}}$$

where $\theta(t) = \theta_0 \cdot e^{-t/\hat{\theta}}$ defines the neighborhood size decay rule, with θ_0 and $\hat{\theta}$ as custom parameters. The distance $d_{i,j}$ between two nodes is calculated as a Manhattan distance, i.e., $d_{i,j} = |x_i - x_j| + |y_i - y_j|$.

The critical factors for deployment on a constrained EE device are reflected by the number of weights w_i. The size of the SOM map is thus defined by the number of neurons, i.e., the grid $N \times N$, and the number of features p. Consequently, the total model size is $\mathcal{O}(N^2 \cdot p)$. In the following Section, we introduce a feature reduction mechanism to decrease the value of p. Additionally, in Section 7.4, we evaluate various SOM sizes to attain satisfactory classification performance while enabling model deployment on constrained EE devices.

4.2. ANOVA-F feature reduction

Feature reduction is an essential step in building efficient ML models for HAR. Indeed, complex ML models tend to use all possible features as inputs and internally disregard features that have little to no influence in discriminating the predicted class. Vice versa, for wearable devices with constrained resources, it is crucial to reduce the number of features *before* building the final model without compromising its accuracy. Several popular methods can be employed for feature reduction, such as Principal Component Analysis (PCA); however, they demand high computation and storage resources during the inference stage.

In the EE scenario, we chose an offline method that is able to reduce the number of features, i.e. the ANOVA-F technique, which can be conducted a priori and not during the inference phase. This technique is a statistical method that can identify the most significant features of the model [33]. In order to calculate the ANOVA-F value of a feature f we first calculate, for each of the K classes, the variance of that feature within such class, then we divide it by the variance of such feature over the whole dataset. More formally, The ANOVA-F value of feature f for class $c \in K$ is given by:

$$\mathrm{An}_c(f) = \frac{\sigma^2(f)_c}{\sigma^2(f)}$$

Ideally, a low value of $\mathrm{An}_c(f)$ means that f is highly discriminant for class c, because the values of f do not change significantly among the members of class c compared to how much they change normally in the dataset. Conversely, values of $\mathrm{An}_c(f)$ that approach 1.0 or above signify that feature f is likely to just add noise for class c. In order to come up with a single ANOVA-F value for each feature f, for this reason, an aggregation function over all instances of ANOVA-F values of f for all K classes is implemented. We conducted our experiments by selecting the *average* and the *minimum* as aggregation functions, which are defined as follows:

$$\mathrm{An}_{avg}(f) = \frac{\sum_{c\in C} \mathrm{An}_c(f)}{K},$$

$$\mathrm{An}_{min}(f) = min\{\mathrm{An}_c(f) | c \in K\}$$

The first associates each feature with an ANOVA-F value by taking into account its overall influence over all classes, while the second only considers the class for which the feature is most influential. Once calculated, we can rank all the p features in ascending order by their ANOVA-F value and keep only the ones ranked best. Therefore, we introduce the ANOVA-F Threshold (An_{thr}), which is the ANOVA-F value below which we keep all the features. This variable becomes a relevant parameter of our system, which depends on the available storage on the IoT device and has to be balanced with other parameters in order to achieve the best accuracy, such as the SOM size.

5. Collective context

The architecture we have detailed in Section 3 presents a versatile and adaptive framework for EE-friendly HAR systems. It incorporates SOM techniques and offers both single and cooperative training methodologies to cater to diverse user needs and privacy concerns. Here, by envisioning the usage of the device in a real context, we could imagine that the owner of a device is introduced in a scenario where a number of other individuals are using the same device. In such a scenario, assuming that each device hosts a model with a fixed shape, how do we train each of the models? We could, on the one hand, deliver a clean slate device so that the users shall perform the training process themselves. On the other hand, we could take advantage of the collective intelligence and perform the model training as a community of users, where all of them share their sensor data to train a single, general model. This however raises privacy concerns, which are unacceptable in certain use cases, unless adopting an FL approach.

More formally, in a collective context, we must select one from three different configurations, as outlined in Fig. 2.

Single. In this case, each device is delivered with an untrained model. Users are then committed to performing training in autonomy, generating a local model that is meant to be tailored to the users themselves. Given its specificity, if the training is conducted correctly, the model is likely to show a high accuracy on the activities of the owner and, since the model is local, it also maximizes her privacy.

Centralized. In this case, each device comes with a pre-trained model, which is generated by the collective training of a number of users who shared their training data and made up a sufficiently heterogeneous training dataset. As sensor data is completely disclosed, a central entity can interpret it as a single dataset and perform the training process only once, sharing the model with all participants. This solution allows users to avoid the local training process, however, in certain cases (i.e. healthcare) sharing plain sensor data with a central entity is not acceptable.

Federated. In this case, each device comes with an untrained model, and all users participate in an FL process mediated by a central entity. In this paper, we implement FL with SOM, which is architecturally similar to classic FL implementations on neural networks using FedAvg [34]. However, unlike the classic FedAvg, our aggregation strategy is unweighted, meaning that each user will have the same impact on the results regardless of the size of their dataset. The rationale behind this choice is to avoid excessively large datasets taking over, as they would cause the results to be too representative of a few users. This can be quite counterproductive in HAR systems, given their tendency to be personalized. This method resembles the Centralized structure, still preserving privacy and, presumably, acknowledging a small drop in performance. In this case, the owner of the device is still committed to performing the training process locally, however, the process results in a single model, which is not tied to a single individual.

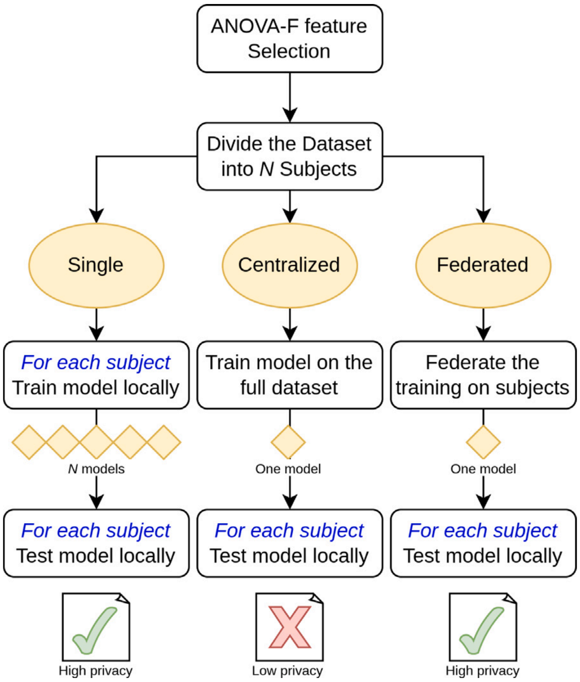


Fig. 2. Flow chart of possible application scenarios: single, centralized, and federated.

Onboarding process

The previous considerations outlined in this Section seem to privilege a *Single* approach, as HAR upon sensor data is strongly dependent on the individual who is performing the training [7]. However, it is likely that such a model would only suit the owner and cannot be efficiently reused. In fact, in a dynamic and realistic scenario, we should consider the onboarding issues. With "onboarding" we mean the process through which a new individual becomes part of the network and, in our case, is given a new device. Now, with a *Single* approach, this means that users would need to train the model from scratch, which, in certain cases, can be either a hassle or unfeasible, because maybe the users are not expert enough to conduct a meaningful training process. This is when a collective model (either through a *Centralized* or a *Federated* approach) comes in handy because it allows a newcomer to utilize a product off-the-shelf, versus performing a tedious and difficult configuration process. For this reason, as reflected in Section 7 we evaluate the collective approach, both in (i) an *offline* context, where all participants take part in the training process, and in (ii) an *onboarding* context when the system is at steady state and a new user, incapable of training a model, enters the system. More in detail, the onboarding process is conducted in the following way, with respect to the approaches:

Single The newcomer onboards by adopting a randomly chosen model from one of the existing users.
Centralized The newcomer onboards by adopting the centralized model.
Federated The newcomer onboards by adopting the federated model.

6. Implementation

The deployment of the HAR system depicted in Fig. 1 is based on the *M5Stack*,[1] which is a modular, stackable, and programmable device designed to support pervasive IoT applications. It is equipped with multiple IMU sensors, including an accelerometer, gyroscope, and magnetometer, making it ideal for HAR data collection tasks. The computational unit is based on the popular ESP32 microcontroller with 240 MHz dual-core processors and 320 kB of RAM memory. We choose this device to validate our HAR architecture due to its compact and energy-efficient profile; however, we emphasize that our firmware can be easily adapted for other IoT boards that support the Arduino SDK.

During the experiments, we positioned the device on users' ankles as this location provides optimal gait analysis, as reported by [35]. Generally speaking, the development of ML models, including the SOM, on micro-controllers is highly challenging and requires optimizing the code in order to fit the memory constraints, which, for the case of the M5Stack, is less than 400 kB. Traditional approaches involve quantization techniques [11], which aim to reduce the size of an ML trained out of the

[1] https://docs.m5stack.com/en/core/gray.

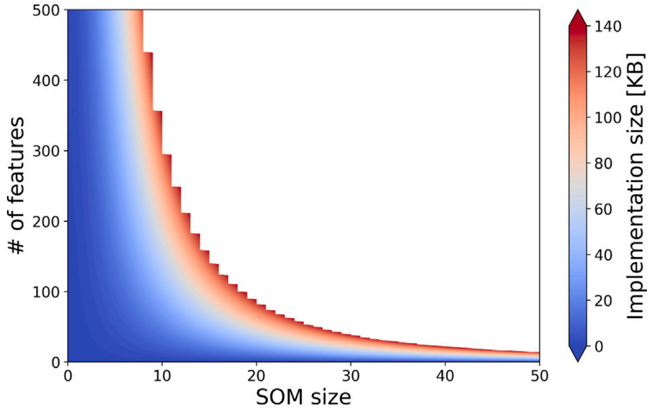


Fig. 3. Impact of the SOM size and the number of features on the deployed firmware. The color indicates how close the memory size of the system is to the limits imposed by the microcontroller. Areas not reached by a bar indicate that the system is not deployable on the device.

microcontroller by reducing the precision of floating point coefficients [12]. To this aim, frameworks like TensorFlow Lite [36] support the quantization process and the exportation of the model in a binary format; the latter is loaded by an interpreter running on the microcontroller. However, we were unable to follow such an approach as the generated code still exceeded the memory capacity of the M5Stack device.

To overcome these challenges, we opted to implement a SOM library from scratch in *C++* for microcontrollers, taking into account the hardware characteristics of the latter. The SOM library for the Arduino SDK is available online[2] and can be easily customized to support different microprocessors than the ESP-32 one.

Through our implementation, the size of the final deployed SOM model is restricted to several tens of kilobytes. The M5Stack has 320 kB of RAM memory, and after an initial code optimization phase that encompasses sensor reading and wireless communication, the maximum memory available for the SOM model is 140 kB only. As described in Section 4.1, the SOM size depends on two variables: the number of features and the map size. In Fig. 3 we depict the relationship between these two variables and the final size of the SOM model to be installed in the wearable device. In the Figure, only feasible configurations are shown. In Section 7.4 we investigate the trade-off between the number of input features, the map size, and the overall accuracy of the HAR task.

7. Performance evaluation

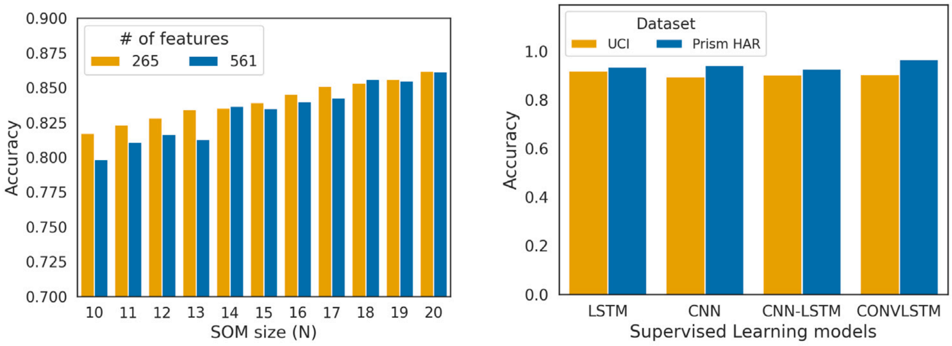
In this Section, we evaluate the performance of the proposed HAR system for resource-constrained IoT devices from different points of view. The evaluation is organized into five phases: (*i*) defining the HAR datasets, (*ii*) analyzing the classic supervised learning techniques on the HAR datasets to establish a reference baseline in a standalone context, (*iii*) assessing the effectiveness of the ANOVA-F technique for feature reduction, (*iv*) analyzing the standalone performance of the SOM on the two datasets, to evaluate the accuracy and memory size occupation for different map sizes, and (v) analyzing the collective approach in both offline and onboarding contexts.

7.1. Description of the datasets

We evaluated our proposed HAR system on two different datasets, one from the literature and a custom one, obtained by gathering data from a few volunteers through the *M5Stack* device.

Regarding the first, we used the well-known *University of California Irvine (UCI) HAR* dataset [21]. It was collected using the accelerometer and gyroscope sensors of a smartphone carried by 30 volunteers performing six different activities: *walking*, *walking upstairs*, *walking downstairs*, *sitting*, *standing*, and *lying down*. The dataset includes time-series data for the three axes of each sensor, as well as the corresponding activity label. The dataset has been widely used in research for developing and evaluating HAR models based on ML techniques. More in detail, each record in the dataset is provided with a vector composed of 561 features, calculated in the time and frequency domain. The features were sourced from the accelerometer and gyroscope 3-axial raw signals, which were sampled at a constant rate of 50Hz. To eliminate noise, a median filter and a 3rd-order low pass *Butterworth filter* with a corner frequency of 20Hz were employed. Another low-pass Butterworth filter with a corner frequency of 0.3Hz was used to split the acceleration signal into body and gravity acceleration signals. After that, the body's linear acceleration and angular velocity were used to derive *Jerk signals* in time. The magnitude of these three-dimensional signals was determined using the Euclidean norm. Finally, a Fast Fourier Transform (FFT) was applied to some of these signals.

[2] https://github.com/UniBO-PRISMLab/extreme-edge-som.



(a) (b)

Fig. 4. 4(a) compares the accuracy of SOMs of different sizes for the UCI HAR dataset, when using all 561 features or a subset of 265 features only, excluding the frequency domain. 4(b) shows a comparison between different supervised techniques over the two datasets.

However, not all of these features are equally essential for accurate HAR systems. Some may be redundant or noisy, while others may be more informative. In [1], we assessed the performance of SOMs of various sizes over the UCI HAR dataset using different feature subsets, which suggests that the frequency-based features did not significantly improve the performance of our HAR system, at least not for the prediction models taken into account in this paper. An accurate result is shown in Fig. 4(a), which demonstrates that the prediction accuracy using the entire set of 561 features and the sole subset of 265 features, obtained by excluding frequency-domain features, behave quite similarly. Quite counterintuitively, results show that a smaller set of features tends to perform slightly better, which suggests that the full set of features is likely oversized for a simple model like our SOM. As a result, we decided to remove the frequency-based features and focus solely on time-domain and statistical features in our deployment. This results in a dataset with 10,299 rows with 265 features each.

The second dataset was built by the authors, aiming to resemble similar conditions to the ones observed in the UCI HAR dataset, in order to perform a meaningful comparison. We attached the M5Stack wearable device on the ankle of 8 different volunteers, who then performed the same six activities present in the UCI HAR dataset. Data was sampled with a rate of 41.29Hz and underwent the same procedures used on the UCI dataset, such as Butterworth filters and Jerk calculation. We then calculated the same selected 265 features, skipping the calculation of the frequency-based ones, obtaining a dataset with 16,976 rows and 265 features each. In our case, the dataset is intentionally unbalanced, with 10,916 rows associated with the most contributory subject, and 560 rows with the least contributory one. As the dataset is collected within the IoT-Prism Lab at the University of Bologna, we named the dataset **Prism HAR** and made it publicly accessible.[3]

7.2. Supervised learning for HAR

In this Section, we show the performance of state-of-the-art supervised learning methods for HAR applications. These techniques rely on labeled training data to build a model that can predict the activity performed by the user, while unsupervised learning techniques aim to discover patterns or structures in the data without any labeled information. Due to the constrained EE device used in our HAR system, in the following Sections, we will deeply focus on evaluating the performance of unsupervised learning techniques instead. This is because unsupervised learning techniques, such as SOMs, are lightweight and computationally efficient, making them well-suited for deployment on constrained devices. However, to have a meaningful comparison perspective, we analyze different supervised learning techniques for HAR applications as a baseline. Specifically, we evaluated the following techniques: long short-term memory (LSTM), convolutional neural network (CNN), CNN–LSTM, and convolutional LSTM (CONVLSTM).

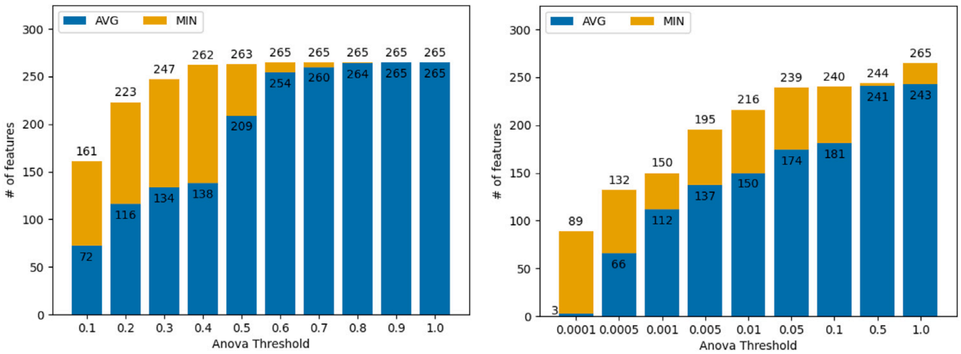
Fig. 4(b) shows that supervised learning techniques are very effective in classifying human activities, with all four models achieving high accuracy. The Figure depicts the results for both datasets. Here, the CNN–LSTM model achieved the highest overall performance, with an accuracy of 99% on our Prism HAR dataset. This evaluation shows also the effectiveness of placing the sensors in the right position, *i.e.*, the ankle, instead of using the sensors inside the smartphone. The results of the supervised learning techniques over the UCI HAR dataset are around 8% less accurate than the one executed on our dataset.

It is however necessary to remark that, while supervised learning methods are highly effective in the classification of HAR systems, they are not suitable for deployment in EE devices. Our M5Stack device, in fact, has 140 kB available memory only. On the contrary, the supervised learning models require significant computational resources and storage capacity.

7.3. Features reduction with ANOVA-F

In this Section, we present the results of the ANOVA-F feature reduction method on both the UCI HAR dataset and our custom dataset. We analyzed the two different variations of the ANOVA-F method described in Section 4.2: one using the average as the

[3] https://github.com/UniBO-PRISMLab/Prism-HAR-dataset.



(a) UCI HAR Dataset (b) Prism HAR Dataset

Fig. 5. Feature reduction using *AVG* and *MIN* for different value of An_{thr}.

aggregation function and another using the minimum as the aggregation function, in the following called *AVG* and *MIN*, respectively. The aim of this analysis was to identify the most relevant features for our ML model and reduce their number to minimize storage usage, as requested in previous Section 6. Our results show that both variations of the ANOVA-F method were effective in reducing the number of features used in our classification model. Figs. 5(a) and 5(b) display the outcomes of features reduction obtained by applying distinct An_{thr} on both the UCI HAR and Prism HAR datasets. It should be noted that the scales on the x-axes are different in the Figures (UCI dataset in Fig. 5(a) and Prism dataset in Fig. 5(b)). This disparity in scaling is because in the Prism dataset, setting the An_{thr} to 0.5 resulted in almost all of the input features being included without any feature reduction. Therefore, to evaluate the Prism dataset, we used exponential steps for the An_{thr}.

The results show that the variation using *AVG* as the aggregation function resulted in a greater reduction in the number of features compared to the *MIN* version, especially for low levels of the An_{thr}. Based on these results, we selected the *AVG* variation of the ANOVA-F method for the next evaluations in order to achieve a significant reduction in the number of features in input to our model.

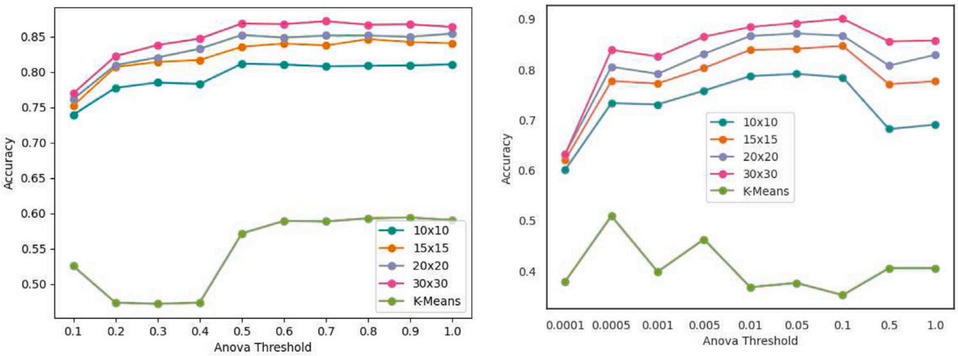
7.4. SOM for extreme edge devices

In this Section, we present the evaluation process undertaken to determine the optimal configuration for the SOM and the ANOVA-F threshold (An_{thr}) in our HAR model. Our primary goal was to achieve a satisfactory classification performance while maintaining a compact model size that is suitable for deployment on the M5Stack device. To ensure the robustness and generalizability of our model, we analyze the performance of SOMs with the legacy *K-Means* algorithm for comparison purposes. We conducted a series of experiments to assess the performance of our HAR model by varying the size of the SOM and the An_{thr}. The SOM sizes tested ranged from small (e.g., 10×10) to larger maps (e.g., 30×30), while the An_{thr} values were varied between 0.0001 and 1 for the Prism dataset and between 0.1 and 1 for the UCI dataset. This range of values was chosen to cover a wide spectrum of model complexities, thereby providing a comprehensive understanding of the trade-off between model size and performance.

Upon analyzing the results in Figs. 6(a) and 6(b) from both the UCI HAR dataset and the Prism HAR dataset, we observed that the HAR model's performance improved by increasing the SOM size. However, for sizes greater than 15×15, the increase in accuracy is not so evident. Regarding An_{thr}, we notice a decrease in performance for low values in both datasets and for high values only in the Prism dataset. Here, we can notice that the optimal performance is different for the two datasets. The UCI dataset performs well starting from An_{thr} of 0.5, i.e., from about 209 features (see Fig. 5(a)). The Prism dataset performs well with An_{thr} values between 0.0005 and 0.1, i.e., with a value between 66 and 181 features. It is clear that with larger SOMs and higher An_{thr}, and hence more features, the model would also demand more computational resources and storage, which may not be available on the M5Stack device.

Taking these considerations into account, we determined that the optimal configuration for our HAR model, when deployed locally, is a SOM size of 15×15 and An_{thr} of 0.005. For the deployment, we considered the results from our custom dataset. This configuration demonstrated a satisfactory balance between model accuracy and computational complexity, making it well-suited for deployment on our wearable device which has a limit of 140 kB.

To further evaluate the practical feasibility of deploying our HAR models on resource-constrained devices, we conducted energy consumption tests comparing the power usage of the SOM algorithm (with a 10×10 size) with the K-Means algorithm. This analysis is crucial as wearable devices have constrained battery capacities and additional energy consumption can impact their lifetime. The results depicted in Fig. 7 indicate that the SOM method consumes approximately 0.035mJ more per classification operation compared to the K-Means algorithm. Although there is a minor increase in energy consumption, this trade-off is justified by the significant improvement in classification accuracy provided by the SOM-based model.



(a) UCI HAR Dataset (b) Prism HAR Dataset

Fig. 6. The accuracy evaluated with *K-Means*, different SOM sizes by varying An_{thr} for the UCI dataset is shown in Fig. 6(a). The accuracy evaluated with *K-Means*, different SOM sizes by varying An_{thr} for our custom dataset is shown in Fig. 6(b).

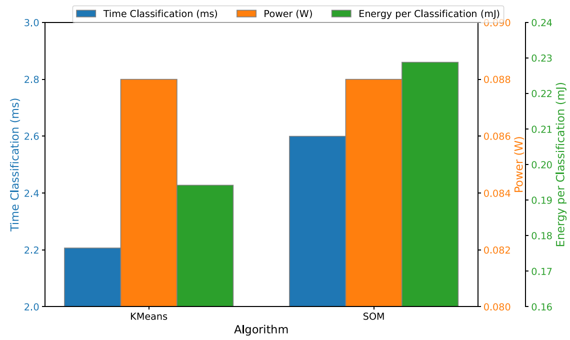
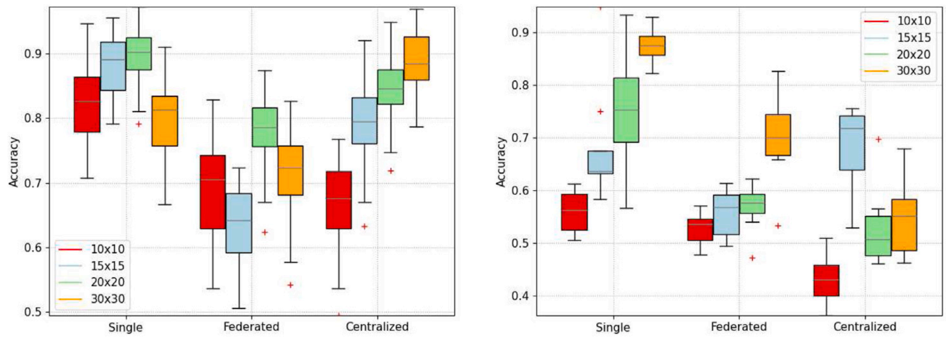


Fig. 7. Energy consumption comparison between K-Means and SOM per classification operation.



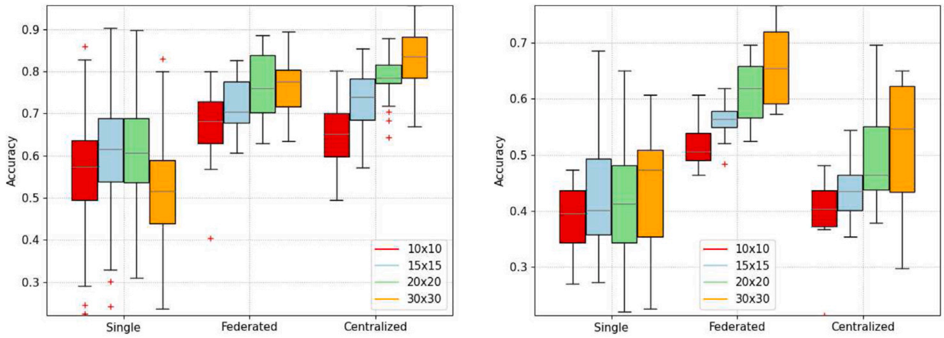
(a) UCI HAR Dataset (b) Prism HAR Dataset

Fig. 8. Evaluation of the collective offline approach.

7.5. Evaluating our SOM on the collective context

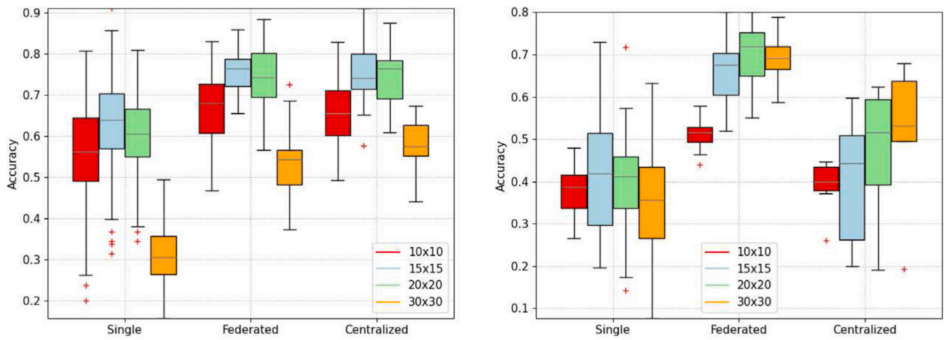
In this last stage of our performance evaluation, we aim to show how our HAR proposed system behaves in a real scenario, where it is evaluated on single users. In particular, the outcome of this performance evaluation demonstrates whether the results outlined in the previous section, which encompass an ideal case, would be impacted by a real scenario.

First of all, we evaluated our system under the three different configurations outlined in Section 5: Single, Centralized, and Federated considering an *offline approach*, thus assuming that every user participated in the training process and no user joins at a later stage. This is an assumption that serves as a baseline for the collective context, as we do not deem such a setup as realistic. The results of the evaluation are reported in Fig. 8 for both datasets and for SOM sizes of 10×10, 15×15, 20×20, and 30×30,



(a) UCI HAR Dataset (b) Prism HAR Dataset

Fig. 9. Evaluation of the collective onboarding approach.



(a) UCI HAR Dataset (b) Prism HAR Dataset

Fig. 10. Evaluation of the collective onboarding approach with the maximum number of features fitting the EE device.

compatibly with our standalone experiments. In this test, we used all 265 features, which means that the models are not necessarily deployable on an EE device. We will remove this assumption later on, here we wanted to focus the evaluation on the different collective configurations, without introducing further variables. Even though in this case the experiment shows slightly different trends on the two datasets, we still can state that the Single configuration yields the best results for almost all SOM sizes. This justifies our earlier statement, that HAR highly depends on the single individual, while a generalized model tends to have worse performance, as people have different movement patterns when they perform physical activities. Furthermore, we notice that the Centralized configuration performs better than the Federated one for the UCI HAR dataset, while it happens almost the opposite for the Prism HAR dataset across all experiments. This happens because the Prism HAR dataset is highly unbalanced, with ∼65% of it attributed to a single user. This implies that the Centralized configuration will most likely train the model accounting greatly for that single user, while the others will have a negligible impact. The testing phase notably outputs the unweighted average accuracy across the users, resulting in the majority of them using a model that is mostly associated with someone else who will likely walk, sit, or run differently from them. Instead, the Federated configuration gives the same importance to all users regardless of the size of their contribution, showing that, even in the presence of fewer data points for the training process, this has a positive influence on the results.

Next, we evaluated the *onboarding approach* by assuming that one of the users is onboarding the system and cannot perform any training. We simulated such a scenario by performing a Leave-One-Out Cross Validation (LOOCV), which results in one user being removed from the training phase. Then, once the training phase is over, whatever the configuration, the user onboards and adopts a pre-trained model as explained in Section 5. In all cases, we repeated the LOOCV by leaving out each of the users in the dataset and then performing the onboarding. For the Single case, because the model adopted by the onboarding user is randomly chosen among all the others, we further repeated the experiment for each of the other users. Fig. 9 shows the results, which display an expected behavior: the Single configuration now has the worst performance because the onboarding user uses a model that is specifically trained on another user, that is, not generalized. Furthermore, as expected, the performance of both Federated and Centralized configurations is quite similar to the offline case, suggesting that both of them are minimally affected by a new user onboarding the system. It is also noticeable how, for smaller SOMs, the performance of the Federated approach increases with respect to its centralized counterpart, which leads to our next result.

Our last experiment repeats the LOOCV on both datasets, this time reducing the number of features from 265 to the maximum number that is acceptable for a SOM to be deployed into the EE system, always selecting the ones with the smallest ANOVA-f value. In particular, the number of considered features is as follows:

- SOM size: 10 - max features: 356 (we keep all 265 features here)
- SOM size: 15 - max features: 158
- SOM size: 20 - max features: 89
- SOM size: 30 - max features: 39

Results, shown in Fig. 10, have a similar trend to the previous test, with the difference that here the Federated configuration performs in line with the Centralized one for the UCI HAR dataset, while still performing better in the Prism HAR dataset. In fact, decreasing the number of features has a much more negative impact on the Centralized configuration, while the Federated one seems more resilient — this is more evident for the bigger models because the number of features drops more significantly. This suggests that, in a realistic scenario, when a new user onboards the systems and is given an off-the-shelf model, then a Federated configuration yields the best results, provided the model is sized appropriately to fit a constrained architecture.

8. Conclusions

In this paper, we presented the design, implementation, and validation of an Extreme Edge (EE)-aware Human Activity Recognition (HAR) system. The system aims to classify human activity on constrained and wearable IoT devices equipped with inertial sensors, by considering the unique challenges posed by the computational environment, such as the scarcity of hardware resources and the impracticability of a data labeling phase for some individuals. Our HAR system incorporates a feature selection mechanism, based on the ANOVA-F technique, to reduce the dimensionality of HAR features at the input stage. Then, it utilizes an unsupervised classification technique based on Self-Organizing Maps (SOMs) to enable data separation and effective activity classification. We developed a C++ SOM library for the Arduino SDK and validated our system through an M5Stack IoT wearable board. The experimental results take place in two phases and use two HAR datasets: the UCI HAR dataset and the Prism HAR dataset built through our IoT prototype. First, our standalone experiments demonstrated that the proposed SOM solution is capable of outperforming other unsupervised approaches and achieves performance close to state-of-the-art DL techniques while generating a model small enough to fit the limited memory capabilities of EE devices. Second, our collective experiments show the advantages of a Federated Learning (FL) approach when considering users that onboard the system without the capacity to train their model, while still considering privacy implications. From our experiments it is also evident that HAR systems are highly personalized, that is, if an individual can train his/her own HAR model, then this will likely perform better than any collective one. This implies that any user who can participate in the model training will not take advantage of the collective model, as he/she would be better off using the one trained locally without any external influence. However, because this assumes that the training is conducted properly, which is not a trivial task, then inexperienced users who are incapable of conducting extensive training would be better off taking advantage of the collective model, especially if they onboard the system at a later stage. In this latter case, we have shown that a centralized approach suffers from unbalanced datasets, while an unweighted FL solution makes sure that all contributing users have the same impact regardless of the size of their dataset and it is in general a better choice.

In the future, there might be a focus on enhancing the system through exploring different methods for selecting features and adapting our approach to suit various wearable devices and application areas.

CRediT authorship contribution statement

Angelo Trotta: Writing – original draft, Visualization, Validation, Project administration, Data curation. **Federico Montori:** Writing – original draft, Methodology, Formal analysis, Conceptualization. **Leonardo Ciabattini:** Writing – review & editing, Resources, Investigation. **Giulio Billi:** Visualization, Software, Data curation. **Luciano Bononi:** Supervision. **Marco Di Felice:** Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have shared the link in the text of the paper.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to improve language and readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Acknowledgments

The work has been supported by the PRIN PNRR 2022 project SORTT ("An IoT-Serviced and Ontology-based Remote human digital Twin for physioTherapy: a feasibility study", project id: P2022H74YP) and by the PRIN 2022 project I-TROPHYTS ("IoT and humanoid RObotics for autonomic PHYsio-Therapeutic monitoring, coaching and supervising in smart Spaces: a feasibility study", project id: 20224TAETP).

References

[1] A. Trotta, F. Montori, G. Vallasciani, L. Bononi, M. Di Felice, Optimizing IoT-based human activity recognition on extreme edge devices, in: 2023 IEEE International Conference on Smart Computing, SMARTCOMP, IEEE, 2023, pp. 41–48, http://dx.doi.org/10.1109/SMARTCOMP58114.2023.00023.
[2] Y. Chen, L. Yu, K. Ota, M. Dong, Robust activity recognition for aging society, IEEE J. Biomed. Health Inf. 22 (6) (2018) 1754–1764, http://dx.doi.org/10.1109/JBHI.2018.2819182.
[3] M.A. Khatun, M.A. Yousuf, S. Ahmed, M.Z. Uddin, S.A. Alyami, S. Al-Ashhab, H.F. Akhdar, A. Khan, A. Azad, M.A. Moni, Deep CNN-LSTM with self-attention model for human activity recognition using wearable sensor, IEEE J. Transl. Eng. Health Med. 10 (2022) 1–16, http://dx.doi.org/10.1109/JTEHM.2022.3177710.
[4] Y.-L. Hsu, S.-C. Yang, H.-C. Chang, H.-C. Lai, Human daily and sport activity recognition using a wearable inertial sensor network, IEEE Access 6 (2018) 31715–31728, http://dx.doi.org/10.1109/ACCESS.2018.2839766.
[5] D. Bouchabou, S.M. Nguyen, C. Lohr, B. LeDuc, I. Kanellos, A survey of human activity recognition in smart homes based on IoT sensors algorithms: Taxonomies, challenges, and opportunities with deep learning, Sensors 21 (18) (2021) URL https://www.mdpi.com/1424-8220/21/18/6037.
[6] R. Salpietro, L. Bedogni, M. Di Felice, L. Bononi, Park here! a smart parking system based on smartphones' embedded sensors and short range communication technologies, in: 2015 IEEE 2nd World Forum on Internet of Things, WF-IoT, 2015, pp. 18–23, http://dx.doi.org/10.1109/WF-IoT.2015.7389020.
[7] A.R.M. Forkan, F. Montori, D. Georgakopoulos, P.P. Jayaraman, A. Yavari, A. Morshed, An industrial IoT solution for evaluating workers' performance via activity recognition, in: 2019 IEEE 39th International Conference on Distributed Computing Systems, ICDCS, IEEE, 2019, pp. 1393–1403.
[8] G. De Leonardis, S. Rosati, G. Balestra, V. Agostini, E. Panero, L. Gastaldi, M. Knaflitz, Human activity recognition by wearable sensors: Comparison of different classifiers for real-time applications, in: 2018 IEEE International Symposium on Medical Measurements and Applications, MeMeA, IEEE, 2018, pp. 1–6.
[9] G. Alam, I. McChesney, P. Nicholl, J. Rafferty, Open datasets in human activity recognition research—Issues and challenges: A review, IEEE Sens. J. 23 (22) (2023) 26952–26980, http://dx.doi.org/10.1109/JSEN.2023.3317645.
[10] J. Dai, B. Saghafi, J. Wu, J. Konrad, P. Ishwar, Towards privacy-preserving recognition of human activities, in: 2015 IEEE International Conference on Image Processing, ICIP, IEEE, 2015, pp. 4238–4242.
[11] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, P. Hui, Edge intelligence: Architectures, challenges, and applications, 2020, http://dx.doi.org/10.48550/ARXIV.2003.12172, URL https://arxiv.org/abs/2003.12172.
[12] A. Ghibellini, L. Bononi, M. Di Felice, Intelligence at the IoT edge: Activity recognition with low-power microcontrollers and convolutional neural networks, in: 2022 IEEE 19th Annual Consumer Communications & Networking Conference, CCNC, 2022, pp. 707–710, http://dx.doi.org/10.1109/CCNC49033.2022.9700665.
[13] R. Presotto, G. Civitarese, C. Bettini, Fedclar: Federated clustering for personalized sensor-based human activity recognition, in: 2022 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, 2022, pp. 227–236.
[14] T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480, http://dx.doi.org/10.1109/5.58325.
[15] K. Sozinov, V. Vlassov, 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.
[16] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, B. Zhao, A federated learning system with enhanced feature extraction for human activity recognition, Knowl.-Based Syst. 229 (2021) 107338.
[17] P.-E. Novac, A. Castagnetti, A. Russo, B. Miramond, A. Pegatoquet, F. Verdier, Toward unsupervised human activity recognition on microcontroller units, in: 2020 23rd Euromicro Conference on Digital System Design, DSD, IEEE, 2020, pp. 542–550.
[18] E. Kim, A. Helal, D. Cook, Human activity recognition and pattern discovery, IEEE Pervasive Comput. 9 (2010) http://dx.doi.org/10.1109/MPRV.2010.7.
[19] D. Riboni, F. Murru, Unsupervised recognition of multi-resident activities in smart-homes, IEEE Access 8 (2020) http://dx.doi.org/10.1109/ACCESS.2020.3036226.
[20] F. Demrozi, G. Pravadelli, A. Bihorac, P. Rashidi, Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey, IEEE Access 8 (2020) 210816–210836, http://dx.doi.org/10.1109/ACCESS.2020.3037715.
[21] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, Comput. Intell. Mach. Learn. ESANN (2013).
[22] S. Zhang, Y. Li, S. Zhang, F. Shahabi, S. Xia, Y. Deng, N. Alshurafa, Deep learning in human activity recognition with wearable sensors: A review on advances, Sensors 22 (4) (2022) http://dx.doi.org/10.3390/s22041476, URL https://www.mdpi.com/1424-8220/22/4/1476.
[23] L. Chen, Y. Zhang, S. Miao, S. Zhu, R. Hu, L. Peng, M. Lv, SALIENCE: An unsupervised user adaptation model for multiple wearable sensors based human activity recognition, IEEE Trans. Mob. Comput. (2022) 1, http://dx.doi.org/10.1109/TMC.2022.3171312.
[24] H. Amrani, D. Micucci, P. Napoletano, Unsupervised deep learning-based clustering for human activity recognition, in: 2022 IEEE 12th International Conference on Consumer Electronics, ICCE-Berlin, 2022, pp. 1–6, http://dx.doi.org/10.1109/ICCE-Berlin56473.2022.9937141.
[25] P. Ariza Colpas, E. Vicario, E. De-La-Hoz-Franco, M. Pineres-Melo, A. Oviedo-Carrascal, F. Patara, Unsupervised human activity recognition using the clustering approach: A review, Sensors 20 (9) (2020) http://dx.doi.org/10.3390/s20092702.
[26] A.R. Sanabria, F. Zambonelli, J. Ye, Unsupervised domain adaptation in activity recognition: A GAN-based approach, IEEE Access 9 (2021) 19421–19438, http://dx.doi.org/10.1109/ACCESS.2021.3053704.
[27] R. Presotto, G. Civitarese, C. Bettini, Semi-supervised and personalized federated activity recognition based on active learning and label propagation, Pers. Ubiquitous Comput. 26 (5) (2022) 1281–1298, http://dx.doi.org/10.1007/s00779-022-01688-8.
[28] R. Presotto, G. Civitarese, C. Bettini, Federated clustering and semi-supervised learning: A new partnership for personalized human activity recognition, Pervasive Mob. Comput. 88 (2023) 101726, http://dx.doi.org/10.1016/j.pmcj.2022.101726, URL https://www.sciencedirect.com/science/article/pii/S1574119222001390.
[29] P. Prakash, J. Ding, R. Chen, X. Qin, M. Shu, Q. Cui, Y. Guo, M. Pan, IoT device friendly and communication-efficient federated learning via joint model pruning and quantization, IEEE Internet Things J. 9 (2022) 13638–13650, http://dx.doi.org/10.1109/jiot.2022.3145865.
[30] F. Luo, S. Khan, Y. Huang, K. Wu, Binarized neural network for edge intelligence of sensor-based human activity recognition, IEEE Trans. Mob. Comput. 22 (3) (2023) 1356–1368, http://dx.doi.org/10.1109/TMC.2021.3109940.

[31] A. Moschetti, L. Fiorini, D. Esposito, P. Dario, F. Cavallo, Daily activity recognition with inertial ring and bracelet: An unsupervised approach, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, 2017, pp. 3250–3255, http://dx.doi.org/10.1109/ICRA.2017.7989370.

[32] J. Han, M. Kamber, J. Pei, Data Mining Concepts and Techniques, Third Edition, Morgan Kaufmann Publishers, Waltham, Mass., 2012.

[33] X. Li, J. Lin, Linear time complexity time series classification with bag-of-pattern-features, in: 2017 IEEE International Conference on Data Mining, ICDM, IEEE, 2017, pp. 277–286.

[34] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[35] A.R. Anwary, H. Yu, M. Vassallo, Optimal foot location for placing wearable IMU sensors and automatic feature extraction for gait analysis, IEEE Sens. J. 18 (6) (2018) 2555–2567, http://dx.doi.org/10.1109/JSEN.2017.2786587.

[36] R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang, et al., Tensorflow lite micro: Embedded machine learning for tinyml systems, Proc. Mach. Learn. Syst. 3 (2021) 800–811.