On-device Semi-supervised Activity Detection: A New Privacy-aware Personalized Health Monitoring Approach

Avirup Roy , Hrishikesh Dutta , Amit Kumar Bhuyan , Subir Kumar Biswas *

Posted Date: 11 June 2024

doi: 10.20944/preprints202406.0523.v1

Keywords: Semi-supervised learning; privacy-preserving; personalized machine learning; human activity detection; wearable devices; on-device learning; health-monitoring.

 doi:10.20944/preprints202406.0523.v1

Article

On-Device Semi-Supervised Activity Detection: A New Privacy-Aware Personalized Health Monitoring Approach

Avirup Roy, Hrishikesh Dutta, Amit Kumar Bhuyan and Subir Biswas *

Department of Electrical and Computer Engineering, Michigan State University; royaviru@msu.edu (A.R.); duttahr1@msu.edu (H.D.); bhuyanam@msu.edu (A.K.B.)

* Correspondence: sbiswas@msu.edu

Abstract: This paper presents an on-device semi-supervised human activity detection system that can learn and predict human activity patterns in real time. The proposed semi-supervised learning (SSL) framework uses sparsely labelled user activity events acquired from Inertial Measurement Unit (IMU) sensors installed as wearable devices. The objective is to learn classification of multiple classes of human activities in real-time. The proposed cluster-based learning model in this approach is trained with data from the same target user, thus preserving data privacy while providing personalized activity detection services. Two different cluster labelling strategies, namely, population-based, and distance-based, are employed to achieve the desired classification performance. A comparative study of these strategies has been presented in terms of classifiability and classification accuracies. The proposed system is shown to be computationally efficient, which is relevant in the context of limited computing resources on typical wearable devices. The trade-off between classification accuracy and computation complexity is analyzed for different algorithmic hyper-parameters and other system parameters. Extensive experimentation and simulation study have been conducted on multi-user human activity data from the public domain in order to validate the proposed learning paradigm.

Keywords: Semi-supervised learning; privacy-preserving; personalized machine learning; human activity detection; wearable devices; on-device learning; health-monitoring

1. Introduction

An increasingly sedentary lifestyle in modern societies has compounded many health problems such as diabetes, obesity, and high blood pressure. High resolution monitoring of daily activities and providing personalized feedback are found to be effective in reducing sedentary lifestyle. There are many commercially available wearable products such as the Apple Watch [1] and Fitbit watch [2] can monitor and classify human activities using supervised machine learning methodologies. Such learning typically requires model training using pre-labeled data collected from many users. Once centrally trained, the model is loaded in those wearable devices before consumers can use them.

While this approach of model training works, the design cycle can be improved in the following two fronts. First, the activity data for supervised learning model training can potentially raise privacy concerns. That is because the activity data from a large population needs to be recorded for training. Inadvertent access to such personal fitness and health-oriented data can be abused by insurance companies, and organizations during job placement and hiring. Second, the model used in commercially available wearable is trained using data collected from a generalized population. This contrasts with model training in a subject-specific manner, which is known to provide better classification accuracies for activity monitoring applications.

This paper sets out to address these two design issues by employing a self-supervised activity classification approach in which activity data is recorded from the same subject for whom a classification model is trained iteratively. By not relying on data from others, it addresses the privacy

 doi:10.20944/preprints202406.0523.v1

concerns, and by training a semi-supervised model with the same person's activity data, thus making it personalized. In the proposed approach, the entire learning process happens on-device, thus eliminating the needs for: i) transferring data over network links, and ii) storing and processing it on an out-of-device system. These further add to privacy preservation. The tradeoff is that the mechanism needs to be computationally and storage-wise light enough to be able to run on low-power embedded devices

The on-device semi-supervised learning (SSL) [3] approach is specifically designed for learning with sparsely labelled datapoints. This light-weight mechanism is fully implemented in the embedded wearable devices. The targeted application scenario is as follows. When a user starts using such a device, the semi-supervised learning algorithm in it starts "training itself" with the activity data collected from the device-integrated inertial measurement sensors (IMU) such as accelerometers. Based on pre-stored templates, if the algorithm can classify a current activity, it labels that data. As the device is used, this mechanism is used to label a small subset of all collected activity data. The self-supervised training and classification are performed simultaneously in an iterative manner. As the device is worn by the user for longer durations, the objective is to improve the classification performance over time. Special care is taken to reduce the computational and storage complexity of the iterative mechanism to keep it suitable for low-power wearable embedded devices. Furthermore, since the training and data collection both happen on the same device, the target application setting ensures full privacy by not allowing the user data to ever leave the device. Personalization is achieved since the algorithm that classifies a user's activities is trained mainly by data from the same user.

The main contributions of the paper are as follows. First, it introduces a privacy-preserving and personalized semi-supervised learning mechanism that can perform human activity classification with accuracies that are comparable to neural network based supervised learning. It also explores how this can be achieved with fewer labeled data, and with low computational resources represented by computational and memory complexities. Second, two SSL variants, namely, population-based, and distance-based, are explored and evaluated for the application considered in the paper. Finally, all the above mechanisms are experimentally analyzed and evaluated from the standpoints of classifiability, classification accuracy, and computational/storage complexity. Accuracy figures are compared with those from benchmark supervised learning models.

2. Related Work

Human activity detection (HAD) in commercially available devices such as smartphones and smart watches is often performed with embedded sensors such as accelerometers and gyroscopes [4]. Different combinations of sensing modalities and their positions on the human body have been explored in the literature [5–8]. On the machine learning front, different forms of HAD systems have been realized using a variety of classifiers based on supervised learning [7,9]. The above works explore different elements of HAD systems like intraclass variability and interclass similarity. The challenges of disparity in relevant and irrelevant data have also been discussed.

To deal with person-specific activity types, which are often affected by individual physical attributes [10,11] and such, personalized human activity detection has been explored in [12–21]. While [12] deals with it using biometry, [13–21] deal at a classifier level. The works in [13] and [16] train supervised learning models using personalized data from the target user. Both of these works involve transfer of the data collected from the target user to be transferred to an external server for supervised training. This method affects user privacy since the data is shared to an external device. [17–19] consider transfer learning in a way of domain adaptation methods. Transfer learning has the convenience where explicit training data is not required from the target user device but has the disadvantage of increased likelihood of selecting incorrect class data for training and thus degrading the overall classification performance. [20,21] present similar works which consider personalized learning models for classification tasks.

On-device learning is an effective solution for applications related to personalized applications. [22] uses similar pre-trained supervised learning models for on-device hydration-tracking. This hydration tracking system also preserves the privacy of the target user data, but the usage of the

 doi:10.20944/preprints202406.0523.v1

supervised learning model requires a huge amount of training data. At times enough labelled data may not be available from the target user to train a supervised learning model. In this way, either privacy or personalization or both for a classifier model may be sacrificed which becomes a point of major concern for these kind of applications which require both privacy and personalization.

Semi-supervised learning (SSL) [3] is a possible solution for on-device learning min which classifiers can be trained on the device from a specific target user's data, thus preserving the personalization feature of the model as discussed in [23]. SSL is also useful in this context because unlike in fully supervised learning, learning with SSL does not rely on the availability of extensive labelled data. This was shown in [24] that the SSL models can potentially learn with sparsely labelled datapoints. This mechanism has proven to be effective for on-device applications involving limited computational power. [25] uses an SSL method based on the combination of deep learning and transfer learning. The usage of a deep learning model involves training using data from different users other than the target user. This affects the personalization feature of our premise in this paper. The work in [26] develops a human activity detection system using a bi-view semi-supervised learning to detect semantic human activities like having dinner, shopping, etc. This method also uses a windowed datapoint extraction technique and clustering mechanism as the basis of the classifier model, but not on the same device where the sensors are present. This method also involves a two-layered framework for the classification task which becomes computationally expensive thus not very suitable for an on-device self-training and classification.

The above works in the literature provide many definitive ideas for classifying human activities with or without using personalized classifiers. Semi-supervised learning has been proposed as an effective tool to train personalized classifiers of HAD, but none of these works considers the feasibility of these training mechanisms on devices with computational constraints. The work in this paper addresses that issue by developing a low-computation self-training mechanism using a semi-supervised learning algorithm. Both the classification accuracy and computational complexity of the proposed approach are considered as the evaluation parameters. The different hyper-parameters and system parameters have been analyzed based on their relevance in implementation on a wearable device with computational constraints.

3. System and Data Model

This section outlines the adopted system and data model used by the proposed semi-supervised algorithms. Figure 1 depicts the entire system where the wearable device on the wrist of the user reads the human motion using the embedded IMU sensors. The different classes of human activities are classified using the Semi-supervised learning model self-trained using the motion data from the same user on the same wearable device.

 doi:10.20944/preprints202406.0523.v1

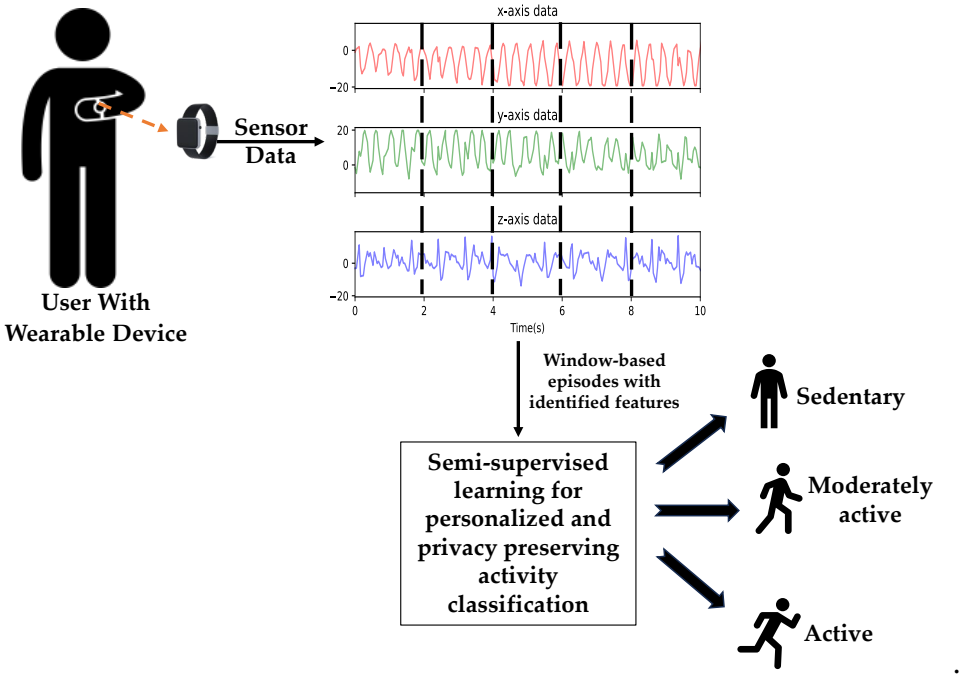


Figure 1. A wearable on-device Human Activity Detection system.

3.1. Sensing Modality and Dataset

Activity classification is performed on data collected from a wearable embedded device containing Inertial Measurement Unit (IMU) sensors, storage, and a micro-controller. The IMU sensors capture user movements in terms of acceleration of the wearable unit along three orthogonal axes. We have used a public-domain human activity dataset (i.e., Wireless Sensor Data Mining Lab-WISDM lab dataset from University of California, Irvine [27]) generated from such a wearable unit. The dataset contains accelerometer time-series data collected from a smartwatch as 36 subjects perform 6 activities.

3.2. Segmentation and Feature Extraction

As a first processing step, the raw accelerometer data is pre-processed as follows. First, l^2 norm [28] is applied to standardize the data across all axes in order to ensure that the data is bounded within the range of -1 to +1. Subsequently, a window-based approach is used for segmenting data into 2-second-long episodes. For each episode, two features are defined for each of the accelerometer axes. The first feature is coefficient of variation, which reflects the acceleration variability across 40 samples spanning the 2 seconds long episode. The second feature is the number of mean crossing points within an episode. Physically, this one indicates the frequency of movements of the subject from its mean position. Based on their physical implications, these two features are selected for their discriminatory abilities towards activity classification.

3.3. Class Definition

User activities are categorized into three distinct classes, namely, sedentary, moderately active, and active. The sedentary class encompasses stationary activities such as standing and sitting. Moderately active activities involve ambulatory motion, primarily represented by walking. Finally, the active class pertains to more vigorous physical activities, such as jogging. The correlation among all the features over the entire activity dataset is depicted in Figure 2.

 doi:10.20944/preprints202406.0523.v1

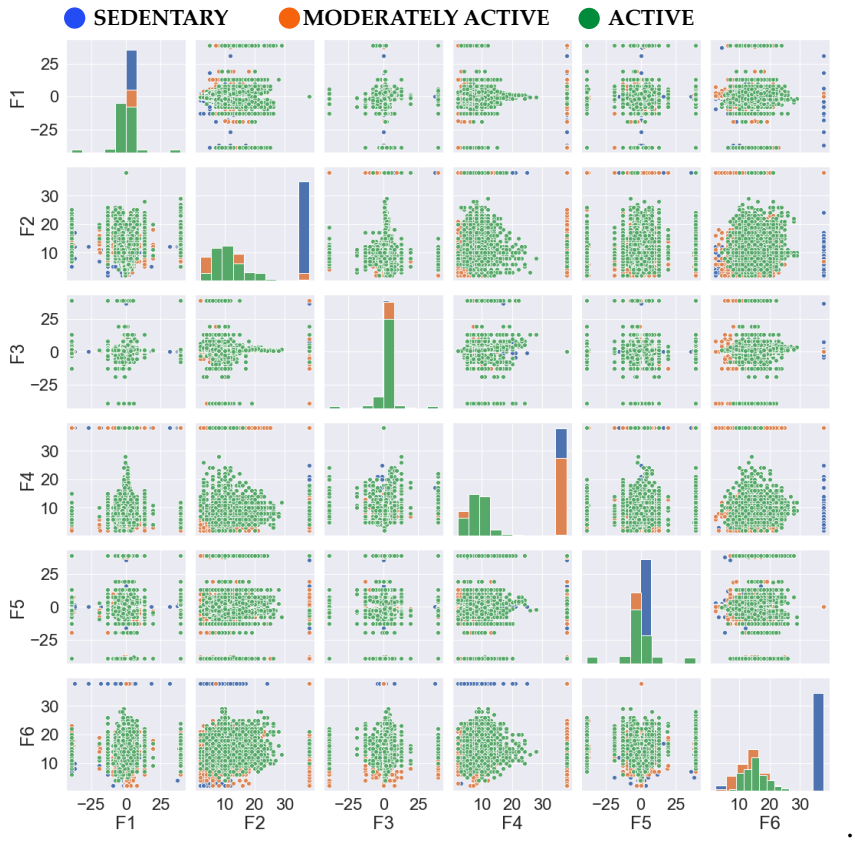


Figure 2. Feature distribution of the three classes.

3.4. Processing Pipeline

Figure 3 depicts the entire processing dataflow for the proposed semi supervised learning mechanism. It starts with acceleration sensing with a sampling rate of 20Hz. The resulting time series is then segmented into 2-seconds long episodes. The data in each windows is subsequently normalized before extracting features as described above. The dimensions of the extracted features are then reduced for managing computational complexity of processing. Such dimensionality reductions are achieved by choosing the top three principal components after applying Principal Component Analysis (PCA) [29]. The proposed iterative semi-supervised learning (ISSL) is then applied in the presence of a small pool of pre-labeled data. Details of the ISSL process including adaptive data clustering and dynamic cluster-labelling are presented in the next section. This processing pipeline enables continuous improvement in activity prediction accuracy over time, thus providing a real-time mechanism to classify human activity with two-second temporal granularity.

 doi:10.20944/preprints202406.0523.v1

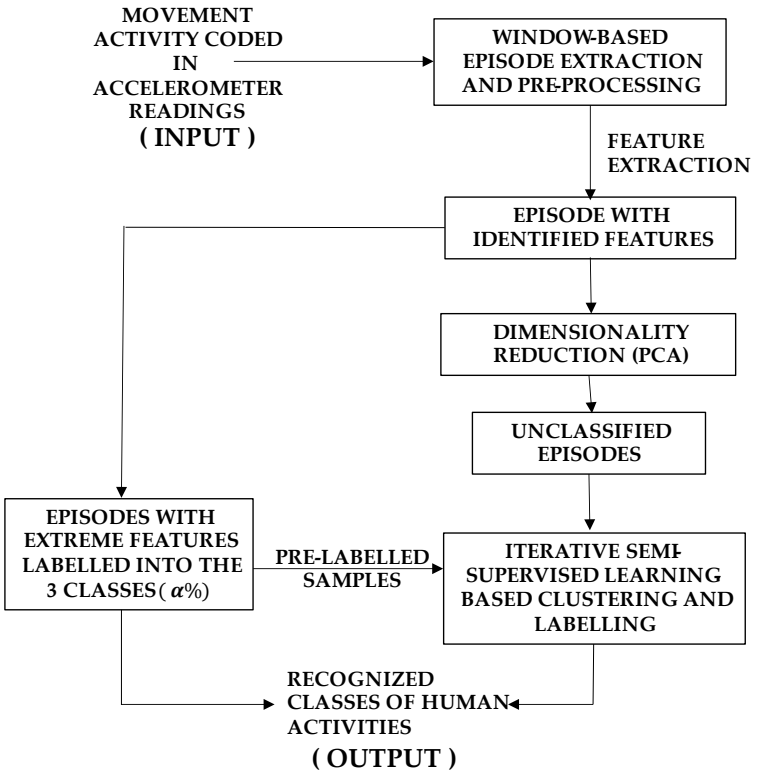


Figure 3. Pre-processing and classification pipeline.

4. Semi-Supervised Learning with Sparsely Labelled Datapoints

Figure 4 depicts the iterative semi-supervised learning framework which forms the basis of the classification model for human activities. The pre-processed accelerometer data arrives in the form of episodes of 2 seconds worth of data. The episodes are represented by the extracted features followed by a dimensionality reduction step. Reducing the dimensionality helps to reduce the computational load of the learning system, as described in section 3. Each unlabeled incoming episode is added to a data pool ($\boldsymbol{S_{pool}}$). A very small percentage ($\boldsymbol{\alpha}$) of incoming episodes which have extreme feature values beyond a threshold are pre-labelled with a designated activity class and added to a pre-labelled data pool ($\boldsymbol{LS_{pool}}$). These are considered pre-labeled because they can be easily identified to be parent of a class based on their extreme feature values matching to the corresponding class. These pre-labelled episodes are used in the SSL training phase for the cluster labelling step. The SSL training and activity prediction is performed on the unlabeled data pool ($\boldsymbol{S_{pool}}$). The training is performed after every $\boldsymbol{N_e}$ incoming episodes. In the training step, first all the existing episodes in the $\boldsymbol{S_{pool}}$ are grouped into $\boldsymbol{N_c}$ number of clusters based on their feature similarity in the clustering step. This clustering is performed by some popular clustering methods like k-means [30] or Gaussian Mixture Model (GMM) [31]. The $\boldsymbol{N_c}$ clusters are labelled using the pre-labelled episodes ($\boldsymbol{LS_{pool}}$) using any of the two cluster labelling methods: *population-based labelling* and *distance-based labelling*.

 doi:10.20944/preprints202406.0523.v1

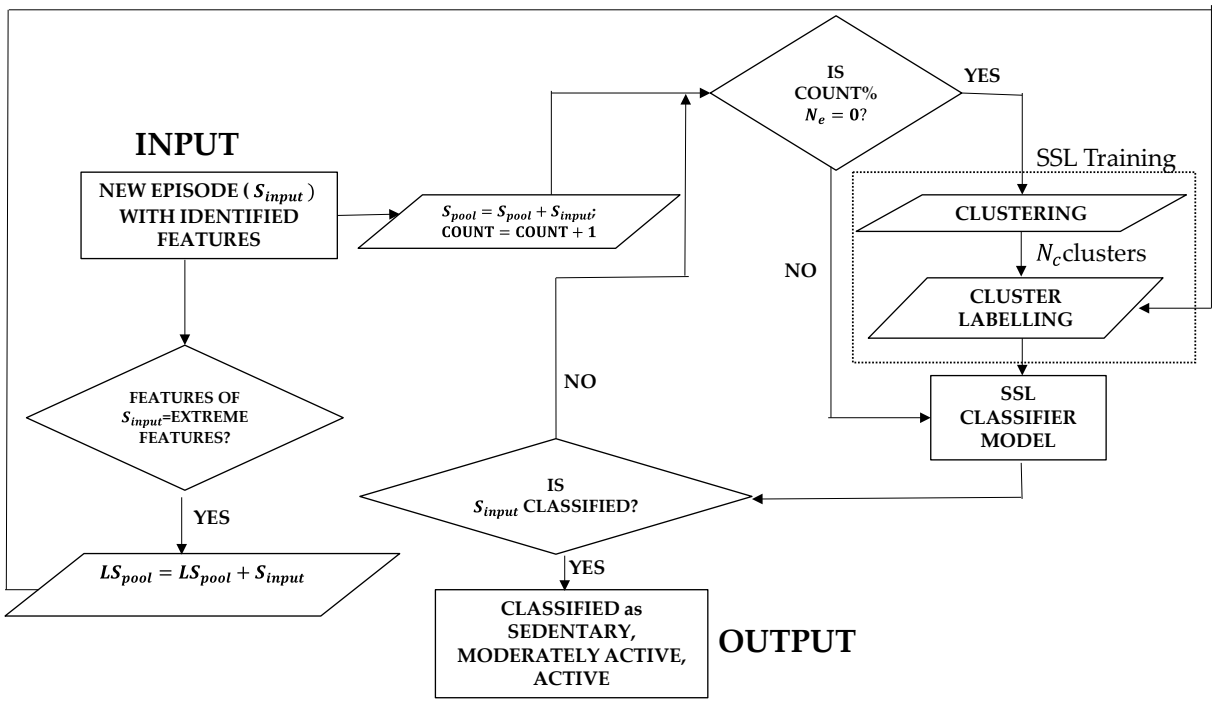


Figure 4. Iterative Semi-supervised learning framework.

In the population-based labelling, the cluster is labelled on the basis of the highest number of pre-labelled episodes present in that cluster. For example, if cluster $\boldsymbol{i}$ has $\boldsymbol{M_1}$ pre-labelled episodes from class $\boldsymbol{A}$ and $\boldsymbol{M_2}$ pre-labelled episodes from class $\boldsymbol{B}$, and if $\boldsymbol{M_1 > M_2}$, then cluster $\boldsymbol{i}$ gets labelled as class $\boldsymbol{A}$. If cluster $\boldsymbol{i}$ does not have a highest number of pre-labelled episodes class of pre-labelled episodes or does not have a pre-labelled episode at all, it remains unlabeled. Thus, all the episodes in that clusters remain unclassified. This cluster labelling mechanism is shown in Algorithm 1.

Algorithm 1: Population-based cluster labelling.

1. FOR $i = 1$ *to* N_c
2. CALCULATE the # of pre-labelled samples in i -th cluster M_1 , M_2,…,M_k, for k classes.
3. IF max (M) exists:
4. LABEL cluster i with label of max(M)
5. ELSE:
6. Cluster i remains unlabeled.
7. END IF-ELSE
8. END FOR

In the distance-based labelling, the cluster is labelled based on label of the closest pre-labelled episode. For example, if cluster $\boldsymbol{i}$ has the minimum distance from the $\boldsymbol{j}$-th pre-labelled episode which belongs to class A, then $\boldsymbol{i}$ is labelled class $\boldsymbol{A}$. No clusters remain unlabeled in this variant of cluster labelling. This cluster labelling mechanism is shown in Algorithm 2.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2024 doi:10.20944/preprints202406.0523.v1

Algorithm 2: Distance-based cluster labelling.

1. FOR $i = 1$ *to* N_c
2. INITIALIZE the distance of cluster rep. i from pre-labeled samples as D;
3. FOR $j = 1$ *to* LS_{pool}
4. Calculate distance of pre-labelled sample j from the cluster-rep i as D_j
   END FOR
5. $D(i) \leftarrow min\,(D_j)$;
6. LABEL cluster 'i' with the label of 'j'-th pre-labelled episode
7. END FOR

After the clusters are labelled using any one of the above two cluster labelling algorithms, the labelled clusters act as the trained classifier model. When a newly arrived and unseen episode comes in between training iterations, the episode is classified based on the most recently trained model, which are the labeled clusters. The new episode gets the class label of the *nearest* cluster in the feature space. A few representative cluster models trained with incoming movement data from a user are shown in Figure 5. It can be seen how the model with 20 clusters expands and the classification accuracy improves with more incoming episodes. The quantitative performance of the classifier is formally presented in the next section.

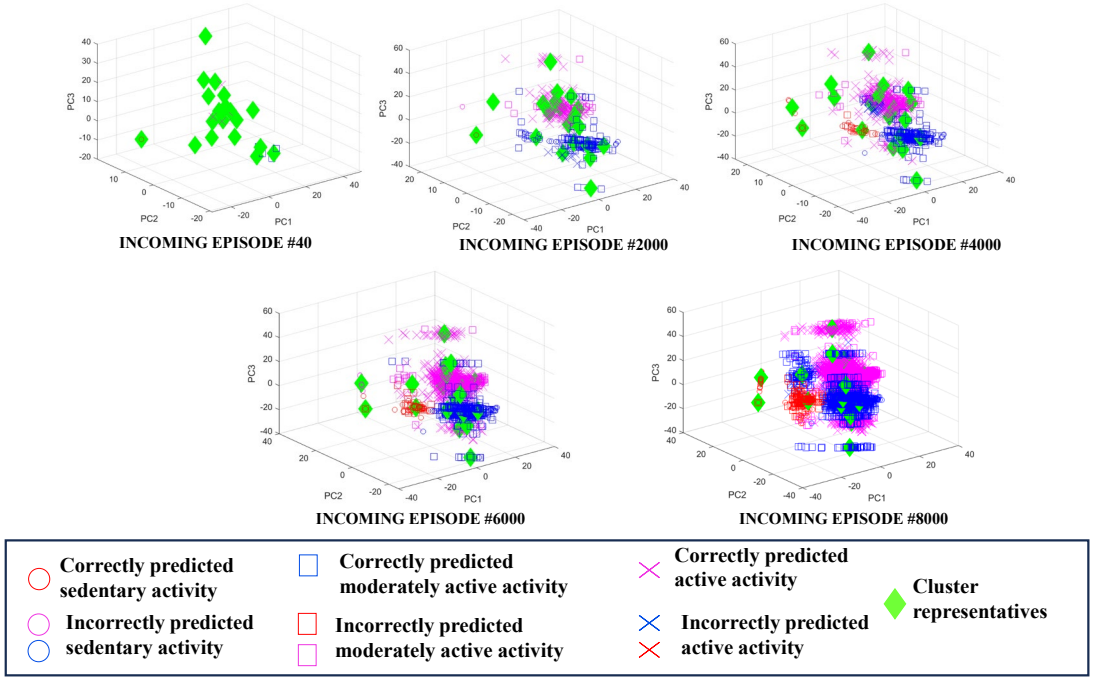


Figure 5. Example cluster model evolution with incoming episodes self-trained using iterative semi-supervised learning paradigm.

Personalization and Privacy: The 2-second episodes represented by the extracted features are classified by the semi-supervised model embedded in a wearable device. The model is iteratively trained as more labelled episodes, which are very infrequent, come in. It also classifies unlabeled episodes in between those infrequent training episodes. Since both the training and classification happens for the same user, the semi-supervised model acts in a very personalized manner. Since in this process no data needs to be transferred outside the embedded wearable device, which performs both model training and classification, privacy of the user's movement data is automatically preserved. This makes the approach personalized and privacy-preserving at the same time.

 doi:10.20944/preprints202406.0523.v1

5. Experiments and Results

An open-source human activity dataset (i.e., Wireless Sensor Data Mining Lab-WISDM lab dataset from University of California, Irvine [27]) has been used for training and validation of the iterative semi-supervised learning model. Three broad classes of activities viz. *sedentary, moderately active and active* have been used from testing the proposed model as discussed in section 3. The time-series data have been pre-processed and windowed into two-minutes' episodes. Six features, as described in section 3, are extracted from each episode. 8262 episodes (2754 from each class of activities) have been used for the model evaluation experiments of the proposed iterative semi-supervised learning framework explained in section 4.

The SSL model is trained using three principal components (i.e., using (PCA) [29]) extracted from the six features discussed earlier. Experiments involving different clustering algorithms, the hyper-parameter n*umber of clusters* ($\boldsymbol{N_c}$), and the system parameter $\boldsymbol{\alpha}$ (i.e., *% of pre-labelled datapoints*) have been performed to evaluate the accuracy and complexity of the proposed SSL model. The accuracy metrics are *true positive (TP)*, *false positive (FP)*, *overall accuracy (Acc)*, and the *classification rate*, which is defined as the percent of classified data-points among the actual number of datapoints. The computational complexity of the model is represented by the *computational time* and the *CPU memory usage* for each learning cycle. The accuracy and complexity are analyzed based on the performance of the SSL algorithm in order to train 8262 episodes (datapoints) for 100 runs. All the results involving the SSL algorithm are mean of the results collected from 100 independent runs.

5.1. Pre-trained supervised learning model as a benchmark

A pre-trained supervised model evaluated with a 10-fold cross-validation using the same dataset has been used as a benchmark for the proposed semi-supervised learning framework. The six extracted features mentioned in section III, have been directly used as the input to the NN model. The NN model is trained using the hyper-parameters mentioned in Table 1.

Table 1. Hyper-parameters for pre-trained Neural Network (supervised learning) model.

# of input features	6
# of hidden layers	1 (128 neurons)
Activation function	tanh (Hidden layers), soft-max (Output layer)
Optimizer	Adam
Loss function	Categorical Cross Entropy

The mean performance (as mentioned in Table 2) of each accuracy parameter for all the 10-fold cross-validation is used as a performance benchmark for the classification accuracy of the proposed model. It has been observed that the NN model has a mean true positives of 98.47 %, 97.41% and 98.19% for the three classes (i.e., sedentary, moderately active, and active) respectively. The mean overall accuracy is 98.03%.

 doi:10.20944/preprints202406.0523.v1

Table 2. 10-fold validation results for a pre-trained Neural Network model.

Folds	Class 1: Sedentary		Class 2: Moderately active		Class 3: Active		Overall Accuracy(%)
	True Positive (%)	False Positive (%)	True Positive (%)	False Positive (%)	True Positive (%)	False Positive (%)	
1	98.6	0.7	97.2	1.5	98.4	0.7	98.1
2	98.2	0.5	97.2	1.4	99.0	0.9	98.2
3	98.6	0.7	96.6	1.3	98.8	1.0	98.0
4	98.9	0.8	97.4	1.5	98.1	0.5	98.1
5	98.2	0.6	98.4	2.3	97.1	0.2	97.9
6	98.4	0.8	97.6	1.8	98.0	0.5	98.0
7	98.5	0.8	97.4	1.6	98.2	0.5	98.0
8	99.0	1.2	96.8	1.9	97.3	0.4	97.7
9	98.3	0.6	97.2	1.5	98.6	0.8	98.1
10	98.1	0.5	98.2	1.7	98.4	0.4	98.2
Mean	**98.5**	**0.7**	**97.4**	**1.7**	**98.2**	**0.6**	**98.0**

5.2. Impacts of feature dimensionality reduction on SSL

The primary motivation of the proposed self-training algorithm is its suitability for the wearable system and its privacy-preserved personalized applications. Ideally, the semi-supervised learning-based self-training model needs to be computationally light due to the resource-constrained nature of the embedded wearable devices. To that end, PCA has been used to reduce the number of features from six to three. Figures 6 and 7 depict the differences in accuracy parameters (TP, FP, Acc, λ) for training the SSL model with and without using PCA, using the population-based and distance-based SSL alorithms, respectively. Results are presented for the two presented algorithm viz. population-based and distance-based. 20 clusters formed by k-means clustering algorithm are used for these experiments.

 doi:10.20944/preprints202406.0523.v1

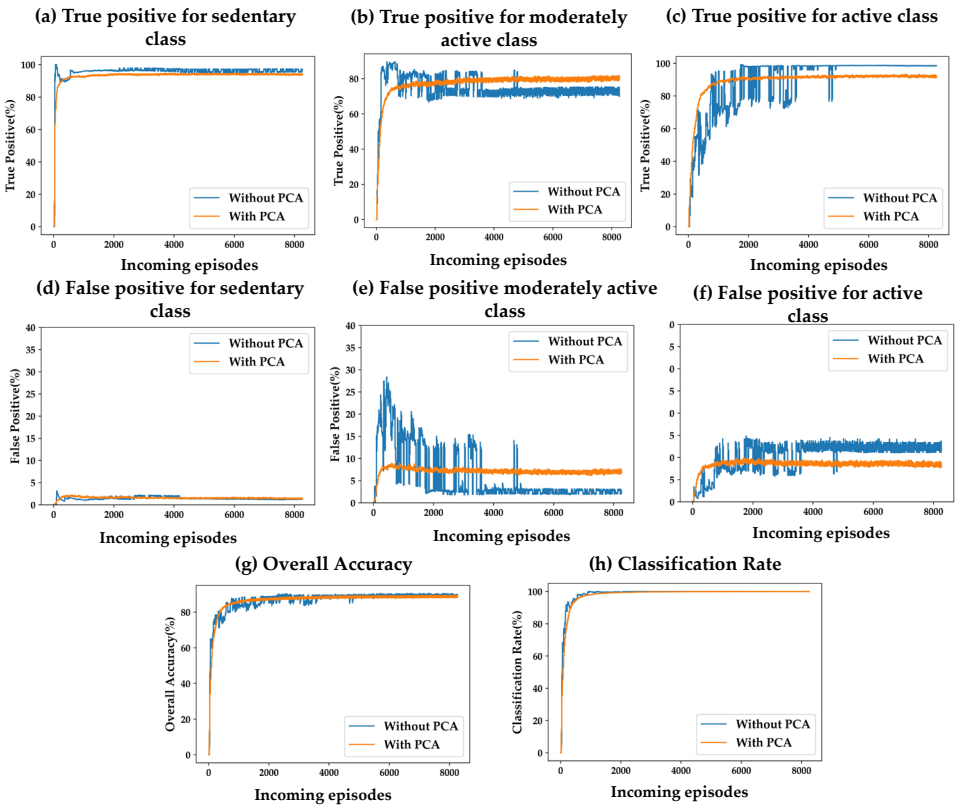
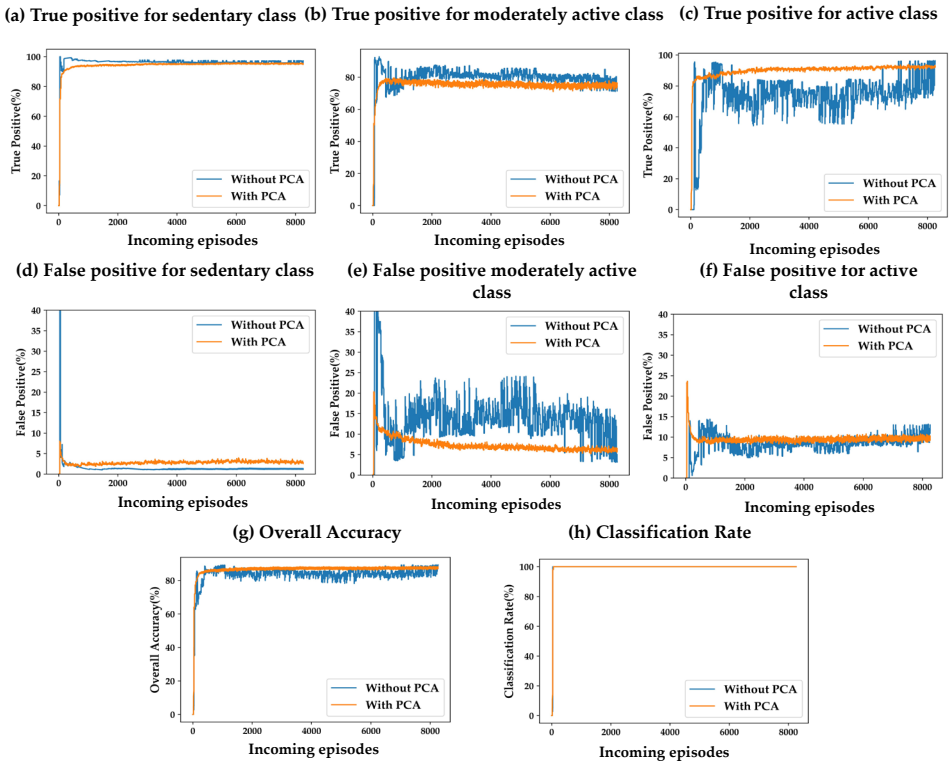


Figure 6. (a)-(h) Impacts of PCA on accuracy parameters for population-based SSL; **(a)-(c):** True positive for the three classes; **(d)-(f)** False positive for the three classes; **(g)** Overall accuracy; **(h)** Classification rate.



 doi:10.20944/preprints202406.0523.v1

Figure 7. (a)-(h) Impacts of PCA on accuracy parameters for distance-based SSL; **(a)-(c)** True positive for the three classes; **(d)-(f)** False positive for the three classes; **(g)** Overall accuracy; **(h):** Classification rate.

For both SSL algorithms, true positives (TP), false positives (FP) for all the three classes are almost same for both the scenarios of with and without using PCA. The overall accuracy (Acc) and the classification rate (λ) are also the same for both algorithms. The only observable difference is in the stability in TP, FP, and Acc performance with more incoming episodes. The performances with PCA have less oscillatory behavior after convergence as compared to the case of without using PCA. This makes it evident that a feature-space with less dimensions makes the cluster models more definite in less time.

Figures 8 and 9 present the impacts of dimensionality reduction from the six original features to three principal components. It can be observed that more dimensions (features) of a data sample results in higher computational complexity. Thus, for both the types of SSL algorithms, both computational time and CPU memory usage go up when PCA is not used. Thus, the experiments to analyze the different hyper-parameters and system parameter of the SSL model are carried out using the three principal compenents obtained by using PCA on the original six features.

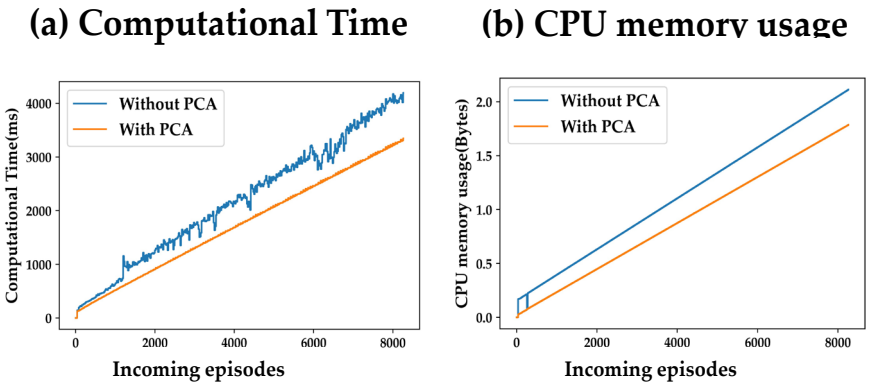


Figure 8. Impact of PCA on complexity for population-based SSL; **(a)** Computational Time; **(b)** CPU memory usage.

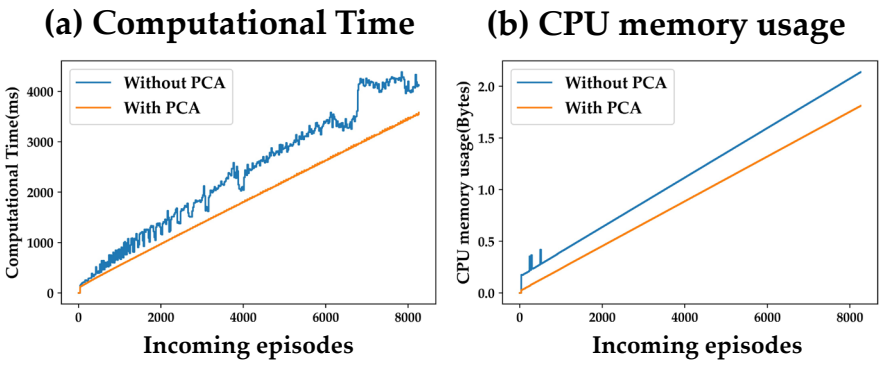


Figure 9. Impact of PCA on complexity for distance-based SSL; **(a)** Computational Time; **(b)** CPU memory usage.

5.3 Impacts of pre-labelled data volume on SSL performance

Figure 10 depicts the TP, FP, overall accuracy, and classification rate for SSL operating with different amount of pre-labelled data, which is represented as α (in percentage). These experiments have been performed with 20 clusters formed by k-means clustering method in the SSL model. It can be observed that all the accuracy parameters converge faster and reaches at better performance

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2024 doi:10.20944/preprints202406.0523.v1

values with increasing values of α. For the population-based SSL, lower α (i.e., 0.05%, 0.1%, 0.5%, and1%) values are unable to achive 100% classification rate. The reasons are as follows. First, many clusters may lack predominance of prelabelled datapoints from a particular class. Second, a cluster may not have any pre-labelled epiosde altogether. All the datapoints in those clusters remain unlabeleld, thus unclassified. Figure 10(f) show that using the distance-based SSL, all the clusters and their datapoints are always labelled since this variant of SSL does not depend on the presence of any pre-labelled datapoints in a cluster. Rather, a cluster is labelled according to the label of the nearest pre-labelled datapoint in the distance-based SSL. The labelling in this mechanism is not impacted by the fact whether the nearest pre-labelled datapoint is present inside or outside the cluster. Also, it can be observed that the convergence is achieved faster in the distance-based SSL than in the population-based SSL.

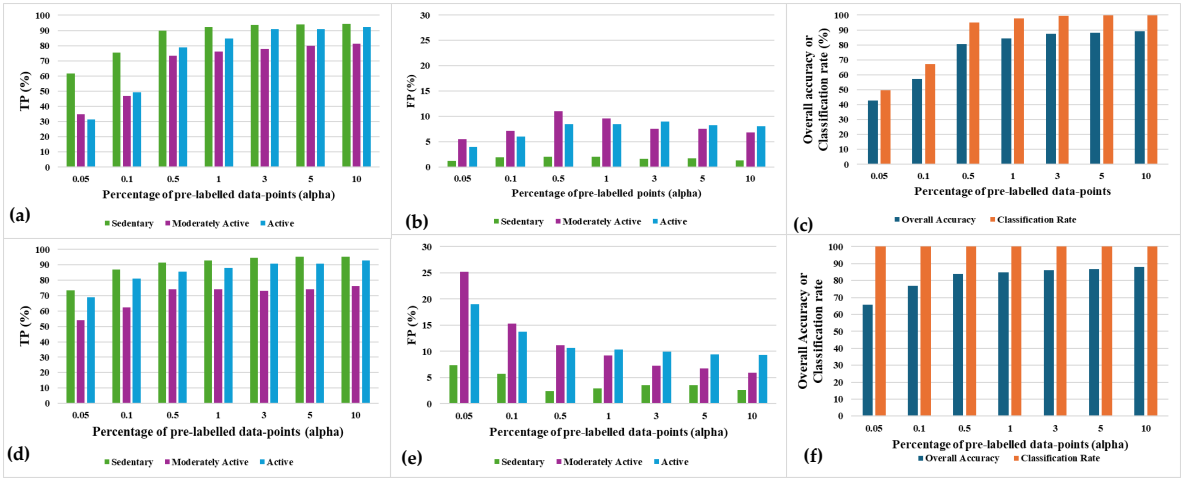


Figure 10. Post convergence accuracy parameters' results for **(a)-(c)** Population-based SSL and **(d)-(e)** Distance-based SSL, with varying α.

Figures 11(a)-(b) depict the total semi-supervised learning computational complexity for both the variants of SSL. Complexity is represented by the total computational time in seconds and CPU memory usage in Bytes. It can be observed that the computational time and CPU memory usage are not dependant on the amount of pre-labelled data-points for the population-based SSL since the number of pre-labelled datapoints vary in different clusters for different runs of the learning algorithm. On the otherhand, it can be observed that more computational time is required for the distance-based SSL to learn 8262 episodes since more the number of pre-labelled datapoints, the more number of distance calculations are required.

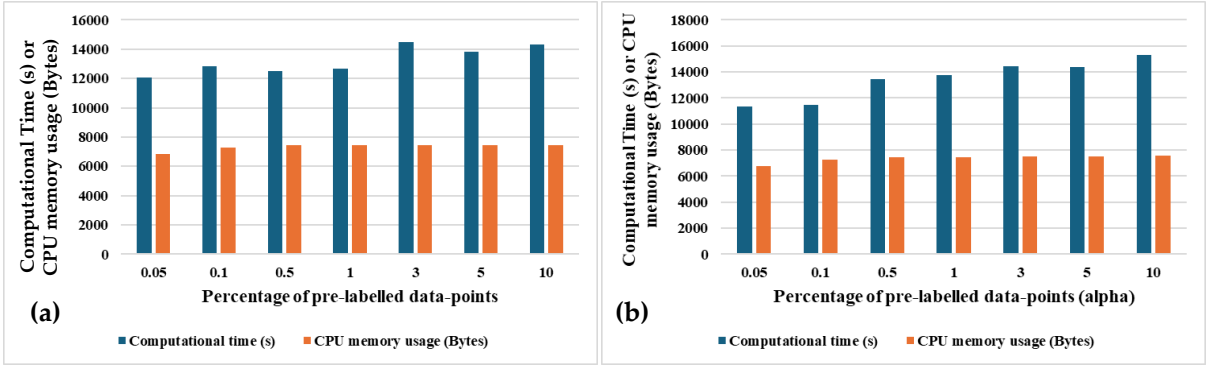


Figure 11. Total learning time and CPU memory usage for all the episodes using **(a)** population-based SSL and **(b)** distance-based SSL, with varying α.

 doi:10.20944/preprints202406.0523.v1

5.4 Impacts of the number of clusters (N_c) on SSL performance

The classification in the proposed model occurs on the basis of the clusters which actually conatins datapoints based on their similarities. The number of clusters in the learning model is an important hyper-parameter which can determine the precison of classification as well as the computational complexity to train the cluster model. Figure 12 depicts the accuracy parameters after learning convergence. These results represent the accuracy parameters for both population-based and distance-based SSL algorithms with varying number of clusters in the model. These above results are obtained from experiments performed by setting the system parameter α to 10% and k-means as the clustering algorithm.

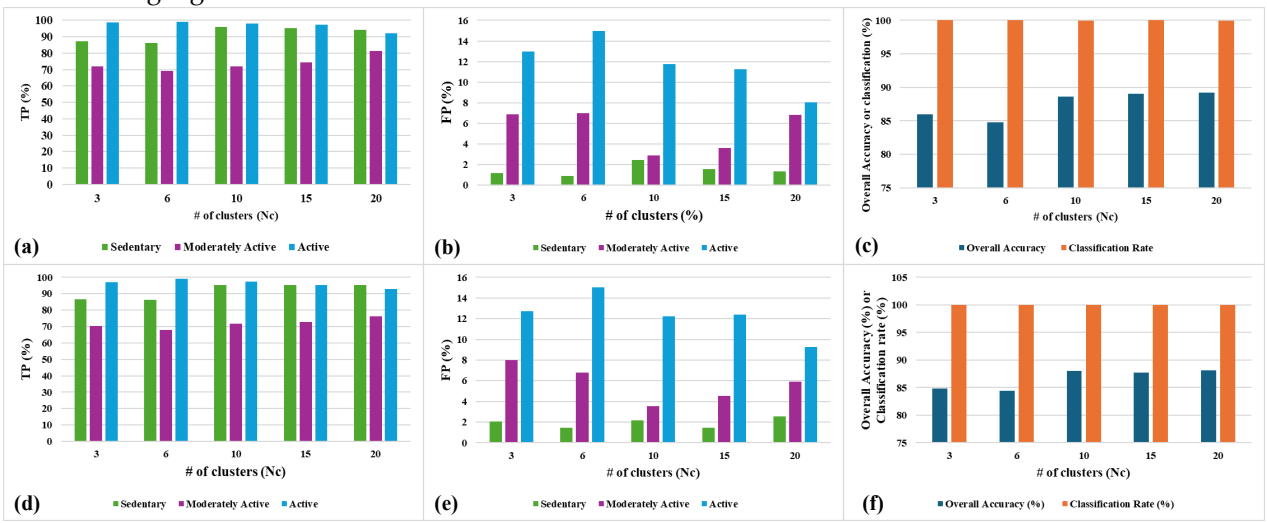


Figure 12. Post convergence accuracy parameters' results for (a)-(c) Population-based SSL and (d)-(e) Distance-based SSL, with varying *Nc*.

It can be observed that true positive (TP) improves and false positive (FP) decreases with higher N_c for all three classes, and for both the variants of SSL. As a result, the overall accuracy improves with higher number of clusters (N_c). The overall accuracy performance does not improve much beyond 10 clusters for both population-based and distance-based SSL. Also, the population-based SSL has marginally better accuracy (90%) as compared to the distance-based SSL (88%). Both the variants of SSL reach 100% classification rate but distance-based SSL reaches faster as all the clusters are labelled from the first cycle of learning as opposed to the population-based SSL.

In the Figures 13 (a)-(b) it can be observed that both the varaints of SSL becomes computaionally expsensive with higher number of clusters. Total CPU memory usage required to train all the episodes of activities is the same for both population-based and distance-based SSL, for each values of N_c. The total computational time required for the iterative learning is marginally higher for the distance-based SSL as compared to the population-based SSL. This is mainly because population-based SSL only counts the pre-labelled episodes in a cluster which is computaionally less expensive than calculating the distance from each pre-laballed datapoint from a cluster center.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2024 doi:10.20944/preprints202406.0523.v1

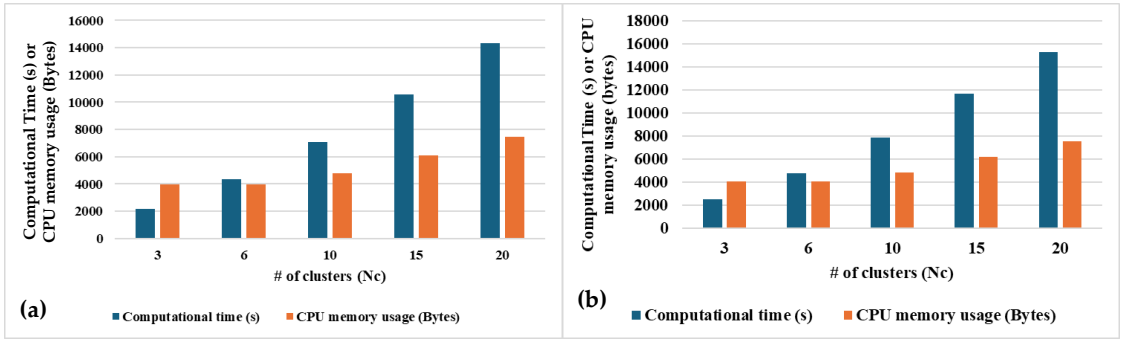


Figure 13. Total learning time and CPU memory usage for all the episodes using **(a)** population-based SSL and **(b)** distance-based SSL, with varying N_c.

5.4. Impacts of different clustering algorithms

Among the popular clustering methods, k-means (KM) and Gaussian Mixture Model (GMM) are the best methods for the current approach. GMM has three useful variants based on the relevant covariance types: spherical (GMM_s), full (GMM_f) and diagonal (GMM_d). Based on the accuracy and complexity performances using the different clustering methods mentioned above, the best clustering method suited for the ideal SSL performance have been analyzed.

Figures 14 (a)-(b) and 14 (d)-(e) depicts the TP and FP for all the three classes of activities along with the overall accuracy and classification rate for different clustering methods. The experiments are performed to provide 20 clusters ($N_c = 20$) using the different clustering methods, with $\alpha = 10\%$. It turns out that GMM_f has the highest TP and the least FP for all the three classes. Subsequently, GMM_f has the best overall accuracy (Acc) for both the population-based and the distance-based SSL variants as shown in figures 14(c) and 14(f). This is because the full covariance type has each component with their own general covaraince matrix which makes the grouping of datapoints belonging to the same class of activities more precise. Population-based SSL has marginally better overall accuracy as compared to distance-based SSL.

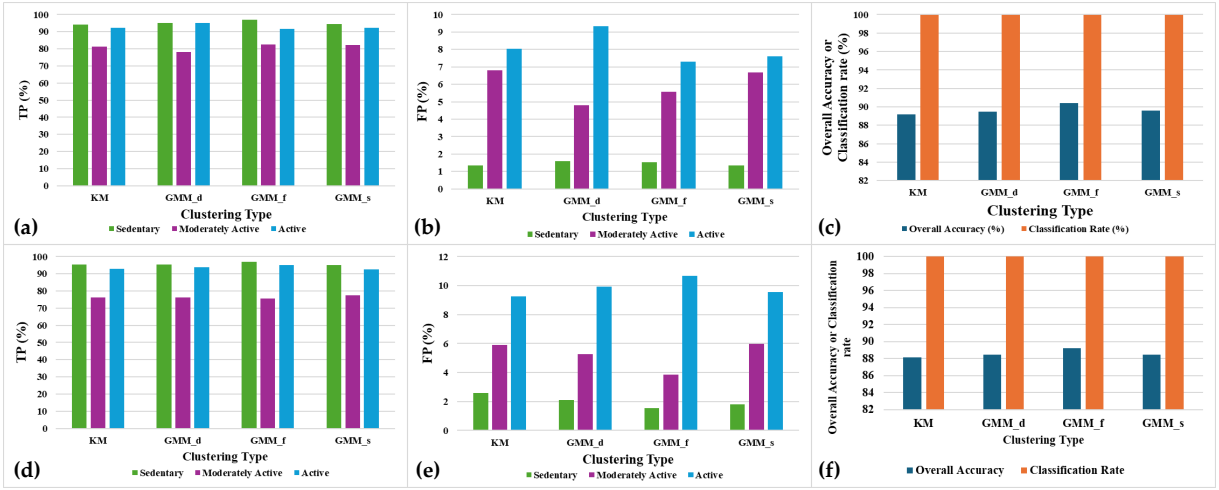


Figure 14. Post convergence accuracy parameters' results for **(a)-(c)** Population-based SSL and **(d)-(e)** Distance-based SSL, for different clustering methods.

Figures 15 (a) and (b) show that GMM_f is computationally expensive with very high total computational time and CPU memory usage for learning the entire dataset (8262 episodes). GMM_s has the second best overall accuracy after GMM_f with less computational time. But the CPU memory usage is still very high. K-means requires much less memory usage as compared to the GMM

 doi:10.20944/preprints202406.0523.v1

clustering types. Thus, when the SSL is implemented on the werable device with limited CPU memory, the K-means should be preferred over GMM.

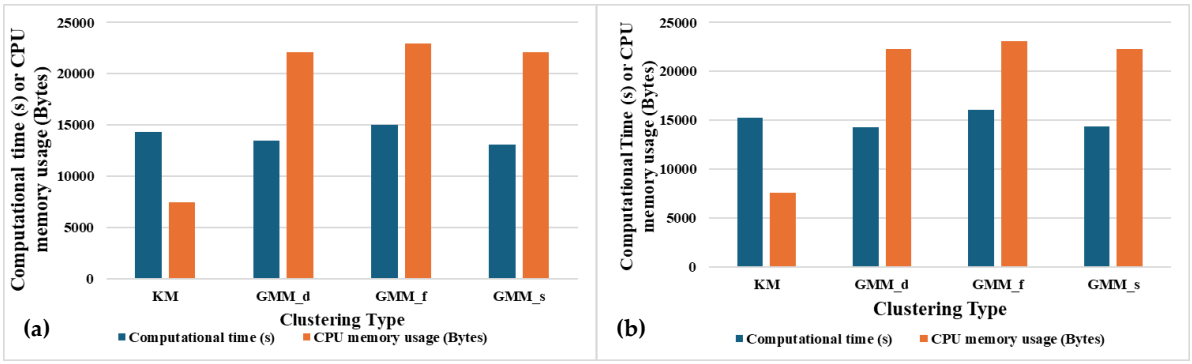


Figure 15. Total learning time and CPU memory usage for 8262 episodes using **(a)** population-based SSL and **(b)** distance-based SSL, for different clustering methods.

6. Discussion and Conclusions

This paper presents a fully on-device approach for human activity detection using a semi-supervised learning paradigm. The proposed SSL mechanisms self-trains in a fully personalized and privacy-preserving manner. The performance of the mechanism is compared with that of a pre-trained supervised learning model which forms a comparison benchmark. Advantages of using an unsupervised dimensionality reduction technique prior to the SSL algorithm are also discussed. The proposed SSL model is analyzed and validated based on different learning hyper-parameters and systems parameters. The results suggest that 5% or higher pre-labelled activity datapoints add precision to the model, therefore improving classification performance. It has been also observed that using 20 clusters in the model with GMM-full type clustering yields an overall accuracy of 90% with 100% classification rate after learning convergence. However, since GMM_full is a computationally expensive model, a K-means clustering method is also explored for resource-limited scenarios. With 4.17 hours of training time for 800 activity episodes the K-means clustering based SSL approach consumes at most 20KB of CPU memory space, while providing a maximum accuracy of 90% with 100% classification rate. These results indicate the feasibility of the proposed SSL paradigm for practical Human Activity Detection towards personalized and privacy-preserving health monitoring system.

Author Contributions: Conceptualization, A.R., H.D., A.B. and S.B.; methodology, A.R.; software, A.R.; validation, H.D., A.B. and S.B.; formal analysis, A.R.; investigation, A.R.; resources, A.R., S.B.; data curation, A.R.; writing—original draft preparation, A.R.; writing—review and editing, H.D., A.B. and S.B.; visualization, A.R.; supervision, S.B.; project administration, S.B.; funding acquisition, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: An open-source human activity dataset (i.e., Wireless Sensor Data Mining Lab-WISDM lab dataset from University of California, Irvine) has been used for the experiments of this research. WISDM dataset can be found at: *https://archive.ics.uci.edu/dataset/507/wisdm+smartphone+and+smartwatch+activity+and+biometrics+dataset.*

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

 doi:10.20944/preprints202406.0523.v1

References

1. G. Y. Lui, D. Loughnane, C. Polley, T. Jayarathna, and P. P. Breen, "The Apple Watch for Monitoring Mental Health–Related Physiological Symptoms: Literature Review," *JMIR Ment. Health*, vol. 9, no. 9, p. e37354, Sep. 2022, doi: 10.2196/37354.
2. J. A. BUNN, J. W. NAVALTA, C. J. FOUNTAINE, and J. D. REECE, "Current State of Commercial Wearable Technology in Physical Activity Monitoring 2015–2017," *Int. J. Exerc. Sci.*, vol. 11, no. 7, pp. 503–515, Jan. 2018.
3. M. F. A. Hady and F. Schwenker, "Semi-supervised Learning," in *Handbook on Neural Information Processing*, M. Bianchini, M. Maggini, and L. C. Jain, Eds., in Intelligent Systems Reference Library. , Berlin, Heidelberg: Springer, 2013, pp. 215–239. doi: 10.1007/978-3-642-36657-4_7.
4. A. Rasekh, C.-A. Chen, and Y. Lu, "Human Activity Recognition using Smartphone." arXiv, Jan. 30, 2014. doi: 10.48550/arXiv.1401.8212.
5. L. Atallah, B. Lo, R. King, and G.-Z. Yang, "Sensor Positioning for Activity Recognition Using Wearable Accelerometers," *IEEE Trans. Biomed. Circuits Syst.*, vol. 5, no. 4, pp. 320–329, Aug. 2011, doi: 10.1109/TBCAS.2011.2160540.
6. A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, p. 33:1-33:33, Jan. 2014, doi: 10.1145/2499621.
7. Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," *Expert Syst. Appl.*, vol. 137, pp. 167–190, Dec. 2019, doi: 10.1016/j.eswa.2019.04.057.
8. N. Twomey *et al.*, "A Comprehensive Study of Activity Recognition Using Accelerometers," *Informatics*, vol. 5, no. 2, Art. no. 2, Jun. 2018, doi: 10.3390/informatics5020027.
9. H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, vol. 105, pp. 233–261, Sep. 2018, doi: 10.1016/j.eswa.2018.03.056.
10. M. B. D. Rosario *et al.*, "A comparison of activity classification in younger and older cohorts using a smartphone," *Physiol. Meas.*, vol. 35, no. 11, p. 2269, Oct. 2014, doi: 10.1088/0967-3334/35/11/2269.
11. M. V. Albert, S. Toledo, M. Shapiro, and K. Koerding, "Using Mobile Phones for Activity Recognition in Parkinson's Patients," *Front. Neurol.*, vol. 3, Nov. 2012, doi: 10.3389/fneur.2012.00158.
12. G. M. Weiss, K. Yoneda, and T. Hayajneh, "Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living," *IEEE Access*, vol. 7, pp. 133190–133202, 2019, doi: 10.1109/ACCESS.2019.2940729.
13. G. M. Weiss and J. W. Lockhart, "The Impact of Personalization on Smartphone-Based Activity Recognition".
14. A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, "On the Personalization of Classification Models for Human Activity Recognition," *IEEE Access*, vol. 8, pp. 32066–32079, 2020, doi: 10.1109/ACCESS.2020.2973425.
15. G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: A machine learning approach," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Feb. 2016, pp. 426–429. doi: 10.1109/BHI.2016.7455925.
16. M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke, and M. Beigl, "ActiServ: Activity Recognition Service for mobile phones," in *International Symposium on Wearable Computers (ISWC) 2010*, Oct. 2010, pp. 1–8. doi: 10.1109/ISWC.2010.5665868.
17. P. Siirtola, H. Koskimäki, and J. Röning, "From user-independent to personal human activity recognition models using smartphone sensors," jultika.oulu.fi. Accessed: May 09, 2024. [Online]. Available: https://oulurepo.oulu.fi/handle/10024/22078
18. D. Roggen, K. Förster, A. Calatroni, and G. Tröster, "The adARC pattern analysis architecture for adaptive human activity recognition systems," *J. Ambient Intell. Humaniz. Comput.*, vol. 4, no. 2, pp. 169–186, Apr. 2013, doi: 10.1007/s12652-011-0064-0.
19. D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: a survey," *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 537–556, Sep. 2013, doi: 10.1007/s10115-013-0665-3.
20. S. Horiguchi, S. Amano, M. Ogawa, and K. Aizawa, "Personalized Classifier for Food Image Recognition," *IEEE Trans. Multimed.*, vol. 20, no. 10, pp. 2836–2848, Oct. 2018, doi: 10.1109/TMM.2018.2814339.
21. Y. Cho, A. Lee, J. Park, B. Ko, and N. Kim, "Enhancement of gesture recognition for contactless interface using a personalized classifier in the operating room," *Comput. Methods Programs Biomed.*, vol. 161, pp. 39–44, Jul. 2018, doi: 10.1016/j.cmpb.2018.04.003.
22. A. Roy, H. Dutta, H. Griffith, and S. Biswas, "An On-Device Learning System for Estimating Liquid Consumption from Consumer-Grade Water Bottles and Its Evaluation," *Sensors*, vol. 22, no. 7, p. 2514, 2022.
23. G. Singh, M. Chowdhary, A. Kumar, and R. Bahl, "A Personalized Classifier for Human Motion Activities With Semi-Supervised Learning," *IEEE Trans. Consum. Electron.*, vol. 66, no. 4, pp. 346–355, Nov. 2020, doi: 10.1109/TCE.2020.3036277.

 doi:10.20944/preprints202406.0523.v1

24. A. Roy, H. Dutta, A. K. Bhuyan, and S. K. Biswas, "Semi-Supervised Learning Using Sparsely Labelled Sip Events for Online Hydration Tracking Systems," in *2023 International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2023, pp. 1799–1804. doi: 10.1109/ICMLA58977.2023.00273.
25. S. Oh, A. Ashiquzzaman, D. Lee, Y. Kim, and J. Kim, "Study on Human Activity Recognition Using Semi-Supervised Active Transfer Learning," *Sensors*, vol. 21, no. 8, Art. no. 8, Jan. 2021, doi: 10.3390/s21082760.
26. M. Lv, L. Chen, T. Chen, and G. Chen, "Bi-View Semi-Supervised Learning Based Semantic Human Activity Recognition Using Accelerometers," *IEEE Trans. Mob. Comput.*, vol. 17, no. 9, pp. 1991–2001, Sep. 2018, doi: 10.1109/TMC.2018.2793913.
27. G. Weiss, "WISDM Smartphone and Smartwatch Activity and Biometrics Dataset." [object Object], 2019. doi: 10.24432/C5HK59.
28. M. Yang, Z. Meng, and I. King, "FeatureNorm: L2 Feature Normalization for Dynamic Graph Embedding," in *2020 IEEE International Conference on Data Mining (ICDM)*, Nov. 2020, pp. 731–740. doi: 10.1109/ICDM50108.2020.00082.
29. M. Greenacre, P. J. F. Groenen, T. Hastie, A. I. D'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nat. Rev. Methods Primer*, vol. 2, no. 1, pp. 1–21, Dec. 2022, doi: 10.1038/s43586-022-00184-w.
30. K. P. Sinaga and M.-S. Yang, "Unsupervised K-Means Clustering Algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.
31. Y. Zhang *et al.*, "Gaussian Mixture Model Clustering with Incomplete Data," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 17, no. 1s, p. 6:1-6:14, Mar. 2021, doi: 10.1145/3408318.
32. D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella, "The 'K' in K-fold Cross Validation," *Comput. Intell.*, 2012.