Correlation Analysis-Based Classification of Human Activity Time Series

Akshay Malhotra, Ioannis D. Schizas, and Vangelis Metsis

***Abstract*— Segmentation of sequential sensor data streams and classification of each segment are common steps in tasks dealing with the detection of events of interest in such data. In this paper, we introduce two correlation analysis-based methods for classifying time series data generated by sensors. Our first method is a lightweight supervised approach utilizing principal component analysis to jointly segment data and classify each segment into a class corresponding to an event of interest. The second method relies on unsupervised canonical correlation analysis to segment time series by clustering together consecutive data points that belong to the same event. Both methods operate without the need for prior feature extraction from the data. The theoretical model of the two methods and the solution to the resulting optimization problem are presented in detail. Classification of human activity from inertial measurement unit sensor data is used as a case study to demonstrate the applicability and effectiveness of the proposed methods.**

***Index Terms*— Activity detection, correlation analysis, segmentation, classification.**

I. Introduction

EVENT detection from sensor data appearing in the form of time series is a problem that is often encountered due to the advent of multimodal sensing systems that collect heterogeneous data streams about a process or phenomenon of interest. Such data points can also be regarded as a signal which has been digitally sampled at a given frequency, thus forming a sequence of discrete-time data. The problem of detecting events of interest in such data can be viewed as the problem of classifying one or more segments from the sequence to one of a set of predefined classes.

A well-studied application is that of human activity recognition from sensors which are attached either to the human subject or the environment [1]. One of the most commonly used sensor types used in such applications is inertial measurement units (IMUs), which often come in the form of simple accelerometers [2]. However, the types of signals that can be represented as time series can vary greatly, depending on the sensor properties and the source that generates them. Another obvious example is physiological biosignals collected in various situations including medical applications, sports, and psychophysiological tests.

Signal segmentation is usually a challenging but necessary step of the event detection process, as the total duration of the signal cannot be treated as a whole. In most existing works, researchers take the straightforward approach of splitting the signal into windows (segments) of fixed size, and classifying each segment in isolation. However, such an approach is not optimal for a number of reasons. First, the size of the segments has to be manually specified by the user; second, occurring events may occupy only a small portion of one segment or may exceed the segment size; third, an event may span across multiple segments. Thus, an adaptive segmentation approach is desirable, which can split the signal into segments of different sizes, based on the properties of the signal.

Even though image segmentation has attracted a lot of attention and a number of methods have been proposed, existing time series segmentations techniques [3]–[7] are limited both in number and in utility, because i) they usually rely on the signal amplitude to perform segmentation; or ii) cannot handle multi-channel heterogeneous biosignals acquired by multimodal sensors; or iii) are sensitive in the presence of noise.

For instanse, the scheme proposed in [7] gives a general procedure for segmenting audio signals by first extracting a sequence of short-term and mid-term feature vectors, then normalizing the extracted features, and finally computing a dissimilarity measure for each pair of successive feature vectors to detect the local maxima. The locations of the maxima are the endpoints of the detected segments. However, the need for prior feature extraction alone is a disadvantage of the method, as the engineering of features to be extracted is a manual process and there are no universally acceptable features for every signal type.

From the classification perspective, activity detection has seen increased interest in the last few years. Much of the recent work have focused on Support Vector Machine (SVM)-based approaches. Reyes-Ortiz *et al.* [8] and Micucci *et al.* [9] have collected datasets from a range of activities and presented results of an SVM-based classifier. More specifically, Reyes-Ortiz *et al.* [8] have combined an SVM-based classifier with a heuristic filtering approach to propose a probabilistic SVM. Micucci *et al.* [9] have presented results with four different classifiers including a K-nearest neighbor classifier, SVM,

Manuscript received June 26, 2018; revised July 24, 2018; accepted July 24, 2018. Date of publication August 7, 2018; date of current version September 12, 2018. This work was supported in part by the National Science Foundation under Grant NSF 1509780 and in part by the Texas State University MIRG Award. The associate editor coordinating the review of this paper and approving it for publication was Prof. Vedran Bilas. *(Corresponding author: Vangelis Metsis.)*

A. Malhotra and I. D. Schizas are with the Department of Electrical Engineering, University of Texas, Arlington, TX 76010 USA (e-mail: akshay.malhotra@mavs.uta.edu; schizas@uta.edu).

V. Metsis is with the Department of Computer Science, Texas State University, San Marcos, TX 78666 USA (e-mail: vmetsis@txstate.edu).

Digital Object Identifier 10.1109/JSEN.2018.2864207

a neural network and a random forrest classifier. A similar analysis over multiple clustering schemes for the data in [8] can be found in [10]. The schemes presented in these papers rely on some form of a priori segmentation and/or feature extraction stage.

Gupta *et al.* [11] and Gusain *et al.* [12] have proposed strategies considering an ensemble of SVMs for classification. The misclassified data of each SVM's output is used to train a new SVM. Further, an Adaboost type weighting strategy is used for giving weights to each of the SVMs. Such techniques provide slight improvement in performance over traditional SVMs at the cost of increased training resources, but are more suited for online learning applications where the data arrives in batches and SVMs are trained over only a small section of the data. Zdravevski *et al.* [13] have tried to leverage the information from gyroscopes and body accelerometers to supplement the total accelerometer data and presented their results over multiple supervised clustering algorithms. Tan and Huang [14] have proposed a half cosine fuzzy clustering scheme, this scheme too is supplemented by a prior feature extraction and a PCA-based feature dimensionality reduction stage. A survey of different techniques applied towards the classification task in activity detection can be found in [15].

In this work, we introduce two novel correlation analysis-based methods that work directly on the captured signal and do not require any a priori segmentation or feature extraction step prior to performing classification. Since the proposed schemes can directly classify the time series data, segmentation is automatically achieved as part of the process. The first method is a supervised approach which relies on PCA decomposition to segment time series data into segments of varying length, and classify each of those segments into one of a predefined set of classes. As in any supervised learning approach, this method requires prior training on labeled data. The second method, which is based on canonical correlation analysis (CCA), is an extension of our previously published framework of linear CCA [16]. Here we extend the linear scheme to its kernelized variant which is capable of handling nonlinear dependencies in the data. The correlation framework is further penalized with a norm-one regularization to induce sparsity required for clustering. This method operates in an unsupervised manner in the sense that it does not require prior training. The only input parameter from the user is the number of existing classes. This is on par with certain clustering algorithms, which require the user to specify the number of clusters. Each segment generated by the CCA-based method can be subsequently classified by mapping its corresponding cluster to one of a known set of classes or by using one of the many existing classification algorithms.

Depending on the nature of the application, either the supervised or the unsupervised approach may be preferable. The supervised, PCA-based approach is lightweight and can be suitable for real-time signal segmentation and classification on low-power or embedded devices. The unsupervised, CCA based approach can be utilized when training data are not readily available. It should be pointed out that both methods described here do not require the extraction of features from the signals being analyzed to achieve the segmentation and classification tasks.

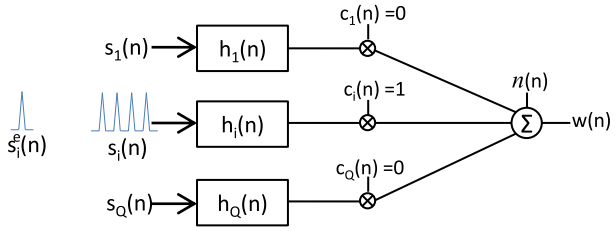

Fig. 1. A block diagram representation of the signal model considered in (1). Here it has been considered that the i-th activity is being performed by the user, thus $c_i(n) = 1$ whereas, all others are zero.

We tested the utility and performance of our methods on two publicly available human activity recognition datasets, and one similar dataset generated by our team, and our findings are presented here. The classification performance of our supervised approach is competitive to the existing state-of-the-art approaches which use more elaborate and heavyweight feature extraction and classification algorithms. We also evaluated the classification performance of our unsupervised approach on the same datasets, even though its direct output is not class labels but rather segmentation and assignment of each segment to a cluster. The classification, as explained in section III-C2, was performed by mapping each cluster to one of the known classes.

The rest of this paper is organized as follows. Section II formulates the activity detection process as a segmentation and classification problem. In Section III, the proposed correlation analysis based algorithms are discussed. In section IV, the algorithm effectiveness is demonstrated by applying it in three different datasets. Finally, Section V provides the concluding remarks.

II. Problem Formulation

A. Problem Statement

Consider a discrete time series $w(n)$, which is assumed to be the output of a sensor (e.g. an accelerometer signal). The objective is to find the time instants corresponding to an activity that results in a change of the characteristics of the signal $w(n)$, modeled here as

$$w(n) = \sum_{i=1}^{Q} c_i(n) \sum_{n_c=0}^{L-1} h_i(n_c) s_i(n - n_c) + \nu(n) \qquad (1)$$

where, $s_i(n)$ represents the signal corresponding to the i-th activity. The signal $s_i(n)$ can be further visualized as a series of epochs generated by the user while performing an activity, and it can be represented as

$$s_i(n) = \sum_{\tau=1}^{T} s_i^e(n - \tau P),$$

where, $s_i^e(n)$ can be considered to be an epoch. The total number of distinct activities (e.g. walking, sleeping, climbing up the stairs, etc.) is represented as Q. $c_i(n)$ is a binary

variable, it is one for the activity that is happening at the time instant n and zero for all other $Q-1$ activities. $\nu(n)$ represents the noise in the signal. The source to sensor relationship is considered as an impulse response, $h_i(n_c)$, that models the channel imperfections when recording an activity signal. We choose to represent the source to sensor relationship using a convolution based model to cover a more generic set of source to sensor relationships that may be applicable in applications beyond the activity detection based framework presented here. The impulse response h_i can be a scalar (a single tap filter) or it may even be the same for all of the different signals. It must be noted that the type and length of impulse response has no bearing on the algorithms proposed here and is not required as part of these algorithms. Also, it must be noted that since $w(n)$ is the output of the sensor recorded by an analog-to-digital converter (ADC), we consider a discrete representation of time in 1 and throughout the rest of the paper.

To identify the segments of $w(n)$ corresponding to different activities, the signal is split into frames of P samples each. Note that, the signal $w(n)$ can have segments in which none of the activities are happening (i.e $c_i(n) = 0 \ \forall i \in 1, ..., Q$). Such segments are irrelevant to the activity segmentation and classification objective. Therefore the signal is passed through a pre-processing stage to filter out the frames unrepresentative of any activity before being used by the two proposed algorithms.

Consider the signal in Fig. 2(a), corresponding to the accelerometer output of a smartphone while the user is performing different activities. Fig. 2(c) and 2(d) show the zoomed in versions of the signal where the user is walking and climbing-up the stairs, respectively. As it can be seen, the data has repetitive patterns or epochs during each of these activities which are distinctively different from the epochs corresponding to the other activity. Lets say there are N_e frames of relevance. The signal can thus be considered as a set of N_e vectors each having P entries. Thus, we can utilize the statistical correlation between these vectors corresponding to the same activity to identify the vectors pertaining to the same activity. For the purpose of identifying the correlation between signals we propose the use of the following two techniques: 1) a Principal Component Analysis (PCA) based framework and 2) a Canonical Correlation Analysis (CCA) based framework [17]. The theoretical basis of the two frameworks is explained in detail in the following section.

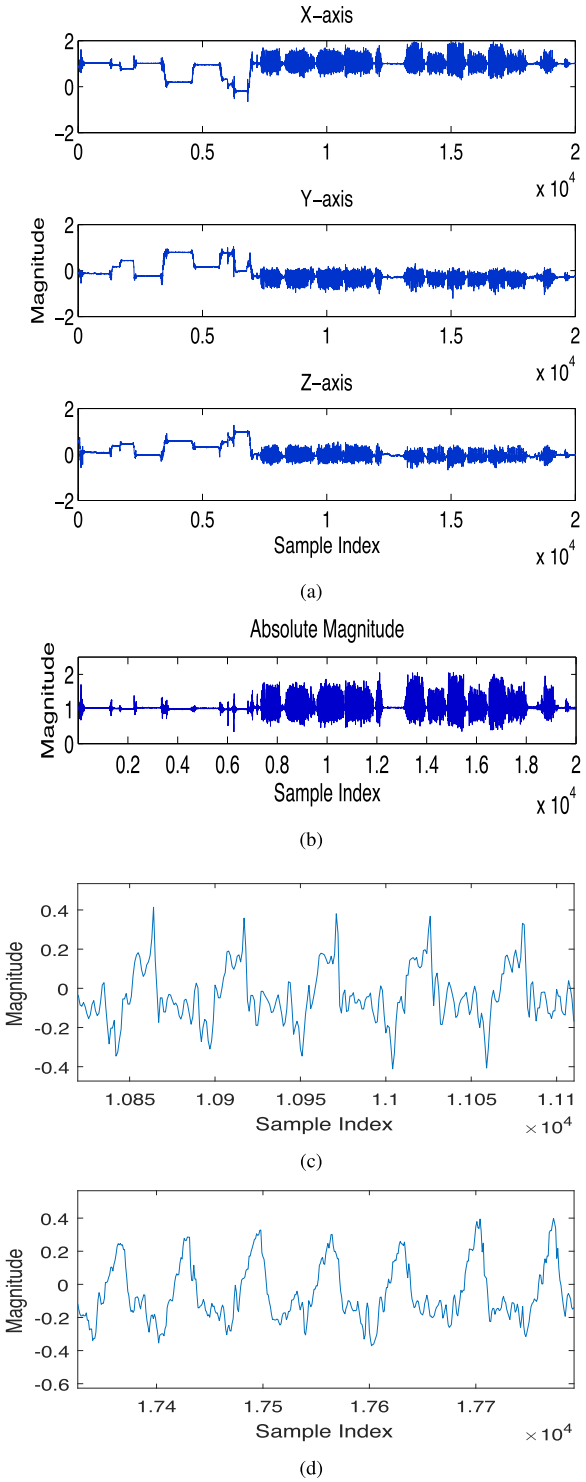

Fig. 2. Example of a sample acceleration signal over multiple activities. Fig. (a) shows the signal from the three accelerometer channels (X, Y, Z) separately, Fig. (b) shows the absolute value (magnitude) of the signal vector across the XYZ components, Fig. (c) zooms in on the portion of the Z-axis signal with walking as the activity, and Fig (d) zooms in on the portion of the Z-axis signal with climbing up-stairs as the activity. The repetitive epochs for the two activities can be clearly seen in the signal. The epochs from the two activities have different structure, this structural differences are utilized by CCA for classification.

III. Methods

A. Pre-Processing

The objective here is to find the frames representing the repetitive structures or the epochs and use them as vectors for the CCA and PCA-based classification. It is important to understand that the epochs or the frames need to be synced along a common reference point to effectively apply the correlation analysis-based algorithms. Since all the epochs have a distinguishable peak, Fig. 2(c) and 2(d), we center the data around this peak and select a set of samples around it. There are multiple schemes present in the literature (eg. [18], [19]) to isolate epochs. In this work, since our primary objective and the contribution is not the epoch-based classification, we use a rather simplistic approach to isolate the epochs to

form the frames with a nominal accuracy and a rather low computational complexity.

The first step is to remove the parts of the signal which are devoid of any activity. As seen in Fig. 2(a), the first 7000 samples consists of approximately constant valued signals, corresponding to user standing or sitting states. During this period the variance of the signal, computed over a window, is small in magnitude.

$$\bar{w}(n) = \begin{cases} w(n), & if \;\; \varrho^2(w(n)) \geq thresh_1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where, the function $\varrho^2(.)$ represents the variance of the signal calculated over a time window $[n - L/2, ..., n, ...n + L/2 - 1]$ with L being the window length and $thresh_1$ indicating a predetermined threshold.

Next, a simple peak detection algorithm is employed that finds the peaks and isolates P samples around it to form the frame. The signal magnitude is first compared against a predetermined threshold, $thresh_2$. The parts of the signal that are greater than the threshold are checked for local peaks (i.e neighboring samples have a magnitude lesser than the current sample). Also, since epochs extend over at least a few samples, two consecutive epochs (or two consecutive peaks) have to be separated by at least a certain number of samples, ϑ_1. Thus, all the peaks which are under ϑ_1 samples away are discarded. The value of L used for evaluating the variance cannot be too small since it should be able to observe the signal statistics over a period comparable to the epoch and again it cannot be too large so that a large number of consecutive samples, devoid of any activity, get coupled with an epoch and are considered for the analysis. Although many combinations of L and $thresh_1$ may give reasonable results, we suggest the use of values in the range $P/2 < L < 2P$. The frames starting at, or around, the peak of the epoch are then used as vectors for the CCA or PCA-based classification. Thus the n-th frame is given by, $\boldsymbol{\omega}_n = \{\bar{w}(n_t - \hat{P} + 1), ..., \bar{w}(n_t), ..., \bar{w}(n_t + (P - \hat{P}))\}^T$, where n_t is the time instant corresponding to the peak of the t-th epoch and $\hat{P}$ is an integer such that $\hat{P} \in [0, P - 1]$. The above mentioned approach performs reasonably well. Since the focus of the paper is not the epoch detection and frame construction algorithm, we move our attention to the two classification approaches proposed here. The pre-processing stage for vectorization can be easily replaced with any other approach from the literature (like [18] and [19]) and can be used as an input to the algorithms presented later on.

B. PCA-Based Activity Classification

Principal component analysis is used as an approach for dimensionality reduction to identify the set of principal basis vectors on which the data can be projected and thus, represented in a low dimensional space. In the current case we have data vectors of P dimensions each, we use PCA to find ψ principal components and represent this data in a ψ dimension space. Projecting the data onto a lower dimension space causes some loss of information. This loss or error can be measured by projecting the data back into the P dimension space to get a reconstructed estimate of the data and evaluate the mean square error (MSE). The PCA is utilized here for activity detection and classification by identifying the principal components for each activity and then, for any new data vector, finding a reconstructed estimate using the principal eigenvectors for each of the considered activities. The MSE between the reconstructed data vector and the original data vector (or the reconstruction error) will be the lowest for the activity the data vector represents.

The proposed technique is divided into a training phase and an online phase. For training, it is assumed that the labeled sensor data are available and this data are used to learn their structure and the underlying data correlation. The trained model is then used during the online stage to identify which frames of the given signal correspond to different activities.

1) Training: After the signal has been pre-processed and the individual epochs have been generated, a set of epochs corresponding to each of the activities to be identified are selected to form the training set. For a given size of a dataset, the percentage of the total epochs used for training has a direct correspondence to the accuracy of the activity classification, as larger training sets yield better accuracy. Another factor that affects the performance is the length of the epoch. As explained earlier we select samples around the peak of the epoch to form the data vector, if the length is too small, the accuracy is low. Similarly, if the length is too large, the samples from the neighboring epochs get included in the data vector. Epochs resulting from isolated events and epochs at the beginning and end of an activity segment have data points belonging to other activities, and thus are not similar to the other data vectors of the same activity, and are therefore misclassified.

Let set Φ_q contain all $\boldsymbol{\omega}_n$ corresponding to the activity q and let $\Phi_q^t \subset \Phi_q$ be the set containing the training vectors and $\Phi_q^v \subset \Phi_q$ be the set containing the validation vectors, where $|\Phi_q^t| + |\Phi_q^v| = |\Phi_q|$ and $\Phi_q^t \cup \Phi_q^v = \Phi_q$. Thus, we form the covariance matrix as

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{q}} = \frac{1}{N_e^q} \sum_{n=1}^{N_e^q} [\boldsymbol{\omega}_n - \mathbf{m_q}][\boldsymbol{\omega}_n - \mathbf{m_q}]^T, \boldsymbol{\omega}_n \in \Phi_q^t \tag{3}$$

where $\mathbf{m_q} := \frac{1}{N_e^q} \sum_{n \in \Phi_q^t} \boldsymbol{\omega}_n$ denotes the sample-average estimate for the mean of ω_n and $N_e^q = |\Phi_q^t|$.

This is followed by principal component analysis (PCA) of the averaged $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{q}}$ to identify the eigenvectors of the training data. The eigenvectors corresponding to the ψ highest eigen values, i.e. $\{\boldsymbol{v}_1^q, ..\boldsymbol{v}_i^q, ..\boldsymbol{v}_\psi^q\}$ are selected for the reconstruction in the validation phase. The eigenvectors $\boldsymbol{v}_i^q$ are P dimensional. This procedure is repeated for all Q classes.

2) Validation: In the validation phase the reconstruction error is used as a metric to allocate a class to the data vectors $\boldsymbol{\omega}_n \in \bigcup_{q=1}^{Q} \Phi_q^v$. The data vectors $\boldsymbol{\omega}_n$ are projected onto the eigenvectors $\{\boldsymbol{v}_1^q, ..\boldsymbol{v}_i^q, ..\boldsymbol{v}_\psi^q\}$ for each class.

$$\zeta_i^q = (\boldsymbol{\omega}_n - \mathbf{m}_q)^T \boldsymbol{v}_i^q \quad \forall \; i \in \{1, ..., \psi\} \;\&\; q \in \{1, ..., Q\} \tag{4}$$

where, ζ_i^q is the projection of the data vector onto the i-th principal eigenvector representing class q.

The data is then reconstructed using the eigenvectors as:

$$\hat{\boldsymbol{\omega}}_n^q = \sum_{i=1}^{\psi} \boldsymbol{\nu}_i^q \zeta_i^q \tag{5}$$

Next, the reconstruction error for each class is evaluated as:

$$\epsilon_q = ||(\boldsymbol{\omega}_n - \mathbf{m}_q) - \hat{\boldsymbol{\omega}}_n^q||_2^2 \tag{6}$$

where, ϵ_q represents the error using the principal eigenvectors corresponding to each class or activity. Each data vector is classified to the class for which the reconstruction error is the lowest.

Algorithm 1 PCA-Based Activity Classification

1: Epoch detection and vectorization of $w(n)$ to form $\boldsymbol{\omega}_n$.
2: Form $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{q}}$ using the framed data $\boldsymbol{\omega}_n$ from Φ_q^t using (3).
3: Find the ψ principal eigenvectors of $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{q}}$.
4: Calculate the reconstruction error and assign the class with the lowest ϵ_q.

3) Multiset PCA: In the algorithm discussed above, one set of principal components ν_i^q were used for each of the Q classes. To achieve better performance, multiple set of principal components can be used to represent each of the Q classes. Having multiple set of principal component for each of the classes provides the flexibility of accommodating for different activity patterns across users.

Consider the case where for a given activity q, the training set Φ_q^t is divided into B subsets, such that we have $\Phi_q^t = \bigcup_{\beta=1}^{B} \Phi_q^{\beta t}$. Thus during the training stage, the covariance matrix $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{q}}^{\boldsymbol{\beta}}$ should be calculated for each of the B training subsets separately similar to the operation in (3). Thus post PCA, for the q-th class we will have B sets of eigenvectors, where the β-th set of principal eigenvectors is given by $\{\nu_1^{q^\beta}, ..\nu_i^{q^\beta}, ..\nu_\psi^{q^\beta}\}$.

For the validation stage, the reconstruction error has to be evaluated $Q \times B$ times instead of Q, as was the case for the single set PCA. As before, the data vector is classified to the class for which the reconstruction error is the lowest.

Since each of the epochs is classified based on the activity, the activity/class label for neighboring epochs can be used to identify the segments of the signal where a particular activity was being performed by the user.

C. CCA-Based Clustering and Classification

Unlike the PCA-based approach, where we have a training phase and require labeled data, the CCA is an unsupervised algorithm. It needs no training data or any priors about the distribution. The only information required is the number of classes, Q, in which the data should be segregated. The objective for the CCA-based scheme is to cluster consecutive epochs (belonging to the same activity type) together, thus forming segments. Given the set of N_e epoch vectors that contain the information about q activities, the objective is to classify the N_e vectors into Q classes.

Given the $N_e \times P$ data matrix, containing the data from all the epochs, consider splitting the rows in two non-overlapping groups of data vectors $\mathbf{x} \in \mathbb{R}^{N_e^x \times P}$ and $\mathbf{y} \in \mathbb{R}^{N_e^y \times P}$ where $N_e^x + N_e^y = N_e$. This formulation makes the following underlying assumptions: 1) the number of classes Q, is lesser than the number of epoch vectors, N_e; and 2) each group of data vectors, $\mathbf{x}$ and $\mathbf{y}$, have at least one representative vector from each class. These assumptions are easily fulfilled for any real-world application, as N_e is much larger than Q. Also, for forming the non-overlapping sets $\mathbf{x}$ and $\mathbf{y}$, alternate vector/epochs appearing in the series are grouped together. Thus the cardinality of the two sets $\mathbf{x}$ and $\mathbf{y}$, (i.e. N_e^x and N_e^y) is approximately the same ($\approx N_e/2$). Such a division process ensures that the cardinality of each set is much larger than Q. Further, in any real world situation, it is plausible to assume that the user will not discontinue an activity after every epoch, but will rather continue in the same activity for several epochs, alternatingly allocating vectors/epochs into different groups ensures that epochs from each activity are present in both the groups $\mathbf{x}$ and $\mathbf{y}$.

To cluster the N_e epoch vectors according to the activity they sense, we make use of the statistical correlations that vectors in $\mathbf{x}$ and $\mathbf{y}$ groups exhibit when representing the same activity. Using CCA we can identify the entries of the two vectors, $\mathbf{x}$, and $\mathbf{y}$, that are maximally correlated and by imposing a ℓ_1 norm regularization, as shown in [16], we can utilize the CCA framework as a clustering algorithm. The modified CCA framework with the ℓ_1 norm regularization is given as:

$$(\hat{\mathbf{E}}, \hat{\mathbf{D}}) = \arg\min_{E,D} P^{-1} \sum\nolimits_{\tau \in \mathcal{P}} ||\mathbf{y}_\tau - \mathbf{E}\mathbf{D}\mathbf{x}_\tau||_2^2 + \sum\nolimits_{\rho=1}^{Q} \lambda_\rho^{\mathbf{E}} ||\mathbf{E}_{:\rho}||_1 + \sum\nolimits_{\rho=1}^{Q} \lambda_\rho^{\mathbf{D}} ||\mathbf{D}_{\rho:}||_1, \tag{7}$$

where the subscript τ indicates the time index of the vector and $\mathbf{x}_\tau$ and $\mathbf{y}_\tau$ are the $\mathbb{R}^{N_e^x \times 1}$ vectors containing the epoch vectors with time index $\tau = 1, ..., P$. The set $\mathcal{P} = \{1, ..., P\}$ contains all the time instances between 1 and P. $\hat{\mathbf{D}} \in \mathbb{R}^{Q \times N_e^x}$ and $\hat{\mathbf{E}} \in \mathbb{R}^{N_e^y \times Q}$ are the sparse matrices indicating the clustering. The sparsity is introduced as a result of the ℓ_1 norm regularization part in the modified CCA framework. The support (non-zero entries) of each row of $\mathbf{D}$ in Eq. (7) will indicate which entries in $\mathbf{x}_\tau$ contain information about the same activity. Note that the rows of $\mathbf{D}$ are expected to be sparse since not all entries of $\mathbf{x}_\tau$ correspond to the same activity. Thus, it is pertinent to impose sparsity across each row of $\mathbf{D}$ that represents a different activity. This enables activity clustering among the entries of $\mathbf{x}_\tau$. Similarly, the columns of $\mathbf{E}$ can be forced to be sparse and their support will point to these entries of $\mathbf{y}_\tau$ that represent the same physical activity. Ideally, the matrix $\hat{\mathbf{D}}$ should have a single non-zero entry in each of the N_e^x columns and similarly for the matrix $\hat{\mathbf{E}}$ there should be only one non-zero entry in each of the N_e^y rows. However, in practice the row-entry with the strongest amplitude is treated as the non-zero entry pointing to the activity, whereas the entries of negligible amplitude are treated as zeros. The position of the strongest in amplitude non-zero entry indicates which of the Q activities does the epoch vector relate to. It should be noted

that $\mathbf{x}_\tau$ and $\mathbf{y}_\tau$ are not used to represent the epoch vectors but are a means to divide the entire set of epoch vectors into two sets of vectors, which is required for applying CCA. The operator $\|\cdot\|_1$ refers to ℓ_1 norm, while the parameters $\lambda_\rho^{\mathbf{D}}$ and $\lambda_\rho^{\mathbf{E}}$ correspond to sparsity controlling coefficients in $\mathbf{D}$ and $\mathbf{E}$, respectively. Interestingly, sparsity across rows for $\mathbf{D}$ (columns for $\mathbf{E}$) can be viewed as sparsity across columns for $\mathbf{D}$ (rows for $\mathbf{E}$) after rearranging terms.

To improve the performance and to take into account any non-linearities in the epochs over a period of time, we will utilize a nonlinear kernel mappings in the formulation in (7). Such a formulation offers more robustness to minor changes in the data while keeping the computation cost under control by employing the kernel trick [20].

Thus, the non linear mapping $\boldsymbol{\phi}(\cdot)$ (where, $\boldsymbol{\phi} : \mathbb{R}^{N_e^x} \rightarrow \mathbb{R}^{N_e^x \times F}$ and F represents the dimensionality of a higher dimensional feature space) is applied independently across each dimension of the input vectors $\boldsymbol{x}_\tau \rightarrow \boldsymbol{\phi}(\boldsymbol{x}_\tau)$ and $\boldsymbol{y}_\tau \rightarrow \boldsymbol{\phi}(\boldsymbol{y}_\tau)$, respectively. Thus, $\boldsymbol{\phi}(\boldsymbol{x}_\tau) = [\hat{\phi}(\boldsymbol{x}_\tau^{\mathbf{1}}), ..., \hat{\phi}(\boldsymbol{x}_\tau^{\boldsymbol{N_e^x}})]^T$, where $\hat{\phi} : \mathbb{R} \rightarrow \mathbb{R}^F$. Therefore, the modified form of equation (7) is given as

$$(\hat{\mathbf{E}}, \hat{\mathbf{D}}) = \arg\min_{E,D} P^{-1} \sum\nolimits_{\tau \in \mathcal{P}} ||\boldsymbol{\phi}(\mathbf{y}_\tau) - \mathbf{E}\mathbf{D}\boldsymbol{\phi}(\mathbf{x}_\tau)||_2^2 + \sum\nolimits_{\rho=1}^{Q} \lambda_\rho^{\mathbf{E}} ||\mathbf{E}_{:\rho}||_1 + \sum\nolimits_{\rho=1}^{Q} \lambda_\rho^{\mathbf{D}} ||\mathbf{D}_{\rho:}||_1. \quad (8)$$

The cost function in (8) consists of two parts, the first one representing the CCA framework (henceforth mentioned as $J^s(\mathbf{E}, \mathbf{D})$) and the second being the ℓ_1 norm regularization part, denote as $J^{reg}(\mathbf{E}, \mathbf{D})$). To minimize the cost in (8) we utilize the gradient descent method to update the clustering matrices $\mathbf{E}$ and $\mathbf{D}$. Since it is an iterative approach the k-th iteration for the update is found by utilizing the following recursive update rule:

$$\hat{\mathbf{E}}_k = \hat{\mathbf{E}}_{k-1} - c\nabla^{\mathbf{E}}_{\hat{\mathbf{E}}_{k-1}, \hat{\mathbf{D}}_{k-1}} J(\mathbf{E}, \mathbf{D}), \quad (9a)$$

$$\hat{\mathbf{D}}_k = \hat{\mathbf{D}}_{k-1} - c\nabla^{\mathbf{D}}_{\hat{\mathbf{E}}_{k-1}, \hat{\mathbf{D}}_{k-1}} J(\mathbf{E}, \mathbf{D}) \quad (9b)$$

where $c > 0$ is the step-size. Note that $\nabla^{\mathbf{E}}_{\hat{\mathbf{E}}_{k-1}, \hat{\mathbf{D}}_{k-1}} J(\mathbf{E}, \mathbf{D})$ and $\nabla^{\mathbf{D}}_{\hat{\mathbf{E}}_{k-1}, \hat{\mathbf{D}}_{k-1}} J(\mathbf{E}, \mathbf{D})$ refer to the partial derivatives of $J(\mathbf{E}, \mathbf{D})$ with respect to $\mathbf{E}$ and $\mathbf{D}$, respectively. It should be noted that since the update equations (9a) and (9b) are dependent on $\mathbf{E}$ and $\mathbf{D}$, at the k-th iteration we utilize the matrix values obtained for these matrices at the $(k-1)$-th iteration. Here we have utilized a Jacobi type update where the new value is dependent only on the previous state. A Gauss-Seidel type update, where if $\hat{\mathbf{E}}_k$ is evaluated first, then for evaluating $\hat{\mathbf{D}}_k$ the current value of $\hat{\mathbf{E}}_k$ can be used instead of using $\hat{\mathbf{E}}_{k-1}$. Nonetheless, the performance was essentially the same for both types of updates.

For ease of notation we replace $\hat{\mathbf{E}}_{k-1}$ and $\hat{\mathbf{D}}_{k-1}$ with $\mathbf{E}$ and $\mathbf{D}$ to get the following:

$$\nabla^{\mathbf{E}} J(\mathbf{E}, \mathbf{D}) = \frac{\delta J^s(\mathbf{E}, \mathbf{D})}{\delta \mathbf{E}} + \frac{\delta J^{reg}(\mathbf{E}, \mathbf{D})}{\delta \mathbf{E}}, \quad (10a)$$

$$\nabla^{\mathbf{D}} J(\mathbf{E}, \mathbf{D}) = \frac{\delta J^s(\mathbf{E}, \mathbf{D})}{\delta \mathbf{D}} + \frac{\delta J^{reg}(\mathbf{E}, \mathbf{D})}{\delta \mathbf{D}}. \quad (10b)$$

The second term in eqs. (10a) and (10b) represent the sub-gradient of the ℓ_1 norm regularization part of equation (8). The sub-gradients are given as:

$$\frac{\delta J^{reg}(\mathbf{E}, \mathbf{D})}{\delta \mathbf{E}} = \text{sgn}(\mathbf{E})\text{diag}(\boldsymbol{\lambda}^{\mathbf{E}}), \quad (11a)$$

$$\frac{\delta J^{reg}(\mathbf{E}, \mathbf{D})}{\delta \mathbf{D}} = \text{diag}(\boldsymbol{\lambda}^{\mathbf{D}})\text{sgn}(\mathbf{D}). \quad (11b)$$

where the matrices $\text{diag}(\boldsymbol{\lambda}^{\mathbf{D}})$ and $\text{diag}(\boldsymbol{\lambda}^{\mathbf{E}})$ are diagonal matrices whose ρ-th elements are $\lambda_\rho^{\mathbf{D}}$ and $\lambda_\rho^{\mathbf{E}}$, which are the sparsity controlling coefficients. The operator $\text{sgn}(\cdot)$ is the element wise sign operator.

The first term in (10a) and (10b), $J^s(\mathbf{E}, \mathbf{D})$ can be written as:

$$J^s(\mathbf{E}, \mathbf{D}) = tr(\hat{\mathbf{C}}_{\mathbf{y}} - 2\cdot\mathbf{E}\cdot\mathbf{D}\cdot\hat{\mathbf{C}}_{\mathbf{xy}} + \mathbf{E}\cdot\mathbf{D}\cdot\hat{\mathbf{C}}_{\mathbf{x}}\cdot\mathbf{D}^T\cdot\mathbf{E}^T) \quad (12)$$

and thus correspondingly we have the non linearly mapped expression:

$$J^s(\mathbf{E}, \mathbf{D}) = tr(\hat{\mathbf{K}}_{\mathbf{y}} - 2\cdot\mathbf{E}\cdot\mathbf{D}\cdot\hat{\mathbf{K}}_{\mathbf{xy}} + \mathbf{E}\cdot\mathbf{D}\cdot\hat{\mathbf{K}}_{\mathbf{x}}\cdot\mathbf{D}^T\cdot\mathbf{E}^T) \quad (13)$$

where $\hat{\mathbf{K}}_x, \hat{\mathbf{K}}_y, \hat{\mathbf{K}}_{xy}$ denote the cross covariance matrix after the non-linear mapping $\boldsymbol{x}_\tau \rightarrow \phi(\boldsymbol{x}_\tau)$ and $\boldsymbol{y}_\tau \rightarrow \phi(\boldsymbol{y}_\tau)$.

$$\hat{\mathbf{K}}_{\mathbf{y}} = P^{-1} \sum_{\tau \in \mathcal{P}_s} \boldsymbol{\phi}(\mathbf{y}_\tau)\boldsymbol{\phi}^T(\mathbf{y}_\tau), \quad (14a)$$

$$\hat{\mathbf{K}}_{\mathbf{x}} = P^{-1} \sum_{\tau \in \mathcal{P}_s} \boldsymbol{\phi}(\mathbf{x}_\tau)\boldsymbol{\phi}^T(\mathbf{x}_\tau), \quad (14b)$$

$$\hat{\mathbf{K}}_{\mathbf{xy}} = P^{-1} \sum_{\tau \in \mathcal{P}_s} \boldsymbol{\phi}(\mathbf{x}_\tau)\boldsymbol{\phi}^T(\mathbf{y}_\tau). \quad (14c)$$

Thus we can obtain the first term of the equations (10a) and (10b), the partial sub gradients of $J^s(\mathbf{ED})$ w.r.t $\mathbf{E}$ and $\mathbf{D}$ by partial derivative of (13). The partial derivatives are given as

$$\frac{\delta J^s(\mathbf{E}, \mathbf{D})}{\delta \mathbf{E}} = -2\cdot\hat{\mathbf{K}}_{\mathbf{xy}}^T\cdot\mathbf{D}^T + \mathbf{E}\cdot\mathbf{D}\cdot\hat{\mathbf{K}}_{\mathbf{x}}^T\cdot\mathbf{D}^T + \mathbf{E}\cdot\mathbf{D}\cdot\hat{\mathbf{K}}_{\mathbf{x}}\cdot\mathbf{D}^T,$$

$$\frac{\delta J^s(\mathbf{E}, \mathbf{D})}{\delta \mathbf{D}} = -2\cdot\mathbf{E}^T\cdot\hat{\mathbf{K}}_{\mathbf{xy}}^T + \mathbf{E}^T\cdot\mathbf{E}\cdot\mathbf{D}\cdot\hat{\mathbf{K}}_{\mathbf{x}}^T + \mathbf{E}^T\cdot\mathbf{E}\cdot\mathbf{D}\cdot\hat{\mathbf{K}}_{\mathbf{x}}.$$

Next we discuss some details about obtaining the kernel covariance matrices and how can we achieve the zero mean behavior (or kernel centering) in the mapped non-linear space.

1) Algorithmic Details: As part of the kernelized CCA framework we utilize one of the most commonly used kernel, the Gaussian radial basis function (RBF) kernel. The entries of this kernel matrix are given as follows:

$$\mathbf{k}_x^{i,j}(\tau, \tau') = \exp\left(-\frac{||\mathbf{x}_\tau^i - \mathbf{x}_{\tau'}^j||^2}{2\sigma^2}\right), \quad (15)$$

where $\mathbf{x}_\tau^i$ and $\mathbf{x}_{\tau'}^j$ correspond to the i-th and the j-th component of the vector $\mathbf{x}_\tau$ and $\mathbf{x}_{\tau'}$, respectively. The variance σ^2 decides the spread upto which the data (in this case the magnitude of difference, $||\mathbf{x}_\tau^i - \mathbf{x}_{\tau'}^j||^2$ is relevant or produces a kernel entry that can impact the CCA process. This is one of the most important parameter for the kernelized CCA and has to be appropriately selected as it directly and significantly impacts the clustering performance.

Another important step in the kernel matrix calculation is the centering. In the linear CCA case (12), the covariance matrix $\hat{\mathbf{C}}_{\mathbf{x}}$ is evaluated with zero mean data vectors. Similarly, post the non-linear transformation of the data to obtain $\boldsymbol{\phi}(\mathbf{x}_\tau)$ and $\boldsymbol{\phi}(\mathbf{y}_\tau)$, the average quantities $\bar{\boldsymbol{\phi}}(\mathbf{x}) := P^{-1}\sum_{\tau\in\mathcal{P}}\boldsymbol{\phi}(\mathbf{x}_\tau)$ and $\bar{\boldsymbol{\phi}}(\mathbf{y}) := P^{-1}\sum_{\tau\in\mathcal{P}}\boldsymbol{\phi}(\mathbf{y}_\tau)$ have to be subtracted respectively from the two quantities, [21]. Since the values $\boldsymbol{\phi}(\mathbf{x}_\tau)$ and $\boldsymbol{\phi}(\mathbf{y}_\tau)$ are never explicitly calculated the centering or the zero mean operation can be effectively achieved as follows:

$$\hat{\mathbf{K}}_x = P^{-1}\sum\nolimits_{\tau\in\mathcal{P}}[\boldsymbol{\phi}(\mathbf{x}_\tau)-\bar{\boldsymbol{\phi}}(\mathbf{x})][\boldsymbol{\phi}(\mathbf{x}_\tau)-\bar{\boldsymbol{\phi}}(\mathbf{x})]^T$$

where the (i,j)th entry of $\hat{\mathbf{K}}_x$, namely $[\hat{\mathbf{K}}_x]_{i,j}$ can be written as $[\hat{\mathbf{K}}_x]_{i,j} = P^{-1}\sum_{\tau\in\mathcal{P}}[\hat{\mathbf{K}}_x(\tau)]_{i,j}$ with

$$\begin{aligned}[\hat{\mathbf{K}}_x(\tau)]_{i,j} :&= [[\boldsymbol{\phi}(\mathbf{x}_\tau)-\bar{\boldsymbol{\phi}}(\mathbf{x})]_i]\cdot[[\boldsymbol{\phi}(\mathbf{x}_\tau)-\bar{\boldsymbol{\phi}}(\mathbf{x})]_j]^T\\ &= \mathbf{k}_x^{i,j}(\tau,\tau) - P^{-1}\sum\nolimits_{\tau'\in\mathcal{P}}\mathbf{k}_x^{i,j}(\tau,\tau')\\ &\quad -P^{-1}\sum\nolimits_{\tau'\in\mathcal{P}}\mathbf{k}_x^{j,i}(\tau,\tau')\\ &\quad +P^{-2}\sum\nolimits_{\tau',\tau''\in\mathcal{P}}\mathbf{k}_x^{i,j}(\tau',\tau''),\end{aligned}\tag{16}$$

as evident from the above equation, $\mathbf{k}_x^{i,j}(\tau,\tau') := [\boldsymbol{\phi}(\mathbf{x}_\tau)]_i \cdot \boldsymbol{\phi}(\mathbf{x}_{\tau'})]_j = \hat{\phi}(\mathbf{x}_\tau^i)\cdot\hat{\phi}(\mathbf{x}_{\tau'}^j)$.

Since the cost function in (8) is non-convex and a gradient descent scheme can potentially get stuck in a local minima, the initialization of matrices $\mathbf{D}$ and $\mathbf{E}$ plays an important role. To find a reasonable estimate of $\mathbf{D}$ and $\mathbf{E}$ matrices, and hence the clusters, we initialize the matrices with multiple set of random entries at iteration $k = 0$ in (9a) and (9b). The set of initialization entries that achieves the lowest value for the cost function in (8) is considered as the final output. The nonzero entries in $\mathbf{D}$ and $\mathbf{E}$ matrices are then used to find the corresponding cluster for each of the input vectors. Since, each column of $\mathbf{D}$ and row of $\mathbf{E}$ may have more than one non-zero entries, each vector in $\mathbf{x}$ (or $\mathbf{y}$) is assigned to the cluster corresponding to the entry with the highest absolute magnitude in the corresponding column of $\mathbf{D}$ (row of $\mathbf{E}$). A summary of the clustering algorithm can be found in Algorithm 2. Further details regarding the selection of $\lambda_\rho^{\mathbf{E}}$ and $\lambda_\rho^{\mathbf{D}}$ can be found in [16].

Algorithm 2 Kernelized CCA-Based Classification

1: Initialize matrices $\hat{\mathbf{D}}_0$ and $\hat{\mathbf{E}}_0$ by random values.
2: Calculate the kernel matrices $\hat{\mathbf{K}}_{\mathbf{x}}$, $\hat{\mathbf{K}}_{\mathbf{y}}$ and $\hat{\mathbf{K}}_{\mathbf{xy}}$ utilizing (14a),(14b),(14c), the kernel trick, (15) and (16).
3: Update $\mathbf{E}$, $\mathbf{D}$ recursively using eqs. (9a) and (9b). Repeat multiple times with different random initializations at $k = 0$.
4: Cluster the epochs using the nonzero entries' indices of the q rows of $\mathbf{D}$ and q columns of $\mathbf{E}$.

2) Segment Classification: Since CCA is an unsupervised clustering scheme, there are no pre-defined labels associated with each cluster. Thus, to report classification accuracy, in accordance with existing schemes in [22] and [23], we further classify each detected segment into one of the known classes as follows. First, we map each cluster to one of the known classes by looking at the known (ground truth) label of each segment. The label of the majority of the segments assigned to each cluster decides the class that maps to that cluster. Subsequently, the segments whose ground truth label does not agree with the label (class) assigned to its cluster is counted as a misclassification. Fig. 3 shows a sample CCA output and its comparison with the ground truth labels.

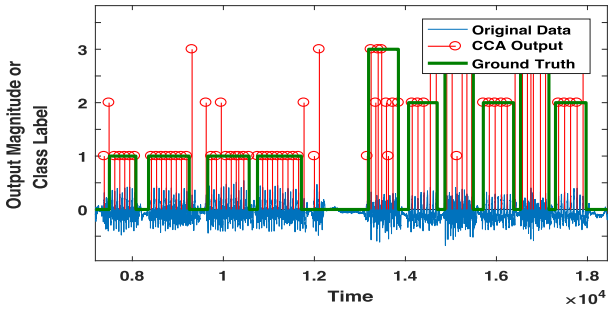

Fig. 3. A sample CCA output for activity detection for a given data stream taken from the SBHAR data set. For the CCA output and the ground truth, a magnitude of 1 represents the activity of walking, 2 represents the activity of climbing-up the stairs and 3 represents the activity of going down the stairs.

IV. Case Studies

In this section we test the performance of the proposed approaches, the kernel regularized canonical correlation analysis (CCA) and the PCA-based approach on 3 different activity data-sets.

For the CCA based algorithm we have used a Gaussian kernel with a variance of $\sigma^2 = 10^{-1.5}$, 10^{-2} and 10^{1} for the SBHAR [8], UniMiB [9] and Texas State datasets, respectively. The $\lambda_i^{\mathbf{D}}$ and $\lambda_i^{\mathbf{E}}$ value were kept fixed at 0.1 and a step size of $c = 5\times10^{-4}$ was used. The values should be appropriately selected such that each column of $\mathbf{D}$ and each row of $\mathbf{E}$ have at least one non zero element, or the scheme suggested in [16] can also be used. The gradient descent iterations were executed up-till the error drops below a factor of 10^{-6}. For each trial of CCA, the $\mathbf{D}$ and $\mathbf{E}$ clustering matrices are randomly initialized.

In the PCA algorithm we have considered the number of eigenvectors used for reconstruction as $\psi = 15$. To understand the impact of the size of the training data on the accuracy of the scheme we average out the results over 10 Monte Carlo iterations while randomly selecting different epochs from the signal for training and keeping the rest of them for validation. We elaborate on the impact of having different epoch lengths and on using different amounts of training data in the later sections.

For the simulations, in the case of CCA, the algorithm is applied towards each user signal separately and the results presented are averaged across all the users for each of the data-sets being considered. In the case of PCA we obtain the results using a 5-fold cross validation over the entire dataset. Tables II-VII show the confusion matrices obtained, including the Precision, Recall and F1-Score per class, as well as the macro-average over all classes and the total classification accuracy.

TABLE I

CONFUSION MATRIX FOR A THREE-CLASS (A, B, C) CLASSIFICATION PROBLEM. PRECISION (P), RECALL (R), AND F1-SCORE (F1) METRICS ARE ALSO SHOWN

		Predicted A	B	C	P_i	R_i	$F1_i$
Actual	A	TP_A	E_{AB}	E_{AC}	P_A	R_A	$F1_A$
	B	E_{BA}	TP_B	E_{BC}	P_B	R_B	$F1_B$
	C	E_{CA}	E_{CB}	TP_B	P_C	R_C	$F1_C$
		Accuracy: $TP/Total$			P_{Avg}	R_{Avg}	$F1_{Avg}$

Assuming a three-class problem with classes A, B and C, the confusion matrices are presented in the format shown in Table I. The Precision (P), Recall (R) and F1-score ($F1$) for class i is calculated respectively as:

$$P_i = TP_i/(TP_i + FP_i),$$
$$R_i = TP_i/(TP_i + FN_i),$$
$$F1_i = 2 \times P_i \times R_i/(P_i + R_i),$$

where TP = True Positive predictions, FP = False Positive, FN = False Negative, and E_{ij} is the error rate of instances of class i missclasified as j.

For example, for class A:

$$FP_A = E_{BA} + E_{CA}, \qquad FN_A = E_{AB} + E_{AC},$$
$$P_A = TP_A/(TP_A + FP_A) = TP_A/(TP_A + E_{BA} + E_{CA}),$$
$$R_A = TP_A/(TP_A + FN_A) = TP_A/(TP_A + E_{AB} + E_{AC}),$$
$$F1_A = 2 \times P_A \times R_A/(P_A + R_A).$$

The total Accuracy is calculated as:

$$Accuracy = \sum_i TP_i/(\sum_i TP_i + \sum_{ij} E_{ij})$$

A. SBHAR Dataset

The first data-set is the Smartphone-based Human Activity Recognition (SBHAR) dataset [8]. The data-set contains a set of 3 major activities (walking, walking-up the stairs and walking down the stairs). In addition to this, there is data for other stationary events like standing, sitting and laying, which we don't consider as events of interest for this study. In our simulations we utilize the XYZ-axis of the accelerometer data to recognize the three major activities in this data. The signals are recorded at a sampling frequency of 50Hz.

For a few of the data samples from this dataset the average value of the signal shifts significantly with time and this interferes with the thresholds used by the pre-processing stage, thus a few of the epochs were not picked up by the pre-processing stage and were not used for the classification stage. The epoch structure in these signals is perfectly fine and thus the PCA and CCA based techniques suggested in this paper are still valid to these signals by adapting the pre-processing part or by using a different scheme for epoch detection. The files are around 6.4 minutes in length on an average. Thus a total of almost 5400 epochs across all the signals have been considered.

TABLE II

CONFUSION MATRIX FOR PCA WITH 5-FOLD VALIDATION USING THE SBHAR DATASET. W = WALKING, U = UPSTAIRS, D = DOWNSTAIRS

		Predicted W	U	D	P_C	R_C	$F1_C$
Actual	W	2334	9	13	99.1	99.1	99.1
	U	21	1079	11	98.4	97.1	97.8
	D	1	8	1909	98.8	99.5	99.1
		Accuracy: 98.8			98.8.9	98.6	98.7

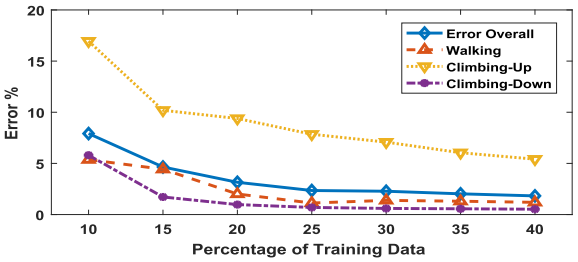

Fig. 4. Percentage error for the PCA based scheme in the SBHAR data-set with different number of training samples.

TABLE III

CONFUSION MATRIX FOR CCA USING THE SBHAR DATASET. W = WALKING, U = UPSTAIRS, D = DOWNSTAIRS

		Predicted W	U	D	P_C	R_C	$F1_C$
Actual	W	1649	355	352	90.1	70.0	78.8
	U	59	975	77	60.2	87.8	71.4
	D	123	289	1506	77.8	78.5	78.2
		Accuracy: 76.7			76.0	78.7	76.1

TABLE IV

COMPARISON WITH OTHER METHODS IN LITERATURE FOR SBHAR DATASET

Classification Method	Accuracy
Proposed Multiset PCA	98.8%
Proposed CCA	76.7%
Probabilistic SVM [8]	98%
LDA	87.1%
LDA with Shrinkage [24]	87.3%
1D-LBP [25]	81.3%

For the PCA based approach, as it can be seen from Fig. 4 the error decreases as the number of training samples increases. The overall error is less than 3% with training data being limited to 25% of the full data-set, beyond that the error slowly decreases as more training data is used. Even with a minimal training data set of 10% the overall error is just 8%. The error is considerably higher with the climbing upstairs data. This is mainly because the epoch structure for climbing down the stairs is not as well formed or structured in its shape as compared to the epochs during walking or climbing down portions of the signal.

Table II gives the classification performance measures for our PCA-based approach when applying a 5-fold cross validation. The confusion matrix for the CCA case is given in Table III. The overall classification accuracies achieved by our algorithms are 98.8% and 76.7% respectively.

Reyes-Ortiz *et al.* [8] report a similar accuracy of 98% on the same dataset using their Probabilistic-SVM approach

TABLE V

CONFUSION MATRIX FOR PCA WITH 5-FOLD VALIDATION USING THE UNIMIB DATASET. SUS = STANDING-UP FROM SITTING, SUL = STANDING-UP FROM LYING-DOWN W = WALKING, R = RUNNING, U = UPSTAIRS J = JUMPING, D = DOWNSTAIRS, LDS = LYING-DOWN FROM STANDING, SD = SITIING-DOWN

		Predicted											
		SUS	**SUL**	**W**	**R**	**U**	**J**	**D**	**LDS**	**SD**	$\mathbf{P_C}$	$\mathbf{R_C}$	$\mathbf{F1_C}$
Actual	**SUS**	133	8	0	0	0	0	0	7	2	83.1	88.7	85.8
	SUL	17	171	2	0	1	0	3	19	2	90.0	79.5	84.4
	W	0	0	1685	3	32	0	15	0	0	95.0	97.1	96.0
	R	0	0	1	1968	1	0	15	0	0	98.2	99.1	98.7
	U	1	0	56	3	829	0	30	1	0	89.5	90.1	89.8
	J	0	0	5	1	8	708	23	0	0	98.2	95.0	96.6
	D	0	0	25	29	52	13	1201	0	0	93.2	91.0	92.1
	LDS	3	10	0	0	2	0	1	252	27	84.6	85.4	85.0
	SD	6	1	0	0	1	0	0	19	173	84.8	86.5	85.6
		Accuracy: 94.1									90.7	90.3	90.5

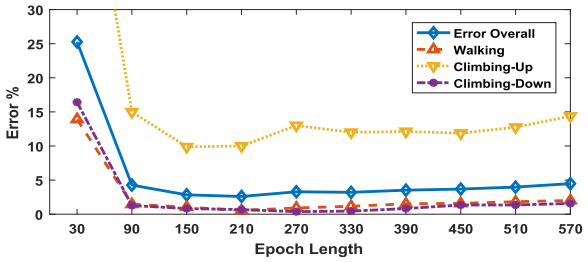

Fig. 5. Percentage error with PCA for different epoch lengths in the SBHAR data-set.

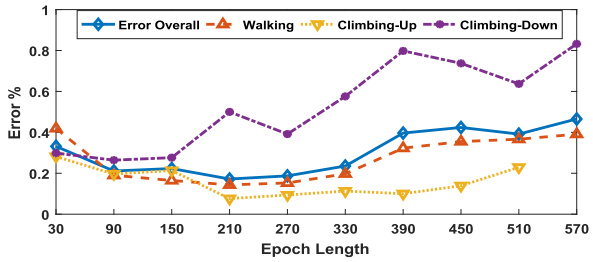

Fig. 6. Percentage error with CCA for different epoch lengths in the SBHAR data-set.

when considering the same three activities (walking, walking upstairs, walking downstairs), after they filter out segments that represent postural transitions. However, their approach relies on prior feature extraction (561 features per window) and a fixed-size window segmentation (2.56 sec). In addition to this, we have also reported results from two techniques which have been popular for classification with EEG signals and in audio signal processing. Blankertz *et al.* [24] have utilized a linear discriminant analysis (LDA) with shrinkage based classifier for classifying the EEG signals. Kaya *et al.* [25] use a more heuristic scheme of finding local binary patterns (LBP) for feature extraction, this is followed by an SVM classifier. The LBP based schemes have also been used in speech processing [26] and for face recognition [27], but for the purpose of comparison given here we have implemented the scheme as it is used with the EEG signals in [25]. The results have been summarized in Table IV.

Another parameter that impacts the performance of the algorithm is the epoch length or the data vector length. As seen in Fig. 5, with the epoch length being small, the error is high since the correlations cannot be accurately observed. As the length of the epoch is increased the accuracy increases. Once the epoch length is increased beyond a threshold, in this case 210, the error starts to increase, as mentioned before this is due to the epoch containing the samples from the neighboring epochs. Similar behavior can be observed even when the length is varied in the case of CCA, as evident in Fig. 6.

B. UniMiB Dataset

The second data set is the University of Milano Bicocca Smartphone-based Human Activity Recognition (UniMiB) dataset [9]. This set has data from 30 users performing a much wider range of activities, a total of 17 activities including 9 daily activities like walking, running, climbing the stairs. The data is pre-split to represent the individual epochs from each of the activities and thus no pre-processing has been done on this data. To showcase the performance over a wider set of activities, we use data pertaining to all the 9 daily activities for this data set. The signals are recorded at a sampling frequency of 50Hz and the dataset for these 9 activities amounts to a total of 7565 epochs. The data from XYZ-axis of the accelerometer is utilized towards classification.

TABLE VI

CONFUSION MATRIX FOR CCA USING THE UNIMIB DATA-SET. (EQUAL EPOCHS) W = WALKING, R = RUNNING, J = JUMPING

		Predicted					
		W	**R**	**J**	$\mathbf{P_C}$	$\mathbf{R_C}$	$\mathbf{F1_C}$
Actual	**W**	515	46	99	64.0	78.0	70.4
	R	144	446	70	79.6	67.6	73.1
	J	145	68	447	72.6	67.7	70.1
		Accuracy: 71.1			72.1	71.1	71.2

The accuracies of the PCA and CCA schemes for the UniMiB dataset are given in the form of a confusion matrix in Tables V and VI, respectively. For the set of daily activities, the PCA based scheme gives an overall accuracy of 94.1%. The macro average accuracy (MAA), or the average of the accuracies of individual activities is 90.3%. In table VII we have compared the performance of the two proposed schemes against the results from Micucci *et al.* [9] who have reported results on the same dataset with a k-nearest neighbor (KNN) classifier, a support vector machine (SVM) based classifier, an artificial neural network (ANN) and a random-forest (RF) classifier. Along with this, we have also reported the accuracies of the two EEG classification schemes in the table.

The number of PC sets used plays a significant role in the accuracy of the PCA based scheme. In Fig. 8 we can see the impact of increasing the number of principal component sets on the macro average accuracy (MAA) across all the classes.

TABLE VII

COMPARISON WITH OTHER METHODS IN LITERATURE FOR UNIMIB DATASET

Classification Method	MAA
Proposed Multiset PCA	90.3%
Proposed CCA	71.1%
SVM [9]	81.6%
KNN [9]	87.8%
ANN [9]	72.1%
RF [9]	88.4%
LDA	40%
LDA with Shrinkage [24]	41.7%
1D-LBP [25]	67.6%

TABLE VIII

CONFUSION MATRIX FOR PCA WITH 5-FOLD VALIDATION USING THE TEXAS STATE DATA-SET. W = WALKING, U = UPSTAIRS, D = DOWNSTAIRS

		Predicted W	D	U	P_C	R_C	$F1_C$
Actual	W	765	13	26	94.0	95.1	94.6
	D	28	157	11	88.2	80.1	84.0
	U	21	8	136	78.6	82.4	80.5
		Accuracy: 90.8			86.9	85.9	86.3

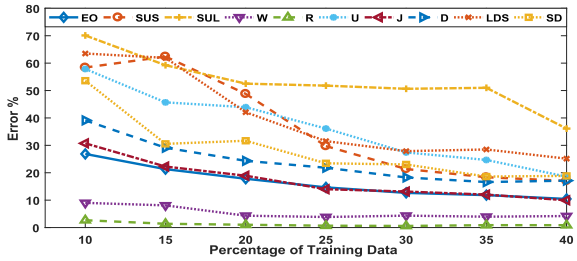

Fig. 7. Percentage error for the PCA based scheme in the UniMiB data-set with different number of training samples.

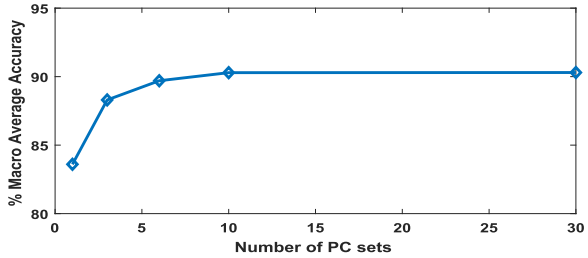

Fig. 8. Percentage macro average accuracy (MAA) for the PCA based scheme in the UniMiB data-set with different number of principal component (PC) sets used.

The MAA increases utill the number of sets is increased to 10 but remains constant after that. This improvement in accuracy though comes at an increased computational cost, especially in regard to the validation. While using B set PCA, the computational cost is increased by a factor of B compared to the single set PCA. This happens because for any incoming epoch, the reconstruction error has to be computed for each of the B PC sets, as explained in section III-B3. But as seen in Fig. 8, the improvement in accuracy saturates pretty quickly and thus for practical purposes, B has a small value. Also, since validation primarily involves the projection of a vector on to the principal components, which is not an expensive operation even if it is done B times, the complexity of the multiset PCA remains considerably low from a practical use standpoint.

CCA being unsupervised, does not require any training data as existing supervised methods, though it is challenged by the presence of many different classes that need to be separated in a dataset. For the CCA based unsupervised clustering approach, we have only presented the clustering accuracies for 3 classes. We are currently improving the approach to reduce computational complexity and enable data clustering in the presence of larger number of clusters. The CCA based approach gives an overall accuracy and a MAA of 71.1% each.

TABLE IX

CONFUSION MATRIX FOR CCA USING THE TEXAS STATE DATA-SET

		Predicted W	D	U	P_C	R_C	$F1_C$
Actual	W	460	134	190	93.9	58.7	72.2
	D	18	132	44	44.6	68	53.9
	U	12	30	126	35.0	75.0	47.7
		Accuracy: 62.6			57.8	67.2	57.9

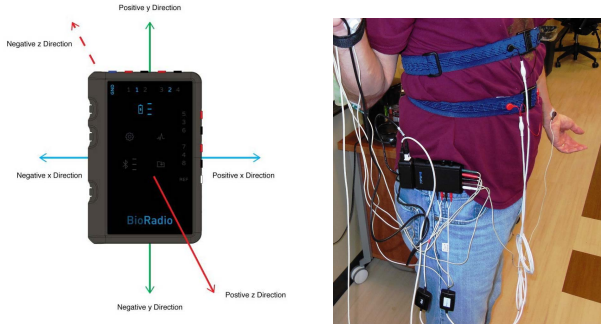

Fig. 9. The accelerometer axes of the Bioradio [28] physiological monitoring device (left) and a photo showing a person wearing the device on their waist during the data collection session (right).

C. Texas State Dataset

The third dataset that we tested was collected by our team at Texas State University, and it is a relatively small data-set, including 3 users performing 5 activities (i.e. walking on a level surface, going downstairs, going upstairs, standing and sitting). The user repeats the activity sequence five times. The total duration of each run was about 2 minutes and 40 seconds. 3-dimensional acceleration data at a sampling rate of 250 Hz were collected using a BioRadio device [28] mounted on the waist of each subject (see Fig. 9). This data-set was mainly created to facilitate the development stages of our methods. Nonetheless, our experimental results are illustrated here as an extra evaluation resource. The accuracy is lower compared to the other datasets due to the small size of the dataset.

Based on the results provided for the proposed methods and in comparison with other existing solutions tested across the three datasets, it can be concluded that the PCA-based scheme gives similar or higher accuracy compared to competitive solutions. It has the added advantage of being cost-effective as only few principal components are used for validation (15 for this case study). The only drawback here is with regard to the need for labeled training data, which may be constrained in some applications. The CCA-based scheme, even-though has lower accuracy compared to PCA (although it outperforms LDA based schemes in some datasets), is completely unsupervised. It does not require any training data, which is a basic requirement for all the other schemes mentioned here, and this gives CCA an obvious advantage.

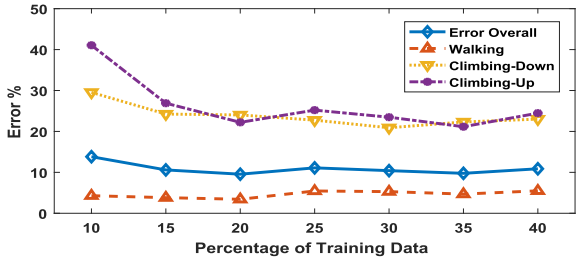

Fig. 10. Percentage error for the PCA based scheme in the Texas State data-set with different number of training samples.

V. Discussion and Conclusion

In this work, we have presented two novel correlation analysis-based approaches for classifying signals representing time-series data. Depending on the application requirements, the proposed approaches are suitable for both supervised or unsupervised learning applications respectively. The advantages of the proposed methods compared to existing approaches can be summarized in the following:

- Both methods described here do not require the extraction of features from the signals being analyzed to achieve the segmentation and classification tasks.
- The signals do not need to be segmented in advance to be used for classification. The segmentation process is part of the overall optimization problem.
- The supervised, PCA-based approach is lightweight and can be suitable for real-time signal classification on low-power or embedded devices.
- The unsupervised, CCA-based approach can be utilized when training data are not readily available.

As part of future work we plan to explore optimization schemes to reduce the computational complexity of the CCA based scheme. Since the optimization involved with CCA is bi-convex in nature, [29]–[31] may provide relevant leads towards solving this problem.

References

[1] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[2] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," *Pervasive Comput.*, pp. 1–17, 2004.

[3] R. Agarwal and J. Gotman, "Adaptive segmentation of electroencephalographic data using a nonlinear energy operator," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Orlando, FL, USA, May/Jun. 1999, pp. 199–202.

[4] A. Procházka, M. Mudrová, O. Vyšata, R. Háva, and C. P. Suárez-Araujo, "Multi-channel EEG signal segmentation and feature extraction," in *Proc. 14th Int. Conf. Intell. Eng. Syst.*, Las Palmas, Spain, May 2010, pp. 317–320.

[5] H. Azami, K. Mohammadi, and B. Bozorgtabar, "An improved signal segmentation using moving average and Savitzky–Golay filter," *J. Signal Inf. Process.*, vol. 3, no. 1, pp. 39–44, 2012.

[6] C. Panagiotakis and G. Tziritas, "Signal segmentation and modelling based on equipartition principle," in *Proc. 16th Int. Conf. Digit. Signal Process.*, Jul. 2009, pp. 1–6.

[7] T. Giannakopoulos and A. Pikrakis, *Introduction to Audio Analysis: A MATLAB Approach*. Amsterdam, The Netherlands: Elsevier, 2014.

[8] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, Jan. 2016, doi: 10.1016/j.neucom.2015.07.085.

[9] D. Micucci, M. Mobilio, and P. Napoletano. (2017). "UniMiB SHAR: A new dataset for human activity recognition using acceleration data from smartphones." [Online]. Available: https://arxiv.org/abs/1611.07688

[10] J. C. Quiroz, A. Banerjee, S. M. Dascalu, and S. L. Lau, "Feature selection for activity recognition from smartphone accelerometer data," *Intell. Automat. Soft Comput.*, pp. 1–9, Jul. 2017. [Online]. Available: https://www.tandfonline.com/action/showCitFormats?doi=10.1080%2F10798587.2017.1342400

[11] A. Gupta, K. Gusain, and D. Kumar, "Novel approach for incremental learning using ensemble of SVMs with particle swarm optimization," in *Proc. 11th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Dec. 2016, pp. 426–430.

[12] K. Gusain, A. Gupta, and B. Popli, "Transition-aware human activity recognition using extreme gradient boosted decision trees," in *Advanced Computing and Communication Technologies*, R. K. Choudhary, J. K. Mandal, and D. Bhattacharyya, Eds. Singapore: Springer, 2018, pp. 41–49.

[13] E. Zdravevski *et al.*, "Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering," *IEEE Access*, vol. 5, pp. 5262–5280, Mar. 2017.

[14] H. He, Y. Tan, and J. Huang, "Unsupervised classification of smartphone activities signals using wavelet packet transform and half-cosine fuzzy clustering," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2017, pp. 1–6.

[15] Shweta, P. Khandnor, and N. Kumar, "A survey of activity recognition process using inertial sensors and smartphone sensors," in *Proc. Int. Conf. Comput., Commun. Automat. (ICCCA)*, May 2017, pp. 607–612.

[16] J. Chen and I. D. Schizas, "Online distributed sparsity-aware canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 64, no. 3, pp. 688–703, Feb. 2016.

[17] D. R. Brillinger, *Time Series: Data Analysis and Theory*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 2001.

[18] J. L. Navarro-Mesa, E. Lleida-Solano, and A. Moreno-Bilbao, "A new method for epoch detection based on the Cohen's class of time frequency representations," *IEEE Signal Process. Lett.*, vol. 8, no. 8, pp. 225–227, Aug. 2001.

[19] P. Xanthopoulos *et al.*, "A novel wavelet based algorithm for spike and wave detection in absence epilepsy," in *Proc. IEEE Int. Conf. Bioinformat. BioEng.*, May/Jun. 2010, pp. 14–19.

[20] A. J. Smola and B. Schölkopf, *Learning With Kernels*, vol. 4. GMD-Forschungszentrum Informationstechnik, 1998.

[21] D. Chu, L. Liao, M. Ng, and X. Zhang, "Sparse kernel canonical correlation analysis," in *Proc. Int. Multiconf. Eng. Comput. Sci.*, Hong Kong, 2013, pp. 322–327.

[22] H. Zha, X. He, C. Ding, M. Gu, and H. Simon, "Spectral relaxation for K-means clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 1057–1064.

[23] J. Chen, A. Malhotra, and I. D. Schizas, "Data-driven sensors clustering and filtering for communication efficient field reconstruction," *Signal Process.*, vol. 133, pp. 156–168, Apr. 2017.

[24] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of ERP components—A tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[25] Y. Kaya, M. Uyar, R. Tekin, and S. Yıldırım, "1D-local binary pattern based feature extraction for classification of epileptic EEG signals," *Appl. Math. Comput.*, vol. 243, pp. 209–219, Sep. 2014.

[26] N. Chatlani and J. J. Soraghan, "Local binary patterns for 1-D signal processing," in *Proc. 18th Eur. Signal Process. Conf.*, Aug. 2010, pp. 95–99.

[27] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[28] *GLNT BioRadio Physiological Monitor*. Accessed: Aug. 11, 2018. [Online]. Available: https://glneurotech.com/bioradio

[29] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality," *Math. Oper. Res.*, vol. 35, no. 2, pp. 438–457, 2010.

[30] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.

[31] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: A survey and extensions," *Math. Methods Oper. Res.*, vol. 66, no. 3, pp. 373–407, Dec. 2007.

Akshay Malhotra, photograph and biography not available at the time of publication.

Ioannis D. Schizas, photograph and biography not available at the time of publication.

Vangelis Metsis, photograph and biography not available at the time of publication.