**Acconotate**: Exploiting Acoustic Changes for Automatic Annotation of Inertial Data at the Source

Soumyajit Chatterjee\(^3\), Arun Singh\(^3\), Bivas Mitra, Sandip Chakraborty
Department of CSE, IIT Kharagpur, India
Email: \{sjituit, erarungw12013\}@gmail.com, \{bivas, sandipc\}@cse.iitkgp.ac.in

**Abstract**—Smart infrastructures often intend to provide personalized context-aware services for their residents. These context-aware services, in turn, often rely on sophisticated machine learning algorithms which need vast volumes of costly annotated sensor data. State-of-the-art automated annotation frameworks try to solve this problem by generating annotated sensor data obtained from personal wearables. However, most of these approaches — (a) either need visual data from the environment or (b) can only work for environments with a single resident. This paper discusses the design of a first-of-its-kind framework Acconotate which can automatically generate annotated data from dual resident smart environments without requiring any visual information. Acconotate achieves this by exploiting the typical transitions present in complex human activities first to solve the critical problem of the user-to-activity association and then use that to annotate the sensor stream available from both the users. Rigorous evaluation with two real-life datasets collected in two diverse scenarios shows that Acconotate can successfully generate annotated sensor data over the edge without human intervention.

**Index Terms**—Human Activity Annotation, Acoustic Context, Signal Processing

I. INTRODUCTION

**Motivation:** Consider a context-aware application for tracking the progress of dementia. The application is deployed in a smart home with an elderly couple as residents. A typical non-invasive way of providing this service can be through seamless tracking of the general and instrumental activities of daily living (ADL) from the locomotive signatures obtained from wearables \([1]\). However, developing sophisticated human activity recognition (HAR) models often need labeled data, which is extremely hard to obtain. Conventional solution of human-in-the-loop annotation for such pervasive sensing applications is already known to be costly, erroneous, and noisy \([2]–[4]\). Also, incremental solutions like active learning \([2]\) or experience sampling \([5]\) fail in these scenarios as it might not be possible for the elderly residents to validate (or provide) the ground-truth labeling. Furthermore, all these approaches are prone to data privacy breaches in the annotation process, as sharing the information with some third party (or a human annotator) might be required. More recent solutions, like \([6]–[9]\) do discuss the development of automated annotations frameworks, with most of these ideas replacing the human-in-the-loop with videos serving as the auxiliary source of label information. However, it might not be possible to capture continuous videos in every environment, for example, within a smart home. Also, extracting complex activities from videos need additional infrastructure support \([8], [9]\).

**Primary Challenges:** Interestingly, an efficient and privacy non-invasive solution for this problem can be through extracting meaningful information from the environmental acoustic signatures, which, unlike videos, can be processed over the edge (like the smartphone or voice assistant) to identify complex ADL(s) \([10]\). Works like \([4], [11]\) exploited this idea of cross-modal learning using audio to develop lightweight, edge-friendly and accurate annotation frameworks, albeit these cannot be extended to environments with more than one user. The reason behind this restriction is the generic problem of activity to user association that becomes a bottleneck for any automated annotation system without having any visual cue from the environment. This is because it is incredibly challenging for the system (or even for a human annotator) to understand “who is doing what” without any visual information coming from the environment.

**Opportunity:** Notably, unlike machine-generated noises, sounds generated from human activities are not continuous and contain intermediate gaps. This is primarily because several soundless micro-activities exist under a broad activity signature, and only some micro-activities generate sounds. For example, during chopping vegetables, sounds are generated only when the knife hits the chopping board. We term these intermediate soundless durations as **acoustic gaps** during an activity. Fig. 1 shows such acoustic gaps from a pilot study we conducted in workshop and kitchen environments, respec-

---

\(^3\)Both the authors were working at the Department of Computer Science and Engineering, IIT Kharagpur, when the work was done.
Fig. 2: The workshop setup. The smartwatch used for collecting the sensor data is visible on the wrist of the user. It is paired with a COTS smartphone that captures the acoustic signatures to avoid the problem of time-drift.

Intuitively, in the workshop scenario, the sound is generated only when the hammer hits a nail, and we observe periodic acoustic gaps. Similarly, in a kitchen, the sound is produced when the utensils clink, and acoustic gaps are observed in between. Interestingly, if spotted correctly, such acoustic gaps allow us to identify the person who has taken a gap. It can potentially help us perform the activity to user association for the other user present in the environment, even without the necessity of having any visual information.

Our Contribution: We in this paper exploit this opportunity presented by the general phenomena of acoustic gaps across human activities to develop the automated framework called Acconotate. Acconotate can automatically annotate the personalized IMU streams obtained from a two-user smart environment without using any visual information or human intervention. To the best of our knowledge, this work is the first of its kind that investigates the possibilities of developing an automatic annotation system for smart environments with more than one user without using any visual cue from the environment. To achieve this, Acconotate in the backend uses lightweight signal-processing techniques to identify the acoustic gap from the globally convoluted audio with the assistance from individual IMU streams and use that to perform the activity to user association. Additionally, it uses pre-trained audio models to generate annotated IMU data over the edge without violating the data boundary. We evaluate Acconotate over in-house datasets across two different contexts of a workshop and a kitchen. Rigorous experimentation with different acoustic and environmental settings shows that Acconotate can generate highly accurate annotated datasets which can be used for developing supervised HAR models.

II. DATASETS

To the best of our knowledge there are no public datasets from a realistic smart infrastructure environment that contain both – (a) IMU from personal wearables along with acoustic data and (b) with more than one users working performing different ADL(s) together in the environment. Hence, we opt to rely on an in-house data collection, where individuals can perform the given tasks without external control and with minimum interference from the connected devices.

We rely on a minimal setup where the IMU data is collected using a Moto-360 smartwatch (sampling rate = 50Hz) worn on the preferred arm of the participant formal approval taken for the corresponding human-centric experiments from the Institute Ethical Committee (analogous to IRB). A COTS Smartphone is utilized to capture the audio generated from the environment (sampling rate = 44.1kHz). The smartwatch was paired apriori with the smartphone over Bluetooth for inter-modality synchronization. We obtain the data from two different environments, namely Workshop and Kitchen, involving a total of 8 volunteers in this experiment. We collected the data in a single-user setup, where every volunteer has been asked to perform one single activity at a time, independently & freely, without having any external constraints. In the Workshop environment, we involve 4 volunteers and ask them to separately perform two primary activities – (a) hammering a wooden plank or a metal pipe, & (b) cutting a wooden plank or metallic pipe using a saw. During this process, the participants organically perform other auxiliary micro-activities, like picking up the nails, or fitting the plank, etc. Similarly in Kitchen environment, the other 4 volunteers are asked to separately conduct two primary activities – (a) chopping vegetables with a knife on a chopping board, & (b) cooking the vegetables in a frying pan or a cast iron wok. For both the Workshop and the Kitchen environments, we captured timestamped videos with a frame rate of 30fps and used them to annotate the ground-truth activity labels.

Next, we use the collected realistic single-user datasets and augment them by time-synchronizing the IMU signals and convoluting the acoustic signals among volunteers to create an augmented dataset adapting the strategy discussed in [12]. This strategy can effectively mimic the multi-user setup with two users performing simultaneous activities. Additionally, the use of state-of-the-art (SOTA) frameworks like Pyroomacoustics [13] allows us to evaluate further the impact of physical factors like distance from the microphone. Finally, following this data augmentation strategy, we create a set of 12 (TABLE I) and 5 (TABLE II) unique augmented combinations for each subject-activity pair in the two setups, respectively.

### TABLE I: Augmented Dataset Details – Workshop

| Combination Id | Activities | IMU Instances | Global Audio Duration (secs) |
|---------------|------------|---------------|-----------------------------|
| C1            | U1, U2     | 6730          | 134.739                     |
| C2            | U1, U2     | 2629          | 52.628                      |
| C3            | U1, U4     | 7652          | 153.222                     |
| C4            | U3, U1     | 6957          | 139.27                      |
| C5            | U4, U1     | 6957          | 139.27                      |
| C6            | U2, U1     | 6957          | 139.27                      |
| C7            | U4, U3     | 6730          | 134.739                     |
| C8            | U4, U2     | 2629          | 52.628                      |
| C9            | U2, U3     | 6730          | 134.739                     |
| C10           | U5, U2     | 2629          | 52.628                      |
| C11           | U3, U4     | 7652          | 153.222                     |
| C12           | U2, U4     | 7652          | 153.222                     |

### TABLE II: Augmented Dataset Details – Kitchen

| Combination Id | Activities | IMU Instances | Global Audio Duration (secs) |
|---------------|------------|---------------|-----------------------------|
| C1            | U3, U4     | 20983         | 420.305                     |
| C2            | U2, U1     | 24351         | 487.787                     |
| C3            | U5, U2     | 20981         | 420.305                     |
| C4            | U4, U3     | 11596         | 232.304                     |
| C5            | U4, U1     | 11597         | 232.304                     |
III. SOLUTION OVERVIEW

Let two users $U_m$ and $U_n$ perform two different primary activities $p_i$ (denoted as $U_m \rightarrow p_i$) and $p_j$ (denoted as $U_n \rightarrow p_j$), respectively, for a time period $[0, T]$. Let $I_m(0, T)$ and $I_n(0, T)$ be the unlabeled IMU data collected through the wearable from each of the two users (see Fig. 3), respectively, and $A(0, T)$ be the audio signal for the entire duration $[0, T]$ captured from a VPA deployed in the environment.

A. Problem Definition and Key Idea

The objective of 

\textit{Acconotate} is to develop a framework to annotate the completely unlabeled IMU data $I_m(0, T)$ and $I_n(0, T)$ utilizing the acoustic information extracted from $A(0, T)$. Our key idea is that there are granular auxiliary micro-activities, say \{a_{m,n}^1, a_{m,n}^2, \ldots \}, within a primary activity $p_i$, some of which do not generate a distinctive sound, thus producing a gap in the acoustic signal. Accordingly, we define a term “Acoustic Gaps” as follows.

\textbf{Definition 1 (Acoustic Gap):} Let a user $U_m$ performs a primary activity $p_i$ for a period $[0, T]$. We define an acoustic gap as the intermittent time duration $[t, t+\Delta]$, ($\Delta \geq 1$s), when a different acoustic context is produced due to an interleaved auxiliary micro-activity $a_{m,n}^k$.

We aim to leverage the acoustic gaps extracted from $A(0, T)$ during primary activities $\{p_i, p_j\}$ to opportunistically annotate the completely unlabeled IMU data $I_m(0, T)$ and $I_n(0, T)$. We consider that the label-space of the primary activities $\{p_i, p_j\}$ is known and included in the label-space of a pre-trained audio-based HAR model like [10]. However, the user-to-activity mappings $U_m \rightarrow p_i$ and $U_n \rightarrow p_j$ are not known. Similar to previous works like [11], we also assume that IMU and audio signals are time-synchronized using any standard approaches like RTP or NTC [14], [15].

\textbf{B. Acconotate in a Nutshell}

By exploiting the acoustic gaps, \textit{Acconotate} first maps the primary activities $\{p_i, p_j\}$ to the users \{U_m, U_n\} and then annotates the IMU data $I_m(0, T)$ and $I_n(0, T)$ with the corresponding primary activities $\{p_i \rightarrow U_m, p_j \rightarrow U_n\}$. As shown in Fig. 4, the entire framework is divided into three major stages. In Stage 1, \textit{Acconotate} independently detects signal changes in $I(0, T)$ and $A(0, T)$ using unsupervised approaches. In Stage 2, \textit{Acconotate} extracts the acoustic gaps from the signal change points to map the primary activities to the corresponding users, i.e. $p_i \rightarrow U_m$ and $p_j \rightarrow U_n$. Specifically, \textit{Acconotate} relies on a pre-trained audio-based activity recognition module for identifying the primary activities $\{p_i, p_j\}$, which allows us to avoid human intervention and detect activities in an automated manner. However, lack of an appropriate number of acoustic gaps and presence of multiple acoustic sources may confuse \textit{Acconotate}, and thus the framework may generate conflicting mappings. To resolve this, \textit{Acconotate} applies a conflict resolution technique that allows it to judiciously map an activity label to a user during the entire duration $[0, T]$.

Ultimately, in Stage 3, \textit{Acconotate} collates all the information from the previous two stages to finally output the annotated IMU data for both the users. The details follow.

IV. STAGE 1: SIGNAL CHANGE DETECTION

The first stage of \textit{Acconotate} uses an unsupervised approach to detect the instances when the distributions of the input signals, both for $I(0, T)$ and $A(0, T)$, show a change in their patterns, indicating that an activity change has happened in the environment for one of the users. A. Detecting Changes in IMU

The objective of this step is to find out windows of duration $[\nu, \eta], 0 \leq \nu < \eta \leq T$ for each user $U_n$, such that each $I_m(\nu, \eta)$ corresponds to one of the activities from $\{p_u, a_{m,n}^1, a_{m,n}^2, \ldots \}$.

\textbf{B. Calculating Change-Points}

To achieve this, we first rely on the statistical \textit{Change-point Detection} [16], [17] approach to evaluate the changes in the IMU data stream. Formally, a change-point represents a point in time where a time series or a stochastic process has changed its probability distribution. Change-point scores quantify these changes in terms of numerical scores. \textit{Acconotate} computes the change-point scores to quantify the change in the IMU data stream considering the input from a tri-axial accelerometer. Say, $x_i$ represents the input from a tri-axial accelerometer at time $t$. Then, to compute the change-point scores, we create windows in the IMU data, such that a window $X_t = \{x_t, x_{t+1}, \ldots, x_{t+f-1}\}$, where, $f$ is the window size $1$. Subsequently, we compute the change-point score $\mu_i$ between any two consecutive IMU windows $X_t$ and $X_{t+f}$ using the $\alpha$-relative Pearson Divergence Estimation (PE), following a similar procedure as discussed in [18].

1) Identifying the Actual Changes

Although the change-point scores can quantify the amount of changes in an IMU stream; however, they do not explicitly demarcate the actual event changes; instead, they give a relative score of changes in the distribution. Ideally, the actual activity changes within the IMU signal should produce relatively higher values of the change-point scores.
change-point scores. However, as the IMU data is unlabeled, we cannot determine an empirical threshold on the change-point score. To solve this problem, we apply an unsupervised approach, where we cluster the obtained change-point scores into two sets—one corresponding to the actual activity changes in the IMU data and the other containing the rest of the change-point scores. Thus, we perform a $k$-means clustering (with $k = 2$) on all the scores obtained from pairwise consecutive IMU windows. Subsequently, we demarcate the set of scores belonging to the cluster with a higher mean as the actual activity change scores.

B. Detecting Changes in Acoustic Data We split the audio signal into 1 second segments for extracting the audio change points and compare the two consecutive segments to detect changes. However, measuring the change in acoustic context is not straightforward, as the environmental acoustic context is a convoluted signal from different activities and other acoustic sources [19]. Therefore, existing methods like [20], which use techniques like Mel Frequency Cepstral Coefficients (MFCC), fail to locate the change points correctly in our context. It is known that MFCC becomes ineffective in the presence of multiple noise sources [21], especially for non-speech environmental acoustic signatures [22].

1) Observing Changes in the Environmental Acoustic Context Notably, recent works like [22] have pointed out that power spectral densities can appear as a more relevant alternative to cases where there are limitations on the usage of MFCC due to random noise components introduced by hardware or external sources. Based on these understandings, we choose Cross Power Spectral Density (CPSD) to identify changes in the environmental acoustic context. Formally, CPSD estimates the similarity (or relationship) between two time-domain signals [23] and is obtained by performing Fourier transform on the cross-correlation of the two signals. Since CPSD returns the power density across all the frequency bins present in both the signals, we consider the sum of absolute output values across all frequency bins to obtain the final CPSD value between two consecutive audio segments. In this context, an important observation is that unlike change-point scores computed for IMU signatures, higher CPSD values indicate lesser chances of any change between two consecutive segments. However, these are also mere values only, and we need to cluster them to demarcate actual changes. The detail follows.

2) Identifying the Actual Changes in the Acoustic Context Similar to clustering the change-point scores for demarcating activity changes in the IMU data, here as well, we cluster the CPSD values. However, a primary problem with acoustic signatures is the presence of different random noise components. Additionally, we also observe that some acoustic signature changes are also introduced due to the behavioral changes in performing the activity. For example, a user may change how she holds the saw according to her ease in a workshop environment. Depending on that, the sound generated while cutting the plank can also change. Due to all these reasons, there can be a huge number of uneventful change-points if we cluster them into two sets only, which can negatively impact Acconotate’s performance. Thus to avoid this, we first obtain the optimal number of clusters for CPSD values using the Silhouette score [24]. For this, we cluster the CPSD values into $c$ clusters, where $c \in [2, C]$ and choose $c$ with maximum Silhouette score as the optimal number of clusters. Once the clustering is done, we mark the cluster with minimum mean CPSD value as the cluster containing actual activity changes.

V. STAGE 2: ACTIVITY TO USER ASSOCIATION

Acconotate maps primary activities $\{p_i, p_j\}$ to users $\{U_m, U_n\}$ in two stages. (a) Using $I_{\text{IMU}}, u \in \{m, n\}$, it first identifies $U_m, u \in \{m, n\}$ who might have taken a break in her primary activity during the time segment $[\nu, \eta], 0 \leq \nu < \eta \leq T$ (thus produces an acoustic gap in $A(\nu, \eta)$ for $U_m, u \in \{m, n\}$). (b) In the next step, it uses $A(\nu, \eta)$ to obtain the activity labels from a pre-trained audio-based HAR model [10] and maps those activities to the individual users based on the timing analysis over acoustic gaps.

A. Extracting Acoustic Gaps One of the major challenges in correlating IMU and audio is that the change-points computed individually from them are not synchronized. Notably, whenever a difference observed in the IMU signal because of the change in primary activity, a change is also observed in the acoustic signal, albeit with a slightly smaller window compared to the acoustic change. This is because when the user stops performing the primary activity, the acoustic signature drops immediately, while the IMU signatures still record the transition. For example, when a user resumes using a saw, the acoustic signature captures this instantly; however, IMU changes a bit earlier when the user just picks up the saw. Therefore, Acconotate uses an opportunistic approach to exploit the acoustic gaps by combining the observations from both modalities. For this, we use the notion of Exclusive Change defined as follows.

Definition 2 (Exclusive Change): Say, at some time-interval $[\nu, \eta], 0 \leq \nu < \eta \leq T$, we observe a change in $I_{\text{IMU}}(\nu, \eta)$ for user $U_m$. We define this change as an exclusive change if and only if the following two conditions are met.

1. $\exists$ a time-interval $[\theta, \zeta], 0 \leq \theta \leq \nu < \eta \leq \zeta \leq T$, where there is a change in $A(\theta, \zeta)$.
2. For the entire time interval $[\theta, \zeta]$, we do not observe any change in $I_{\text{IMU}}(\theta, \zeta)$ for the other user $U_n$.

Based on this definition, Fig. 5 shows an exclusive change for the user $U_m$ over the IMU signal $I_{\text{IMU}}$. The exclusive changes indicate the presences of an acoustic gap. Once we determine these exclusive changes for the individual users, we identify the activity labels from the acoustic context and map them to individual users as follows.

B. Associating Activities to Users The acoustic gaps during the exclusive changes help us to find out a unique mapping from the activity labels $\{p_i, p_j\}$ to the users $\{U_i, U_j\}$. Let, $[\beta, \gamma]$ be a continuous time interval, and $[\nu, \eta], \beta \leq \nu < \eta \leq \gamma$ be an exclusive change detected for the user $U_m$ from

---

2We choose $C = 6$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Let $A_m(\beta, \gamma)$ and $A_n(\beta, \gamma)$ be the pure acoustic signal components generated from the activities (primary or auxiliary) being performed by $U_m$ and $U_n$, respectively, during the time interval $[\beta, \gamma]$. Further, consider that $N(\beta, \gamma)$ is the environmental noises generated from non-human activities (like the sound from AC, dog barking, etc.). Then, $A(\beta, \gamma) = A_m(\beta, \gamma) \oplus A_n(\beta, \gamma) \oplus N(\beta, \gamma)$, where $\oplus$ is the signal convolution operator. It can be noted that $A(\beta, \gamma)$ should contain change-points near $\nu$ and $\eta$, as the primary activity for $U_m$ changes at $\nu$ and $\eta$; therefore, $A_m(\beta, \nu + \Delta_1)$, $A_m(\nu + \Delta_1, \eta + \Delta_2)$, and $A_m(\eta + \Delta_2, \gamma)$ should have different distributions in their Power Spectral Density (PSD). Here, $\Delta_1$ and $\Delta_2$ are small adjustment windows, as the IMU change-points and the acoustic change-points may not be perfectly time-synchronized. However, $U_n$ does not change her activity during $[\beta, \gamma]$, and therefore $A_n(\beta, \gamma)$ should not ideally contain any change-points.

**Activity detection from acoustic context:** We first detect the activities from the environmental acoustic signature $A(\beta, \gamma)$ during the entire duration $[\beta, \gamma]$. As one of the main objectives of **Acconotate** is to minimize human-in-the-loop, therefore, we rely on the concept of pre-trained models for audio-based activity recognition. Specifically, we adapt the model suggested in [10] for this purpose, which is pre-trained using YouTube-8M [25] dataset and uses context information to filter out noisy activity labels effectively. For an input acoustic signature, the pre-trained model returns a set of detected activities with an associated confidence level indicating the model’s confidence over the detected activities. Notably, the label space of [10] also contains the workshop and kitchen activities defined in our datasets.

**Separating activities:** Let $A(\beta, \gamma)$ be the set of activities returned by the pre-trained model [10] during the time interval $[\beta, \gamma]$. $A(\beta, \gamma) = A_m(\beta, \gamma) \cup \{p_j\} \cup A_N(\beta, \gamma)$, where $A_m(\beta, \gamma)$ is the set of activities (including the primary activity and the auxiliary activities) being performed by $U_m$, $p_j$ is the single primary activity being performed by $U_n$ during the entire duration $[\beta, \gamma]$, and $A_N(\beta, \gamma)$ is the set of non-human noisy activities. $A_N(\beta, \gamma)$ is easily separable as they do not belong to the target activity set. $A_m(\beta, \gamma)$ should contain one primary activity $p_i$ within the duration $[\beta, \nu + \Delta_1]$ & also within $[\eta + \Delta_2, \gamma]$, and an auxiliary activity $a_m$ within the duration $[\nu + \Delta_1, \eta + \Delta_2]$, as detected from the pre-trained acoustic-based activity recognition model [10].

**Mapping the primary activities based on the IMU changes:** To map the primary activities $p_i$ and $p_j$ with the corresponding users $U_m$ and $U_n$, we now look into the change-points detected in $I_m(\beta, \gamma)$ and $I_n(\beta, \gamma)$. $I_m(\beta, \gamma)$ should have a change (which is the exclusive change) within the duration $[\nu, \eta]$, whereas, $I_n(\beta, \gamma)$ should not contain any change-points. Consequently, we should observe a break in $p_i$ within the duration $[\nu, \eta]$ (when the auxiliary micro-activity $a_m$ was performed), but there will be no break in $p_j$. Thus, these two cases are easily separable based on the exclusive changes at $U_m$; therefore, **Acconotate** maps $p_i$ to $U_m$ and $p_j$ to $U_n$ unanimously for the window $[\beta, \gamma]$.

The above approach works well unless $p_i = p_j$, i.e., both $U_m$ and $U_n$ performs the same activity at the same instance of time where the system only has to observe the changes in activity signatures and map them to acoustic changes. Therefore, we argue that **Acconotate** can label the IMU data for the majority of the cases, which is also evident from the detailed experiments, as discussed next.

**C. Putting It All Together** We repeat the two steps mentioned above for the entire duration to obtain the activity mappings for all the exclusive changes across both the users. Specifically, we first create a list of exclusive changes $E_m$ and $E_n$ for the users $U_m$ and $U_n$, respectively. Mathematically, the contents of these lists can be defined as follows.

$$E_j = \{(\theta, \zeta)|\forall [\nu, \eta] \in \text{exclusive changes in } I_j\}, j = \{m, n\}$$

Once these lists are obtained for each user, we then separately get the activity mappings for each of the entries in $E_m$ and $E_n$ by querying the audio-based activity recognition model $^3$ with the audio segment corresponding to the given time-interval $[\theta, \zeta]$. However, it can be noted that due to the noise in the acoustic data as well as the confusion with the acoustic-based activity recognition model, different exclusive changes from $E_m$ may result in different activity labels for the other user $U_n$.

However, we consider that a user performs a single primary activity within the entire duration $[0, \theta]$. Therefore, to map a single primary activity label to the user $U_n$, we consider the activity label with the majority from the list of exclusive changes $E_m$ for the user $U_m$.

**D. Conflict Resolution** Although this mapping seems straightforward, several challenges may appear while mapping the activity labels to the users. Notably, we observe that a critical condition may arise when there is a conflict, and the same activity gets mapped to both the users. This appears when both the users have rarely performed the auxiliary micro-activities, or the acoustic context changes are because of the users’ external noise or behavioral changes. Since **Acconotate** is concerned with generating annotations for the IMU data for which we use a pre-trained acoustic model, fine-tuning

---

$^3$We choose the activity label with maximum confidence.
the audio-based activity recognition model is entirely out of scope. However, where the results are conflicting, we apply the following strategy. For resolving the conflict where Acconotate maps the same activity to both the users, we start by defining the opportunistic user from the set of two users.

**Definition 3 (Opportunistic User):** Among two users \( U_m \) and \( U_n \) performing simultaneous activities, \( U_m \) is called to be an opportunistic user if \( |\mathcal{E}_m| > |\mathcal{E}_n| \), i.e., \( \mathcal{I}_m(0,T) \) indicates higher number of exclusive changes than \( \mathcal{I}_n(0,T) \).

In other words, an opportunistic user is the user who has taken more number of breaks during her primary activity and thus provided the framework more opportunities to map the activities to the other user correctly. Subsequently, to resolve the conflict, we assume the decision made by observing the changes in \( \mathcal{E}_m \) to be true and map the inferred activity to \( U_m \).

**VI. STAGE 3: IMU ANNOTATION**

The output of Stage 2 of the framework is the individual activity labels (corresponding to their primary activities) for each user for the entire duration \([0,T]\). Next, the final task is to annotate the unlabeled IMU data for both the users with the respective mapped activity labels. It can be noted that \( \mathcal{I}_m(0,T) \) and \( \mathcal{I}_n(0,T) \) may contain the instances of the auxiliary activities performed by the users; however, we are only interested in annotating the IMU segments when they have performed their primary activities. Although we have a unique activity to user mapping for both users, we cannot use this mapping directly for the IMU annotation because of the following two reasons. (1) The activity labels are returned by a pre-trained acoustic-based activity recognition model [10], and thus the returned activity labels are synchronized with the acoustic change-points. (2) The IMU change-points may not be perfectly time-synchronized with the acoustic change-points, as they are computed independently.

**Acconotate** solves these issues as follows. The pre-trained acoustic-based HAR model [10] returns the activity labels with an associated confidence value. We first find out the acoustic segment, say \( A(\beta,\gamma) \), which returns the primary activity \( p_i \rightarrow U_m \) with the maximum confidence label. We then extract the corresponding IMU segment \( I_m(\beta,\gamma) \), termed as the **Key Segment**. We first label the key segment \( I_m(\beta,\gamma) \) with the activity label \( p_i \). Based on the IMU change-point detection technique, we have segmented \( I_m(0,T) \) where IMU change-points mark the start and the end of each segment. We first fit all these IMU segments \( I_m(\theta,\zeta) \) to an unsupervised nearest neighbor algorithm, which learns the patterns of these IMU segments. As similar types of segments should come closer, and they collectively indicate the same activity (because the same activity should produce a similar variety of signal patterns for a user). We now use the key segment \( I_m(\beta,\gamma) \) to find out the \( z \) number of nearest neighbors of that segment and annotate all those segments using the activity label \( p_i \) (see Section VII).

**TABLE III: Accuracy and Volume of Annotations – Workshop.** Rows highlighted in red depict the erroneous cases of activity to user association.

| Id | % Annotation Accuracy | % Annotation Volume |
|----|-----------------------|---------------------|
|    | z = 9                 | z = 11               | z = 13               | z = 15               |
|    | Ham. Saw | Ham. Saw | Ham. Saw | Ham. Saw | Ham. Saw |
| C1 | 59.2 | 95.6 | 62.7 | 96.3 | 59.6 | 96.9 | 57.8 | 94.7 | 62.8 | 97.9 |
| C2 | 67.9 | 99.0 | 70.5 | 88.4 | 67.3 | 79.2 | 61.3 | 84.3 | 68.4 | 89.2 |
| C3 | 100.0 | 80.4 | 95.8 | 80.9 | 93.9 | 84.7 | 93.2 | 76.9 | 100.0 | 80.4 |
| C4 | 98.1 | 34.3 | 98.2 | 92.9 | 97.1 | 68.3 | 97.5 | 68.2 | 98.1 | 34.3 |
| C5 | 81.7 | 87.3 | 81.1 | 80.1 | 77.9 | 81.5 | 67.1 | 80.1 | 81.7 | 87.3 |
| C6 | 98.7 | 100.0 | 97.9 | 100.0 | 97.9 | 99.9 | 97.1 | 99.9 | 98.7 | 100.0 |
| C7 | 64.6 | 90.6 | 65.1 | 91.9 | 57.7 | 94.5 | 59.5 | 94.9 | 64.6 | 90.6 |
| C8 | 92.9 | 95.9 | 92.2 | 94.5 | 88.3 | 93.1 | 87.3 | 89.1 | 92.9 | 95.9 |
| C9 | 83.8 | 100.0 | 83.2 | 100.0 | 85.3 | 100.0 | 85.6 | 100.0 | 83.8 | 100.0 |
| C10 | 38.6 | 100.0 | 38.2 | 100.0 | 36.2 | 100.0 | 38.2 | 100.0 | 38.6 | 100.0 |
| C11 | 38.6 | 100.0 | 38.2 | 100.0 | 36.2 | 100.0 | 38.2 | 100.0 | 38.6 | 100.0 |
| C12 | 38.6 | 100.0 | 38.2 | 100.0 | 36.2 | 100.0 | 38.2 | 100.0 | 38.6 | 100.0 |

**VII. EVALUATION**

We evaluate the performance of **Acconotate** in **Kitchen** and Workshop scenarios (see Section II for the in-house collected dataset) from multiple perspectives, starting with the quality of activity labels generated by **Acconotate**, to its utility in developing supervised models, with a glimpse on its resource consumption.

A. Accuracy of Activity to User Association

TABLE III and TABLE IV indicate that **Acconotate** can correctly map activities to users for 10 out of 12 cases in the Workshop environment and 3 out of 5 cases in the Kitchen environment, respectively. Additionally, we observe that the conflict resolution technique used in **Acconotate** helps resolve Case C3 in the Kitchen.

Evidently, **Acconotate** assigns incorrect activities to the users for combination C3 and C8 in **Workshop** and C2 and C4 in the Kitchen environment. Close inspection reveals that in C3 for **Workshop**, the user U4 performed sawing at a stretch, taking only a single break in the entire period. This provides fewer opportunities to **Acconotate** for annotating user U1 (performing hammering). On the other hand, in C8, the framework cannot provide any conclusive activity annotation for both the users, mostly due to the external noises present in the environment. Albeit **Acconotate** could identify 41 acoustic
TABLE IV: Accuracy and Volume of Annotations – Kitchen. Rows highlighted in red depict the erroneous cases of activity to user association. For combination C3, we first observe a conflict with both users annotated with labels Cooking, however, the conflict resolution approach chooses U2 as the opportunistic user with maps U3 with ‘Cooking.’

| Id | % Annotation Accuracy | % Annotation Volume |
|----|-----------------------|---------------------|
|    | z = 9                | z = 11              | z = 15              |
| C1 | Chop                  | Chop                | Chop                |
|    | 100.0                 | 100.0               | 98.6               |
|    | [6.6]                 | [6.3]               | [7.7]              |
|    | Cook                  | Chop                | Chop                |
|    | 100.0                 | 100.0               | 97.3               |
|    | [6.6]                 | [6.6]               | [8.1]              |
| C2 | Both Cooking \(\Rightarrow\) U2 Opportunistic \(\Rightarrow\) U1: Cooking | Chop | Chop |
|    | 100.0                 | 100.0               | 100.0              |
|    | [3.6]                 | [8.9]               | [13.2]             |
| C3 | Chop                  | Chop                | Chop                |
|    | 100.0                 | 100.0               | 100.0              |
|    | [5.1]                 | [10.1]              | [12.1]             |
|    | Chop                  | Chop                | Chop                |
|    | 100.0                 | 100.0               | 100.0              |
|    | [6.3]                 | [6.6]               | [13.2]             |
| C4 | No Activities Detected for any User | Chop | Chop |
|    | 100.0                 | 100.0               | 100.0              |
|    | [8.1]                 | [8.1]               | [19.1]             |

chances, the audio-based activity recognition model fails to recognize any meaningful activities for 38 of them. Side by side, we notice a general performance drop of Acconotate in Kitchen environment. This attributes to the fact that the kitchen, in general, exhibits an inherently complex nature of activities, such as cooking and chopping. This results in a jumbled patterns in the acoustic contexts, which adversely affects the audio-based activity recognition model of the framework. For example, we observe that the model incorrectly detects cooking using the sound of handling utensils, which can occasionally occur while chopping as well.

B. Accuracy in Annotating IMU Data Next, we dive deep and investigate the performance of Acconotate in correctly annotating the IMU stream \(I_m\), generated by a specific user \(U_m\), with a correct activity label \(p_i\) (say, \(p_1 \rightarrow U_m\)). We measure the accuracy of the annotations for a user \(U_m\) by comparing the overlap of the annotated instances with the ground-truth activity labels for each time instance. For every overlap, we assign a score of 1 and a 0, otherwise.

Concerning the Workshop dataset, from TABLE III, we observe that for most of the users, the framework generates annotations (for some value of \(z\)) with accuracy > 70%, albeit the volume of annotations is less than 20% for some instances, especially in the case of users performing sawing. The IMU data from sawing exhibits frequent change-points, and the IMU change patterns within those change-points depend on factors like the holding the saw and the speed. Consequently, we see significant variations in the per-window IMU pattern. As we rely on a nearest-neighbor strategy for labeling the IMU data, such variations in the patterns result in few IMU windows showing similarity with the key segment (Section VI), resulting in a lower volume of annotated data. However, for Kitchen (see TABLE IV), we observe that Acconotate performs with better accuracy. This improvement can be attributed to the inherent nature of kitchen activities like cooking and chopping, which by default require intermediate stops, thus providing many opportunities for the framework to identify the patterns accurately. However, one important point regarding similarity of patterns also reveals that common activities like cooking often have high variability. For example, users may change the way they use the cooking spud depending on the item being cooked. These variations over time cause frequent small change windows impacting the volume of annotation.

One critical observation is that the IMU patterns heavily depend on the context. E.g., auxiliary micro-activities are much less during workshop actives. On the contrary, kitchen activities inherently provide a better opportunity for annotation. Therefore, Acconotate can widely be used for annotating ADLs; nevertheless, it can also annotate non-ADL workshop activities to a satisfactory extent.

C. Impact of Unequal Distance from the Microphone We next evaluate the impact of physical factors like unequal distance from the microphone on the overall accuracy of the framework. To achieve this, we use the SOTA virtual acoustic environments available in [13], where we first design a virtual workshop of dimension \((50 \times 50 \times 50)\)m\(^3\) and position the microphone at the location having coordinates \((4, 2.5, 1.2)\). Accordingly, we then fix one of the activities (say hammering) at a distance of 2m and vary the distance for the other activity within the room (say using saw). TABLE V and TABLE VI show that performance of Acconotate with varying distance. In most cases, the framework can effectively annotate to a distance of 5m. However, after 5m, the system performance drops. This is mostly because the outcome of the acoustic activity recognition module [10] gets heavily affected with the increase in distance from the microphone.

Furthermore, for two combinations (for C8 and C11), we observe that although the framework can figure out the correct activity mapping for one user, the framework cannot map the other user to any relevant activity. From deeper investigation, we find that for these cases, the acoustic activity recognition module failed to provide an instance where it could potentially detect the other activity with significant confidence. Similarly, we also observe instances where irrelevant activities like ‘Hazard Alarm,’ ‘Vaccum in Use,’ and ‘Drill in Use’ are detected due to an increase in noise with the increase in distance from the microphone.

D. Benchmarking Annotated Data The primary purpose of annotating sensor data is to create training data for supervised models. Understanding this objective, we assess the quality of our annotated data in a cross-user setup. This setup allows us to benchmark the annotated data and provides insight if Acconotate can be used to get bootstrap data for a few users which can be later used to obtain the label for other unlabeled data. To evaluate this, we use a leave-one-out strategy to test a particular users’ original data with supervised models trained using the annotated labels (\(z = 15\)) and the labeled IMU streams as features from the other users in the setup. For the workshop, we thus generate four sets (Set 1 to Set 4) of training data by leaving out one of the four users in each of the cases and use the left-out user to test the model accuracy. For the Kitchen, we had data for both the activities only for user U2 whose data we use as the test dataset and thus create two training datasets with a mixture of chopping data from U1, cooking data from users U3 (Set 1) and U4 (Set 2), respectively. From the results with two standard supervised learning algorithms (Random Forest and
TABLE V: Cases where the person performing the activity of 'Hammering' worked at different distances from the microphone. The person performing 'Using Saw' stood at a fixed distance of 2m from the microphone. The cells marked in red are the cases where Acconotate fails to associate both the users with any activities. The cells marked in yellow are the cases where Acconotate makes wrong associations between the activities and the users.

| Combination | Hammer Distance = 5m | Hammer Distance = 10m | Hammer Distance = 20m | Hammer Distance = 30m |
|-------------|----------------------|-----------------------|-----------------------|-----------------------|
|             | Hammer: | Saw: | Hammer: | Saw: | Hammer: | Saw: | Hammer: | Saw: | Hammer: | Saw: |
| C1          | U1: Drill in Use | U3: Saw | U1: None | U3: Drill In Use | 51.01 | 94.85 | 59.21 | 94.85 |
| C2          | 61.05 | 81.55 | U1: None | U2: None | 51.75 | 97.75 |
| C3          | 64.27 | 98.34 | U1: None | U4: None | U1: None | U4: None | 96.25 | 98.63 |
| C7          | 97.14 | 99.95 | 97.14 | 99.95 | U4: Saw | U3: None | U4: None | U2: None |
| C8          | U1: None | U2: None | U4: Hazard Alarm | U2: Saw | U4: None | U2: None | U4: None | U2: None |
| C9          | 58.41 | 95.36 | 95.36 | 95.36 | 95.36 | 95.36 | 95.36 | 95.36 |

TABLE VI: Cases where the person performing the activity of 'Using Saw' worked at different distances from the microphone. The person performing 'Hammering' stood at a fixed distance of 2m from the microphone. The cells marked in red are the cases where Acconotate fails to associate both the users with any activities. The cells marked in yellow are the cases where Acconotate makes wrong associations between the activities and the users.

| Combination | Saw Distance = 5m | Saw Distance = 10m | Saw Distance = 20m | Saw Distance = 30m |
|-------------|-------------------|-------------------|-------------------|-------------------|
|             | Saw: | Hammer: | Saw: | Hammer: | Saw: | Hammer: | Saw: | Hammer: | Saw: |
| C4          | U1: None | U3: None | U1: None | U3: None | U1: None | U3: None | U1: None | U3: None |
| C5          | 99.09 | 68.17 | U4: Saw | U1: Hammering | U4: Hazard Alarm | U2: None | U4: None | U1: None |
| C6          | U2: Saw | U1: Hammering | U2: Saw | U1: Hammering | U2: Saw | U1: Hammering | U2: Saw | U1: Hammering |
| C7          | 88.64 | 82.71 | 88.63 | 82.71 | 95.18 | 94.21 |
| C8          | 82.71 | 95.18 | 95.18 | 94.21 |

Fig. 6: Cross-user activity recognition accuracy with the annotated data produced by the framework in the (a) Workshop and (b) Kitchen setups, respectively.

SVM) shown in Fig. 6, we observe that for most of the cases the supervised models attain > 0.70 F1-score, in correctly predicting the activity labels in the cross-user setup, which demonstrates the quality of annotation by Acconotate.

E. Resource and Time Consumption Acconotate uses audio as an auxiliary modality to make the overall framework lightweight so that the privacy-sensitive data can be kept within its territory by running the model on the edge. To assess this, we profile the stages of the framework individually on a per-module basis for time and memory consumption (using Linux proc filesystem) on a Dell Inspiron 14 5410 laptop with 16GB primary memory running a Linux environment (kernel version: 5.14.0-1042-oem) on an Intel Core i5 processor. The results particularly for the C1 dataset from the Workshop, as shown in Fig. 7a, highlights that except for the audio-activity-recognition module, all the remaining modules take < 200MB memory which allows them to run on resource-constrained edge devices. Albeit the audio-based HAR module consumes a significant amount of memory, however, it is known to run on edges as shown in [10].

Concerning the time consumption, the total running time for Acconotate on Workshop-C1 is 969.15 seconds, with the majority of time being consumed to detect changes in the IMU data (≈ 80.80%). However, this total time consumed is for the entire dataset containing the IMU streams from both users, which may be provided in sessions to reduce data volume and processing time. Notably, the module-wise mean time consumption across the workshop dataset shows that the maximum time is consumed for detecting the changes in the IMU for each user with a mean time consumption of 549.40(±202.29) seconds, followed by the acoustic change detection which consumes 2.94(±0.74) seconds. Nevertheless, we observe that Acconotate is capable of running on any resource-constrained edge device which allows any smart-infrastructure using Acconotate to secure the data boundary and reduce the necessity of uploading privacy-sensitive data to third-party clouds or servers.

F. Impact of $z$ An additional investigation in analyzing reveals that indeed for certain activities like using saw and cooking the patterns change over a time (see Fig. 8) which leads to larger Euclidean distances for the set of unsupervised nearest neighbors, albeit without hampering the overall accur...
Fig. 8: Variation of Euclidean distance across different $z$ for (a) Hammering, (b) Using saw, (c) Cooking, and (d) Chopping.

accuracy however, with a decrease in volume as explained above.

VIII. CONCLUSION AND FUTURE WORK

Obtaining annotated datasets required to train the ML algorithms is the primary bottleneck for any smart infrastructure provider. Undoubtedly automated annotation frameworks can be a major tool to mitigate this challenge and reduce the overall cost. However, SOTA automated annotation frameworks lack the necessary qualities and either depend on visual cues from the environments or are restricted to single-user smart environments only. In this paper, we design the framework \textit{Accomodate} that extracts meaningful information from the gaps taken within any complex ADL to solve the complex problem of activity to user association and use that to generate annotated locomotive data without any human intervention. Notably, \textit{Accomodate} also does not require any visual information from the environment and is lightweight enough to be run over the edge; thus is completely privacy non-invasive. Nevertheless, an important future step for this work would be to extend it to more complex environments with more than two users.

REFERENCES

[1] S. R. Ramamurthy, S. Chatterjee, E. Galik, A. Gangopadhyay, N. Roy, B. Mitra, and S. Chakraborty, “Cogax: Early assessment of cognitive and functional impairment from accelerometry,” in \textit{IEEE PerCom}, 2022, pp. 66–76.

[2] H. Hossain and N. Roy, “Active deep learning for activity recognition with context aware annotator selection,” in \textit{ACM KDD}, 2019, p. 1862–1870.

[3] M. Zeni, W. Zhang, E. Bignotti, A. Passerini, and F. Giunchiglia, “Fixing mislabeling by human annotators leveraging conflict resolution and prior knowledge,” \textit{ACM MMWUT}, pp. 32:1–32:23, 2019.

[4] S. Chatterjee, A. Chakma, A. Gangopadhyay, N. Roy, B. Mitra, and S. Chakraborty, “LASO: Exploiting locomotive and acoustic signatures over the edge to annotate imu data for human activity recognition,” in \textit{ACM ICMI} 2020, 2020, p. 333–342.

[5] G. Laput and C. Harrison, “Sensing fine-grained hand activity with smartwatches,” in \textit{ACM CHI} 2019, 2019, pp. 338:1–338:13.

[6] M. Benndorf, F. Rungleben, T. Haenselmann, and B. Yadav, “Automated annotation of sensor data for activity recognition using deep learning,” \textit{INFORMATIK}, 2017.

[7] F. Cruciani, I. Cleland, C. Nugent, P. McCullagh, K. Synnes, and J. Hallberg, “Automatic annotation of human activity recognition in free living using a smartphone,” \textit{Sensors}, vol. 18, 2018.

[8] N. P. Owoh, M. M. Singh, and Z. Zaaba, “Automatic annotation of unlabeled data from smartphone-based motion and location sensors,” \textit{Sensors}, vol. 18, 2018.

[9] V. Roy, P. Hevesi, O. Kovalenko, and P. Lukowicz, “Let there be imu data: Generating training data for wearable, motion sensor based activity recognition from monocular rgb videos,” in \textit{ACM UbiComp 2019 and ISWC 2019}, 2019, pp. 699–708.

[10] G. Laput, K. Anuja, M. Goel, and C. Harrison, “Ubicoustics: Plug-and-play acoustic activity recognition,” in \textit{ACM UIST} 2018, 2018, pp. 213–224.

[11] T. Xing, S. Sandha, B. Balaji, S. Chakraborty, and M. Srivastava, “Enabling edge devices that learn from each other: Cross modal training for activity recognition,” in \textit{ACM EdgeSys} 2018, 2018, p. 37–42.

[12] S. Chatterjee, A. Singh, B. Mitra, and S. Chakraborty, “Poster abstract: Realistic multiuser, multimodal (IMU, acoustic) HAR data generation through single user data augmentation,” in \textit{ACM/IEEE IPSN} 2022, 2022.

[13] R. Scheibler, E. Bezzam, and I. Dokmanic, “Pyroomacoustics: A python package for audio room simulation and array processing algorithms,” in \textit{IEEE ICASSP} 2018, 2018, pp. 351–355.

[14] E. Spriggs, F. De La Torre, and M. Hebert, “Temporal segmentation and activity classification from first-person sensing,” in \textit{IEEE CVPR Workshops} 2009, 2009, pp. 17–24.

[15] A. Fleury, M. Vacher, and N. Noury, “Svm-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results,” \textit{Trans. Info. Tech. Biomed.}, vol. 14, pp. 274–283, 2010.

[16] S. Aminikhanghahi, T. Wang, and D. Cook, “Real-time change point detection with application to smart home time series data,” \textit{IEEE Trans Knowl Data Eng}, pp. 1010–1023, 2018.

[17] Y. Kawahara, T. Yairi, and K. Machida, “Change-point detection in time-series data based on subspace identification,” in \textit{IEEE ICDM} 2007, 2007, pp. 559–564.

[18] M. Alam, N. Roy, A. Gangopadhyay, and E. Galik, “A smart segmentation technique towards improved infrequent non-speech gestural activity recognition model,” \textit{PMC}, vol. 34, pp. 25–45, 2017.

[19] M. Islam and S. Nirpon, “Sound-adapter: Multi-source domain adaptation for acoustic classification through domain discovery,” in \textit{ACM IPSN} 2021, 2021, p. 176–190.

[20] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y. Chen, J. Li, and B. Firner, “Crowd4+: Unsupervised speaker count with smartphones,” in \textit{ACM UbiComp} 2013, 2013, pp. 43–52.

[21] X. Zhao and D. Wang, “Analyzing noise robustness of mfcc and gfc features in speaker identification,” in \textit{IEEE ICASSP} 2013, 2013, pp. 7204–7208.

[22] F. Atlas, J. Socoró, and X. Sevillano, “A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds,” \textit{Applied Sciences}, vol. 6, 2016.

[23] C. S. Analysis, “Cross power spectral density spectrum for noise modelling and filter design,” https://resources.system-analysis.cadence.com/blog/mma2020-cross-power-spectral-density-spectrum-for-noise-modelling-and-filter-design, 2020.

[24] P. Rousseveau, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” \textit{J. Comput. Appl. Math.}, vol. 20, pp. 53–65, 1987.

[25] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadaran, and S. Vijayanarasimhan, “Youtube-8m: A large-scale video classification benchmark,” \textit{arXiv preprint arXiv:1609.08675}, 2016.