GroupSense: A Lightweight Framework for Group Identification

Snigdha Das, Soumyajit Chatterjee, Sandip Chakraborty, and Bivas Mitra

Abstract—In an organization, individuals prefer to form various formal and informal groups for mutual interactions. Therefore, ubiquitous identification of such groups and understanding their dynamics are important to monitor activities, behaviors, and well-being of the individuals. In this paper, we develop a lightweight, yet near-accurate, methodology, called *GroupSense*, to identify various interacting groups based on collective sensing through users' smartphones. Group detection from sensor signals is not straightforward because users in proximity may not always be under the same group. Therefore, we use acoustic context extracted from audio signals to infer the interaction pattern among the subjects in proximity. We have developed an unsupervised and lightweight mechanism for user group detection by taking cues from network science and measuring the cohesivity of the detected groups regarding modularity. Taking modularity into consideration, *GroupSense* can efficiently eliminate incorrect groups, as well as adapt the mechanism depending on the role played by the proximity and the acoustic context in a specific scenario. The proposed method has been implemented and tested under many real-life scenarios in an academic institute environment, and we observe that *GroupSense* can identify user groups with on an average $0.9(\pm 0.14)$ F_1-Score even in a noisy environment.

Index Terms—Group detection, smartphone, collective sensing

1 Introduction

Workplace meetings and team formation among the individuals are key factors behind organizational efficiency. People formally as well as sporadically meet, interact and form groups for various purposes, which include information sharing [1], teaching and learning [2], problem solving and decision making [3], brainstorming [4], socialization [5], etc. Tracking the dynamics of group formation facilitates various utilities; for instance, organizational leaders may prefer to monitor the formation of teams, which benefit the overall efficiency and activeness of the organization [6]; course instructors in flipped classrooms [7] may like to know how the students form groups among themselves to solve assignments. Unlike regular & pre-scheduled team meetings, the formation of sporadic and instantaneous groups (often observed in office breaks, conferences etc.) make the problem challenging. On the other hand, increasing the availability of sensor-rich smartphones provides a unique opportunity for collecting wide sensor information in a seamless manner. In this backdrop, we investigate the potential of smartphones to develop a lightweight ubiquitous system for identifying and monitoring group formation. Notably, in this paper, we primarily concentrate on the *meeting* groups where co-located group members occasionally interact with each other. In this line, we capture different types of real-life meeting group scenarios such as outdoor roadside informal meeting; informal outdoor cafe meet, formal and informal laboratory meeting, and classroom interaction as shown in Fig. 1.

Identification of a meeting group primarily relies on the location proximity [8] of the group members, which (apparently) can be conceptualized as a localization problem [9]. In that direction, prior art explores the following three modalities – GPS, Bluetooth, and WiFi for identification of the location similarity in supervised [8] as well as unsupervised [10] manner. In our context of group detection, vanilla localization based solutions demand high accuracy, which pushes the system towards complex processing. Notably, location proximity alone is insufficient to correctly discriminate and identify the meeting groups. For instance, consider a large conference hall where multiple meeting groups get formed simultaneously; here members of different groups may exhibit location similarity among themselves, which makes the group detection challenging. Close inspection reveals that albeit similarity in location proximity, context [11] of the members participating in individual meeting groups play a critical role in identifying groups; for instance all the members of a specific group in the conference hall share a substantial amount of contextual similarity (room illumination, ambient noise, member interactions, magnetic fluctuations) [11], [12]. However, identifying suitable contextual information, which is computationally lightweight as well as carries the signature of a meeting group, is an important problem.

We propose *acoustic context*, extracted from the audio signals received by individual smartphones, as a key context indicator. The initial attempts consider the physical features of the audio signal, which primarily capture the particular aspects of temporal & spectral properties such as amplitude, audio pressure and frequency components of the signal. Due to the multipath effect [13], [14], the signal can be delayed,

• *The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal 721302, India. E-mail: snigdhadas@sit.iitkgp.ac.in, sjituit@gmail.com, {sandipc, bivas}@cse.iitkgp.ac.in.*

Manuscript received 12 Apr. 2018; revised 20 Nov. 2018; accepted 29 Nov. 2018. Date of publication 12 Dec. 2018; date of current version 31 Oct. 2019. (Corresponding author: Sandip Chakraborty.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2886333

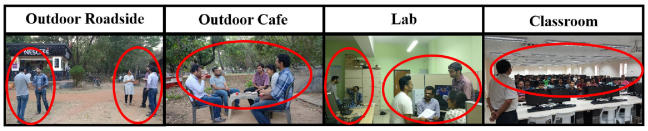


Fig. 1. Setup of different meeting group scenarios.

and the temporal features are highly affected by the delayed signal.[1] On the other side, frequency components are affected by the echo and reverberation, which introduces additional frequency components in the original audio signal.[2] Hence, we move to the high-level features such as *tone* to compute *acoustic context*, which consumes the perceptual information of the signal. One can apply standard *Mel-frequency Cepstral Coefficients* (MFCC) [15] on the recorded audio signals for measuring the tone & pitch. However, this solution comes with multiple challenges. (a) The process of using MFCC usually follows a supervised approach which needs individuals' pre-trained information. (b) MFCC is computationally expensive, which makes it inappropriate for developing a lightweight system. (c) MFCC technique is quite sensitive to noise, hence most suitable for the unidirectional microphone with the stereo channel. Unfortunately, most of the commercial smartphones are equipped with omnidirectional microphones, which makes them prone to noise and corrupting the speaker identification process.[3]

In Next2Me [16], Baker and Efstratiou attempted to detect social groups considering WiFi and microphone fingerprints. First, WiFi signal strengths are used for detecting the co-located population; next, this filtered population is fed to the audio module for finding out the social groups. The audio module considers the top n frequencies of all the co-located the individuals and computes the pairwise similarities. However, in the real-life environment, getting the actual top n frequencies is challenging, and little variation in the selection of frequencies exerts a huge impact on the similarity computation. Additionally, the audio signals captured on different smartphones can be time drifted, even if a single speaker acts as the audio source, since the subjects (devices) may be at different distances from the speaker. Once the co-located population has been identified, and audio based context information has been extracted, the state of the art techniques perform naive component analysis [16] and community detection [17] to identify social groups. However, in most of the cases, quality (cohesivity) of the discovered groups have been overlooked, which leads to the detection of incorrect communities (false positives).

In this paper, we develop *GroupSense*, a smartphone-driven ubiquitous platform for automatic detection of meeting groups. The proposed method is lightweight, unsupervised, hence equipped to detect instantaneously formed groups without any pre-training. First, we determine the co-located population using standard localization techniques [8], [16], [18]. For this, we rely on the WiFi-based proximity schemes; nevertheless, this can be extended to Bluetooth and GPS based techniques as well. The crux of the proposed method is the

1. https://www.music.mcgill.ca/ gary/618/week2/effects.html
2. https://www.eit.lth.se/fileadmin/eit/courses/etin80/2016/reports/sound-effects.pdf (last accessed Oct 1, 2018)
3. https://www.fabathome.org/best-smartphone-microphone/ (last accessed Apr 12, 2018)

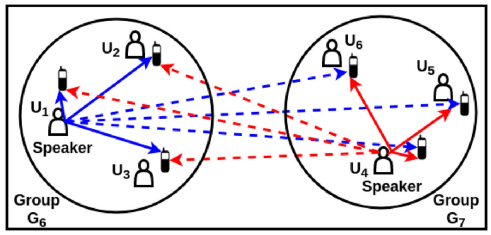


Fig. 2. Impact of audio signals in group detection - two speakers from two different groups talk simultaneously.

computation of *acoustic context* of the identified co-located population, which is based on the following key intuition. Interactions between participants of a meeting group switches from one speaker to another; where, at a time, there exists (mostly) one dominating speaker. Hence, power of the dominating tone (say α_1) captured by the smartphones (subjects[4]) in one group (say G_6 in Fig. 2) is significantly different from the power of the dominating tone (α_4) captured by the devices of another group G_7. If both the groups G_6 and G_7 in Fig. 2 are closely located, then all the devices might capture both the tones with varying power. However, for the devices in group G_6, the power of the dominating tone α_1 should be higher than α_4, whereas exactly the opposite is likely to happen for the group G_7. By discriminating against the power of the dominating tone, one can differentiate the acoustic context of the members of different groups. Finally, leveraging on the proximity of the co-located population and their acoustic context, we propose *GroupSense*, a community-driven group detection model. The advantages of this model are manifold. (1) The model is unsupervised and lightweight. (2) This model can perform group detection even in the absence of proximity indicators (say, WiFi etc.). (3) We take cues from network science and measure the cohesivity of the detected groups with the help of *modularity*. Taking modularity into consideration, *GroupSense* can efficiently eliminate incorrect groups as well as adapt the algorithm depending on the role played by the proximity and the acoustic context in a specific scenario.

The organization of the paper is the following. In Section 2, we formally define the meeting group and state the problem of group detection. We introduce two primary indicators – (a) proximity and (b) audio signal, and explore the related literature. We also conduct pilot experiments to highlight the current challenges. In Section 3, we propose an audio signal processing approach that can capture the acoustic context even with low power microphones available with the smartphones. In Section 4, we develop *GroupSense*, a group detection model leveraging on the community detection algorithms. In Section 5, we show that *GroupSense* can detect such groups with more than 90 percent accuracy while incurring low computation overhead compared to the existing state of the art methods.

2 Problem Definition & Background Study

In this section, first, we define the meeting group and state the problem of group detection in the context of smartphone-based sensing. Next, we identify the primary indicators

4. In this paper, we use the term 'subject' to indicate a participant, a member or a smartphone, interchangeably.

(say, proximity, acoustic context etc.) facilitating the group detection and explore their potential in the light of state of the art endeavours. Finally, we concentrate on the acoustic context and conduct a pilot study to highlight the challenges of group detection from audio signatures.

2.1 Problem Statement

We start with the definition of a *Meeting Group* and subsequently state the problem of group detection.

Definition 1 (Meeting Group). *Given a population of subjects $\mathbb{U}$, we define a meeting group $\mathcal{G}^{[t,t+T]} \subseteq \mathbb{U}$ for the time period $[t, t+T]$ as the collection of co-located individuals $\{u_i \in \mathbb{U}\}$ sharing similar context.*

For instance, two subjects u_i and u_j participate in a group $\mathcal{G}^{[t,t+T]}$ iff u_i and u_j are located in close proximity and share similar context for time duration $[t, t+T]$ [10], [19]. In this paper, we hypothesize that sound profile, observed by the group members, defines the *acoustic context* of a group. For instance, sensing the verbal interactions between the participating members can discriminate one meeting group from another. Notably, in the acoustic context, we only concentrate on the *tone* information. Consider each subject $u_i \in \mathbb{U}$ carries a smartphone equipped with various sensors. We collect the sensor log s_i from each subject u_i and populate the data in a central repository $\mathcal{X}$. The sensor log s_i comprises of the location information p_i and acoustic information α_i. The location information may come from various signals for indoor and outdoor localization techniques based on GPS, wireless signals etc. [9], [20], [21], 22, 23]; similarly acoustic information can be extracted from the audio signals captured by the smartphones [24], [25], [26]. We aim to discover the meeting group $\mathcal{G}^{[t,t+T]}$ formed during the period $[t, t+T]$ from the logged sensor repository $\mathcal{X}$.

2.2 Primary Indicators and Respective Prior Art

The definition of the meeting group mainly relies on two sensing modalities – (a) *Location Proximity* and (b) *Acoustic Context*. We explore the recent attempts in this direction and highlight their potential & challenges in group detection.

2.2.1 Location Proximity

Localizing the subjects within their proximity is the initial step towards the group identification. In this line, the past literature explores mainly three modalities – GPS, Bluetooth, and WiFi. GPS [23] is an important modality (albeit energy-hungry) for localization and detecting population within proximity. Although GPS performs well in outdoor environments, its accuracy sharply falls in indoor environments due to the interruption in the signal [20]. On the other side, the Bluetooth-based study is one of the earliest attempts for localization in indoor environments. However, Bluetooth scanning is power hungry [8]. Moreover, many of the Android smartphones (starting from ver. 4.4) have partial support for Bluetooth Low-Energy (BLE), which are capable of only detecting other BLE devices [11]. Additionally, the Bluetooth signal as a medium of information is considered to be unreliable and noisy.

Recently, attempts have been made to detect proximity from WiFi fingerprint [8]. WiFi consumes significantly less power compared to Bluetooth and GPS. Although BLE appears as an alternative to WiFi regarding power consumption, nevertheless BLE suffers from data loss and fluctuations with increasing distance [27]. Furthermore, WiFi can work in any environment irrespective of whether the device location is indoor or outdoor. Each modality has its positive and negative aspects in the context of localization. Hence, the selection of modalities is highly dependent on the application for which the proximity is computed. In [8], the authors have developed a supervised based learning approach for person-to-person proximity detection using WiFi fingerprints, like access point (AP) coverage and signal strength measurements. On the other hand, the authors in [16] have developed an unsupervised learning based approach for proximity detection using a novel WiFi based metric computed using *Manhattan distance*, which is the average of the pairwise signal strength difference among the APs, from which the subject receives signals. For group identification, any of these existing schemes can be used.

2.2.2 Acoustic Context

A microphone is an important indicator to identify the meeting group members. Participants, in general, avoid talking simultaneously in a meeting; although there can be a small overlap when the discussion switches from one speaker to another (utterance duration). Therefore the voice properties, such as pitch and tone of the current speaker in a group dominates in the audio signals captured by individual subjects in that group [28]. *Pitch* defines the perceived fundamental frequency of the sound [29], whereas *tone* is the variation or thickness of the pitch, indicating the quality of the sound [30]. Fig. 2 explains the intuition behind using human voice characteristics for group identification. The blue audio signal dominates for the subjects of G_6, whereas the red signal dominates for the subjects of G_7. Therefore, human voice characteristics (aka acoustic context) may show a strong feature similarity, if the subjects belong to the same group.

Audio pitch and tone extraction from human voice signal is a well-studied problem in the literature [29], [30]. YIN [29] is a simple time-domain pitch calculation algorithm which is used in many existing applications such as counting the crowd from human voice signals [30]. Although the pitch is a good indicator for speaker identification, however, pitch alone fails to differentiate the relative distance of the speakers from other subjects, since it only concentrates on the central frequency of the audio signal. Therefore, tone information needs to be extracted along with the pitch, and *Mel-frequency Cepstral Coefficients* based techniques [15] with Gaussian Mixture Model (GMM) [31] can be applied for this purpose. However, in smartphones, the use of unidirectional microphones with the stereo channel is rare. A smartphone may capture the voice signals from the subjects of the other nearby groups, apart from the primary speaker of its group, as shown in Fig. 2. Further, the environmental noise generated from the variety of external sources may collude the recorded audio signal. For instance, the humming noise generated from the ACs and other machines (indoor) and vehicles (outdoor) may collude the collected audio signals and make the group detection challenging. Additionally, for instantaneous group detection, there is no apriori knowledge of the group members' tone information. Therefore, group identification from MFCC

TABLE 1
Pilot Study Minutiae

Group ID	Member IDs	Location	Primary Speaker
G_1	U_1, U_3, U_4	SMR Lab	U_4
G_2	U_2, U_5, U_6	Class C-118	U_2
G_3	U_1, U_2, U_3, U_4	Cafeteria	U_3
G_4	U_2, U_5	SMR Lab	U_5
G_5	U_1, U_2, U_3, U_4	Way to Cafeteria	U_4
G_6	U_1, U_2, U_3	Outdoor Roadside	U_1
G_7	U_4, U_5, U_6	Outdoor Roadside	U_4

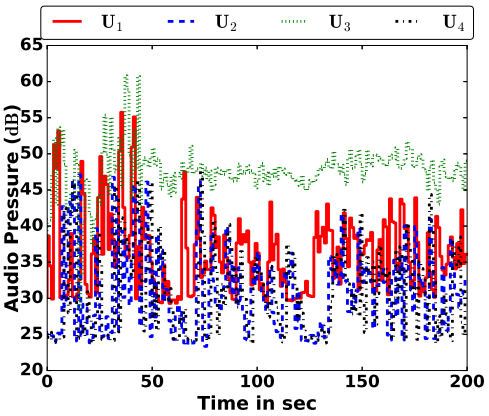


Fig. 3. Impact of audio pressure among the subjects of same group: U_1, U_2, U_3, U_4 in G_3.

based audio processing along with some supervised techniques may pose some additional challenges, although they work well for applications like crowd count [30]. In the following, we explore these challenges from the observations over a pilot experiment.

2.3 Pilot Study: Challenges in Audio Signal Processing

We launched a pilot study to examine the potential of the acoustic context in identifying the meeting groups amidst challenging scenarios. We developed an Android app for collecting the audio signal log from the smartphones. We recruited six volunteers in this experiment for two weeks, installed the app on their smartphones and instructed them to occasionally form pre-designed meeting groups (multiple times) for around $T \geq 15$ minutes. Subjects have been asked to record the group formation instances manually for validation. The detailed overview of the formed groups in this study is listed in Table 1. During the experiments, we have captured 16 bit audio signal at 44.1 kHz sampling rate using smartphones. Notably, while forming those controlled groups, we pay special attention towards incorporating two fundamental challenges – (1) device heterogeneity and (2) noisy environments. To capture device heterogeneity, we have used smartphones from four different makes and models – 2 Moto X, 1 Moto G 2nd Gen, 2 OnePlus3, 1 Samsung Note5. The noise environment can be summarized in the following generic scenarios.

(a) *Low noise environment:* This includes the formation of meeting groups where the surrounding environmental noise is low (audio amplitude ≤ 40 dB [32]). Subjects forming groups inside classrooms (group G_2), while moving inside the laboratory (G_1, G_4) etc. can be categorised as this.

(b) *Noisy environment:* In this scenario, subjects are forming groups in the noisy environment (audio amplitude ≥ 40 dB). Subjects forming groups in cafeterias (group G_3), marketplace (G_5) etc. fall in this scenario.

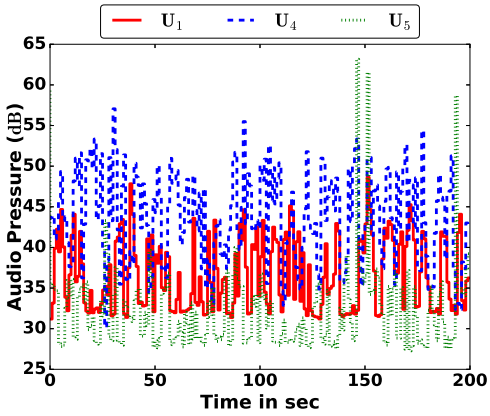


Fig. 4. Impact of audio pressure among the subjects of different groups: U_1, U_4 in G_1 & U_5 in G_4.

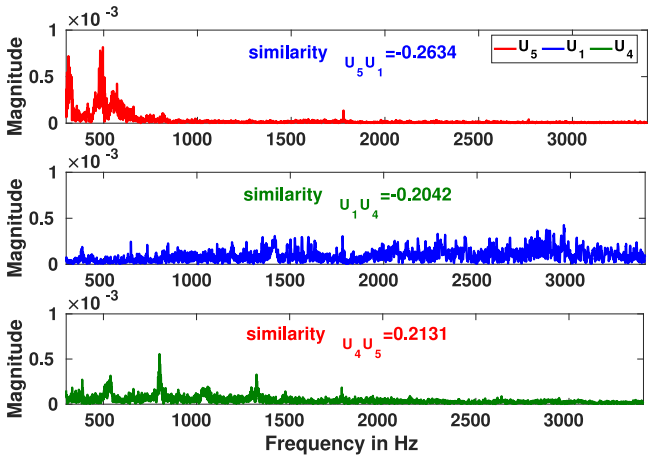
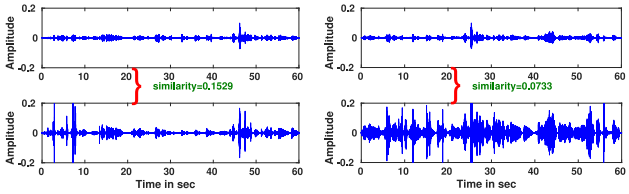


Fig. 5. Deviations of frequencies in groups.

2.3.1 Observations

We first normalize the amplitude of the audio signal and then compute the audio pressure as an indicator of the volume of the audio signal received by individual devices. We concentrate on the meeting group G_3 where subject U_3 primarily speaks while other group members mostly remain silent. In Fig. 3, we plot the audio pressure received from the individual subjects (U_1, U_2, U_3, U_4) of group G_3. We observe that subjects (U_1, U_2, U_4), participating in the same group G_3, exhibit similar audio pressure. However, the audio pressure of U_3 deviates from the rest of the subjects since the user is moving while speaking. Therefore, the values are slightly different than the other group members.

The scenario gets compounded when we consider two groups G_1 (U_1, U_3, U_4) & G_4 (U_2, U_5) which get formed inside the same laboratory during the similar time period. Fig. 4 highlights the fact that although audio pressure of the subjects (U_1, U_4) participating in same group (G_1) exhibit similar behavior, however, the same indicator fails to show clear discrimination between the subjects (say U_1, U_5)



(a) Same Build: U_2 & U_5 in G_2 (b) Different Build: U_2 & U_6 in G_2

Fig. 6. Audio amplitude in same & different builds devices.

TABLE 2
Device to Device Audio Amplitude Cross-Correlation Similarity

Heterogeneous Devices	MotoX	Samsung Note5	OnePlus3	MotoG
Moto X	**0.3247**	0.1178	-0.2781	-0.1138
Samsung Note5	0.1178	**0.2977**	0.0896	0.1287
Oneplus3	-0.2781	0.0896	**0.5671**	-0.0653
Moto G	-0.1138	0.1287	-0.0653	**0.5822**

participating in two different groups (G_1, G_4). For further investigation, we move to the frequency domain based analysis. Importantly, Fig. 5 shows that the frequency component present in the subject U_1 exhibits contrasting behaviour from the subject U_5, belonging to a different group. However, the frequency components of subjects U_1 & U_4 present in the same meeting group (G_1) exhibit (albeit minor) difference (due to environmental noise), posing a new challenge. Last but not the least, Fig. 6 demonstrates the variation of amplitude (raw version of audio pressure) due to device heterogeneity. The smartphone microphones use *automatic gain control* (AGC) circuit, which exaggerates the variation of amplitude for the same audio signal captured through different devices. In group G_2, the subjects U_2 & U_5 carrying same make & model devices whereas another subject U_6 carries a different build. Although all three of them belong to the same meeting group, Fig. 6b exhibits a dissimilarity in amplitude for the subjects U_2 & U_6 (nevertheless, similarity computed using cross-correlation can be observed for subjects U_2 & U_5 (Fig. 6a)). The detailed comparison of the devices, located within the same group, is listed in Table 2, depicting the cross-correlation similarity index of the captured audio signals, generated from a single source. We observe that the cross-correlation similarity index is sometimes quite low for two different devices from two different makes and models.

2.3.2 Lessons Learnt

We observe that audio signals provide us with a good indicator to capture the acoustic context of a group. However, due to omnidirectional nature of smartphone microphones, a significant audio pressure from the speakers of the nearby groups is also getting captured, as we observe in Fig. 4 (formation of G_1 & G_4 in the same lab). Moreover, the physical features of the signal parameters, such as amplitude and frequency components, which represent the time and frequency domain features respectively, fail due to the presence of the device heterogeneity and environmental noise. In summary, although the microphone provides important signature uncovering group membership, however, it is inadequate in its given form for handling all scenarios.

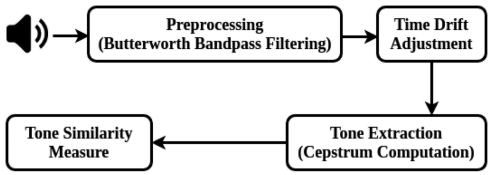
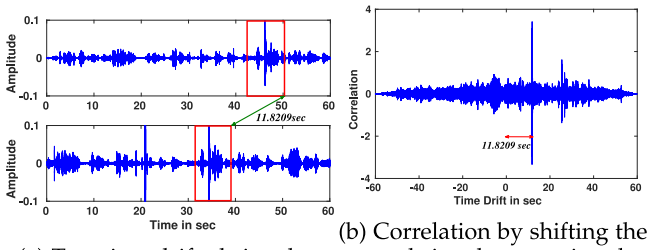


Fig. 7. Audio signal processing flowchart.



(a) Two time drifted signals from same audio source

(b) Correlation by shifting the second signal concerning the first signal

Fig. 8. Computation of time drift.

3 Measuring Acoustic Context of Meeting Groups

From the pilot study, we demonstrate that audio signals are rich sources to capture the context of a meeting group. However, we also comprehend that the naive audio processing techniques are not sufficient to extract reliable information under various complicated scenarios. In this section, we develop a methodology for computing acoustic context from smartphone audio signals, as shown in Fig. 7. The different steps in this procedure are as follows.

3.1 Preprocessing of Vanilla Audio Signals

For audio-based feature extraction, we collect the audio data α_i from all the subjects u_i at a sampling rate of f_s, continuously for t units with an interval of $\hat{t}$ units of time, where t and $\hat{t}$ are specified by the application developers. We first extract the human speech signal between 300 to 3400 Hz using Butterworth bandpass filtering. The human speech signals captured from different smartphones are used for further processing.

3.2 Time Drift Adjustment

The audio signals captured from different devices can be time drifted, even if a single speaker acts as the audio source. There are broadly two reasons for this – (a) the clocks at different devices may not be time synchronized, and (b) the subjects may be at different distances from the speaker, which introduces propagation lag to the signals. Fig. 8a shows the time drifted signals with a single speaker, captured from two different subjects. To compare two signals, we need to place both the signals at the same time reference frame, and therefore eliminating the time drift is an important task for audio processing.

Although some existing studies have developed techniques for time drift adjustment of audio signals captured in hand-held devices [33], they employ smoothing techniques over the raw signal and thus tend to lose the physical properties of the signal, such as tone and pitch of the signal. However, such physical properties are important to capture the nature of human voice, which are essential for extracting

acoustic context. Therefore, we introduce a simple technique in this paper to mitigate the time drift introduced in the signals coming from a single audio source.

To eliminate the time drift, we apply the concept of similarity measure between the signals in the time domain. Consider the audio signal coming from a single source, but captured at two different devices. Ideally, when both the signals are placed at the same reference frame, at the time domain (considering drift as zero), the similarity between them should be maximum. To measure the similarity between the signals, we use statistical correlation. The procedure works as follows. We fix one signal as the reference, and then shift another signal for the one-time unit at every step, and measure the correlation between the signals. Fig. 8b plots the correlation between the signals shown in Fig. 8a, concerning the amount of time shift applied over the second signal, while considering the first signal as the reference. A positive time shift indicates that the second signal has been shifted towards the time clock, and a negative time shift represents that the signal has been shifted backwards the time clock. In the example, we observe that the correlation is maximum when the second signal is shifted 11.8209 seconds, indicating that the drift is 11.8209 seconds. Once the drift is calculated, one signal is shifted to make the drift zero concerning the reference signal.

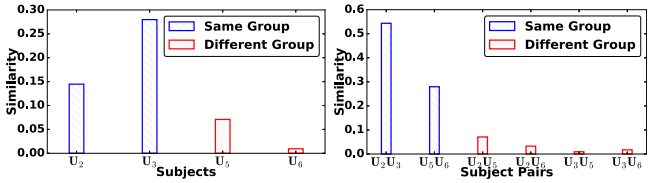
3.3 Audio Tone Extraction

The audio tone of the members of a meeting group should exhibit high similarity among themselves whereas tone dissimilarity indicates different groups. Hence, pairwise tone similarity between the group members should be an important property to determine the acoustic context of that group. Considering that group participants, in general, avoid talking simultaneously in a meeting, intuitively, there exists one dominating tone that gets captured at the smartphones of all the subjects in a meeting group. Audio tone extraction is a well-studied problem [29], [30] and *Mel-frequency cepstral coefficients* based techniques [29] are widely applied for tone extraction from audio signals. However, we face the following challenges while extracting the tone from smartphone audio signals. (a) Smartphone microphones are omnidirectional, and they capture environmental noise along with the human voice. Moreover, the devices are heterogeneous. MFCC fails in the face of the noisy environment and with device heterogeneity [34]. (b) The device heterogeneity is in general handled through various energy-based normalization techniques [28], [30], however they fail for smartphone microphones due to the nonlinearity gain of amplifiers and the presence of *automatic gain control* circuits.[5] (c) As MFCC mostly follows a supervised scheme, the approach may require the voice samples from each user for the correct identification of pitch and tone. However, most of the members in the instantaneous groups are new and appear for the first time. Hence, pretraining is impossible in most of the scenarios.

In this paper, we apply *Complex Cepstrum* (CCEP) to perform tone extraction. CCEP of a signal $\mathcal{S}$ is computed as

$$\text{CCEP}(\mathcal{S}) = \text{IFT}\ (\log(\text{FT}(\mathcal{S})) + j2\pi\ell), \tag{1}$$

5. https://www.fabathome.org/best-smartphone-microphone/ (last accessed Apr 12, 2018)



(a) Single Speaker Multiple Groups Scenario Similarity with Speaker $\mathbf{U_1}$

(b) Multiple Speaker Multiple Groups Scenario Similarity among non-Speaker

Fig. 9. Audio similarity variation in different scenarios.

where FT(.) is the Fourier transform, IFT(.) is the inverse Fourier transform and $j = \sqrt{-1}$. The imaginary part uses complex logarithmic function, and ℓ is an integer which is required to properly unwrap the imaginary part of the complex log function. CCEP uses the log compression of the power spectrum, and therefore is less affected by the environmental noise, the nonlinearity of amplifiers and the effect of AGC circuits. To extract the tone from an audio signal, we segment the signal into one second units, and then compute the CCEP for the audio segments. The CCEP for segment $\bar{t}$ from subject u_i is denoted as $cep_i^{\bar{t}}$. These CCEP values for all the subjects are then used for tone similarity measure, as discussed next.

3.4 Computing Acoustic Context Feature C_{ij}^t

We compute cross-correlation between the CCEP values to measure tone similarity, thereby high and low cross-correlation indicates similar and dissimilar acoustic context between the pair of subjects, respectively. Let $cep_i^{\bar{t}}$ and $cep_j^{\bar{t}}$ denote the CCEP for segment $\bar{t}$ from two different subjects u_i and u_j. We compute the segment wise cross-correlation between $cep_i^{\bar{t}}$ and $cep_j^{\bar{t}}$ as $cor_{ij}^{\bar{t}}$, and then average it over the time span t. This audio cepstrum cross-correlation is used as the acoustic context similarity C_{ij}^t for subject pair u_i and u_j during time duration t.

In order to demonstrate the role of tone similarity to compute acoustic context of meeting groups, we consider the groups G_6 & G_7 formed in outdoor roadside as shown in Fig. 2. Subjects U_1, U_2, U_3 & U_4, U_5, U_6 form the group G_6 & G_7, respectively. In the first scenario, subject U_1 in group G_6 is the dominating speaker, whereas members of group G_7 are mostly silent. Fig. 9a shows the pairwise context similarity between the individual subject and the dominating speaker U_1 (of group G_6). We observe that subjects in group G_6 (say, U_2 & U_3) exhibit higher similarity with dominating speaker U_1 as compared with the members of the group G_7 (say, subjects in U_5 & U_6). Next, we consider two dominating speakers U_1 & U_4 in two respective groups G_6 & G_7. We compute the context similarity between any pair of (non-speaking) subjects. In Fig. 9b we observe that members belonging to the same group (say U_2 & U_3 in G_6 and U_5 & U_6 in G_7) exhibit higher context similarity compared to non-group pairs. Precisely, the context similarity between the intragroup members is substantially higher (close to 1.0) than the intergroup members (close to 0.0). This result indicates that acoustic context within a single group exhibits substantial similarity.

We also investigate the impact of the position of a subject on her acoustic context. We set up two groups G_6 & G_7, 18m apart in the outdoor environment, with two dominating

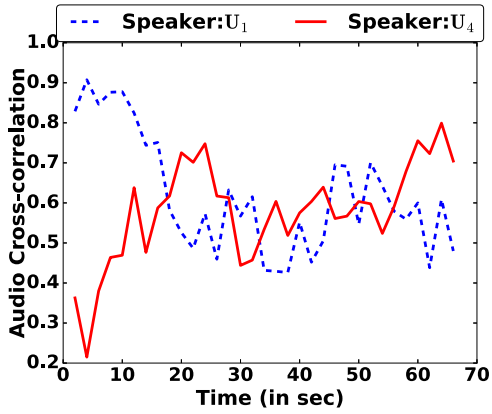


Fig. 10. Audio cross-correlation variation over time with the moving U_2.

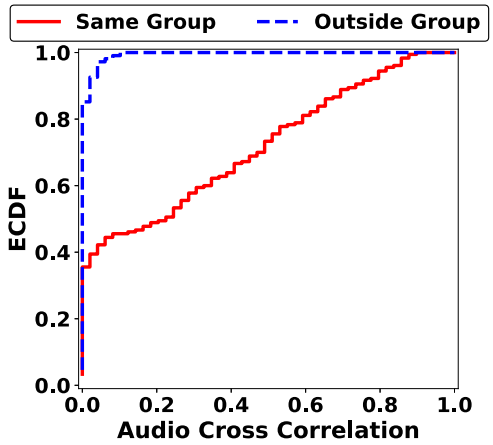


Fig. 11. ECDF of cepstrum cross-correlation.

speakers namely U_1 & U_4 respectively. We consider one moving subject U_2, initially inside the G_6 (from Table 1) and walks towards group G_7 (it takes around 66 sec to reach group G_7 from G_6). Fig. 10 shows the variation in the acoustic context similarity between the subject U_2 and the dominating speaker over time. We observe that, when the subject U_2 is in group G_6, the context similarity between the U_1 and U_2 is high as compared with the U_4. The reverse behaviour is noticed at the end of the experiment when the subject U_2 reaches G_7. However, the context is confusing as the subject located in the middle of both the groups.

Finally, we perform an overall evaluation, considering all the meeting groups formed in the pilot study. In Fig. 11, we plot the empirical cumulative distribution (ECDF) of the acoustic context similarity for the pair of subjects. We observe that the acoustic context similarity in between the intra-group subject pair is high whereas that in between the inter-group subject pair is low. This establishes the fact that tone similarity, computed from cepstrum cross-correlation, reflects the acoustic context of a group and more importantly within a single group.

The aforesaid methodology of extracting acoustic context from smartphone microphone has three broad advantages. First, as the feature is extracted from the dominating tone in an audio signal (captured by cepstrum), it is sufficient if at least one subject in a group talks for a duration. Hence, this method can be able to detect meeting groups where most of the group members do not prefer to interact (consider a conference presentation). Second, the proposed model is unsupervised. Hence, there is no need for pre-training of the tone information of the group members, enabling the method to detect instantaneous groups.

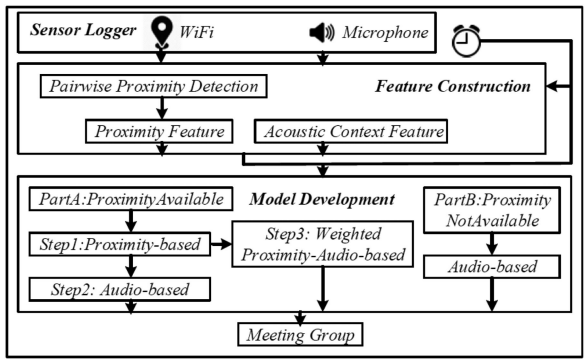


Fig. 12. *GroupSense* model.

4 Design of *GroupSense*

GroupSense is an unsupervised framework for detecting meeting groups leveraging on a subject's proximity and acoustic context. Fig. 12 shows the flow outline of the *GroupSense* framework. First, the sensor logger module records the microphone data along with the proximity indicators followed by the pairwise feature computation.

4.1 Feature Construction

In this module, we first compute the acoustic context similarity $\mathcal{C}_{ij}^t$ between the subject pair u_i & u_j at time t from the collected microphone log (following Section 3). The pairwise proximity similarity feature $\mathcal{F}_{ij}^t$ at time t can be extracted from any of the state of the art techniques [8], [16]. Now considering a subject pair u_i & u_j, we need to compute the aggregated features $\overline{\mathcal{F}_{ij}}$ & $\overline{\mathcal{C}_{ij}}$ respectively for time duration T. One simplest way of aggregation is computing the mean $\overline{\mathcal{F}_{ij}}$ & $\overline{\mathcal{C}_{ij}}$ from the instantaneous features $\mathcal{F}_{ij}^t$ & $\mathcal{C}_{ij}^t$ respectively for time duration T. However, the signal sample collected from the proximity indicator and microphone may suffer from sensitivity and fluctuations. Additionally, the audio signals can also get muffled by obstacles, clothing materials, and are also impacted by the interference. The colluded mean features, computed from all the feature points $\mathcal{F}_{ij}^t$ & $\mathcal{C}_{ij}^t$ for the time duration T, may not provide a clear indication of feature similarities between the subject pair u_i & u_j. Hence, we compute the refined mean features $\overline{\mathcal{F}_{ij}}$ & $\overline{\mathcal{C}_{ij}}$, by eliminating the low-frequency noise component. Here we split all the features points (say, for proximity feature $\mathcal{F}_{ij}^t$) into two clusters (via k-means clustering, with $p_{value} < 0.05$). Eliminating the minor cluster as the noisy component, we compute the mean $\overline{\mathcal{F}_{ij}}$ from the feature points in the major cluster (see Algorithm 1). However, in case of $p_{value} \geq 0.05$, we compute the mean $\overline{\mathcal{F}_{ij}}$ considering all the feature points in the single cluster. Similarly, we compute the refined mean acoustic context feature $\overline{\mathcal{C}_{ij}}$ from Algorithm 1.

4.2 Model Development

Finally, leveraging on the aforementioned features, we develop an unsupervised model for meeting group detection. We denote the population with proximity information

as $\mathbb{U}_o$ and the population without any proximity information as $\mathbb{U}_{no}$. The model executes Part *A* (Algorithm 3) or Part *B* (Algorithm 4) depending on the availability of the proximity information. If the subject possesses proximity information, the model exploits both proximities as well as acoustic features in Part *A*. Otherwise, the model only relies on the acoustic information in Part *B*. The outcome of the model is all the meeting groups detected by both the individual parts. The model outline is described in Algorithm 2.

Algorithm 1. Feature_Construction

Inputs: $\mathbb{F}_{ij}, T$
Output: $\overline{\mathcal{F}_{ij}}$
$[Cl_1, Cl_2] \leftarrow kmeans((\mathbb{F}_{ij}, T), 2)$
if $p_{value} > 0.05$ **then** ▷ *Single Cluster Scenario*
  $\overline{\mathcal{F}_{ij}} \leftarrow (1/|\mathbb{F}_{ij}|)(\sum_{\forall \mathcal{F}^t_{ij}} \mathcal{F}^t_{ij})$
else
  if $|Cl_1| > |Cl_2|$ **then** ▷ *Major Cluster Cl_1 Scenario*
    $\overline{\mathcal{F}_{ij}} \leftarrow (1/|\mathbb{F}_{ij}|)(\sum_{\forall \mathcal{F}^t_{ij} \in Cl_1} \mathcal{F}^t_{ij})$
  else ▷ *Major Cluster Cl_2 Scenario*
    $\overline{\mathcal{F}_{ij}} \leftarrow (1/|\mathbb{F}_{ij}|)(\sum_{\forall \mathcal{F}^t_{ij} \in Cl_2} \mathcal{F}^t_{ij})$
  end if
end if

Algorithm 2. GroupSense: Group_Dection_Algorithm

Inputs: $u_i(p^t_i, \alpha^t_i) \forall u_i \in \mathbb{U}, \delta_{p_1}, \delta_{p_2}, \delta_\alpha$
Output: $\mathbb{G}^T$
$\mathbb{U}_o \leftarrow \emptyset, \mathbb{U}_{no} \leftarrow \emptyset$
if $p^t_i \neq \emptyset$ **then**
  $\mathbb{U}_o \leftarrow \mathbb{U}_o \cup u_i$
end if
$\mathbb{U}_{no} \leftarrow \mathbb{U} - \mathbb{U}_o$
if $\mathbb{U}_o \neq \emptyset$ **then** ▷ *Proximity Available Scenario*
  $\mathbb{G}^T_o \leftarrow$ ProximityAvailable_Function $(u_i(p^t_i, \alpha^t_i) \forall u_i \in \mathbb{U}_o, \delta_{p_1}, \delta_{p_2}, \delta_\alpha)$
end if
if $\mathbb{U}_{no} \neq \emptyset$ **then** ▷ *Proximity Not Available Scenario*
  $\mathbb{G}^T_{no} \leftarrow$ ProximityNotAvailable_Function $(u_i(\alpha^t_i) \forall u_i \in \mathbb{U}_{no}, \delta_\alpha)$
end if
$\mathbb{G}^T \leftarrow \mathbb{G}^T_o \cup \mathbb{G}^T_{no}$

Part A. In this part, we first attempt to extract the cluster of co-locating subjects only based on the pair-wise proximity similarity. If we identify a *highly cohesive cluster* $\mathbb{G}^T_p$ based on proximity only, we consider $\mathbb{G}^T_p$ as a potential meeting group and execute the second step. In the second step, we leverage only on the acoustic context features to detect meeting group(s) $\mathbb{G}^T_\alpha$ from the identified proximity clusters $\mathbb{G}^T_p$. On the other hand, if we identify *moderately cohesive cluster* $\mathbb{G}^T_p$ from the proximity features, the model abandons the cluster $\mathbb{G}^T_p$, considering proximity as a critical albeit weak signal, and moves to the third step. In the third step, we combine both the proximity and acoustic context similarity features together to detect cohesive cluster $\mathbb{G}^T_w$ on the complete population $\mathbb{U}_o$ (where proximity information is available). If $\mathbb{G}^T_w$ exhibits high cohesivity, we assert the cluster $\mathbb{G}^T_w$ as the meeting group. Poor cohesivity in any step rejects the existence of any group in the population. The overall procedure is illustrated in Algorithm 3.

Algorithm 3. ProximityAvailable_Function

Inputs: $u_i(p^t_i, \alpha^t_i) \forall u_i \in \mathbb{U}_o, \delta_{p_1}, \delta_{p_2}, \delta_\alpha$
Output: $\mathbb{G}^T$
Compute $\mathcal{F}^t_{ij}, \mathcal{C}^t_{ij}$ ▷ *Feature Generation*
$\overline{\mathcal{F}_{ij}} \leftarrow$ Feature_Construction $(\mathbb{F}_{ij}, T) \forall u_i, u_j$
$(\mathbb{G}^T_p, \mathcal{M}_p) \leftarrow$ Community_Detection $(\mathbb{U}_o, \overline{\mathbb{F}})$
if $\mathcal{M}_p \geq \delta_{p_1}$ **then** ▷ *Proximity Dominating Scenario*
  $\overline{\mathcal{C}_{ij}} \leftarrow$ Feature_Construction $(\mathbb{C}_{ij}, T) \forall u_i, u_j \in \mathcal{G}, \mathcal{G} \in \mathbb{G}^T_p$
  $(\mathbb{G}^T_\alpha, \mathbb{M}_\alpha) \leftarrow$ Community_Detection $(\mathbb{G}^T_p, \overline{\mathbb{C}})$
  if $\mathcal{M}_\alpha \geq \delta_\alpha \forall (\mathcal{G}_\alpha, \mathcal{M}_\alpha) \in (\mathbb{G}^T_\alpha, \mathbb{M}_\alpha)$ **then** ▷ *Proximity & Audio Influence Scenario*
    $\mathbb{G}^T \leftarrow \mathbb{G}^T \cup \mathcal{G}_\alpha$
  else ▷ *Proximity Influence & Audio Insignificance Scenario*
    Failure
  end if
else
  if $\mathcal{M}_p < \delta_{p_1} \& \mathcal{M}_p \geq \delta_{p_2}$ **then**
    $\overline{\mathcal{C}_{ij}} \leftarrow$ Feature_Construction $(\mathbb{C}_{ij}, T) \forall u_i, u_j$
    $(\mathbb{G}^T_w, \mathcal{M}_w) \leftarrow$ Community_Detection $(\mathbb{U}_o, (1-w) \times \overline{\mathbb{F}} + w \times \overline{\mathbb{C}}) \forall w \in [0, 1]$ ▷ *Weighted Features*
    $(\mathbb{G}^T_w, \mathcal{M}) \leftarrow max(\mathbb{G}^T_w, \mathcal{M}_w) \forall w \in [0, 1]$
    if $\mathcal{M} \geq \delta_\alpha$ **then** ▷ *Proximity Confused & Audio Influence Scenario*
      $\mathbb{G}^T \leftarrow \mathbb{G}^T_w$
    else ▷ *Proximity Confused & Audio Insignificance Scenario*
      Failure
    end if
  else ▷ *Proximity Insignificance Scenario*
    Failure
  end if
end if

Detection of Cohesive Cluster. Consider a weighted network $\mathcal{CG}(\mathbb{U}, \mathbb{E})$, where $u_i \in \mathbb{U}$ is a subject and $\{e_{ij}, w^e_{ij}\} \in \mathbb{E}$ denotes the weighted link e_{ij} between the subject pair u_i & u_j. We apply community detection algorithm [35] on $\mathcal{CG}$ to obtain a partition $\mathcal{K} = \{K_1, K_2, \ldots, K_m\}$ on population $\mathbb{U}$. Essentially the community detection algorithm partitions the network into communities ensuring dense connections within a community and sparser connections between communities. We consider the detected community K_i as a cluster in population $\mathbb{U}$. The cohesivity of the partition $\mathcal{K}$ can be measured with modularity index $\mathcal{M}$ as $\mathcal{M} = \frac{1}{4\varphi}\sum_{ij}(w^e_{ij} - \frac{\rho_i \rho_j}{2\varphi}) f(\sigma_i, \sigma_j)$, which reflects the fraction of the links that fall within a given community, compared to the expected fraction if links are distributed at random [35]. The φ, ρ_i, σ_i, & $f(.)$ represent the sum of all of the edge weights in the network, sum of the edge weight attached to node u_i, the community of node u_i, and delta function, respectively. Notably, modularity of a weighed fully connected graph becomes *zero* if all the nodes form a single community [36]. In this paper, we apply Walktrap community detection algorithm [17]; however, our methodology is not sensitive to any specific (weighted) community detection algorithm. Algorithm 3 comprises the following three steps.

Step 1. We construct a complete proximity graph $\mathcal{PG}(\mathbb{U}_o, \overline{\mathbb{F}})$ where $\mathbb{U}_o$ denotes the population with proximity information and $\{e_{ij}, \overline{\mathcal{F}_{ij}}\} \in \overline{\mathbb{F}}$ is a link between the subject pair u_i & u_j weighted by the proximity feature $\overline{\mathcal{F}_{ij}}$ computed over the time T. We apply the community detection algorithm on the proximity graph $\mathcal{PG}$ to discover the cluster

$\mathbb{G}_p^T$ with modularity $\mathcal{M}_p$. If the $\mathcal{M}_p$ is above a threshold δ_{p_1}, we consider $\mathbb{G}_p^T$ as the candidate meeting group and move to step 2. If $\mathcal{M}_p$ falls below a threshold of δ_{p_2}, we reject the existence of any meeting group in population $\mathbb{U}_o$. Otherwise, we move to step 3. It is pertinent that the proposed model uses the thresholding scheme not for group formation detection, rather for checking the quality of the detected groups based on modularity.

Step 2. We construct the complete acoustic context graphs $\mathcal{IG}(\mathbb{G}_p^T, \overline{\mathbb{C}})$ where $\{e_{ij}, \overline{\mathcal{C}_{ij}}\} \in \overline{\mathbb{C}}$ links between subject pair u_i & $u_j \in \mathbb{G}_p^T$. Essentially, in $\mathcal{IG}$, the link weight $\overline{\mathcal{C}_{ij}}$ depicts the acoustic context similarity over the time T. Similar to step 1, we apply the community detection on $\mathcal{IG}$ to discover the cluster $\mathbb{G}_\alpha^T$ with modularity $\mathcal{M}_\alpha$. If the $\mathcal{M}_\alpha$ is above a threshold δ_α, we confirm $\mathbb{G}_\alpha^T$ as the detected meeting groups. Otherwise, we reject the existence of meeting groups in population $\mathbb{U}_o$.

Step 3. We construct a complete proximity-acoustic context graph $\mathcal{MG}(\mathbb{U}_o, \mathbb{W})$ where $\{e_{ij}, \mathcal{W}_{ij}\} \in \mathbb{W}$ links between subject pair u_i & $u_j \in \mathbb{U}_o$ weighted by $\mathcal{W}_{ij} = (1-w) \times \overline{\mathcal{F}_{ij}} + w \times \overline{\mathcal{C}_{ij}}$. Essentially, in $\mathcal{MG}$, the link weight $\mathcal{W}_{ij}$ carries the information from both acoustic context and proximity feature. Similar to step 1, we apply the community detection on $\mathcal{MG}$ to discover the cluster $\mathbb{G}_w^T$ with modularity $\mathcal{M}_w$. If the $\mathcal{M}_w$ is above a threshold δ_α, we confirm $\mathbb{G}_w^T$ as the detected meeting groups. Otherwise, we reject the presence of any group in population $\mathbb{U}_o$.

Part B. Due to the unavailability of proximity data, in this part, we completely rely on the acoustic context similarity between the subjects. We first construct a complete acoustic context graph $\mathcal{IG}(\mathbb{U}_{no}, \overline{\mathbb{C}})$ where $\{e_{ij}, \overline{\mathcal{C}_{ij}}\} \in \overline{\mathbb{C}}$ links between subject pair u_i & $u_j \in \mathbb{U}_{no}$. Essentially, in $\mathcal{IG}$, the link weight $\overline{\mathcal{C}_{ij}}$ carries the information of the acoustic context feature over the time T. Similar to part *A*, we apply the community detection on $\mathcal{IG}$ to discover the clusters $\mathbb{G}_\alpha^T$ with modularity $\mathcal{M}_\alpha$. If the $\mathcal{M}_\alpha$ is above a threshold δ_α, we confirm $\mathbb{G}_\alpha^T$ as the detected meeting groups. Otherwise, we reject the existence of meeting groups in population $\mathbb{U}_{no}$. The outline of the mechanism is portrayed in Algorithm 4.

Algorithm 4. ProximityNotAvailable_Function

Inputs: $u_i(\alpha_i^t) \forall u_i \in \mathbb{U}_{no}, \delta_\alpha$
Output: $\mathbb{G}^T$
1: Compute $\mathcal{C}_{ij}^t$ ▷ *Feature Generation*
2: $\overline{\mathcal{C}_{ij}} \leftarrow$ Feature_Construction $(\mathbb{C}_{ij}, T)\ \forall u_i, u_j$
3: $(\mathbb{G}_\alpha^T, \mathcal{M}_\alpha) \leftarrow$ Community_Detection$(\mathbb{U}_{no}, \overline{\mathbb{C}})$
4: **if** $\mathcal{M}_\alpha \geq \delta_\alpha$ **then** ▷ *Audio Influence Scenario*
5: $\mathbb{G}^T \leftarrow \mathbb{G}_\alpha^T$
6: **else** ▷ *Audio Insignificance Scenario*
7: Failure
8: **end if**

5 Performance Evaluation

We evaluate *GroupSense* by developing a smartphone-based application and deploying it over IIT Kharagpur campus spreading 8.5 square kilometres, consisting of administrative blocks, approximately 30 academic departments along with campus residential, hostels and market areas. We first illustrate the system implementation followed by the field study and performance comparison with different baselines.

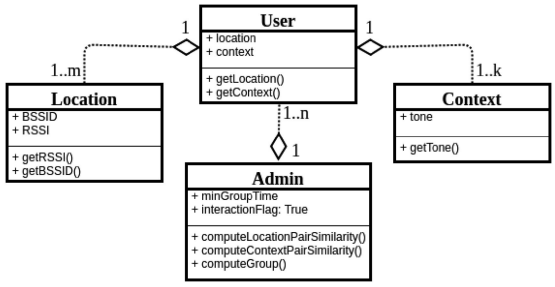


Fig. 13. Class diagram of the developed system.

5.1 System Implementation and Field Study

The *DataGatherer* application consists of a software suite running on the smartphones and a back-end infrastructure hosted on the central server of the laboratory. The software installed on the phones primarily senses the raw signals and transfers that signal to the back-end server. The smartphone app mainly comprises four software components – *location listener*, *audio listener*, *local storage manager* and *upload manager*. Considering the impact of battery consumption on the smartphone, we rely on the interval-based sensing [37] instead of continuous sensing. With interval-based sensing, we capture the data continuously for a period followed by a gap-interval. Following this mechanism, the *location listener* senses the WiFi APs along with the signal strength once in a minute and stores the details of the APs having the signal strength more than -80 dBm (minimum signal strength for basic connectivity[6]). On the other hand, the continuous audio signal is captured at a sampling rate of 44.1 kHz for one-minute time span followed by a gap-interval of three minutes. The selection of the gap-interval is strictly application-dependent. The determination of the trade-off between the gap-interval and system performance is out of the current scope of the work, which we plan to explore as part of future work. In addition to the interval-based sensing, the battery consumption can be further reduced by applying the dynamic sensing mechanism, where the silent detector module (or reverse of speech/ conversation detector module) [38] can be used to identify the non-speech zone of the audio signal and act as an indicator to efficiently store the speech content of the audio signal. The raw sensed data is stored temporarily in the local storage and finally uploaded to the back-end server once in a day through the *upload manager*, which acquires negligible battery consumption. The app is designed by following a threaded architecture for ensuring that any partial failure of a component does not affect other components. On the back-end, Apache server runs for the communication with the storage server to store the collected sensing data. For processing the data, a system comprising proximity module and GroupSense is developed on the central processing server using python. The codes are written by considering four major classes (Fig. 13) such as `user`, `location`, `context`, and `admin`. Each user has a location and a context at a specific timestamp, whereas with time the location and context may vary. The `location` and `context` classes are responsible for capturing the raw sensing data – BSSID, RSSI and tone. The `user` class transfers the location and the context

6. https://support.metageek.com/hc/en-us/articles/201955754-Understanding-WiFi-Signal-Strength (Accessed on Apr 12, 2018)

information to the admin. Finally, the admin class computes the pairwise similarity to obtain the meeting groups.

Data Collection. The Android app, *DataGatherer*, has been launched over the smartphones of 40 subjects consisting of undergraduate and postgraduate students, summer interns, research scholars and faculties of the institute. In our implementation of *GroupSense*, we have considered $T \geq 15$ mins; if an interaction continues for at least 15 minutes, we consider it as a group. Nevertheless, this is an application specific tunable parameter. We have used different models of smartphones, where costs per phone range from USD 150$ to USD 700$ approximately. The data has been collected for approximately seven months.[7] We collect the ground truth meeting group information from the participants for validation. In ground truth data collection, a questionnaire app periodically probes from the participants regarding the (a) start time of the meeting, (b) end time of the meeting, (c) meeting venue and (d) details of the other participants of the meeting. In some cases where a participant misses to provide the ground truth information, we validate the detected meeting groups from the participants by forwarding an email at every two hours each day. Based on the field study collected data, we identify 12 typical meeting group scenarios, which repeatedly occurred (at least once a week) during the seven months of field study. These scenarios are highlighted keeping in mind the critical conditions of group formation that were realized during the pilot study (Section 2). We evaluate the performance of *GroupSense* and compare it with other baselines considering these typical scenarios, as well as the other scenarios observed from the collected data. These scenarios are as follows.

S1 (Indoor: Two Groups at Neighbouring Rooms). 3 subjects attend a lecture in classroom C-119, and 2 subjects have another meeting in the FV Lab opposite to C-119 at the same instance of time.

S2 (Indoor: Three Groups at Different Rooms of the Same Department). 4 subjects interact in the faculty office in the second floor, 2 subjects are in a meeting at the departmental library opposite to that faculty office, and 2 subjects are in another meeting at the lab in the first floor.

S3 (Outdoor: Cafeteria Interactions). Two different groups at the cafeteria, one with 3 subjects in front of the cafeteria and another one with 3 subjects at the back of the cafeteria.

S4 (Indoor: Large Single Group). 7 subjects attend a presentation at the departmental conference room.

S5 (Indoor: Two Different Groups at a Large Lab). 3 subjects meet at cubicle K-1 and another 3 subjects meet in the cubicle K-10 of the lab.

S6 (Indoor: Two Roaming Groups). 3 subjects together and 2 subjects together roam around the corridor of the department, and move from one room to another, forming two indoor moving groups.

S7 (Outdoor: Two Roaming Groups). 5 subjects together and 2 subjects together roam within the campus maintaining a certain distance from each other, forming two outdoor moving groups.

S8 (Indoor: One Formal Group and One Informal Group in a Room). 4 subjects meet together for a formal discussion, and 4 subjects talk loudly side by side in a room.

S9 (Indoor: Two Frequent Crossing Groups). 3 subjects together and another 3 subjects together walking separately for a discussion at the corridor but those two groups encounter each other frequently.

S10 (Indoor: Three Groups in a Lab at the Department). 3 subjects meet at cubicle K-1, another 3 subjects meet near the centre, and 3 subjects meet in the cubicle K-10 of the lab.

S11 (Indoor: Two Group on Different Floors of the Department). 3 subjects together and another 3 subjects together walking at the first and second floors respectively.

S12[8] *(Indoor & Outdoor: One Indoor Group and Another Outdoor Group)*. 2 subjects enjoy social celebration inside the room, and another 2 subjects conduct informal meeting just outside the room.

5.2 Proximity Computation for Group Detection

In *GroupSense*, we implement the existing proximity detection mechanisms that have been well studied in the literature. We focus on the two approaches of proximity detection based on WiFi data, as follows.

(a) *Supervised learning with WiFi-based proximity sensing (SLWP) [8]:* Sapiezynski et al. developed a WiFi access point based *supervised* proximity detection mechanism, where Bluetooth data is considered as the ground truth. In this approach, a set of WiFi-based features has been computed, such as overlapping access points, signal strength from different access points etc., and then a support vector machine (SVM) is used to classify whether two subjects are in proximity or not.

(b) *Next2Me* [16]: This is a smartphone-based unsupervised approach for capturing the social interactions within close proximity users. Next2Me uses WiFi signal information for measuring the pairwise co-located Manhattan distance between the users, and then a threshold over the distance function is used to locate the subjects in proximity.

5.3 Baselines for Audio Based Interaction Detection

We have evaluated the performance of *GroupSense* with the following baselines, which utilize audio signals for acoustic context detection. We use the proximity similarity, followed by the acoustic context to detect various meeting groups.

(a) *Next2Me* [16]: After determining the subjects in proximity, Next2Me utilizes Jaccard similarity over top n audio frequencies to capture the audio fingerprints of various subjects. Finally, they generate social community by applying the Louvain community detection algorithm.

(b) *AudioMatch* [10]: Casagranda et al. implemented a smartphone based group detection system based on the joint usage of GPS and audio fingerprints. First, the GPS information is used for filtering the nearby devices. On top of the GPS based clusters, the audio module is executed for identifying the groups. For that, *AudioMatch* implements a short time Fourier transform (STFT) with overlapping Hamming window. Finally, it computes the Hamming distance between the pair of devices for detecting the nearby pairs.

7. The ethical guidelines for human experiments have been followed, and necessary permissions have been obtained.

8. In this experiment, Wi-Fi APs are not available.

TABLE 3
Performance Comparison (The Green & Blue Cells Represent Scenariowise Highest & Lowest F_1-Score)

ID	SLWP						Next2Me					
	Next2Me		GroupSense		AudioMatch		Next2Me		GroupSense		AudioMatch	
	F_1-Score	Modularity	F_1-Score	Modularity	F_1-Score	Modularity	F_1-Score	Modularity	F_1-Score	Modularity	F_1-Score	Modularity
S1	1.0000	0.0412	1.0000	0.2879	0.7273	0.0000	1.0000	0.0412	1.0000	0.2907	0.7273	0.0000
S2	0.9000	0.2030	1.0000	0.1760	0.6667	0.0000	0.9000	0.2030	1.0000	0.1760	0.6667	0.0000
S3	0.5333	0.1261	1.0000	0.3642	0.7273	0.0000	0.5333	0.1261	1.0000	0.3642	0.7273	0.0000
S4	0.8326	0.0772	1.0000	0.0000	1.0000	0.0000	0.8326	0.0772	1.0000	0.0000	1.0000	0.0000
S5	0.8571	0.0732	1.0000	0.3801	0.7273	0.0000	0.8571	0.0732	1.0000	0.3801	0.7273	0.0000
S6	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
S7	0.5833	0.1942	0.6500	0.0976	0.8333	0.0000	0.5833	0.1942	0.6500	0.0976	0.8333	0.0000
S8	0.6519	0.0008	0.6190	0.1859	0.6667	0.0000	0.6519	0.0008	0.6190	0.1859	0.6667	0.0000
S9	0.6250	0.0074	0.8286	0.1640	0.6667	0.0000	0.6250	0.0074	0.8286	0.1640	0.6667	0.0000
S10	0.8036	0.2427	0.8333	0.2775	0.5000	0.0000	0.8036	0.2427	0.8333	0.2775	0.5000	0.0000
S11	0.8286	0.0913	1.0000	0.4255	0.6667	0.0000	0.8286	0.0913	1.0000	0.4255	0.6667	0.0000
S12	0.8889	0.1389	1.0000	0.1489	0.7778	0.0000	0.8889	0.1389	1.0000	0.1489	0.7778	0.0000
ALL	0.8119 ± 0.16	0.0971	**0.9296 ($\pm$ 0.14)**	**0.2122**	0.7846 ($\pm$ 0.14)	0.0000	0.7969 ($\pm$ 0.16)	0.0898	**0.9296 ($\pm$ 0.14)**	**0.2123**	0.7846 ($\pm$ 0.14)	0.0000

Next, we discuss the experimental procedure by combing the WiFi-based proximity detection and audio based acoustic context detection together.

5.4 Experimental Procedure

For evaluating the performance of *GroupSense* under various circumstances, we consider the following strategy to combine proximity (P) and acoustic context (I) detection mechanisms. First, we compute the pairwise proximity (P) following the state of the art algorithms SLWP and Next2Me. Next, we combine the pairwise proximity similarity with the acoustic context detection techniques following the framework developed in Algorithm 2. In acoustic context detection (I), we implement Next2Me and AudioMatch as baselines to compare the performance with *GroupSense*. We leverage on the system implementation described in Section 5.1 to collect the required data for the baselines. The different combinations of proximity and acoustic context detection mechanisms used in our experiments are as follows.

I. *SLWP (P) + Next2Me (I)*: In this arrangement, we extract the pairwise proximity information from *SLWP*, and the outcome is directly fed to the *Next2Me* audio model for group detection.
II. *SLWP (P) + GroupSense (I)*: This arrangement uses the pairwise proximity information from *SLWP*. Then, *GroupSense* audio centric context detection is applied on top of the proximity outcome.
III. *SLWP (P) + AudioMatch (I)*: In this arrangement, we apply *SLWP* for pairwise proximity detection. After that, *AudioMatch* is applied to the outcome of the proximity clusters for detecting the pairwise acoustic context from the audio signals. Notably, we have not used GPS for proximity detection (as used in AudioMatch), since GPS gives a weak signal in the indoor scenarios. However, the audio module is implemented following *AudioMatch*, followed by the community detection algorithm to detect groups.
IV. *Next2Me (P) + Next2Me (I)*: This arrangement is analogous to the vanilla *Next2Me* system [16], where both the WiFi based proximity detection and the audio based acoustic context detection are used for group detection.
V. *Next2Me (P) + GroupSense (I)*: In this arrangement, we compute the proximity-based pairwise distance following *Next2Me* proximity module. The pairwise similarity is computed by reversing the pairwise distance value. Then, we apply *GroupSense* Feature Construction Algorithm 1 followed by community detection module on the pairwise similarity value. Finally, *GroupSense* acoustic context module is employed on top of the proximity outcome.
VI. *Next2Me (P) + AudioMatch (I)*: This arrangement uses the proximity information from *Next2Me* like the previous setup. After that, *AudioMatch* is applied to the outcome of the proximity clusters. The pairwise acoustic context information is finally fed to the community detection algorithm for meeting group detection.

5.5 *GroupSense* Performance

We first evaluate the overall performance of *GroupSense* in terms of F_1-Score defined as follows. Let Γ and Υ be the sets of meeting groups in the ground truth data and the ones detected by *GroupSense*, respectively. Then $F_{1_{\kappa\nu}} = \frac{2\times|\kappa\cap\nu|}{|\kappa|+|\nu|}$ where $\kappa \in \Gamma$ and $\nu \in \Upsilon$. This parameter captures the accuracy of the detected group ν in terms of membership overlap with ground truth κ for the meeting duration T. Now, to obtain the final accuracy of *GroupSense* considering all the detected meeting groups, we compute the average F_1-Score as $F_1 = \sum_{\forall\kappa\in\Gamma;\forall\nu\in\Upsilon} F_{1_{\kappa\nu}}/|\Upsilon|$.

Table 3 summarizes the performance of *GroupSense* in terms of F_1-Score and modularity ($\mathcal{M}$) for 12 representative scenarios as well as for all the observed scenarios combined. We fix the model thresholds (δ) based on the best performance obtained for the overall system and consider those thresholds throughout the paper for both *GroupSense* and other baselines. The modularity $\mathcal{M}$ indicates the cohesiveness of the detected groups; hence even a low F_1-Score with high modularity contributes more to identify maximum participants in a meeting group. Although *GroupSense* performs marginally worse for certain scenarios, such as outdoor groups with mobility, due to the high environmental noise, we observe that on an average the system achieves more than 0.9 (± 0.14) F_1-Score.

5.5.1 Baseline Comparison

Table 3 compares the performance of *GroupSense* with *AudioMatch* and *Next2Me* while combined with two different

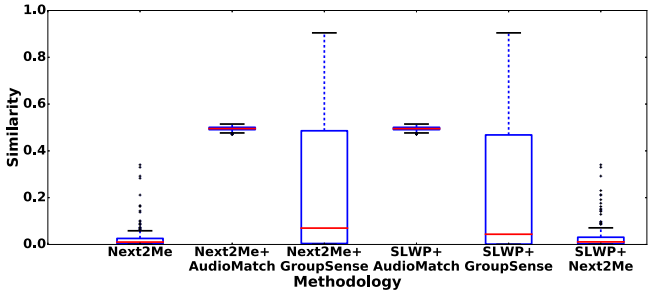
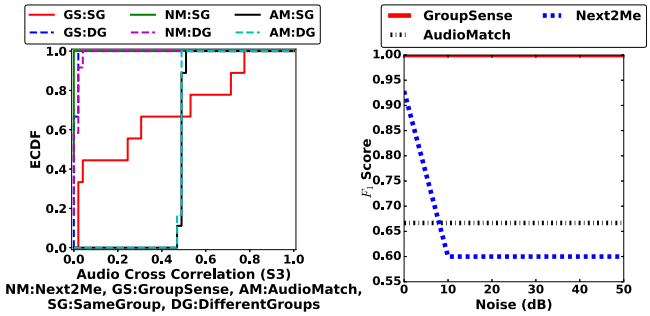


Fig. 14. Difference of similarity in comparison to baselines.



(a) Audio correlation comparison for scenario $S3$

(b) Effect of Noise in *GroupSense* and *Next2Me*

Fig. 15. Performance analysis at different environments.

proximity schemes (*SLWP* & *Next2Me*). For all the three scenarios when *SLWP* is used for proximity measure, we observe that *GroupSense* outperforms the other baselines. Although *Next2Me* uses audio based features to capture the social interaction among the subjects (which is similar to the acoustic context for *GroupSense*), it uses Jaccard similarity among top n audio frequencies, which is susceptible to environmental noise. For example, in an outdoor environment, the sound frequencies originated from external entities, such as moving vehicles, can fall within the top n frequency components. As a consequence, we observe that although *Next2Me* manages to perform well in indoor scenarios, it poorly performs in the outdoor environment. On the other side, *AudioMatch* applies hamming distance measure of the logarithmic amplitude of the audio signal for suppressing the noise component. Although this scheme works for artificially generated Gaussian noise, it poorly performs in the presence of real environmental noise. The impact is visible in the outdoor scenarios. Additionally, we observe that *AudioMatch* produces a single group consisting of all the subjects from the population, resulting in zero modularity.

Next, we have tested the scenarios with three baselines for acoustic context measurement along with *Next2Me* proximity measure. The results of these are similar to the results from the scenarios with *SLWP* proximity measure, except for the proximity dominant scenario $S1$ in *GroupSense*. The similar results also claim that the audio features are more dominant compared to the proximity features for meeting group detection. Moreover, this validates the importance of Algorithm 4 in the meeting group detection mechanism.

5.5.2 Robustness of Acoustic Context Measure

For investigating the variations in the performance of different schemes, we report the box-plot of the pairwise feature similarity values for the acoustic context, shown in Fig. 14. The box plot depicts that there are significant mean differences between the various schemes. In the box plot, the medians for different schemes are shown in red lines. Focusing on the upper and the lower halves from the median, the results show that *GroupSense* captures significant variations in the pairwise similarity between the subjects. As we consider multiple meeting group scenarios, the variation of the pairwise similarity between the subjects is justified. It can be noted that the median is biased towards the lower values because the pairwise feature similarity becomes very close to zero whenever the two subjects in the pair are from different groups. However, a wide variation of similarity values greater than 0.1 is observed when both the subjects are in the same group. On the contrary, although we have considered the same multiple scenarios for the baselines, *Next2Me* and *AudioMatch* show the minimal difference in the upper and the lower halves from the median. Therefore, in the presence of multiple groups, the minimal difference in the lower and upper range of the pairwise similarity implies the incapabilities of the acoustic features used in the baselines for separating out multiple groups in proximity. Hence, the F_1-Score significantly drops for the baselines. Additionally, we also observe that the median value is closer to the first quartile. As we capture the proximity and audio signatures of the subjects in various environments, the similarity values between each pair of subjects significantly varies over the different meeting groups, causing the dense zone towards the lower halves from the median. The wide variation of the pairwise similarity values in different groups further interprets that the simple thresholding based scheme is not suitable for detecting various types of meeting groups in the diverse environment.

5.5.3 Dissecting the Methodologies

Next, we look into the performance of the various competing methodologies by exploring their internals. From the above experiments, we observe that baselines perform poorly for the scenario $S3$. Therefore, we further study the top n frequency based similarity, thresholding based hamming distance measure, and cepstrum similarity for that scenario. From Fig. 15a, we found that audio cross-correlation for 'same group' (target subjects are within a group) and 'different groups' (target subjects are from different groups) pairs of subjects are more distinct in *GroupSense* as compared with the other two baseline methods. The outdoor environment like the cafeteria (scenario $S3$) are noisy due to the presence of the non-member group voice and noise from the environment. As *Next2Me* considers only the top 6 frequency components (as per our implementation $n = 6$), it unknowingly considers those frequencies, resulting in the similar audio correlation values for 'same group' and 'different groups'. On the other side, *AudioMatch* compares the logarithmic amplitude of STFT of the audio signal with its neighbouring points to generate 16-bit fingerprint. Therefore, the 16-bit fingerprint generation completely relies on the centre comparing the amplitude value. If the centre value is corrupted due to the environmental noise, the entire 16-bit fingerprint is prone to be corrupted. Those spurious fingerprints are further used for computing the Hamming

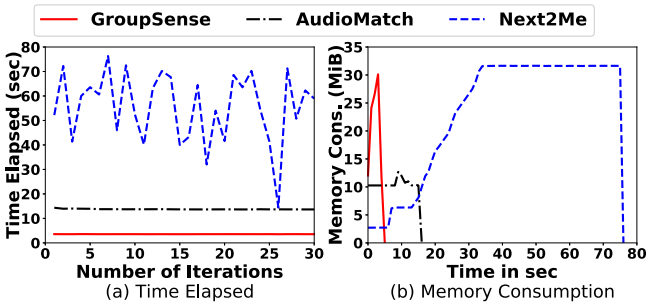


Fig. 16. Performance in terms of computational cost.

distance between the pair of subjects, resulting in identical behaviour for the audio correlation values in 'same group' and 'different groups'. As *GroupSense* considers cepstrum containing the tone information for computing the audio correlation, the correlation values are more close to one for 'same group' and close to zero for 'different groups'. Therefore, the audio features of *GroupSense* can distinguish this scenario. Consequently, although *Next2Me* and *AudioMatch* fail to separate out the groups based on the audio features, *GroupSense* can correctly differentiate the groups.

We further evaluate *Next2Me*, *AudioMatch* and *GroupSense* in varying noise environment. As simulating complete random noise is nearly impossible, we generate continuous Gaussian noise at different levels and superimpose the noise with the captured audio signal. Fig. 15b shows that *GroupSense* is more noise resistant than *Next2Me*, whereas *AudioMatch* is as noise resistant as *GroupSense*. Analogous to noisy environmental scenarios, *Next2Me* performs poorly in the presence of statistically generated Gaussian noise due to the improper selection of top 6 frequencies. In case of *AudioMatch* the generation of 16-bit fingerprint causes the drop though it is much less prone to noise as compared to *Next2Me* because of considering the logarithmic amplitude of STFT. Similar behaviour is also found for intermittent and impulsive noises as these are the subset of continuous noise. Next, we compare the three acoustic context detection mechanisms regarding computational resource requirements, as shown in Fig. 16. We measure these performance statistics in a standard Linux (Kernel version: 4.4.0) based workstation (Dell Precision Tower 7,810) using the `free` command to obtain the primary memory consumption of the different methodologies. We compute the total execution time and the overall memory consumption during the execution of the three methods. We observe that (i) *GroupSense* takes very less time per iteration during the computation process compared to *Next2Me* and *AudioMatch* (Fig. 16a); (ii) the memory consumption for *GroupSense* is less than *Next2Me* (Fig. 16b). *GroupSense* enjoys the benefit of lower resource consumption primarily because it computes only cepstrum component for a few segments over the entire interaction time, whereas *Next2Me* uses several windowing operations along with smoothing and FFT computations. *AudioMatch* calculates audio spectrogram using short time Fourier transform with a highly overlapping hamming window, causing higher elapsed time than that of *GroupSense*. In a nutshell, we observe that *GroupSense* can detect various meeting groups generically and in a device independent way, however, can provide better group detection F_1-score with less resource usage compared to the baselines.

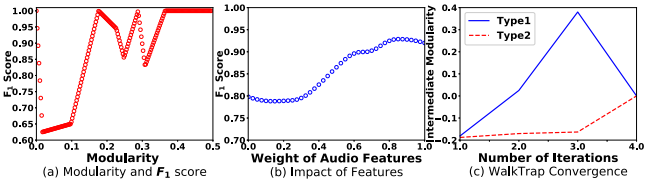


Fig. 17. *GroupSense* insights.

5.5.4 GroupSense *Internals*

In this section, we discuss the importance of the modularity value in *GroupSense*, and how the proximity and acoustic context features improve the modularity of the proposed group detection mechanism. We plot the F_1-Score for the modularity, as shown in Fig. 17a. We observe that the F_1-Score converges to 1.0 when the modularity is more than 0.35. Hence a group is detected with high accuracy when the cohesiveness is also high. This indicates the importance of modularity index in *GroupSense*. Therefore, the community detection algorithm used in *GroupSense* tries to optimize the modularity in successive iterations. In this line, Fig. 17b highlights the importance of Step 3 of *GroupSense* model, where we plot F_1-Score with respect to the weight (w) of the audio feature. The figure indicates that the system achieves maximum accuracy regarding F_1-Score when both the proximity and the acoustic context attain non-zero weights, indicating that both the features are important for correct detection of meeting groups. However, the importance of acoustic context is more prominent over the proximity feature.

Next, we look into the iteration-wise modularity variations of the *GroupSense* model. As mentioned earlier, modularity of a weighted fully connected graph converges to zero when all the nodes form a single community [36]. Accordingly, the group detection algorithm converges with two cases – (a) $\mathcal{M} > 0.0$, when there are multiple groups in the population of subjects ($Type1$) and (b) $\mathcal{M} \approx 0.0$, when there is a single group consisting of all the subjects from the population ($Type2$). Fig. 17c plots the change in modularity value to the number of iterations performed in the algorithm, for these two cases. We observe that for a $Type1$ group (scenario $S5$), we get the maximum modularity close to 0.4 with 3 iterations, whereas, for a $Type2$ group (scenario $S4$), the modularity starts with a negative value and converges to $zero$ with iteration 4.

6 Conclusion

In this paper, we have developed *GroupSense*, a smartphone based light-weight methodology to infer various meeting groups by sensing the acoustic context around the users in proximity. From the pilot study, we have observed that although audio signals captured at the smartphones provide a good indication of the acoustic context of the environment, a significant audio pressure from speakers of the nearby groups also gets captured due to the omnidirectional nature of smartphone microphones. We have developed an unsupervised methodology to process audio signals to capture the context and used the concept of cohesivity from network science to identify the groups based on context information. The implementation and thorough testing of *GroupSense* shows that it can significantly improve meeting

group detection accuracy compared to other baselines, and the method is independent of scenarios or devices used to capture signals. However, our understanding is that *GroupSense* can perform well when the underline groups are sufficiently cohesive; it may fail in the scenarios when multiple groups are overlapped in space, or a group is spatially overlapped with individuals who are not part of that group, for example, small groups in a crowded space. We envisage the application of *GroupSense* in the domain of workplace team formation, team tracking, as well as studying the dynamics of the team members. In such scenarios, *real time* group detection is not the major objective. Hence, we conduct the model execution *offline* on the central server, whereas the smartphones are used mostly for data collection. However, current implementation of the system may have privacy concerns as we transfer the raw audio signals to the server for further processing. Such privacy concerns can be handled by directly extracting the tone information at the smartphones and apply data-perturbation before sending the data to the server, such that the raw audio signals cannot be extracted from the processed data. Importantly, the battery consumption is another major concern in smartphone sensing. The dynamic sensing mechanism can be useful in energy saving, where the sensor awakes only during some predefined event. Such techniques need further investigation, which we keep as a part of the future extension of this system.

REFERENCES

[1] K. A. McComas, "Citizen satisfaction with public meetings used for risk communication," *J. Appl. Commun. Res.*, vol. 31, no. 2, pp. 164–184, 2003.

[2] T. Clark, "Teaching students to enhance the ecology of small group meetings," *Bus. Commun. Quart.*, vol. 61, no. 4, pp. 40–52, 1998.

[3] K. McComas, L. S. Tuite, L. Waks, and L. A. Sherman, "Predicting satisfaction and outcome acceptance with advisory committee meetings: The role of procedural justice," *J. Appl. Social Psychology*, vol. 37, no. 5, pp. 905–927, 2007.

[4] B. A. Reinig and B. Shin, "The dynamic effects of group support systems on group meetings," *J. Manage. Inf. Syst.*, vol. 19, no. 2, pp. 303–325, 2002.

[5] A. P. Horan, "An effective workplace stress management intervention: Chicken soup for the soul at Work™ employee groups," *Work*, vol. 18, no. 1, pp. 3–13, 2002.

[6] K. Jayarajah, Y. Lee, A. Misra, and R. K. Balan, "Need accurate user behaviour?: Pay attention to groups!" in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 855–866.

[7] M. B. Gilboy, S. Heinerichs, and G. Pazzaglia, "Enhancing student engagement using the flipped classroom," *J. Nutrition Edu. Behavior*, vol. 47, no. 1, pp. 109–114, 2015.

[8] P. Sapiezynski, A. Stopczynski, D. K. Wind, J. Leskovec, and S. Lehmann, "Inferring person-to-person proximity using WiFi signals," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 2, 2017, Art. no. 24.

[9] C. Wu, J. Xu, Z. Yang, N. D. Lane, and Z. Yin, "Gain without pain: Accurate WiFi-based localization using fingerprint spatial gradient," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 2, 2017, Art. no. 29.

[10] P. Casagranda, M. L. Sapino, and K. S. Candan, "Audio assisted group detection using smartphones," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2015, pp. 1–6.

[11] R. Sen, Y. Lee, K. Jayarajah, A. Misra, and R. K. Balan, "GruMon: Fast and accurate group monitoring for heterogeneous urban spaces," in *Proc. 12th ACM Conf. Embedded Netw. Sensor Syst.*, 2014, pp. 46–60.

[12] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. 10th IEEE Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 197–210.

[13] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "BeepBeep: A high accuracy acoustic ranging system using cots mobile devices," in *Proc. 5th ACM Conf. Embedded Netw. Sensor Syst.*, 2007, pp. 1–14.

[14] W. Wang, A. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. 22nd ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 82–94.

[15] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in *Readings in Speech Recognition*. San Mateo, CA, USA: Morgan Kaufmann, 1990, pp. 65–74.

[16] J. Baker and C. Efstratiou, "Next2Me: Capturing social interactions through smartphone devices using WiFi and audio signals," in *Proc. ACM EAI Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Services*, 2017, pp. 412–421.

[17] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006.

[18] S. Das, S. Chatterjee, S. Chakraborty, and B. Mitra, "An unsupervised model for detecting passively encountering groups from WiFi signals," in *Proc. IEEE Global Commun. Conf.*, Dec. 2018.

[19] S. Das, "A framework for group identification using smartphone and wearables," in *Proc. ACM Joint Conf. Int. Symp. Pervasive Ubiquitous Comput. Wearable Comput.*, 2018, pp. 1847–1850.

[20] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.

[21] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. 15th ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 261–272.

[22] H. Abdelnasser, R. Mohamed, A. Elgohary, M. F. Alzantot, H. Wang, S. Sen, R. R. Choudhury, and M. Youssef, "SemanticSLAM: Using environment landmarks for unsupervised indoor localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1770–1782, Jul. 2016.

[23] H. Aly, A. Basalamah, and M. Youssef, "Accurate and energy-efficient GPS-less outdoor localization," *ACM Trans. Spatial Algorithms Syst.*, vol. 3, no. 2, 2017, Art. no. 4.

[24] T. M. T. Do and D. Gatica-Perez, "GroupUs: Smartphone proximity data and human interaction type mining," in *Proc. 15th IEEE Annu. Int. Symp. Wearable Comput.*, 2011, pp. 21–28.

[25] H. Hong, C. Luo, and M. C. Chan, "SocialProbe: Understanding social interaction through passive WiFi monitoring," in *Proc. 13th ACM Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Services*, 2016, pp. 94–103.

[26] Y. Lee, C. Min, C. Hwang, J. Lee, I. Hwang, Y. Ju, C. Yoo, M. Moon, U. Lee, and J. Song, "SocioPhone: Everyday face-to-face interaction monitoring platform using multi-phone sensor fusion," in *Proc. 11th ACM Int. Conf. Mobile Syst. Appl. Services*, 2013, pp. 375–388.

[27] R. Friedman, A. Kogan, and Y. Krivolapov, "On power and throughput tradeoffs of WiFi and bluetooth in smartphones," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1363–1376, Jul. 2013.

[28] Z. Liu, Z. Zhang, L. W. He, and P. Chou, "Energy-based sound source localization and gain normalization for ad hoc microphone arrays," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2007, pp. II-761–II-764.

[29] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoustical Soc. America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[30] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner, "Crowd++: Unsupervised speaker count with smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 43–52.

[31] T. Song, X. Cheng, H. Li, J. Yu, S. Wang, and R. Bie, "Detecting driver phone calls in a moving vehicle based on voice features," in *Proc. 35th IEEE INFOCOM*, 2016, pp. 1–9.

[32] W. Passchier-Vermeer and W. F. Passchier, "Noise exposure and public health," *Environ. Health Perspectives*, vol. 108, pp. 123–131, 2000.

[33] M. Guggenberger, M. Lux, and L. Böszörmenyi, "An analysis of time drift in hand-held recording devices," in *Proc. Int. Conf. Multimedia Model.*, 2015, pp. 203–213.

[34] M. Narayana and S. Kopparapu, "Effect of noise-in-speech on MFCC parameters," in *Proc. 9th WSEAS Int. Conf. Signal Speech Image Process. 9th WSEAS Int. Conf. Multimedia Internet Video Technol.*, 2009, pp. 39–43.

[35] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E*, vol. 78, no. 4, 2008, Art. no. 046110.

[36] M. E. Newman, "Modularity and community structure in networks," *Proc. Nat. Academy Sci. United States America*, vol. 103, no. 23, pp. 8577–8582, 2006.
[37] D. Liaqat, R. Wu, A. Gershon, H. Alshaer, F. Rudzicz, and E. de Lara, "Challenges with real-world smartwatch based audio monitoring," in *Proc. 4th ACM Int. Conf. Mobile Syst. Appl. Services*, 2018, pp. 54–59.
[38] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. Campbell, "StudentLife: Assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 3–14.

Snigdha Das received the MS (by research) degree from the School of Information Technology, Indian Institute of Technology Kharagpur, India, in 2015. She is currently working toward the PhD degree in the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. Her current research interests include mobile systems and ubiquitous computing.

Soumyajit Chatterjee received the BE degree from the University Institute of Technology, University of Burdwan, in 2012, and the MTech degree in computer science from IIT (ISM), Dhanbad, in 2016. He joined IIT Kharagpur, in 2017, as a research scholar (Doctorate Program). He also has industry experience of one year seven months. Currently, his domain of research is mobile systems and ubiquitous computing.

Sandip Chakraborty received the BE degree in information technology from Jadavpur University, Kolkata, India, in 2009, and the MTech and PhD degrees from the Indian Institute of Technology Guwahati, India, in 2011 and 2014, respectively. Currently, he is an assistant professor with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. His research interests include computer systems and distributed computing.

Bivas Mitra received the PhD degree from IIT Kharagpur, India. He is an assistant professor with the Department of Computer Science & Engineering, IIT Kharagpur, India. Before that, he worked briefly with Samsung Electronics, Noida, as a chief engineer. He did his postdocs with CNRS Paris, France, and UCL, Belgium. His research interests include network science, multilayer network, and mobile affective computing.