Autonomous Learning of Speaker Identity and WiFi Geofence from Noisy Sensor Data

Chris Xiaoxuan Lu, Yuanbo Xiangli, Peijun Zhao, Changhao Chen, Niki Trigoni, and Andrew Markham

Abstract—A fundamental building block towards intelligent environments is the ability to understand who is present in a certain area. A ubiquitous way of detecting this is to exploit unique vocal characteristics as people interact with one another in common spaces. However, manually enrolling users into a biometric database is time-consuming and not robust to vocal deviations over time. Instead, consider audio features sampled during a meeting, yielding a noisy set of possible voiceprints. With a number of meetings and knowledge of participation, e.g., sniffed wireless Media Access Control (MAC) addresses, can we learn to associate a specific identity with a particular voiceprint? To address this problem, this paper advocates an Internet of Things (IoT) solution and proposes to use *co-located* WiFi as supervisory weak labels to automatically bootstrap the labelling process. In particular, a novel cross-modality labelling algorithm is proposed that jointly optimises the clustering and association process, which solves the inherent mismatching issues arising from heterogeneous sensor data. At the same time, we further propose to reuse the labelled data to iteratively update wireless geofence models and curate device specific thresholds. Extensive experimental results from two different scenarios demonstrate that our proposed method is able to achieve 2-fold improvement in labelling compared with conventional methods and can achieve reliable speaker recognition in the wild.

Index Terms—Cross-modal Association; Internet of Things; Speaker Identification

1 Introduction

Speaker recognition and verification are key components of smart spaces, e.g., offices and buildings for determining who is where [1], [2]. Knowing this information allows a wide range of context-aware applications such as personalized heating and cooling, entertainment, behavioral analysis or health sensing.

A vast amount of research over the past decades has focused on the design of bespoke systems for speaker recognition, and with the advent of deep learning, progress has rapidly accelerated. As an example of a state-of-the-art speaker recognizer, X-vector [3] uses a deep neural network (DNN) as the feature extractor and achieves extremely low error rate (e.g., 4.16%) on very challenging datasets through a trained feature embedding. Although X-vector and similar approaches can operate at remarkable levels of performance, transferring them to real-world scenarios is far from trivial. Deploying a reliable speaker recognition in a smart space faces two overarching issues. Firstly, unlike many classification tasks where the class labels are shared with many public datasets, speaker labels in a particular smart environment are inherently out of set, due to their uniqueness. For instance, although VoxCeleb [4], the largest-scale human speech dataset, contains utterances from more than $7,000$ speakers, most of them are celebrities and are orthogonal to the target speaker group in a particular environment. Other domain differences, such as usage of low-fidelity microphones and emotion changes within the speaker could also impact the recognition performance [5]. Therefore, a speaker recognizer trained on public datasets cannot be used directly in a new environment, but has to be re-trained or adapted with the labelled utterances in new environments. This leads to the second issue: labelling utterances in the wild is challenging since subject enrollment incurs significant user effort and cost [6]. Unfortunately, crowdsourcing tools such as Amazon Mechanical Turk [7] are not applicable as conversations include sensitive data and many users are not willing to post them online.

On the other hand, with the recent advent of Internet of Things (IoT), we also witness our physical spaces (e.g., smart buildings) being equipped with more and more sensors and communication infrastructure. For example, a

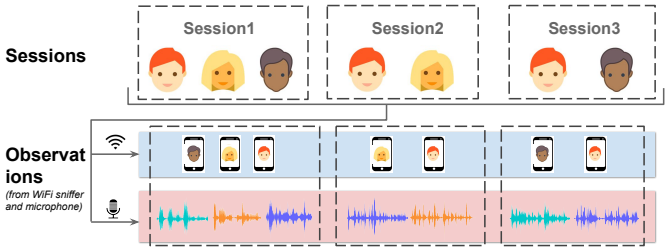


Fig. 1: Given sets of co-located observations (utterances and device identifiers) from multiple sessions, utterances can be associated with device identifiers. With a sufficient number of sessions, this mapping can be made unique, leading to automatic utterance labeling.

Manuscript received March 17, 2019; Revised May 31, 2019; Accepted June 21, 2019.
All authors are with the Department of Computer Science, University of Oxford, OX1 3QD, UK. (e-mail: {xiaoxuan.lu, yuanbo.xiangli, peijun.zhao, changhao.chen, niki.trigoni, andrew.markham}@cs.ox.ac.uk).

- *Chris Xiaoxuan Lu and Yuanbo Xiangli contributed equally.*
- *Corresponding Author: Changhao Chen*

variety of sensors have been deployed in commercial offices, such as microphones, WiFi access points, cameras etc. In this work, we utilize this emerging trend of IoT and propose to automatically achieve high levels of recognition with *zero* user enrollment effort by *harvesting* the voice labels from ambient WiFi data. To this end, we exploit the fact that speakers are usually, but not always, *co-located* with their mobile devices, e.g., smartphones and fitness monitors. To provide ubiquitous connectivity, these devices have some forms of wireless interfaces, e.g., BLE, WiFi, cellular. These provide a unique identifier, ranging from the hardware level, e.g., International Mobile Equipment Identity (IMEI) or MAC addresses, to the network authentication level (e.g., usernames). Our aim is to use a set of such identifiers as ID proxies to label a set of collected speech utterances and use the labeled data to adapt a pre-trained speaker recognizer. The adapted speaker recognition model can then work independently, even in rooms without WiFi MAC scanning. Fig. 1 describes this intuition.

The main challenge to update the model is that the binding between a voice and a wireless identifier is loose and noisy, for two major reasons. Firstly, unlike standard fusion problems where two sensor modalities are both observing temporally evolving systems, the cross-modality data in our case is temporally unaligned. For instance, detecting a WiFi identifier does not imply the device owner is speaking at the exact instant. However, a recognition system cannot be trained without a fine-grained association between the identifiers and utterances, while conventional diarization systems [6] cannot provide explicitly labelled data either and are unable to select high-confident utterance samples. Secondly, the co-located WiFi identifiers are determined by the device received signal strength (RSS) and these collocations are not necessarily accurate. For instance, the presence of a smartphone in the room is decided by a threshold RSS value, which effectively defines a geofence. Due to device heterogeneity, this threshold varies significantly and largely depends on the particular device. Because we could not know a suitable geofence value for each device *a priori*, the detected collocations are uncertain and add noise to the relationship between device and speaker presence.

We hence present `SCAN+`, a novel framework which gradually associates vocalizations with a specific identity through ambiently harvested wireless identifiers. In this way, the developed speaker recognition model is bespoke to the vocal dynamics of a set of users within a particular smart space, without the cost and effort of having to make users enroll into the system. We explicitly assume that the sensed WiFi data and voice features will be noisy, and present a technique that simultaneously clusters and names utterances, yielding accurate, zero-effort speaker recognition. To account for device heterogeneity, we propose an iterative approach to automatically curate the best geofence RSS value for individual devices. We show that this can be further improved by iterating between clustering and naming to minimise the mismatch. In summary, our contributions are:

- We show that co-located side-channel information about likely participants in an event provides valuable, albeit noisy, clues about speaker identity.
- We propose a novel algorithm which simultaneously handles clustering and association, and highlight the benefits of the algorithm compared to handling these problems in a sequential manner.
- We propose an iterative optimization framework to automatically customize the personalized geofence to tackle the device heterogeneity issue, which improves the detection accuracy of collocation.
- We compare `SCAN+` against various baselines in different scenarios and show 2-fold improvements in performance especially in noisy environments.

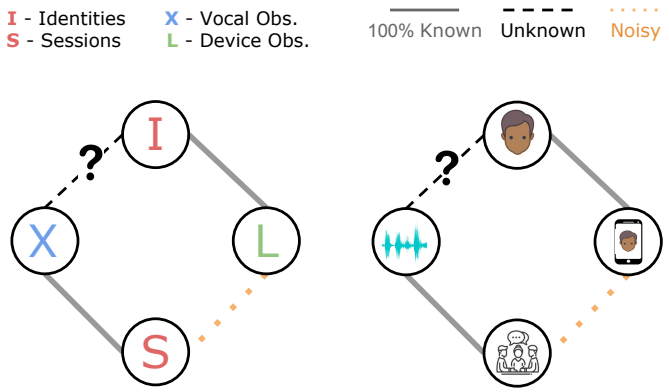
The rest of the paper is organised as follows. Sec. 3 overviews the workflow of our proposed system and Sec. 4 describes the cross-modality labelling module. In Sec. 5 we present a novel curation method to simultaneously identify and localize speakers based on the labelling module. Sec. 6 provides implementation details. Sec. 7 evaluates the proposed system, and compares its performance with the competing approaches. Sec. 2 surveys the related work, while Sec. 9 concludes the paper and outlines future directions.

2 Related Work

Cross-modality Matching: Cross-modal matching has received considerable attention in different research areas. Methods have been developed to establish mappings from images [8], [9], [10] and videos [11] to textual descriptions (e.g., captioning), developing image representation from sounds [12], [13], and generating visual models from text [14]. In cross-modality matching between images and radio signals, however, work is very limited and all dedicated to trajectory tracking of humans [15], [16], [17]. The field of recognizing speaker identities from wireless signals is still a blank space.

Data Association: Our proposed cross-modal labelling approach is also related to data association methods. Given a track of sensor readings, data association aims to figure out inter-frame correspondences between them. Data association is widely used in radar systems, when tracking blips on a radar screen [18], as well as object monitoring of surveillance systems [19]. To find inter-frame correspondences, a lot of Bayesian filtering approaches have been developed, including Nearest-Neighbour Data Association Filter [20], Probabilistic Data Association Filter [21], Joint Probabilistic Data Association Filter [22] and Multiple Hypothesis Tracking [23]. Unlike our work, these approaches rely on state-based models, where both sensors are observing temporally evolving systems. In our problem, detecting a MAC address does not imply that someone will be speaking at that exact instant.

Speaker Recognition: A standard speaker recognition system can be distilled down to two tasks: speaker feature extraction and feature scoring. The unsupervised extractor i-vector [24] is based on a linear Gaussian model. It has dominated speaker recognition tasks and has inspired the design of DNN-based systems in this field. DNN-based embedding extractors are, on the other hand, supervised. Speaker features can be extracted from the last layers of the network. A typical example is the x-vector system proposed in [3]. The difference in model structures makes x-vector more discriminative and superior in short-utterances compared with i-vector. Other DNN-based speaker recognisers



(a) Abstraction (b) Example

Fig. 2: Relationship of heterogeneous sensor observations. (a) Abstract relationship. (b) Example. Given the noisy co-located device observations, SCAN+ aims to accurately harvest vocal observations to the respective speakers.

are also proposed in different application scenarios [25], [26]. Although the above methods have been proven remarkably effective in speaker recognition, the training needs a vast amount of labeled utterances to train the deep speaker recognisers. However, such amount of utterances are always achievable in a particular domain and using a small amount of training data will incur poor generalisation ability in the wild. SCAN+ aims to automatically label utterances in the wild which is complementary to the above work as their label provider.

3 System Overview

We now give an overview of our system model and architecture.

3.1 System Model

We consider a recognition problem with m speakers of interest (SOI) and each speaker owns one WiFi-enabled device, e.g., a smartphone. We denote the identity set as $\mathcal{I} = \{i_j | j = 1, 2, \ldots, m\}$. The set of observed SOI's devices in the target environment is denoted by $\mathcal{L} = \{l_j | j = 1, 2, \ldots, m\}$. We assume the mapping from device MAC address $\mathcal{L}$ to the user identity $\mathcal{I}$ is known and denoted as $\mathcal{L} \Rightarrow \mathcal{I}$. In practice, the mapping between a MAC address to a user is easy to get for smart-space managers. In order to authenticate legitimate WiFi users, say Eduroam[1], device MAC addresses and user account information are usually bound together in the building management system. A collection of vocal observations is collected by microphones in a target environment, which contains the segmented utterances from g sessions of conversations ($\mathcal{S} = \{s_j | j = 1, 2, \ldots, g\}$). Note that, to mimic the real-world complexity, this collection includes voices of both SOI and some non-SOI. Formally, let $\mathcal{X} = \{x_j | j = 1, 2, \ldots, h\}$ denote the utterance collection. In this sense, the *identification problem* addressed in this paper is: given the noisy vocal observations $\mathcal{X}$, find the correct associations between the device ID observations $\mathcal{L}$. Then through the mapping between device IDs to user identities, a database of pairs of vocal observations and identity labels can be developed. Finally, an independent speaker recognizer can be trained on this database to automatically determine identity label i_k given a vocal observation x_j in the future. Fig. 2 provides a simple schematic illustration of this problem.

3.2 System Architecture

In this section, we provide an overview of the system architecture. SCAN+ consists of three modules:

- *Heterogeneous Data Collection*. This module collects audio data (vocal observations) and WiFi data (device ID observations) through microphones and WiFi sniffers[2] in a target environment. The audio data is then preprocessed into homogeneous utterances.
- *Cross-modality Labeling*. This module labels utterances by associating them with the correct device IDs. Device IDs in this work are MAC addresses detected via a collocated WiFi sniffer with the microphone. With an unknown table, MAC addresses can be mapped to user identities, i.e., labels.
- *Cross-modality Curation*. This module refines the geofence RSS value for each device and use the new RSS value to re-estimate collocation sessions. The goal of curation is to diminish the labelling inconsistency between the device observations and utterances.

with the exception of the one-off data collocation module, the labeling and curation modules are iteratively tasked in SCAN+ until the cross-modality observations are sufficiently consistent. The speaker recognition model trained in the final iteration is the model that is uniquely tailored to the environment. A byproduct of SCAN+ is a set of personalized geofence models for individual devices that could be used for future localization tasks at room level. Fig. 3 illustrates the workflow of SCAN+.

4 Cross-modality Labeling

In this section, we introduce the labeling module in SCAN+. The challenge in the labeling module is that collected audio and sniffed WiFi data are temporally unaligned. For example, detecting a device WiFi address does not imply that the device owner will be speaking at the exact instant and vice versa. Such mis-alignment distinguishes our problem for prior sensor fusion problems, where multiple sensors are observing a temporal evolving system.

4.1 Baseline: Sequential Clustering and Association

In order to tackle the above challenge, a naive approach is to leverage the diverse participatory information in multiple sessions and use a two-step procedure in Sequential: a) in the Clustering Step, utterances $\mathcal{X}$ are firstly grouped into clusters across all sessions, each of which represents the vocal samples of a single individual; and then b) in the Data

1. https://en.wikipedia.org/wiki/Eduroam

2. https://www.wireshark.org/

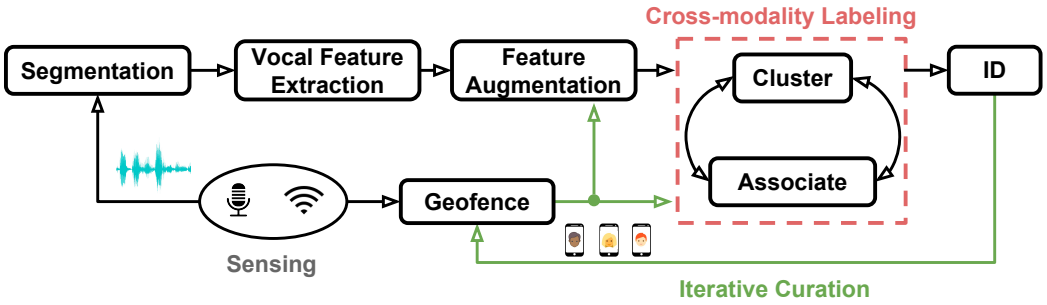


Fig. 3: Workflow of SCAN+.

Association Step, the clusters are assigned with identities based on device ID observations $\mathcal{L}$.

Clustering Step. Given a set of sessions, utterances are first transformed into feature vectors $\mathcal{Z}$, through a speaker recognition model pre-trained f_θ on public datasets. Such extractor is trained with metric losses, e.g., triplet loss so that the learned features could bear a good property for clustering [26]. Based on the extracted features, these utterance samples are then merged into disjoint, non-overlapping clusters. Without loss of generality, we denote the set of derived clusters by $\mathcal{C} = \{c_i | i = 1, 2, \ldots, h\}$. In order to make assignments in the subsequent association step, the number of clusters h must be equal to or greater than the number of SOI, m.

Data Association Step. Based on the similarity of session attendance, voice clusters can be mapped to device IDs by data association. Let $\mathbf{r}_{c_k} = (r^1_{c_k}, r^2_{c_k}, \ldots, r^g_{c_k})$ be the context vector of the k-th voice cluster c_k, where g is the total number of sessions. $r^j_{c_k}$ is set to 1 only if c_k contains utterances from session s_j. At the same time, a SOI's device l_i is also linked with a context vector $\mathbf{r}_{l_i}$, and $\mathbf{r}^j_{l_i}$ is set to 1 only if l_i is detected in session s_j. An edge can be created between a cluster c_k and a device ID l_j, with the edge weight determined by the similarity in terms of context vector. Intuitively, a higher similarity score means that there are more shared session attendance and such pairs of voice clusters and device are more likely belonging to the same identity. Then associating identities with clusters is equivalent to solving the combinatorial optimization problem on the weighted bipartite graph, e.g. using the Hungarian algorithm [27]. Finally, through a mapping table between device ID and user identity, utterances in the same clusters are all labeled with the same user identity. The pre-trained speaker recognizer f_θ can adapt its model parameters θ that are bespoke to a new environment.

Limitations. The above method addresses the identification problem in two Sequential steps: context observations are firstly clustered and then matched to identities by minimizing the combinatorial mismatch. Although this approach is simple and easy to implement, it is not robust to noisy observations. Firstly, errors can occur due to the noise in vocal observations. For example, people's voices may vary considerably across contexts due to illness or emotional influences [5], confusing the clustering step and causing unrecoverable knock-on effects on the ensuing association step. Secondly, and more importantly, errors can also occur due to noisy device observations. For example, a session might contain non-SOI's voices, though their device are not sniffed. As disturbing observations incurred by non-SOI, the number of clusters h is difficult to know for clustering. A misleading clustering result could further degrade the quality of data association.

4.2 SCAN: Simultaneously Clustering And Naming

In this section, we introduce how to mitigate the above limitations in the sequential approach. The key insight of our solution is that the clustering of utterances should not be finalized independently of and in advance of data association, but both tasks should progress in tandem. The proposed **S**imultaneous **C**lustering **A**nd **N**aming (SCAN) algorithm works as follows. Firstly, it compiles sensor observations as an augmented linkage tree, which succinctly encodes the hierarchical clustering plans of context observations across different contexts, and more importantly all possible data association plans given a specific clustering plan. Then our algorithm finds the best clustering and data association plan by solving a constrained optimization problem on the constructed linkage tree.

Feature Augmentation. Instead of merely depending on the vocal feature similarity, session attendance information of devices could bootstrap merge utterances as well. Recall that device attendance already reveals the identities of subjects (in the form of MAC addresses) in a particular session, and the collected utterances may contain the utterances of these speakers accordingly. The overlapped speakers in distinct sessions can be used as a prior that guides the utterance merging. For example, if there are no shared MAC addresses sniffed in two sessions, then it is very likely that the collected utterances in these two sessions should lie in different clusters. Formally, for a session s_i, we denote the device attendance vector as $\mathbf{u}_i = (u^1_i, u^2_i, \ldots, u^m_i)$, where $u^j_i = 1$ if device l_j is detected in session s_i. In this way, we can develop an augmented feature $\widetilde{\mathbf{z}}_k$ for an utterance x_k collected in the session s_i:

$$\widetilde{\mathbf{z}}_k = [\mathbf{z}_k, \mathbf{u}_i] \tag{1}$$

where z_k is the extracted vocal features for utterances as introduced in Sec. 4.1. Then, the similarity of two cross-session utterances is determined by computing the distance between their hybrid feature vectors defined in Eq. 1. The distance is computed as follows:

$$D_{\widetilde{\mathbf{z}}_{k1}, \widetilde{\mathbf{z}}_{k2}} = D_{\mathbf{z}_{k1}, \mathbf{z}_{k1}} + D_{\mathbf{u}_{i1}, \mathbf{u}_{i2}} \tag{2}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2926645, IEEE Internet of Things Journal

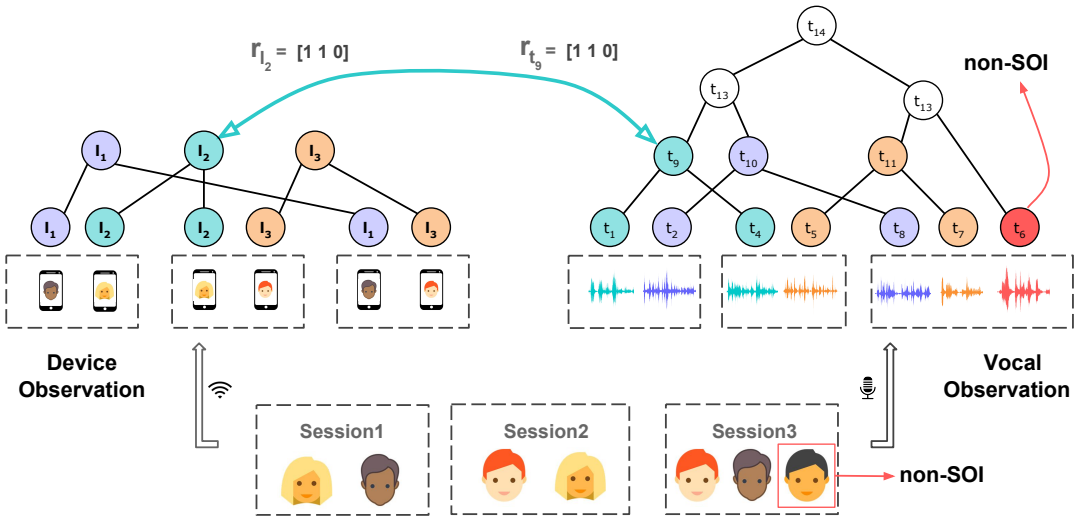


Fig. 4: SCAN: *Simultaneously* Clustering and Naming. It simultaneously performs clustering and association, by directly examining the fitness between a MAC address and an utterance node on the tree, in terms of context vector similarity. SCAN can tolerate disturbances from non-SOI as it associates the pure cluster before it gets contaminated by the biometric samples of non-SOI.

where $k1 \neq k2$. $D_{\mathbf{z}_{k1},\mathbf{z}_{k1}}$ is the Euclidean distance of the vocal features and $D_{\mathbf{u}_{i1},\mathbf{u}_{i2}}$ is the Jacared index of the device attendance vectors. Note that such hybrid features does not affect the grouping of two feature vectors in the same session.

Linkage Tree Construction. Based on the pairwise similarity between hybrid features, the proposed algorithm compiles them into a linkage tree T. The leaf nodes T_{leaf} are the voice samples, while a branch node represents the cluster of all its descendant leaf nodes. Essentially T represents the hierarchical clustering of all utterances in different sessions, and selecting a combination of nodes from the tree will give a specific clustering plan. For example in Fig. 4, selecting nodes t_9 means that leaf nodes t_1 and t_4 should be grouped together (and thus belong to the same individual). Each node t_i in T is associated with a *linkage score* q_{f_i}, describing the feature similarity or compatibility between the data within the cluster it represents.

Augment Linkage Tree with Data Association Scores. Given a linkage tree T, the clustering process of the baseline approach is equivalent to finding the set of nodes in T that maximises the total linkage score. However as discussed in the previous section, this is not reliable due to noisy sensor observations. Therefore, the proposed SCAN algorithm augments the linkage tree by introducing additional data association scores to each of its nodes t_i, which represent the fitness of assigning an identity label to t_i given device observations $\mathcal{L}$. Concretely, let $\mathbf{r}_{t_i}$ be the session context vector of a node t_i, where $r^j_{t_i} = 1$ if t_i contains voice samples collected from session s_j. Similarly, SOI's device l_k is also linked with a context vector $\mathbf{r}_{l_j}$, and $\mathbf{r}^j_{l_j}$ is set to 1 only if l_j is detected in session s_j. Intuitively, for a node t_i and a device l_j, if $\mathbf{r}_{t_i}$ and $\mathbf{r}_{l_j}$ are similar enough, it is very likely that vocal observations under node t_i are actually the voice footprint of the speaker who owns device l_j, since they appear in similar series of contexts and match with each other well. Formally, for a node t_i, we define its data association scores with respect to the device IDs as a vector $\mathbf{q}_{a_i} = (q^1_{a_i}, q^2_{a_i}, \ldots, q^m_{a_i})$, where the j-th score $q^j_{a_i}$ is the Euclidean distance between the node context vector $\mathbf{r}_{t_i}$ and the device context vector $\mathbf{r}_{l_j}$. Together with the feature score, the final score to assign node t_i to device l_j is a composite score function:

$$q^j_i = (1-\omega) * q_{f_i} + \omega * q^j_{a_i} \tag{3}$$

where the parameter ω governs how much we trust the device observations and to what extent we want them to impact the result of clustering.

Optimization Program. With the previously introduced terminology and notations, we formulate the following optimization problem:

$$\max_{\mathbf{A}} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} q^j_i * a_{i,j} \tag{4}$$

$$s.t. \quad \sum_{j=1}^{m} a_{i,j} \leq 1, \forall i \in 1,\ldots,n \tag{5}$$

$$\sum_{i=1}^{n} a_{i,j} = 1, \forall j \in 1,\ldots,m \tag{6}$$

$$\sum_{i\in\Pi_k}\sum_{j=1}^{m} a_{i,j} \leq 1, \forall k \in T_{leaf} \tag{7}$$

$$a_{i,j} \in \{0,1\}, \forall i \in \{1,\ldots,n\}, \forall j \in \{1,\ldots,m\} \tag{8}$$

where $\mathbf{A} = (a_{i,j})_{n\times m}$ is the decision variable and q^j_i is the composite score determined by Eq. (3). T_{leaf} represents the set of all leaf nodes in the linkage tree. The objective function aims to maximize the total scores when selecting m nodes in the linkage tree T with size of n. Intuitively, the selected m nodes are the optimal clusters out of these n utterance samples. The inequality in (5) simply means a node can be assigned to at most one device. The constraint in (6) is used to ensure that device is associated with a single node.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2926645, IEEE Internet of Things Journal

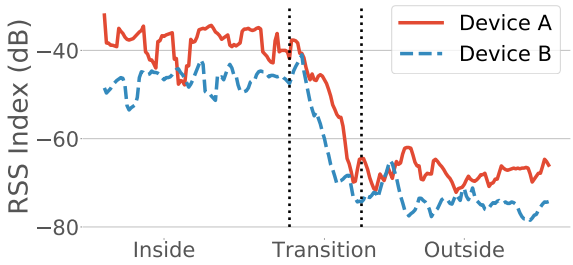


Fig. 5: The impact of device heterogeneity on RSS index. The difference of RSS index of two different devices can be as large as 8dB. A universal geofence value will lead to downstream labelling errors.

Additionally, a node cannot be selected with its ancestors or descendants at the same time since they contain duplicate data. In order to compile this tree structure in optimization, the constraint (7) is enforced to guarantee that on any path leading to a leaf node, at most one node is assigned to a device. Finally, the constraint (8) is there to make sure that decision variable $a_{i,j}$ can take on the integer value 0 and 1 only. The above optimization formulation is essentially a integer linear programming (ILP) problem and can be readily solved by either exact or approximate algorithms [28], [29].

This finishes our SCAN algorithm. Note that, SCAN bypasses the requirement of knowing the number of non-SOI speakers but only depends on the number of SOI to associate. Furthermore, joint clustering and association prevent associating impure clusters. SCAN early selects pure clusters before they merge with wrong samples that contaminate their context vectors for association. In Sec. 7, we will see the significant performance gain of SCAN compared to the baseline approach.

5 Iterative Curation

So far we have seen how to automatically label speaker utterances with co-located device ID information. However, determining a co-located event is non-trivial and can be error-prone. In SCAN, the colocation is determined by comparing a device's average RSS index with a geofence threshold to determine a binary presence/absence indicator. Naturally, any device with an RSS index lower than the threshold is regarded to be outside the room and vice versa. However, as illustrated in Fig. 5, due to device heterogeneity and diversity in WiFi NICs, different devices have different signal strengths even if they are placed in the same place. As such, it is difficult to distinguish in-room devices and out-of-room devices by using a global geofence threshold. In this section, we present SCAN+, an iterative approach to automatically customize individual geofence models that tackles the challenge of device heterogeneity, which we refer to as iterative curation. Through iterative curation, not only can labelling errors be reduced in subsequent iterations, but also geofence models are personalized for better localization accuracy.

5.1 Geofence Initialization

We start by initializing a common geofence value, ζ_{init} for all devices. Then in any session, SCAN+ sniffs WiFi packets and group RSS readings together if they share the same source MAC addresses, i.e. they are emitted by the same device. However, due to environmental dynamics, RSS values can vary significantly even if the device is physically still. In order to reduce the effect of RSS variations [30], [31], we use the median RSS index o_k^j to summarize the RSS readings of device l_j collected in the session s_k. Finally, by comparing o_k^j to the initialized geofence value ζ_{init}, we can distinguish whether device l_j is present at a particular session s_k and construct its context vector $\mathbf{r}_{l_j}$ and device attendance vector $\mathbf{u}_k$ for subsequent cross-modality labeling.

5.2 Device Presence Update

After cross-modality labeling with the common initial geofence threshold, we can update the device context vector of both modalities. Due to uncertainty in the initial geofence value, the resulted device context vector is noisy. Fortunately, we note that the attenuation of sound is much faster, e.g., if a door is closed. This leads to a more precise relationship between the recorded utterance and sessions. Therefore, once a device is associated with an utterance cluster (see Sec. 4.2), we can use the utterance context vector to update the presence information of associated devices. We present two different approaches to determine a threshold value, one hard and one soft (probabilistic), which are discussed below.

Hard Geofence: We firstly introduce a deterministic technique to update the context vector. Given an associated pair of device l_j and cluster t_i, we re-group the RSS readings of a device l_j based on all participating sessions inferred from the cluster context vector $\mathbf{r}_{t_i}$. The median of the collection of RSS indices is selected as the new geofence value ζ_j. Then for a session s_k, we then update the device attendance vector $\mathbf{u}_k$ based on the new geofence value.

Soft Geofence: Due to lack of model adaptation, utterance clusters can be impure, especially in early iterations. Consequently, the context vector of associated clusters might be inaccurate, which significantly affects the deterministic geofence update. To reduce the risks incurred by using impure utterance clusters, we propose to use a probabilistic geofence, under the Gaussian noise model for RSS as suggested by [32], [33]. Similarly, we examine the associated utterance context vector to group the RSS readings of a device l_j across all participated sessions and fit these in-room RSS readings to a normal distribution, denoted by $D_{in}^j \sim \mathcal{N}(\mu_{in}, \sigma_{in}^2)$. Additionally, we fit another normal distribution of the RSS readings sniffed in absent sessions inferred from the utterance context vector. We denote this distribution by $D_{out}^j \sim \mathcal{N}(\mu_{out}, \sigma_{out}^2)$. Then, the presence of a device l_j in session s_k is geofenced by a normalized probability:

$$p_k^j = \frac{p(o_k^j|D_{in}^j)}{p(o_k^j|D_{in}^j) + p(o_k^j|D_{out}^j)} \tag{9}$$

The new context vector of device l_j is a probabilistic vector, where each element represents the presence probability in a particular session:

$$\mathbf{r}_{l_j} = (p_1^j, p_2^j, \ldots, p_g^j) \tag{10}$$

Similarly, the device attendance vector for session s_k is updated to a probabilistic vector as well:

$$\mathbf{u}_k = (p_k^1, p_k^2, \ldots, p_k^m) \tag{11}$$

5.3 Pipeline

We are now in a position to give the pipeline algorithm of SCAN+. as shown in Algorithm. 1. The pipeline takes as inputs vocal and device observations collected across a set of sessions, as well as a voice representation model f_θ pre-trained on public datasets. It starts by initializing a global geofence threshold ζ_{init} for all devices, through empirical observations. With a hard geofence model, the initial device context vectors in terms of session attendance can be inferred. Then the iterative process begins in line 5. First, we extract the vocal features of utterances with speaker model and construct the linkage tree based on the feature similarity. Then, based on the tree structure and the compatibility of device and node context vectors, we use SCAN (see Sec. 4.2) to label the selected m utterance clusters of SOI. The speaker model can adapt its parameters by re-training or fine-tuning with the labeled utterances (line 9). Finally, by referring to the associated clusters context vectors, we can customize the soft geofence model of individual devices and correct their devices context vectors accordingly (line 11 and 12). When the total changes of device context vectors between consecutive iterations are small enough, SCAN+ finishes the cross-modality curation. The speaker recognizer derived in the last iteration are chosen as the adapted model, as well as personalized geofence models for SOI's devices.

Algorithm 1: SCAN+ Pipeline

Input: pre-trained voice feature extractor f_θ, device observations $\mathcal{L}$, vocal observations $\mathcal{X}$, Sessions $\mathcal{S}$, number of SOI m, threshold ϵ and mapping table $\mathcal{L} \Rightarrow \mathcal{I}$

Output: adapted model f_θ^*, personalized geofence models D_{out} and D_{in}

Initialize: a global geofence value ζ_{init}

1 **for** $j \leftarrow 1$ *to* m **do**
2 $\quad \mathbf{r}_{l_j}^{(1)} \leftarrow$ `Hard_Geofence`$(\zeta_{init}, l_j, \mathcal{S})$
3 **end**
4 **for** $k \leftarrow 1$ *to* g **do**
5 $\quad \mathbf{u}_k^{(1)} \leftarrow$ `Hard_Geofence`$(\zeta_{init}, s_k, \mathcal{L})$
6 **end**
7 $\tau = 1$
8 **while** $\sqrt{\frac{1}{|m|}\sum_{j=1}^{m} ||\mathbf{r}_{l_j}^{(\tau)} - \mathbf{r}_{l_j}^{(\tau-1)}||^2} > \epsilon$ **do**
9 $\quad \mathcal{Z}^{(\tau)} = f_\theta(\mathcal{X})$
10 $\quad \widetilde{\mathcal{Z}}^{(\tau)} =$ `hybrid_features`$(\mathcal{Z}^{(\tau)}, \mathcal{U}^{(\tau)})$
11 $\quad T^{(\tau)} =$ `linkage_tree`$(\widetilde{\mathcal{Z}}^{(\tau)})$
12 $\quad \mathbf{A}^{(\tau)} =$ `SCAN`$(m, \mathbf{r}_{l_{1,\ldots,m}}^{(\tau)}, T^{(\tau)})$
13 $\quad$ **for** $j \leftarrow 1$ *to* m **do**
14 $\quad\quad D_{in}^{(\tau),j}, D_{out}^{(\tau),j} \leftarrow$ `Gaussian_model`$(\mathbf{A}^{(\tau)}, l_j)$;
15 $\quad\quad \mathbf{r}_{l_j}^{(\tau+1)} \leftarrow$ `Soft_Geofence`$(D_{in}^{(\tau),j}, D_{out}^{(\tau),j}, l_j, \mathcal{S})$
16 $\quad$ **end**
17 $\quad$ **for** $k \leftarrow 1$ *to* g **do**
18 $\quad\quad \mathbf{u}_k^{(\tau+1)} \leftarrow$ `Soft_Geofence`$(D_{in}^{(\tau),j}, D_{out}^{(\tau),j}, s_k, \mathcal{L})$
19 $\quad$ **end**
20 $\quad \tau \leftarrow \tau + 1$
21 **end**
22 $f_\theta^* \leftarrow$ `speaker_model_update`$(\mathbf{A}^{(\tau)}, \mathcal{X}, \mathcal{L} \Rightarrow \mathcal{I})$

6 Implementation

In this section, we provide implementation details of the sensing module in SCAN+ and the pre-processing process for conversations.

6.1 Sensing Front-end

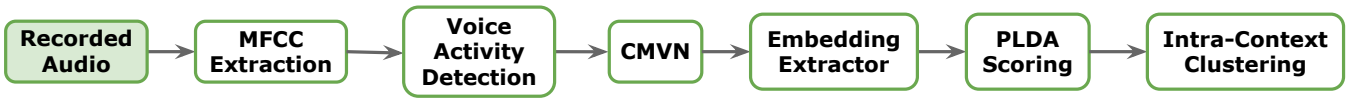
The sensing front-end of SCAN+ collects both wireless identifiers and speech data in the same environment. This module is implemented on a WiFi-enabled laptop running Ubuntu 14.04. Our sniffer uses Aircrack-ng [3] and tshark [4] to opportunistically capture the WiFi packets in the vicinity. The captured packet has unencrypted information such as transmission time, source MAC address and the Received Signal Strengths (RSS). As SCAN+ aims to label utterances for POI, our WiFi sniffer only records the packets containing POI's device MAC addresses and discards them otherwise, so as to not harvest addresses from people who have not given consent. A channel hop mechanism is used in the sniffing module to cope with cases where the POI's device(s) may connect to different WiFi networks, namely, on different wireless channels. The channel hop mechanism forces the sniffing channel to change by every second and monitor the active channels periodically (1 second) in the environment. The RSS value in the packet implies how far away the sniffed device is from the sniffer. We put the laptop in the center of the room to sniff the environment in an unbiased manner. The speech data is recorded via the embedded microphone on commercial smartphones, with the sampling rate of 16KHz. Note that, the positions of smartphones were different in various sessions which mimics the real-world complexity.

6.2 Conversation Processing

Discriminative features are important for downstream tasks in SCAN+. In this section, we describe our implementation approach for speech data processing. During experiments, a recorder is set up to log conversations that took place in the monitored environment. Since SCAN+ operates on the utterance-level, whereas the recorded audio file lasts the entire session, speaker diarization hence comes prior to any further steps. Ideally, with utterances of each speaker, conversation processing moves on to feature vector extraction, or voice embedding extraction. The extracted features are then used for linkage tree construction.

Utterance Segmentation: Utterance segmentation is a product of speaker diarization. We adopted the speaker diarization pipeline implemented in Kaldi toolkit [5]. The underlying

3. https://www.aircrack-ng.org/
4. https://www.wireshark.org/docs/man-pages/tshark.html
5. http://kaldi-asr.org/



(a) Pipeline of utterance segmentation.

(b) Pipeline of X-vector (speaker embedding) extraction.

Fig. 6: Steps of conversation processing. The derived segmentations of (a) are input to (b). Acronyms. MFCC: Mel-frequency cepstral coefficients; CMVN: cepstral mean and variance normalization; PLDA: probabilistic linear discriminant analysis.

speaker diarization system operates as a means of intra-context clustering of overlapping sliding windows on the session-wise audio clip. Sliding windows are processed the same way as in speaker recognition pipeline, namely Mel-Frequency Cepstral Coefficients (MFCC) feature extraction, Cepstral mean and variance normalization (CMVN), followed by speaker feature extraction (x-vector in our case). Intuitively, given two consecutive sliding windows, if the latter one contains a change point while the former window is integral, their feature vectors should be significantly dissimilar. In practice, this similarity is measured with a scoring function (PLDA scoring in our case), and is compared with a pre-defined threshold to determine whether or not a change point should be placed.

Voice Embedding: As mentioned above, voice embedding is critical for both speaker recognition and diarization tasks. DNN-based X-vector architecture proposed in [3] was adopted in our experiment, available in Kaldi [34]. The system uses 24 MFCC banks as input features for a time-delayed neural network. After five time-delay layers, a stats pooling layer is used to aggregate frame-level knowledge into segment-level features. The aggregated vector is then passed through several fully-connected layers to generate a high-level speaker embedding. This feature extractor is trained with a softmax cross entropy loss function, and a PLDA backend is adopted for good discriminative results. In our experiment, we used the X-vector feature extractor trained on the augmented VoxCeleb corpus [35] (augmented with MUSAN [36]), and the PLDA backend initially trained on purely VoxCeleb.

7 Performance Evaluation

7.1 Setup

Datasets. Experiments are conducted on both public datasets and our own collected data, denoted as *Public* and *RealWorld* in the following context.

Public dataset is synthesized from VoxCeleb2 [4]. In 7.2, 50 SOIs and 20 non-SOIs are sampled from VoxCeleb2 and distributed into 100 meetings. For experiments in 7.3, the number of non-SOIs varies from 0 to 100, and the amount of sessions is altered from 50 to 150. On average, there are 11 SOIs in each session while each session comprises 97 utterances from SOIs. To represent device heterogeneity, two Gaussians are preset for each SOI (or SOI's device), corresponding to inside-'geofence' and outside-'geofence' RSSI distribution respectively. Then, for each session, RSS values of each SOI (or SOI's device) are sampled according to their presence to reflect the typical characteristics of real world IoT cross-modal data collection.

RealWorld dataset was collected from 49 meetings (i.e., sessions) from three different rooms, with the area of $60m^2$, $20m^2$ and $25m^2$ respectively. These rooms consist of two office and one meeting room located in a modern building. 29 meetings are recorded in the meeting room and the rest are evenly recorded in the two offices. Our real-world dataset contains conversations contributed by 21 *distinct* SOIs and 9 *distinct* non-SOIs attending meetings with between 3 to 5 participants per meeting. The average number of people per session 3.22 , derived from $\frac{\sum_{i=1}^{S} P_i}{S}$, where S is the number of recorded sessions and P_i represents the number of participants in i-th session. Collected audio recordings were segmented into 3,555 utterances via a diarization system available in Kaldi [34] and ground truth attendance was labeled manually. In particular, conversations were recorded by three different mobile microphones at the sampling rate of 16 kHz. A WiFi sniffer was deployed inside the environment to continuously scan the ambient WiFi identifiers in the vicinity. Observation errors exist due to reasons such as speakers leaving their devices behind, being disconnected from WiFi, or being incorrectly detected when not actually present, e.g., working in the office next to the meeting room. The evaluation purpose of this dataset is to examine the performance in real-world scenarios where speaker and device observations are noisy.

Competing Approaches. The approaches being evaluated are sequential clustering and association (denoted as *Sequential*), Simultaneous Clustering And Naming (denoted as *SCAN*) and `SCAN+`, which augmented SCAN by iterative curation. Depending on which model is adopted in curation, `SCAN+` can be further classified into HardSCAN+ and SoftSCAN+, corresponding to hard and soft geofence modeling respectively. HardSequential and SoftSequential are likewise denoted.

Evaluation Metrics. Our evaluation examines two types of performance, in both offline utterance labeling and online speaker identification. Specifically, we use *precision*, *recall* and *F1 score* to evaluate the purity of utterance labeling, which follows the convention in [37]. In order to find to what extent the automatically labeled utterance can help develop a speaker recognition system, we use equal error rate (EER) to evaluate the developed system on a held-out

	Public				RealWorld			
	Sequential			SCAN	Sequential			SCAN
Algorithm	KMeans	Spectral	Hierarchical	-	KMeans	Spectral	Hierarchical	-
Precision	0.257	0.472	0.568	**0.740**	0.392	0.539	0.598	**0.685**
Recall	0.406	0.526	0.606	**0.715**	0.366	0.649	0.658	**0.705**
F1-score	0.315	0.498	0.587	**0.727**	0.379	0.589	0.626	**0.695**

TABLE 1: Comparing SCAN with the Sequential Clustering and Naming methods using different clustering algorithms.

dataset. EER is the most widely adopted metric in speaker verification and summarizes both false positive and false negative errors [3].

7.2 Core Experiments

Experiments in this section are conducted on both *Public* and *RealWorld* datasets. Particularly, the *Public* dataset contains 100 sessions generated by 50 SOIs and 20 non-SOIs.

Cross-modality Labeling. To start with, we compared SCAN with Sequential method where device heterogeneity was not taken into consideration, i.e., without curation. Different clustering algorithms were applied to perform clustering step in Sequential method, namely K-Means, spectral clustering and hierarchical clustering. Results were presented in Table 1. It can be observed that on both *Public* and *RealWorld* dataset, SCAN outperformed all sequential approaches. Respectively, on *Public* dataset, the F1-score of SCAN is at least 23.9% higher than the baseline sequential method, while on *RealWorld* dataset, SCAN maintained its advantage with over 7% gain in F1-score. Such differences indicate that, when clustering and naming steps are independent, the correctness of association depends greatly on the chosen clustering method and the underlying number of clusters in feature space. In our case, the compound feature is friendly to hierarchical clustering. More importantly, because of noises like non-SOI's and diarization error, uncertainties are also introduced when determining the number of clusters. In contrast, SCAN averts the decision of number of clusters by incorporating the data association plan into clustering. As a result, only clusters that maximize both the linkage the association score are selected, whilst the majority of the membership noise is discarded through optimization. Among all three sequential methods, hierarchical clustering appears to perform better with the hybrid features. Therefore, for the rest of the experiments, the baseline sequential method was chosen as the one with hierarchical clustering algorithm.

Iterative Curation. In this section, experiments focus on iterative curation in `SCAN+`. Two strategies of device geofence threshold adaptation are evaluated, namely hard geofence and soft geofence modeling. Baselines are set to the case where device presence is not updated (e.g. Sequential and SCAN). Results are shown in Fig. 7. From the chart we can see that `SCAN+` is always superior to the sequential method when the same curation strategy is deployed in each variant, indicating that SCAN's advantage is preserved when curation is incorporated. It is also clear that the hard geofence is outperformed by soft geofence modeling on both *Public* and *RealWorld* datasets. This is due to the fact that representing a device's presence discretely when deciding a geofence can amplify the effect of erroneous clustering, which is usually the case. On contrary, soft geofence manages to alleviate the impact of incorrect decisions by treating the presence of a device probabilistically. The probability distribution is derived from the previous clustering results, from which the distributions are approximated. These reflect the characteristics of a device being inside a geofence or outside a geofence. Hence, soft geofence modeling is more tolerant of RSS values that lie on the junction.

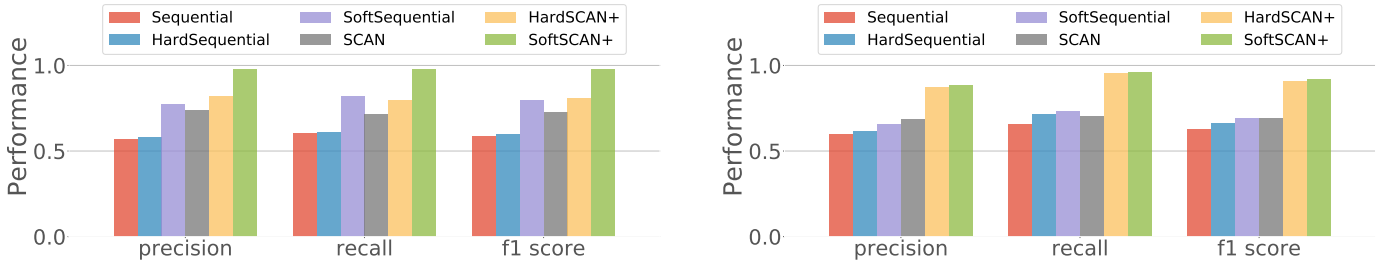
Online Speaker Recognition. We are now in a position to evaluate the effectiveness of `SCAN+` for online speaker recognition in a new domain. In particular, The association results of SoftSCAN+ are utilized to update the feature extractor as it got the best labelling performance among all variants of `SCAN+`. To incorporate associated instances in the feature extractor, we retrained the PLDA backend on the mixture of VoxCeleb and the labelling result of SoftSCAN+. The new feature extractor is evaluated by speaker verification task and speaker identification task on the held-out test set of *Public* and *RealWorld* datasets and we compare it to the following three baselines: (1) non-adaptation, (2) adapted by the limited in-domain labelled data only and (3) unsupervised adaptation [38]. Table. 2 shows that for speaker verification tasks, the feature extractor adapted by SoftSCAN+ outperforms other baseline. Our proposed method is able to achieve EERs of 3.32 and 6.96 on the public and real-world datasets respectively. This accuracy is over 20% better than non-adapted cases. Additionally, we found that the mixture of VoxCeleb data in PLDA adaption can bring a clear advantage ($\sim 6\%$ gain) over purely in-domain data adaptation. This can be ascribed to PLDA's requirement on training data size, where a small-scale dataset cannot be used to effectively estimate a discriminative feature subspace. This observation is also found in [39]. Last, although the unsupervised method achieved the second-best result on public dataset, it struggles on the real-world dataset, due to more significant domain differences. The speaker identification performance showed in Fig. 8 further confirms the superiority of SoftSCAN+, where it consistently outperforms other baselines in all levels of cumulative accuracy. The above results indicate that a PLDA feature extractor trained on a different domain can be adapted to target domain in a weakly supervised manner via `SCAN+`.

Effectiveness of Geofence Personalization Finally, we evaluate the effectiveness of the personalized geofences on determining whether a device is inside or outside the room. In an additional experiment, 10 SOI were asked to stay

	non-adapted	limited Indomain	Unsupervised	Ours
Public	4.41	3.98	3.75	**3.32**
Real World	8.76	7.40	8.44	**6.96**

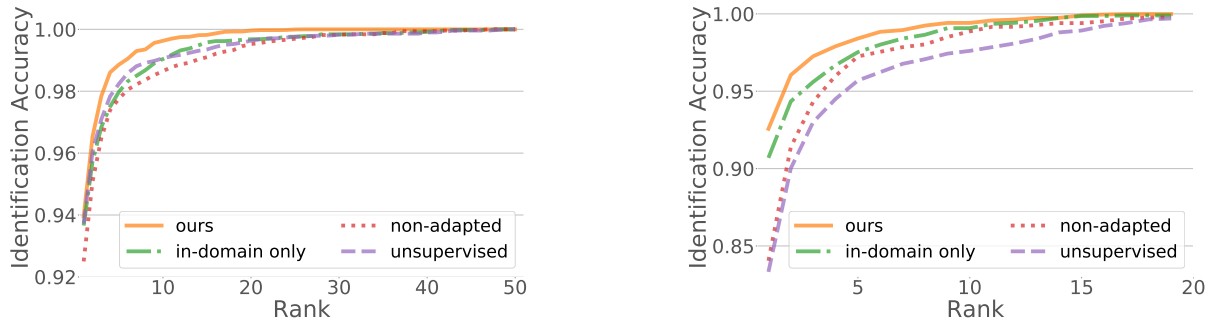
TABLE 2: EER(%) of speaker verification on *Public* and *RealWorld* datasets using different PLDA adaptation methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2926645, IEEE Internet of Things Journal



(a) *Public* dataset. (b) *RealWorld* dataset.

Fig. 7: Overall performance comparison with different curation (none, hard, soft) methods on *Public* and *RealWorld* datasets.



(a) *Public* dataset. (b) *RealWorld* dataset.

Fig. 8: Performance of speaker identification on *Public* and *RealWorld* datasets using different adaptation methods.

inside and then outside our experiment testbed. During the experiment, our sniffing system continuously recorded their device RSS values. By using different geofence models curated in the testbed, we compare their performance on this new data collection. The global geofence model is a threshold calibrated by a single smartphone. As shown in Fig. 9, the best overall accuracy ($\sim 95\%$) can be achieved when adopting a soft geofence model in SCAN+. Using the hard-version of SCAN+ gives the second-best performance. As expected, the global fence is biased to the calibrated smartphone and generalizes poorly to other devices. Despite its superiority in inside-room detection, the global geofence failed to reliably detect outside-room events. Overall, the baseline methods based on sequential clustering and association are inferior, and can be worse than the global threshold model when no personalised geofence is used. This is because the sequential method can incur significant association errors which will drive the geofence update in the incorrect direction.

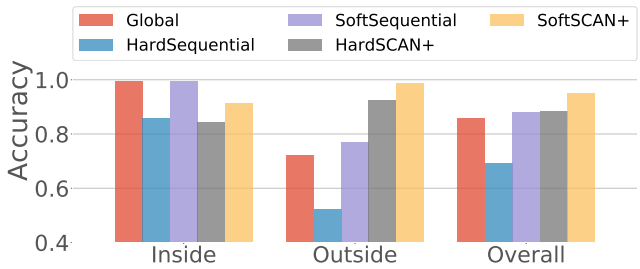


Fig. 9: Effectiveness of Geofence Personalization. 'Global' method means to use a pre-defined RSS value calibrated via a certain device to globally fence all devices.

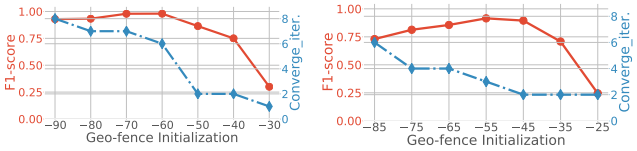
7.3 Sensitivity Analysis

The following experiments are carried out using SCAN+ with soft geofence model. Note that we fix the number of SOI in all the sensitivity tests to ensure fair comparisons.

Impact of Initial Geofence Value. In this section, we varied geofence initialization by using different RSS thresholds at the beginning of curation. Results presented in Fig. 10 indicate that when the initial geofence threshold is set too large. e.g., -30dB for *Public* dataset, -25dB for *RealWold* dataset, association accuracy drops drastically. Under such conservative initialization, multiple devices that were present in the meeting room are falsely ruled out due to their weak signal reception ability. As a result, a large portion of RSS statistics that should be used to update the inside-geofence model is excluded and were undesirably included for outside-geofence model update. In contrast, using a very small geofence threshold (e.g., -85dB) wrongly included absent devices, though its impact is relatively mild. In summary, the accuacy of SCAN+ is relatively insensitive to a sensible initial geofence initialization, and is able to operate comparably when initial values lie in the range of $[-70, -45]$dB. Lastly, we also found that the number of iterations required for convergence decreases when the initial value becomes larger. This behaviour is the natural consequence of the decreased number of SOIs' devices present in sessions incurred by overly-conservative initialization.

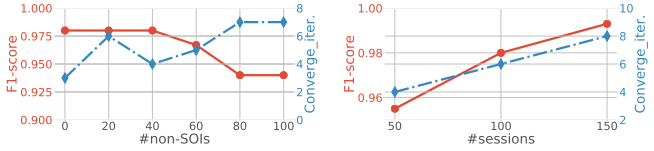
Impact of Number of Non-SOIs. In this section, we varied

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2926645, IEEE Internet of Things Journal



(a) *Public* dataset. (b) *RealWorld* dataset.

Fig. 10: Impact of Geofence initializations on *Public* and *RealWorld* datasets.



(a) Number of non-SOIs. (b) Number of sessions.

Fig. 11: Impact of number of non-SOIs and sessions on *Public* dataset.

the number of non-SOIs in *Public* dataset with each non-SOI contributing 63.75 utterances on average, while the number of sessions remains the same. Results in Fig. 11a show that along with the increase of non-SOIs as well as their utterances, the association accuracy of SCAN+ with soft geofence model slowly degrades. The system achieved a high F1-score at 98% when there are no non-SOI presented, and managed to maintain such performance up to 40 non-SOI. Even though the performance decreased by 1.3% when the amount of non-SOIs increased to 60 and 4.1% when scaled up to 100, SCAN+ managed to discard most of the non-SOI noise and maintain an acceptable association. This can be ascribed to the fact that when the number of non-SOIs grows, the possibility of confusing a SOI with some non-SOI raises. As a result, when the vocal feature of non-SOIs cannot be effectively separated from SOIs, device presence vectors will be erroneous and will in turn affect the similarity between clusters and SOIs. Furthermore, it is also possible that a non-SOI will share the same attendance with some SOI from the beginning or during curation. Hence, these two speakers become indistinguishable. With respect to convergence efficiency, Fig. 11a indicates that the number of iterations until convergence tends to increase with the number of non-SOIs, which is related to data volume and complexity.

Impact of Number of Sessions. In this section, we kept the number of non-SOIs unchanged and varied the number of sessions, simulating the growing number of sessions being recorded in real-life. As depicted in Fig. 11b, the increase in sessions benefits the association result. For *Public* dataset with 50 SOIs and 20 non-SOIs, when 50 sessions are provided, soft SCAN+ achieved about 95.5% F1-score. The performance was improved by 2.6% when twice the number of sessions are provided; and improved by 4.0% when there are 150 sessions. The improvement in performance is easy to comprehend in that with more sessions recorded, the speakers become more distinguishable since more variations are introduced into device presence. On the other hand, more sessions imply a rise in the number of iterations required for convergence as indicated by Fig. 11b.

8 Discussion and Future Work

Privacy Concerns: In practice, SCAN+ requires voice features and device IDs of users to operate, which may impact user privacy, if used without consent. For example, a user may be able to be identified without explicit consent in a new environment, if the owner has the access to the voice or WiFi sniffing data of this user. In this work, we do not explicitly study the attack model in this context. However we note that potential privacy concerns are worth exploring in future work.

Extensibility: Biometric recognition has gained increasing popularity in ubiquitous computing, and applications range from personalized service to secure ubiquitous devices. Unfortunately, an accurate recognition system usually relies on a large amount of labeled biometric data to train a classifier, which is often costly to attain. In this work, we only considered autonomous speaker identification, although there are an increasing number of biometrics, such as gait, height and visual features that are targeted towards widespread passive observations of users in buildings. However, the framework and philosophy of SCAN+ can readily generalize to other biometric features (e.g., facial images and gaits) in smart spaces, by utilizing other co-located digital IDs (e.g., WiFi MAC addresses, email accounts and etc.). Combinations of the above are interesting future research directions.

Utterance Segmentation: Segmented utterances are the smallest units in SCAN+, and the performance of SCAN+ relies on the segmentation quality. An ideal segmentation should only contain the voice data of a single person. In this work, we adopted the segmentation module from the speaker diarization system proposed in [40], which is the state-of-the-art technique in DIHARD 2018 competition [6]. As DIHARD is a challenging contest to evaluate the performance of diarization systems in the wild, it is worth exploring the segmentation modules of other diarization systems with top scores in this competition, such as [41], [42], [43].

9 Conclusion

In this paper, we proposed SCAN+, a system that automatically learns speaker identity and adapts WiFi geofence model via cross-modal labelling. We show that *co-located* microphones and WiFi sniffers can complement the knowledge base for each other, and lead to speaker recognition systems with *zero* user effort. In particular, a novel method is proposed that simultaneously clusters voices and associates device IDs, which addresses the issue of disturbing non-SOI. In addition, an iterative optimization framework is proposed to automatically customize geofence and tackle the device heterogeneity issue. Experimental results in different scenarios indicate that SCAN+ is able to achieve 2-fold improvement compared with conventional methods and can achieve reliable speaker recognition in the wild.

6. https://coml.lscp.ens.fr/dihard/2018/results.php

REFERENCES

[1] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[4] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[5] K. Wang, N. An, B. N. Li, Y. Zhang, and L. Li, "Speech emotion recognition using Fourier parameters," *IEEE Transactions on Affective Computing*, 2015.

[6] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on Audio, Speech, and Language Processing*, 2006.

[7] Amazon. Amazon mechanical turk. [Online]. Available: https://www.mturk.com/

[8] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov *et al.*, "Devise: A deep visual-semantic embedding model," in *Advances in neural information processing systems*, 2013, pp. 2121–2129.

[9] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[10] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 3156–3164.

[11] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729*, 2014.

[12] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient sound provides supervision for visual learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 801–816.

[13] A. Nagrani, S. Albanie, and A. Zisserman, "Seeing voices and hearing faces: Cross-modal biometric matching," *arXiv preprint arXiv:1804.00326*, 2018.

[14] C. L. Zitnick, D. Parikh, and L. Vanderwende, "Learning the visual interpretation of sentences," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1681–1688.

[15] J. Teng, B. Zhang, J. Zhu, X. Li, D. Xuan, and Y. F. Zheng, "Ev-loc: integrating electronic and visual signals for accurate localization," *IEEE/ACM Transactions on Networking*, 2014.

[16] A. Alahi, A. Haque, and L. Fei-Fei, "Rgb-w: When vision meets wireless," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 3289–3297.

[17] S. Papaioannou, H. Wen, Z. Xiao, A. Markham, and N. Trigoni, "Accurate positioning via cross-modality training," in *ACM SenSys*, 2015.

[18] S. S. Blackman, "Multiple-target tracking with radar applications," *Dedham, MA, Artech House, Inc., 1986, 463 p.*, 1986.

[19] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334–352, 2004.

[20] X. R. Li and Y. Bar-Shalom, "Tracking in clutter with nearest neighbor filters: analysis and performance," *IEEE transactions on aerospace and electronic systems*, vol. 32, no. 3, pp. 995–1010, 1996.

[21] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems*, vol. 29, no. 6, 2009.

[22] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *The International Journal of Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.

[23] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[24] N. Dehak and et al., "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[25] H. Bredin, "Tristounet: Triplet loss for speaker turn embedding," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5430–5434, 2017.

[26] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. of Interspeech*, 2017.

[27] R. Jonker and T. Volgenant, "Improving the hungarian assignment algorithm," *Operations Research Letters*, 1986.

[28] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM, 1984, pp. 302–311.

[29] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management science*, vol. 27, no. 1, pp. 1–18, 1981.

[30] H. Zou, B. Huang, X. Lu, H. Jiang, and L. Xie, "A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1252–1266, 2016.

[31] J. Yang, H. Zou, H. Jiang, and L. Xie, "Device-free occupant activity sensing using wifi-enabled iot devices for smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3991–4002, 2018.

[32] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 205–218.

[33] H. Zou, Y. Zhou, J. Yang, and C. J. Spanos, "Unsupervised wifi-enabled iot device-user association for personalized location-based service," *IEEE Internet of Things Journal*, 2018.

[34] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[35] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[36] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[37] C. X. Lu, H. Wen, S. Wang, A. Markham, and N. Trigoni, "Scan: learning speaker identity from noisy sensor data," in *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2017, pp. 67–78.

[38] D. Garcia-Romero, X. Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 378–383.

[39] M. H. Rahman, A. Kanagasundaram, I. Himawan, D. Dean, and S. Sridharan, "Improving plda speaker verification performance using domain mismatch compensation techniques," *Comput. Speech Lang.*, vol. 47, no. C, pp. 240–258, Jan. 2018.

[40] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *et al.*, "Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge," in *Proc. Interspeech*, 2018, pp. 2808–2812.

[41] M. Diez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Zmolıková, O. Novotnỳ, K. Veselỳ, O. Glembek, O. Plchot *et al.*, "But system for dihard speech diarization challenge 2018," in *Proc. Interspeech*, 2018, pp. 2798–2802.

[42] L. Sun, J. Du, C. Jiang, X. Zhang, S. He, B. Yin, and C.-H. Lee, "Speaker diarization with enhancing speech for the first dihard challenge," *Proc. Interspeech 2018*, pp. 2793–2797, 2018.

[43] I. Vinals, P. Gimeno, A. Ortega, A. Miguel, and E. Lleida, "Estimation of the number of speakers with variational bayesian plda in the dihard diarization challenge," in *Proc. INTERSPEECH*, 2018, pp. 2803–2807.