This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

Self-Supervised Association of Wi-Fi Probe Requests Under MAC Address Randomization

Tianlang He, Jiajie Tan, and S.-H. Gary Chan, *Senior Member, IEEE*

Abstract—Wi-Fi-on devices such as smartphones search for network availability by periodically broadcasting probe requests which encapsulate MAC addresses as device identifiers. To protect identity privacy, modern devices embed random MAC addresses in probe frames, the so-called MAC address randomization. Such randomization disrupts the frame association, inadvertently frustrating identity-oblivious statistical analytic efforts such as people counting and trajectory inference. To address that, we propose ***Cappuccino***, a novel privacy-preserving approach that captures the association of probe requests under MAC address randomization. Cappuccino first estimates pairwise frame correlation and then associates frames over time. For frame correlation, it employs a self-supervised estimator that jointly considers multiple modalities, i.e., information elements, sequence number, and received signal strength. For multiple frame association, Cappuccino formulates frames as a minimum-cost flow optimization. To the best of our knowledge, this is the first piece of work that leverages self-supervised learning to estimate frame correlation based on multiple modalities and formulates the probe request association problem as the network flow optimization. We have conducted extensive experiments in a leading and crowded shopping mall for more than three months. Cappuccino achieves remarkable performance in terms of V-measure scores (> 0.85).

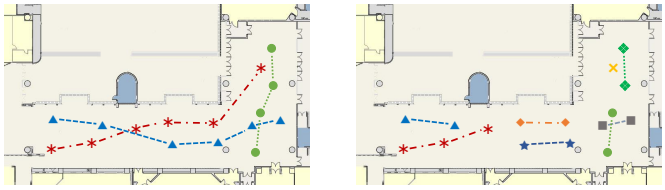
Index Terms—Wi-Fi, MAC address randomization, self-supervised learning, multiple modalities, network flow optimization

1 Introduction

Wi-Fi-on devices periodically search for network availability by broadcasting probe requests. In the past, the request frames used physical media access control (MAC) addresses to communicate with the access points (APs). This fixed MAC address is a unique device identifier and may expose personal information – by collecting the probes with such MAC addresses over time, one may re-identify a device across sites (such as different buildings) and leverage the device trajectory/location [1] to determine its user identity, even though the device is not connected with any network. This has raised grave privacy concerns on user identity and location.

To protect privacy against Wi-Fi sensing, MAC address randomization has been recently implemented in modern commercial devices [2], [3], including most smartphones with operating system iOS [4] and Android [5]. Instead of using the *real* physical MAC address in probe frames, the device generates randomized *virtual* addresses at unpredictable times. In other words, the probe requests emitted from a single device no longer carry the same MAC address but change to some random addresses once in a while.

While protecting privacy, a direct consequence of MAC address randomization is that it breaks the continuity and semantics of probe requests, leading to fragmentation in data analytics. This adversely and inadvertently frustrates device-oblivious statistical gathering efforts such as people counting [6], crowd flow estimation [7], and trajectory inference [8], [9]. As an illustration of the impact, we show in Figure 1 that three users carry Wi-Fi-on smartphones walking through an area. Figure 1(a) depicts the three complete trajectories if MAC addresses were not randomized. In reality, due to address randomization, these trajectories would be fragmented. Figure 1(b) shows a possible sensing result with eight fragmented trajectories despite only three devices.



(a) Complete trajectories for the case without MAC address randomization

(b) Fragmented trajectories in the case of MAC address randomization

Fig. 1. An illustration showing the influence of MAC address randomization in a tracking scenario. The markers represent the positions where probe requests are emitted. Each type of marker corresponds to a MAC address.

In this paper, we study how to *associate* probe requests with their emitter under MAC address randomization – by correlating these frames through this association process, we seek to recover Figure 1(a) from Figure 1(b) for our example above. Such association cannot uncover the underlying real MAC addresses and only associate consecutive probe requests emitted from the same devices, making it impossible to track or re-identify user devices across sites. Thus, it enables more fruitful and meaningful anonymised privacy-protecting data analytics under MAC address randomization.

Without the real MAC address, existing works leverage other modalities from probe requests as alternative to de-

• *The authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. E-mail: {theaf, jtanad, gchan}@cse.ust.hk*

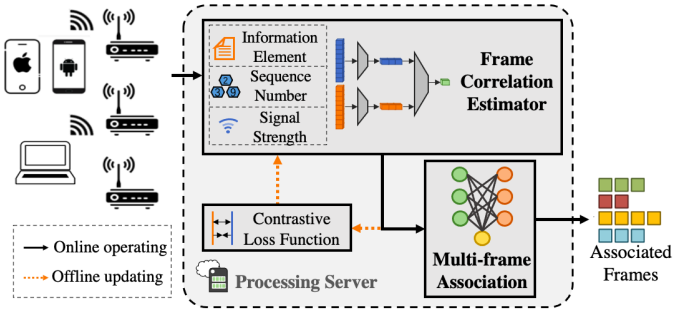


Fig. 2. The system framework of Cappuccino.

termine if two frames are associated – they either leverage location continuity or use one of the frame payloads (such as information element [4] and sequence number [10], [11]). However, these approaches often fail in the real-world cases where many Wi-Fi-on devices are in close proximity to each other. Under such crowded scenarios, neither location nor those single-modal payloads is sufficient to differentiate senders/emitters with a high accuracy. Also, one frame may have many candidate frames to associate in the crowd case; the current methods suffer a significant ambiguity (or conflict) when there are multiple candidates are to be associated to a frame (association conflict).

To address the above, we propose ***Cappuccino***, a simple and novel approach that ***cap***tures ***p***robe req***u***est asso***c***iation effi***c***iently under MAC address ra***n***d***o***mization. Figure 2 shows the system framework of Cappuccino. In the online phase, Wi-Fi sensors deployed in the venue capture probe requests from nearby devices and transmitted them to a processing server, where the probe requests are associated for online operating or stored for offline updating/training. On the server, a frame correlation estimator based on multiple modalities periodically (e.g., every 30 seconds) estimates the pairwise correlation probabilities of the received frames. Then, a multi-frame association module associates these probabilities, alleviating the association conflict. In the offline phase, Cappuccino self-supervise itself by learning from the stored probe requests.

The major novelties of this work are as follows:

1) *Multi-modal association features*: To tackle real-world crowded and complex scenarios, we estimate the pairwise frame association in terms of device, time and space – we jointly capture features from *information element* (IE), *sequence number* and *signal strength*. These multiple modalities, which are widely available in frame attributes, complement with each other, and thus, provide better association features than the single modality.
2) *Self-supervised frame correlation estimator*: We employ a neural network with Siamese structure to capture association features from the multiple modalities. To reduce its training efforts, we leverage contrastive learning to avoid manual data labeling. Specifically, we differentiate frames with real MAC addresses from virtual MAC addresses by the 7^{th} bit of the input frames (which is grounded by the guideline [12]) and use the real MAC addresses to create positive and negative data pairs for estimator training/updating. Thus, our system is self-supervised and can adapt to environmental changes.
3) *Multi-frame association to alleviate association conflict*: Only considering pairwise association may lead to the association conflict; thus, given the pairwise correlation probabilities, we study to achieve a global optimal association over a batch of probe requests. The problem has not been considered before. We formulate it to be a minimum-cost flow problem where a node represents a frame and the edge between nodes are a decreasing function of the pairwise association probability. A flow over an edge indicates that the two frames are associated. By seeking flow with the minimum cost, Cappuccino obtains the optimal association for all the frames in the batch. Besides, to deal with a large batch of frames, Cappuccino further employs a mini-batch design to improve its efficiency.

Cappuccino works well even in crowded cases and is designed for existing Wi-Fi infrastructure without any special hardware beyond regular Wi-Fi sensors or APs. It requires neither external localization system nor manual calibration/labeling beforehand. We have implemented it and conducted extensive experiments in a leading shopping mall for more than three months. The results show that Cappuccino achieves remarkable performance in terms of discrimination accuracy ($> 80\%$) and V-measure scores (> 0.85), which outperforms state-of-the-art by 27% discrimination error reduction and 4% clustering completeness (covered in Section 6.1) average improvement.

The remainder of this paper is organized as follows. We review the related work in Section 2, followed by the preliminaries of Cappuccino in Section 3. Then we present in Section 4 the multi-modal frame correlation estimator, and detail the association algorithm for multiple frames in Section 5. Section 6 illustrates the experimental results. We conclude in Section 7.

2 Related Works

Prior works for frame association leverage device-specified IE contents to fingerprint devices [4], [10], [13], [14]. Specific fields such as transmission rate [13] and service set identifier (SSID) are used to distinguish between devices [10], [14]. Vanhoef *et al.* further explore using the combination of multiple IE fields as fingerprints. They analyze the field discriminability and utilize the most discriminative and stable fields to form fingerprints [4]. Though promising, IE alone is not robust against the diversity of devices because different devices (say, smartphones of the same model) may emit probe requests of the same IEs, and the same device may produce different IEs. By contrast, our work uses the multimodality of probe requests, i.e., information element, sequence number, and received signal strength, to accurately associate frames.

In contrast to content-based fingerprinting mentioned above, some works study inter-frame patterns to distinguish frames from different devices. Clock skew, the inherent drifts of clock in a device, has been explored to differentiate transmitting devices [15], [16]. The works in [11], [17] use the arrival time between frames as a unique pattern to identify devices. Bezawada *et al.* extract features from network traffic to form fingerprints [18]. However, these works require either specialized hardware or a large number

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

of consecutive probe requests; therefore, they are not cost-effective in deployment. On the other hand, the continuity of sequence numbers is another effective indicator for consecutive frames. Frame association can be constructed by predicting the sequence number of the next emitting frame [4], [19], [20]. Cappuccino also utilizes the sequence number as one of the features in frame correlation estimation. In contrast to the deterministic methods given above, Cappuccino proposes a probabilistic scheme in order to adapt to heterogeneous devices and avoid dedicated calibration.

Some works take advantage of the leaks in protocols or system designs to obtain the device identifiers. Probe requests from certain devices may carry the universally unique identifier-enrollees (UUID-Es), which are derived from their MAC addresses. The works in [4], [21] reverse-engineer the UUID-E to recover the original MAC addresses via pre-computed hash tables. The work in [3] infers the real Wi-Fi MAC address from the Bluetooth MAC address as lots of manufacturers assign consecutive MAC addresses for the two interfaces of the same phone. Vanhoef *et al.* propose to set up fake APs with usual SSIDs (e.g., "Starbucks" and "Airport") [4], so that devices would expose their true MAC addresses when they auto-connect to these networks. These schemes assume special conditions and hence cannot be generalized to heterogeneous devices and different scenarios. Furthermore, some of these approaches also raise privacy or security concern because they attempt to forcefully acquire the true MAC addresses of devices. On the contrary, Cappuccino uses only the universal information available on almost every device and fully respects user privacy. The constructed association is not able to link to user identities or does not extract user locations.

Data association is to find matching between two sets of objects. It is conventionally used in correlating measurements with targets in multi-object tracking problems [22], [23], [24]. The applications have been extended to wilder fields such as multi-sensor data fusion [25], simultaneous localization and mapping (SLAM) [26], [27], person re-identification in visual surveillance system [28], [29], etc. However, none of these prior works has considered the association of multiple probe requests. Cappuccino proposes a novel, efficient and self-supervised multi-frame association algorithm by formulating it as a minimum-cost flow problem. To the best of our knowledge, this is the first piece of work that estimates frame correlation based on multi-modal deep learning and formulates the probe request association problem as network flow optimization.

A preliminary version of this work termed Espresso has appeared in [30]. Espresso and Cappuccino share the same multi-frame association, while Espresso's frame correlation estimator is based on Bayes' theorem. As for the multiple modalities, it separately employs a linear regression, a Gaussian mixture model, and a K-means clustering model to estimate correlation probability. However, Espresso's frame correlation estimator requires offline site-dependent parameter tuning, leading to much manual training effort on deployment. Also, Espresso suffers ambiguity when a device emits two probe requests in a short period — for example, the third frame emitted from the same device may get a higher correlation probability to associate the first frame than the second frame. The current work advances from it by proposing a new end-to-end frame correlation estimator based on contrastive learning, which does not require any offline calibration for its deployment. Since it leverages contrastive loss to guide/self-supervise all correlation probability between associated frames to be identical, it is less likely to mismatch frames aiding with a time decay. Therefore, it maintains high clustering completeness and thus outperforms Espresso in accuracy. We also empirically show that Cappuccino's frame correlation estimator is ~ 270 times faster than Espresso with an off-the-shelf GPU.

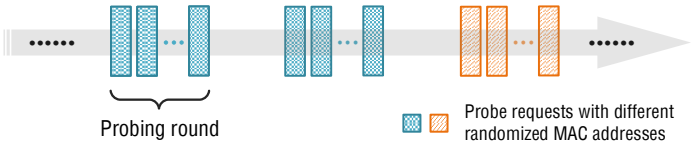


Fig. 3. An illustration of active scan under MAC address randomization over time. The bars represent the probe requests emitted from a single device. Different colors and patterns indicate different randomized MAC addresses in the frames.

3 Preliminaries

In this section, we introduce the preliminaries of probe requests (Section 3.1) and the modeling of frame association (Section 3.2).

3.1 Probe Request

Active scan is one of the major methods that Wi-Fi devices use to discover nearby wireless networks [31]. During the scanning, devices initiate a network search by broadcasting management frames known as *probe request*s.

The scan is generally triggered on a periodic basis in order to reduce energy consumption. Figure 3 illustrates the process of active scan over time. To discover all the networks, devices need to probe every available channel. We refer to the probe request emission of one scan as a *probing round*. The duration of a probing round is about $1\,\mathrm{s}$ to $4\,\mathrm{s}$ subject to the number of scanned channels. In most cases, the MAC addresses remain unchanged during a probing round (whether it is real or virtual). Besides that, the trigger of randomization is not necessarily synchronized with probing rounds, and a device may use the same MAC address in different consecutive rounds. The time interval between two consecutive probing rounds is subject to the factory configurations of devices.

In Cappuccino, we represent the i-th captured probe request as the tuple $\boldsymbol{P}_i = \langle \boldsymbol{I}_i, s_i, \boldsymbol{R}_i, t_i \rangle$, where $\boldsymbol{I}_i$ is the IE vector, s_i is the sequence number, $\boldsymbol{R}_i$ is the RSS vector, and t_i is the transmission time. The IE vector $\boldsymbol{I}_i = \langle I_i^h \,|\, 1 \le h \le |\boldsymbol{I}| \rangle$, where I_i^h is the h-th field in frame i and $|\boldsymbol{I}|$ is the total number of fields. Note that some IE fields (e.g., *Vendor Specific*) may appear multiple times in a frame. We concatenate them by their presenting order and regard the combination as the content of the field. The sequence number s_i is a 12-bit counter indicating the transmission order on the device. It is bounded between 0 and 4095. The RSS vector of frame i is denoted as $\boldsymbol{R}_i = \langle r_i^u \,|\, 1 \le u \le |\boldsymbol{R}| \rangle$, where r_i^u is the signal strength measured by sensor u, and $|\boldsymbol{R}|$ is the total number of sensors.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

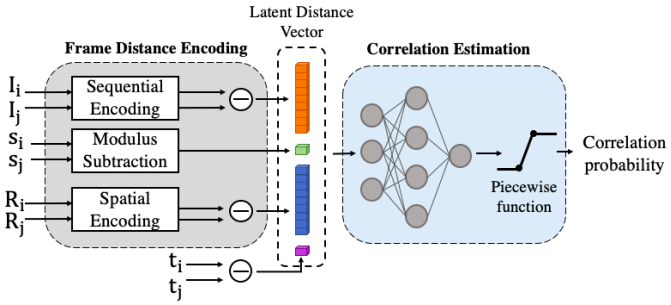


Fig. 4. Model structure of frame correlation estimator.

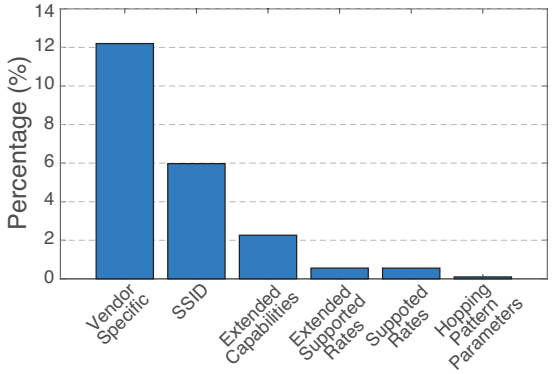


Fig. 5. The ratio of distinct IE fields in the probe request pairs.

3.2 Modeling of Frame Association

As the frames in the same probing round usually share the same MAC address [11], we are more interested in associating frames between different probing rounds. In Cappuccino, we consider frames to be in the same round if they have identical MAC addresses and their adjacent time intervals are less than 1 s. We randomly pick one frame from each probing round to represent it. In the following, we work on the set of selected frames $\boldsymbol{\mathcal{P}} = \{\boldsymbol{P}_i \,|\, 1 \le i \le M\}$, where M is the number of frames (rounds).

Let $\boldsymbol{\mathcal{D}}$ be the set of transmission devices in $\boldsymbol{\mathcal{P}}$. We use $\boldsymbol{\mathcal{S}}_k = \{\boldsymbol{P}_l^k \,|\, 1 \le l \le M_k, \boldsymbol{P}_l^k \in \boldsymbol{\mathcal{P}}\}$ to denote the sequence of probe requests from a device $d_k \in \boldsymbol{\mathcal{D}}$, where M_k is the number of frames in the sequence. In the frame sequence $\boldsymbol{\mathcal{S}}_k$, we name $\boldsymbol{P}_{l-1}^k$ to be the *predecessor* frame of $\boldsymbol{P}_l^k$ ($2 \le l \le M_k$), and $\boldsymbol{P}_{l+1}^k$ as the *successor* frame of $\boldsymbol{P}_l^k$ ($1 \le l \le M_k - 1$).

Given above, we define that two probe requests $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ ($1 \le i < j \le M$) are *associated* if $\boldsymbol{P}_j$ is the successor of $\boldsymbol{P}_i$ (or $\boldsymbol{P}_i$ is the predecessor of $\boldsymbol{P}_j$). We use a binary indicator $\boldsymbol{\Phi}_{ij}$ to denote the association between $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$, where

$$\boldsymbol{\Phi}_{ij} = \begin{cases} 1, & \boldsymbol{P}_j \text{ is associated to } \boldsymbol{P}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Note that the association $\boldsymbol{\Phi}_{ij} = 1$ also implies that $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ are transmitted from the same device. In this work, we aim to find the associations in the probe request set $\boldsymbol{\mathcal{P}}$.

4 Frame Correlation Estimator

In this section, we present the multi-modal correlation estimator between any two probe requests. We first overview in Section 4.1 the structure of the frame correlation estimator based on contrastive learning, which consists of frame distance encoding and correlation estimation. Then we discuss the frame distance encoding for multi-modal inputs in Section 4.2. Finally, we introduce correlation estimation in Section 4.3.

4.1 Estimator Based on Contrastive Learning

The association correlation between probe frames $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ is defined as their association probability $p(\boldsymbol{\Phi}_{ij} \,|\, \boldsymbol{P}_i, \boldsymbol{P}_j)$. In particular, $p(\boldsymbol{\Phi}_{ij} \,|\, \boldsymbol{P}_i, \boldsymbol{P}_j)$ should be significantly high if P_j is the successor frame of P_i, i.e.,

$$p(\boldsymbol{P}_j | \boldsymbol{P}_i, \boldsymbol{\Phi}_{ij} = 1) \gg p(\boldsymbol{P}_j | \boldsymbol{P}_i, \boldsymbol{\Phi}_{ij} = 0). \tag{2}$$

We can achieve this by learning a metric $M_\theta(\boldsymbol{P}_i, \boldsymbol{P}_j) \in \mathbb{R}$ to measure how likely $\boldsymbol{P}_j$ is associated to $\boldsymbol{P}_i$ (namely, frame distance), where θ represents the model parameters. We map frame distance to probability by simply resorting to a piecewise function

$$g_\theta(\boldsymbol{P}_i, \boldsymbol{P}_j) = \begin{cases} 0, & M_\theta(\boldsymbol{P}_i, \boldsymbol{P}_j) < 0, \\ 1, & M_\theta(\boldsymbol{P}_i, \boldsymbol{P}_j) \ge 1, \\ M_\theta(\boldsymbol{P}_i, \boldsymbol{P}_j), & \text{otherwise.} \end{cases} \tag{3}$$

Thus, we can learn the metric parameters by optimizing the contrastive loss [32] on frame association

$$\begin{aligned} \arg\min_\theta \sum_{P_i \in \boldsymbol{\mathcal{P}}} \sum_{P_j \in \boldsymbol{\mathcal{P}}, j > i} & (1 - \Phi_{ij}) g_\theta^2(\boldsymbol{P}_i, \boldsymbol{P}_j) \\ & + \Phi_{ij} \left[1 - g_\theta(\boldsymbol{P}_i, \boldsymbol{P}_j)\right]^2, \end{aligned} \tag{4}$$

where $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ are associated and non-associated frame pairs sampled from historical physical probe requests.

We evaluate the frame distance by analyzing their frame attributes – information element I, sequence number s, RSS vector R, and time interval Δt. Since the first three attributes are independent, we estimate $p(\boldsymbol{\Phi}_{ij} \,|\, \boldsymbol{P}_i, \boldsymbol{P}_j)$ by

$$\begin{aligned} & p(\boldsymbol{P}_j \mid \boldsymbol{P}_i, \Phi_{ij} = 1) \\ & = p(\boldsymbol{I}_j, s_j, \boldsymbol{R}_j, \boldsymbol{t}_j \mid \boldsymbol{I}_i, s_i, \boldsymbol{R}_i, \boldsymbol{t}_i, \Phi_{ij} = 1) \\ & \propto p(\boldsymbol{I}_j \mid \boldsymbol{I}_i, \Phi_{ij} = 1)\, p(s_j \mid s_i, \Delta t_{ij}, \Phi_{ij} = 1) \\ & \quad p(\boldsymbol{R}_j \mid \boldsymbol{R}_i, \Delta t_{ij}, \Phi_{ij} = 1)\, p(\Phi_{ij} \mid \Delta t_{ij}), \end{aligned} \tag{5}$$

where $p(\Phi_{ij} \mid \Delta t_{ij})$ conveys a prior knowledge that a frame is more likely to be associated with a smaller time interval; we assign it as an exponential decay distribution, i.e.,

$$p(\boldsymbol{\Phi}_{ij} \,|\, \Delta t_{ij}) = \begin{cases} \exp(-\lambda \Delta t_{ij}), & \Delta t_{ij} > 0, \\ 0, & \Delta t_{ij} \le 0. \end{cases} \tag{6}$$

where λ is a decay rate.

With the goal of estimating Equation 5, we learn the metric $g_\theta(\boldsymbol{P}_i, \boldsymbol{P}_j)$ based on Siamese structure, which encodes the compared frames into an identical latent space to estimate their distance. We illustrate the model structure of the frame correlation estimator in Figure 4. In frame distance encoding, we first apply different encoding approaches to represent the frame attributes of two input frames. Then, the encoded vectors are fused to be a latent distance vector for frame correlation estimation. In correlation estimation, we employ a learning-based metric to evaluate the correlation probability from the latent distance vector. Notably, both model inference and training are conducted end-to-end, and there is no further adjustment once the model structure is settled.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

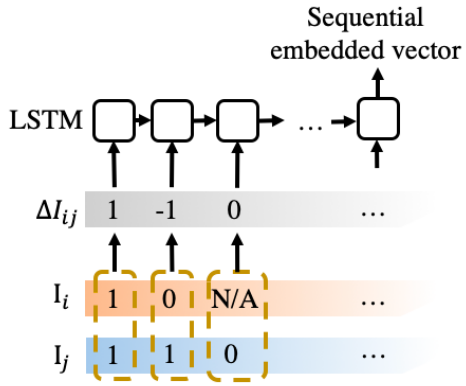


Fig. 6. Illustration on sequential encoding for IE.

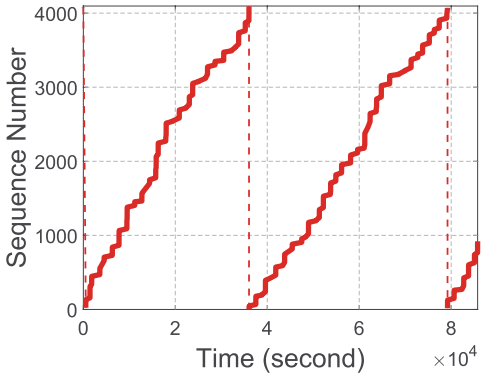


Fig. 7. The growth of sequence numbers over time.

4.2 Frame Distance Encoding

Although the frame attributes – information element (IE), sequence number, and signal strength – shed lights on frame association, their association patterns are complex and heterogeneous. To explore the association patterns, we encode the attributes into a latent space where we can directly measure the frame correlation by calculating the vector distance. Since these attributes possess separate properties, we individually consider them with different encoding structures.

4.2.1 Sequential Encoding for Information Element

Information element (IE) contains device specification and configuration information. However, a device may emit probe frames with diverse IEs, and hence using IE as unique device fingerprints [4] suffers severe false positives. To show this, we compile statistics from $100{,}000$ pairs of probe requests collected from the same device but at different times in Figure 5, which depicts the percentage of distinct IE fields despite the same device.

To effectively utilize IE, we encode IE into a latent space where we can explore frame correlation. Note that IE is a sequence of bits in which each conveys a certain device state. We want to explore which bits are stable while which bits are prone to change. Meanwhile, some bits may correlate with each other, which also indicates the clues of frame association. We use LSTM [33] to encode IE sequence because it is able to capture long-range dependence between any bits in an IE sequence. Also, LSTM is memory-efficient to train and compute, which makes it suitable for this task.

We illustrate the sequential encoding in Figure 6. For two IEs, I_i and I_j, we first compare their contents bit-wisely. In particular, the kth element of comparison result is

$$\Delta I_{ij}(k) = \begin{cases} 1, & [I_i(k) \neq \emptyset] \wedge [I_i(k) = I_j(k)], \\ -1, & I_i(k) \neq I_j(k), \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Then, the produced sequence is processed by LSTM, whose final hidden state serves as the sequential encoded vector that accumulates the information throughout the sequence.

The above encoding are conducted on each IE field. For efficiency concern, we only work on the IE field I^h, if, for two frames $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$,

$$p(\boldsymbol{\Phi}_{ij} = 0 | I_i^h = I_j^h) > \tau, \tag{8}$$

where the probability is evaluated according to all the frames with physical MAC addresses in advance, and the confidence threshold τ is set to be 0.5 in the paper. We evaluate the distance between two sequential encoded vectors by element-wise Euclidean distance.

4.2.2 Modulus Subtraction on Sequence Number

Sequence numbers are used to record the transmission order of frames. Wi-Fi chips normally increase the sequence number by 1 when emitting a new frame; the number will be reset to 0 when reaching the maximum value 4096 [34]). The sequence number in the i^{th} frame emitted from a device is

$$s_i = i \mod 4096. \tag{9}$$

When devices do not connect to any wireless network, they transmit nothing but probe requests for active periodical scans. To illustrate this, we record the probe requests emitted from an iOS device (MAC address randomization enabled) for 24 hours. The device does not connect to any network and keeps its screen off. Frames are captured by a sensor operating on channel 1 of $2.4\,\text{GHz}$ Wi-Fi. Figure 7 shows the growth of sequence numbers during the period. Despite the imperfect linearity due to channel switching and frame loss, the continuity of sequence numbers can be clearly observed.

Based on this observation, we can capture the patterns of sequence number interval when receivers operate on a fixed channel. However, such interval patterns can be affected by many factors, such as frame loss and device heterogeneity. To address it, we apply the modulus subtraction of sequence number as part of the latent distance vector. i.e., for s_i and s_j in two frames,

$$\Delta s_{ij} = (s_j - s_i) \mod 4096. \tag{10}$$

The correlation estimation can then learn the multi-mode pattern of sequence number interval, which will be discussed in Section 4.3.

4.2.3 Spatial Encoding for Signal Strength

Due to the location-dependence nature of RSS, the transitions of signals reflect user movements in the physical space. On the one hand, location can serve as a physical constraint. For instance, a device should not move far away in a short time period (say, several seconds). On the other hand, users'

locations are usually correlated by building functionality. For example, pedestrians' locations mostly perform similar patterns in a pathway. Thus, exploring spatial correlation from RSS is beneficial for estimating the correlation between two frames.

RSS vectors represent the received signal strengths of each sensor from an emitting device. Usually, during an active scan from a device, the number of receiving sensors is much smaller than the total number of available ones in a site (such as a shopping mall). This makes the sparsity of RSS vectors, and hence exploring the spatial correlation between two frames is tricky. To address this, we encode RSS vectors into a compact latent space to indicate their locations so that the spatial correlation can be better estimated.

In nature, RSS vectors represent locations. Thus, we treat it as a whole and encode RSS by a stack of fully connected layers with ReLU as the activation functions. We use the dropout mechanism to reduce overfitting [35]. Note that, this encoding is privacy-preserving because we cannot get actual location from the encoded vectors. Similar to IE sequential encoding, we use element-wise Euclidean distance to evaluate the distance between two spatial encoded vectors.

4.3 Correlation Estimation

Given the latent distance vector, the module then estimates the correlation between the frames. The module is designed based on two intuitions. Firstly, given the latent distance vector, we need a model to estimate frame correlation by jointly considering the encoded distance information from different attributes. Secondly, as mentioned in Equation 5, we want to capture the complex association pattern of RSS and sequence number under different time intervals between frames. Thus, we concatenate the time interval into the latent distance vector and employ a neural network to evaluate frame association (mapping distance to probability is shown in Equation 3).

We argue that a neural network with a simple structure (in terms of layer number) is suitable for the correlation estimation. For one thing, since the frame attribute patterns have already been captured by the frame distance encoding, a small neural network is enough to evaluate frame correlation from the latent distance vector. For another, the neural network with too many layers renders it hard to optimize for the first few layers, so a correlation estimation with a too deep structure would make it hard to learn for the frame distance encoding. Overall, we employ a 2-layer neural network for the correlation estimation module.

5 Multi-frame Association

In this section, we present how Cappuccino efficiently associates multiple frames. Section 5.1 formulates the multi-frame association problem. In Section 5.2, we show that the association problem can be tackled as a minimum-cost flow solution. Finally, we present in Section 5.3 the mini-batch adaptation to process a large dataset.

5.1 Multi-frame Association Formulation

We first present the formulation of the multi-frame association problem. Recall that $\boldsymbol{\mathcal{P}} = \{\boldsymbol{P}_i \mid 1 \leq i \leq M\}$ denotes the set of M frames obtained in a period. Our goal is to find the predecessor frame (i.e., the previous frame sent by the same device) and the successor frame (i.e., the next frame sent by the same device) of each frame $\boldsymbol{P}_i$ in $\boldsymbol{\mathcal{P}}$.

Let $\boldsymbol{\Phi}$ be the $M \times M$ association matrix where the element $\boldsymbol{\Phi}_{ij}$ is defined in Equation 1. In case the successor and/or predecessor of a frame is not in $\boldsymbol{\mathcal{P}}$, we further introduce two indicator vectors of length M, denoted by $\boldsymbol{A}$ and $\boldsymbol{B}$, where

$$\boldsymbol{A}_i = \begin{cases} 1, & \boldsymbol{P}_i \text{ has no successor in } \boldsymbol{\mathcal{P}}, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

and

$$\boldsymbol{B}_i = \begin{cases} 1, & \boldsymbol{P}_i \text{ has no predecessor in } \boldsymbol{\mathcal{P}}, \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

Given a set of probe requests $\boldsymbol{\mathcal{P}}$ and their pairwise association probability given by the correlation estimator (Section 4), the multi-frame association problem seeks an association with the highest joint probability over the entire set. That is,

$$\arg\max_{\boldsymbol{\Phi}} \quad \prod_i \prod_j p(\boldsymbol{\Phi}_{ij} \mid \boldsymbol{P}_i, \boldsymbol{P}_j)^{2\boldsymbol{\Phi}_{ij}} \prod_i \gamma^{\boldsymbol{A}_i} \prod_j \gamma^{\boldsymbol{B}_j}, \tag{13}$$

$$\text{subject to:} \quad \sum_{j=1}^{M} \boldsymbol{\Phi}_{ij} + \boldsymbol{A}_i = 1, \quad \forall i : 1 \leq i \leq M, \tag{14}$$

$$\sum_{i=1}^{M} \boldsymbol{\Phi}_{ij} + \boldsymbol{B}_j = 1, \quad \forall j : 1 \leq j \leq M, \tag{15}$$

$$\boldsymbol{\Phi}_{ij} \in \{0,1\}, \boldsymbol{A}_{ij} \in \{0,1\}, \boldsymbol{B}_{ij} \in \{0,1\}, \quad \forall i : 1 \leq i \leq M, \ \forall j : 1 \leq j \leq M, \tag{16}$$

where γ is the probability that a frame has no predecessor (or successor) in $\boldsymbol{\mathcal{P}}$. The objective function (13) is the joint probability of all the association decisions over $\boldsymbol{\mathcal{P}}$. The constraints (14) and (15) require that each frame has at most one predecessor and at most one successor, respectively. The value of γ can be empirically determined according to historical data.

5.2 Minimum-cost Network Flow Solution

We show that the above multi-frame association problem can be solved efficiently by viewing it as a minimum-cost network flow problem.

Figure 8 illustrates the graph structure of the corresponding flow network $\mathcal{G}$. For each frame $\boldsymbol{P}_i \in \boldsymbol{\mathcal{P}}$, two nodes are added into the network $\mathcal{G}$ — a *sender* node u_i and a *receiver* node v_i. A sender node has a supply of 1 and a receiver node demands 1. Edge $(u_i, v_j) \in E$ ($1 \leq i, j \leq M, i \neq j$) indicates that $\boldsymbol{P}_j$ is a possible successor frame of $\boldsymbol{P}_i$. The cost of edge (u_i, v_j) is assigned by the negative logarithm of the frame correlation, i.e., $-\log p(\boldsymbol{\Phi}_{ij} \mid \boldsymbol{P}_i, \boldsymbol{P}_j)$. We further set the edge capacities to 1 since the association between frames is a binary decision.

Moreover, we add in $\mathcal{G}$ an auxiliary node w to represent the case where frames have no successor and/or predecessor. A flow from a sender node to the auxiliary node indicates that the frame has no successor, while a flow from

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

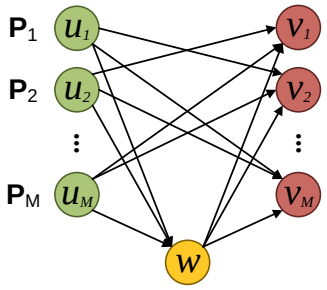


Fig. 8. The graph structure of a flow network.

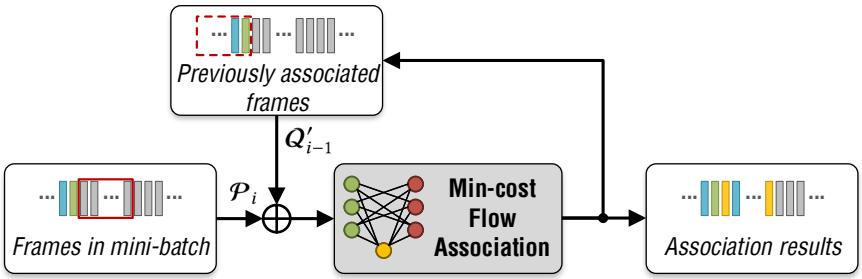


Fig. 9. The workflow of frame association for a large dataset.

it to a receiver implies the case of no predecessor. The edge costs to/from the auxiliary node are $-\log\gamma/2$, and their capacities are 1. The supply on the auxiliary node is 0 due to the following theorem.

Theorem 1. *Let N denote the number of devices that transmit probe requests $\boldsymbol{\mathcal{P}}$ ($N \leq M$). The number of frames with no successor is equal to the number of frames with no predecessor, i.e., $\sum_{i=1}^{M} \boldsymbol{A}_i = \sum_{i=1}^{M} \boldsymbol{B}_i = N$.*

Proof. Let $\boldsymbol{\mathcal{S}}_k = \{\boldsymbol{P}_l^k \,|\, 1 \leq l \leq M_k\}$ be the frame sequence emitted from the k-th device. In $\boldsymbol{\mathcal{S}}_k$, all the frames except for the last one $\boldsymbol{P}_{M_k}^k$ have successors, i.e., $\boldsymbol{A}_{M_k}^k = 1$ and $\boldsymbol{A}_l^k = 0$ ($1 \leq l \leq M_k - 1$); all the frames except for the first one $\boldsymbol{P}_1^k$ have predecessors, i.e., $\boldsymbol{B}_1^k = 1$ and $\boldsymbol{B}_l^k = 0$ ($2 \leq l \leq M_k$). Because $\boldsymbol{\mathcal{P}}$ is the union of frame sequences from the M devices, the number of devices N can be represented by the total number of frames with no successor ($\sum_{i=1}^{M} \boldsymbol{A}_i = \sum_{k=1}^{N} \boldsymbol{A}_{M_k}^k = N$) or the total number of frames with no predecessor ($\sum_{i=1}^{M} \boldsymbol{B}_i = \sum_{k=1}^{N} \boldsymbol{B}_1^k = N$). □

In the flow network above, we can find a maximum flow f with the minimum cost to obtain the optimal association, i.e.,

$$\begin{aligned}
\arg\min_{f} \quad & -\sum_i \sum_j f(u_i, v_j) \log p(\boldsymbol{\Phi}_{ij} \,|\, \boldsymbol{P}_i, \boldsymbol{P}_j) \\
& - \frac{1}{2}\log\gamma \sum_i f(u_i, w) - \frac{1}{2}\log\gamma \sum_j f(w, v_j), \quad (17) \\
\text{subject to:} \quad & \sum_{j=1}^{M} f(u_i, v_j) + f(u_i, w) = 1, \ \forall i : 1 \leq i \leq M, \quad (18) \\
& \sum_{i=1}^{M} f(u_i, v_j) + f(w, v_j) = 1, \ \forall j : 1 \leq j \leq M, \quad (19) \\
& \sum_{i=1}^{M} f(u_i, w) = \sum_{j=1}^{M} f(w, v_j), \quad (20) \\
& f(u_i, v_j) \in \{0, 1\}, \quad \forall (u_i, v_j) : (u_i, v_j) \in E, \quad (21)
\end{aligned}$$

where $f(u_i, v_j)$ represents the flow on the edge (u_i, v_j). Equations (18–20) are the constraints of flow conservation, and Equation (21) specifies the capacities on edges.

The minimum-cost flow problem can be solved efficiently by using algorithms such as Cycle Canceling [36], [37] and Network Simplex [38]. Given the optimal flow f, the corresponding frame association can thus be obtained by

$$\boldsymbol{\Phi}_{ij} = f(u_i, v_j), \quad (22)$$
$$\boldsymbol{A}_i = f(u_i, w), \quad (23)$$
$$\boldsymbol{B}_j = f(w, v_j). \quad (24)$$

Last but not least, we prove the correctness of the proposed minimum-cost flow modeling by the following theorem.

Theorem 2. *The proposed minimum-cost flow problem is equivalent to the multi-frame association problem.*

Proof. The theorem is proven by showing that both the objective functions and the constraints in the two problems are equivalent. We first discuss the objective functions. Let O_{MFA} denote the objective function (13) in the original multi-frame association problem, and O_{MCF} be the objective function (17) of the minimum-cost flow formulation. We can easily verify that $O_{\text{MCF}} = -\log O_{\text{MFA}}/2$. Hence maximizing the objective function (13) is equivalent to minimizing the objective function (17). We then show the equivalence of constraints. By applying the conversion in Equations (22–24), constraints (18–21) can be rewritten to the format of constraints (14–16), respectively. Constraint (20) corresponds to the characteristic of the multi-frame association as described in Theorem 1. Therefore, we conclude that the proposed minimum-cost flow problem is equivalent to the multi-frame association problem. □

5.3 Mini-batch Processing for Large Dataset

Although the above algorithm can construct the association on a given frame set, the efficiency would be affected when processing a large body of frames in a single batch (say, a set of frames in several hours). To address this, we present in the following an efficient processing scheme based on mini-batch.

The basic idea is to divide the dataset into multiple smaller batches and sequentially construct associations for each. Apart from the association within each mini-batch, Cappuccino also considers the association across different mini-batches in order to obtain the complete association over the entire period. Figure 9 illustrates the workflow. The frame set $\boldsymbol{\mathcal{P}}$ is partitioned into multiple mini-batches of the same interval T, i.e., $\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{P}}_1 \cup \boldsymbol{\mathcal{P}}_2 \cup \cdots$. Let $\boldsymbol{\mathcal{Q}}_i$ denote the set of frames without successors after processing the i-th

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

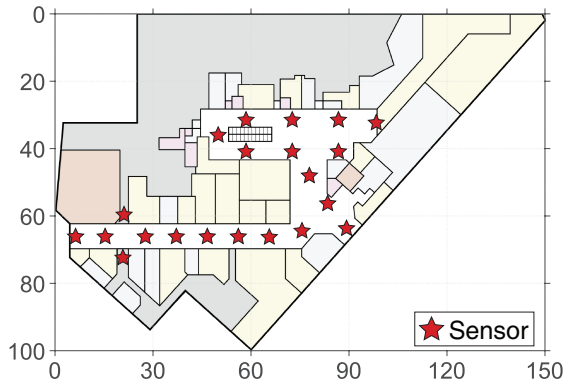


Fig. 10. The floor plan of the experimental site in a large shopping mall (in meter).

mini-batch (initially, $\mathcal{Q}_0 = \emptyset$). To avoid the size exploding of $\mathcal{Q}_i$, Cappuccino only keeps the frames in the last time period of length V, denoted by $\mathcal{Q}'_i$. V is usually the possible maximum interval between two consecutive probe requests ($V = 600$s empirically). In the i-th iteration, given the mini-batch frames $\mathcal{P}_i$ and the pruned set of previously associated frames $\mathcal{Q}'_{i-1}$, we solve the minimum-cost flow problem over $\mathcal{P}_i \cup \mathcal{Q}'_{i-1}$. The ultimate association over the entire set $\mathcal{P}$ can be obtained by the union of the association in all the iterations. Note that the scheme also enables Cappuccino to operate in an online manner with a delay of T.

6 Illustrative Experimental Results

We present in this section the experimental evaluation of Cappuccino. We introduce the experimental settings in Section 6.1 and discuss the illustrative results in Section 6.2.

6.1 Experimental Settings

We have implemented Cappuccino and conducted extensive experiments to validate its performance. The system consists of multiple Wi-Fi sensors and a centralized server. The sensors are implemented on commercial Wi-Fi APs (GL-AR150 with OpenWrt 18.06). They capture probe requests via *libpcap*. Captured frames are then transmitted to the server through Ethernet for storage and processing. The server is built on a PC equipped with Intel Core i7 3.6 GHz CPU and 16 GB RAM. The algorithm is implemented in Python. We solve the minimum-cost flow problem by employing *Google OR-Tools* [39], which implements a cost-scaling push-relabel algorithm. The complexity of the algorithm is $O(n * m^2 * \log(n * c))$ where n, m is the number of node and edge and c is the largest edge cost.

The experiments are conducted on an entire floor of a large shopping mall (in Hong Kong), including public walkway and stores. Figure 10 shows its floor plan ($\sim 8000\,\mathrm{m}^2$) and sensor placement (21 sensors in total). Wi-Fi sensors are installed on the ceiling. Probe requests are often broadcasted in all channels; we hence only need to listen on one channel where we operate on channel 1 of 2.4 GHz in our experiments. The system has been running for more than three months. We use the frames collected in one day for training the frame correlation estimator and the data in another day for evaluation. During a business hour on a typical weekend day, there are $\sim$20,000 frames captured by the system with $\sim$5000 unique MAC addresses.

Obtaining the actual frame association (as ground truth) is a major challenge in the experiments. We introduce two methods to address this. The first solution is to use the probe frames of physical MAC addresses. The real addresses are naturally the device identifiers and hence indicate the true associations of frames. Although the frames come from the devices without MAC address randomization, we can still leverage them to evaluate the overall performance under a large number of simultaneous frames. The second way is to manually collect the emitted MAC addresses. We attach external Wi-Fi sensors close to the transmitting devices, and the frames emitted can be captured with strong signal strength. In particular, we consider the frames with RSS greater than $-40\,\mathrm{dBm}$ coming from the targeting device and thus obtain its frame association. In the experiment, we invite 6 volunteers to carry smartphones (one for each user; the Wi-Fi function is switched on but does not connect to any network) and roam in the site. This can be used to validate the system performance on real devices with MAC address randomization. Unless otherwise stated, the dataset labeled with physical MAC addresses is our default test set.

Our experiment uses the following performance metrics:

- *Discrimination accuracy*: Given a probe request $\boldsymbol{P}_i$ at time t_i and the set of previous frames captured in $[t_i - \tau, t_i)$ that contains at least one predecessor of $\boldsymbol{P}_i$, we regard it as a correct association if the frame of the highest correlation to $\boldsymbol{P}_i$ is emitted from the same device transmitting $\boldsymbol{P}_i$. Discrimination accuracy is defined as the ratio of the correct cases among all the tests. The metric reflects the effectiveness of the frame correlation estimator. In our experiments, discrimination accuracy is estimated from 1000 randomly selected frames and their previous frames.
- *Homogeneity*, *completeness* and *V-measure*: Homogeneity and completeness are widely used in clustering performance analysis by leveraging normalized conditional entropy. We borrow its concept to evaluate the goodness of associated frame sequences by regarding them as clusters. Readers may refer to [40] for a detailed explanation. Homogeneity reflects whether generated sequences contain only frames from the same devices, while completeness implies whether all frames from a device are assigned to the same frame sequence. V-measure gives a comprehensive score which is defined as the harmonic mean of homogeneity and completeness. All the scores are between 0 and 1, and a higher value indicates better performance.

We compare Cappuccino with the following prior schemes:

- *IE fingerprinting (IEFpr)* [10]: The work uses a bit-level fingerprinting approach to distinguish between devices. It performs entropy-based analysis (i.e., variability and stability) on each frame bit to extract the informative ones as the devices' fingerprints.
- *Sequence number thresholding with IE clustering (SeqThresh)* [4]: The scheme proposes a two-stage association based on IEs and sequence numbers. It first clusters probe requests according to their IE fields. Within each

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

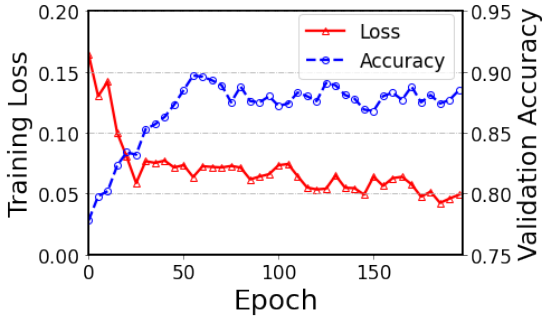


Fig. 11. Training process of frame correlation estimator.

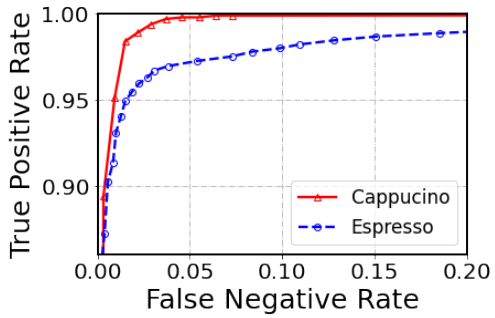


Fig. 12. ROC curves of model components.

TABLE 1
Experimental baseline parameter setting.

Parameter	Value
Sequential model layer	64, 64
Spatial encoding model layer	35, 50, 1
Correlation estimator layer	50, 1
Decay rate λ	10^{-5}
Mini-batch duration T	$30s$
No association probability γ	10^{-5}

cluster, it then links frames if the difference between sequence numbers is below a specified threshold η. We follow the original paper to select IE fields and set $\eta = 64$.

- *Time-based signature (TimeSig)* [11]: The scheme treats the distribution of the inter-frame arrival time (i.e., the time interval of pairwise frames) in probing rounds as the devices' signatures. A distance function is also designed to compare the similarity between two interval distributions.
- *Espresso* [30]: Espresso separately uses linear regression, Gaussian mixture model, and K-means cluster to estimate the frame correlation in terms of information element, sequence number, and signal strength. This is the preliminary version of Cappuccino, as mentioned in Section 2.

Unless otherwise specified, the system parameters are given as follows. As for the hyper-parameter of frame correlation estimator, the sequential encoding model is a 2-layer LSTM with hidden size of 64, the neuron quantities of spatial encoding model are 35, 50, 1, and neuron quantities of correlation estimation are 50 and 1. In training frame correlation estimator, all the associated pairs are used as positive samples while we randomly select non-associated pairs during training process; we use Adam optimizer whose initial learning rate is 0.001. The frame association decay rate λ is 10^{-5}. In the multi-frame association, the mini-batch duration T = 30s and the probability of no predecessor or successor $\gamma = 0.00001$. These parameters are summarized in Table 1.

6.2 Illustrative Results

Figure 11 illustrates the training process of the frame correlation estimator, where we plot the contrastive loss (as shown in Equation 4) and validation accuracy (using all test data) over training epoch. In the experiment, we repeat the training process for five times and show their averaged result. The training loss shows prominent descending with epoch smaller than 50, after that, the curve slowly descends with oscillation caused by the random sampling of non-associated frame pairs. Similarly, the validation accuracy increases before epoch of 50 and flats off after that. The validation result does not suffer severe overfitting since the random sampling renders training data change between epoches. Thus, we stop training process at the epoch of 50 and use the produced model to conduct following experiments.

Figure 12 compares the ROC curves between the frame correlation estimator of Cappuccino and Espresso when the period $\tau = 60$. We view the frame correlation estimator as a binary classifier to predict if a pair of probe requests are emitted from the same device. A frame pair from the same emitter is regarded as "positive"; it is "negative" otherwise. From the plot, Cappuccino shows superiority over Espresso because it extracts high-level features from the multi-modal inputs and make less assumption on the decision model. What's more, since it employs contrastive loss as training objective, most frame pairs are well predicted to be either 0 or 1; thus, Cappuccino shows better performance (27% discrimination error reduction with default τ) as a classifier.

We further compare discrimination performance in the periods of various lengths τ as for the frame correlation estimator of Cappuccino and Espresso. As shown in Figure 14, though Espresso is fairly accurate, Cappuccino further improves its accuracy by a large margin (e.g., Cappuccino reduces around 27% discrimination errors with $\tau = 60$). As shown by the curves, the discrimination accuracy of both approaches decreases as τ grows; while accuracy variation becomes smaller as τ goes larger. This is because the two frames are unlikely to be associated when their time interval is large, so that the frame correlation estimator can easily identify the additional frame candidates, introduced by the increased τ, as non-associated ones.

We verify in Figure 13 about the model substructures of Cappuccino's frame correlation estimator regarding the frame attributes. We, in each time, keep one encoding structure from frame distance encoding (as shown in Figure 4) to

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

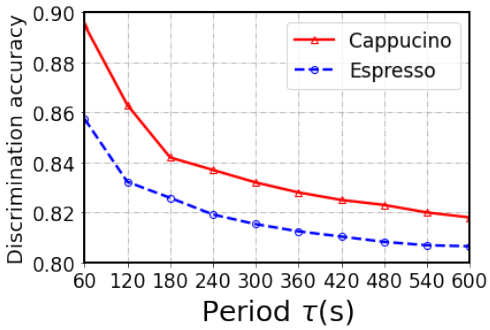


Fig. 13. Discrimination accuracy comparison in different lengths of periods.

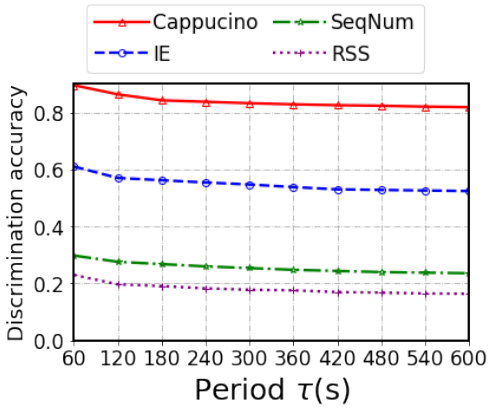


Fig. 14. Discrimination accuracy of model components in different lengths of periods.

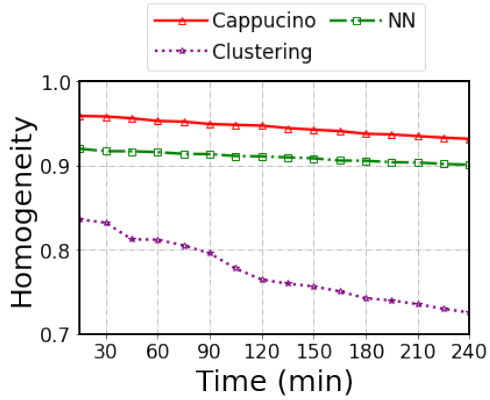


Fig. 15. Homogeneity of the frame sequences constructed by different association methods.

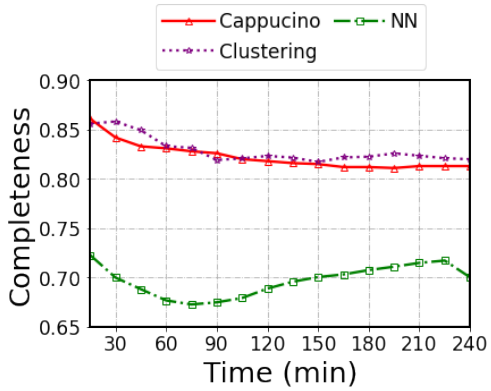


Fig. 16. Completeness of the frame sequences constructed by different association methods.

investigate its contribution to the whole model. The figure shows that the sequential encoding (information element) is the most informative (with discrimination accuracy $\sim 60\%$) on frame association among the three attributes; while spatial encoding (signal strength) and sequence number separately discriminate associated pairs by accuracy $\sim 20\%$ and $\sim 30\%$. Overall, their combination (i.e., Cappuccino) greatly outperforms each of them by at least 30% discrimination accuracy. This shows that Cappuccino is capable of utilizing these frame attributes to complement with each other.

We verify the multi-frame association module in Figures 15 and 16, which illustrate the homogeneity and completeness of the frame sequences constructed by different association methods, respectively. Since we are the first to consider multi-frame association, we involve two additional related comparison schemes for fairness: *NN* refers to the nearest neighbor algorithm which associates a frame with the other one of the highest correlation; *Clustering* denotes the approach that applies DBSCAN [41] to cluster probe requests based on the pairwise correlations. We plot the dynamics of homogeneity (Figure 15) and completeness (Figure 16) under different time periods. Since each probe request can be exclusively associated with another frame (or none), a single mistaken decision may cause a domino effect on the remaining associations. *NN* considers only the local association optimality and hence tends to fragment the frame sequences, resulting in low completeness. By contrast, *Clustering* achieves high completeness but low homogeneity. The reason is that *Clustering* greedily associates frames. Such unconstrained association tends to merge frame sequences and leads to low homogeneity. Cappuccino employs the minimum-cost flow approach to capture the global optimality and thus achieves a balanced performance between homogeneity and completeness.

We further illustrates in Figure 17 and Figure 18 the benefits on multi-frame association from the accurate frame correlation estimator of Cappuccino: while being high homogeneity as Espresso, Cappuccino improves the association completeness (by 2-6% over time). Espresso shows low completeness because it suffers ambiguity when a device emits more than one frames within a mini-batch – the successor of the successor frame may get a higher correlation probability. Cappuccino's frame correlation estimator leverages contrastive learning to guide all the correlation probabilities of associated frames to be one; thus, it is less likely to fail in this scenario with time decay (as shown in Equation 6).

Figure 19 demonstrates the impact of the mini-batch interval T in terms of homogeneity and completeness. The mini-batch interval does not significantly affect the perfor-

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3205924

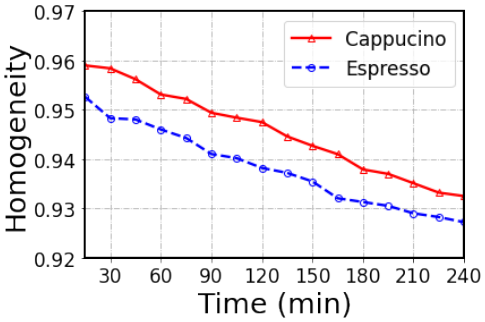


Fig. 17. Comparison to Espresso in terms of homogeneity.

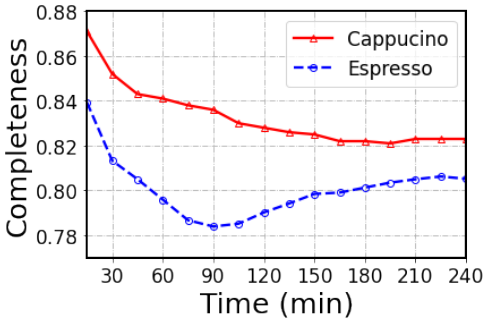


Fig. 18. Comparison to Espresso in terms of completeness.

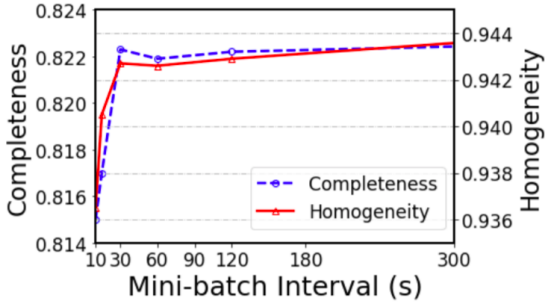


Fig. 19. Homogeneity and completeness against the mini-batch interval T.

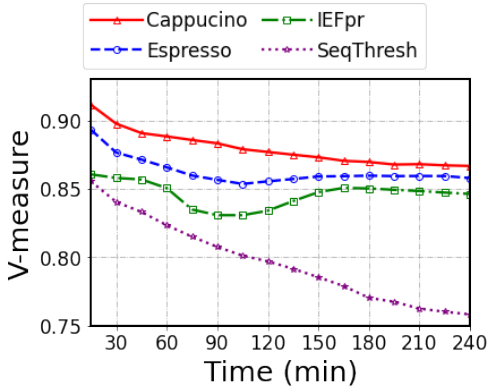


Fig. 20. V-measure of the frame sequences constructed by different schemes.

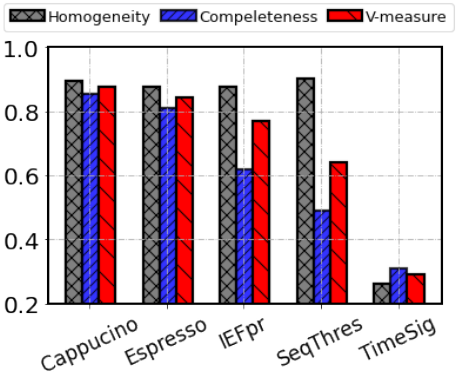


Fig. 21. Performance scores of the frame sequences constructed by different schemes (virtual MAC address).

mance since the overall standard deviations of homogeneity and completeness are low (0.002 and 0.004, respectively). However, we can still observe that both homogeneity and completeness increase rapidly with the growth of T when T is small ($T < 30$s). With a large T ($T \geq 30$s), the dynamics of both metrics become gentle. On the other hand, the period T also affects the system's responsiveness (i.e., the delay of outputting association results). To balance the association performance and responsiveness, we choose $T = 30$s in the system.

Figure 20 demonstrates the V-measures of the frame sequences constructed by different schemes under various periods. V-measure can reflect the comprehensive performance of frame association. We can observe that Cappuccino outperforms the others in all the experimental periods. The score of *IEFpr* declines significantly as time goes on. This is because it cannot accurately distinguish between devices under a large number of ambiguous frames and thus results in low homogeneity. We have not included the curve of *TimeSig* in the figure since its V-measure is extremely low (< 0.1). The possible reason is that *TimeSig* cannot well-characterize the inter-frame interval feature from a limited number of frames in probing rounds since the sensors only monitor a single channel.

We verify Cappuccino by probe requests with virtual MAC addresses (the ground-truth association is captured by external sensors). Figure 21 depicts the homogeneity, completeness and V-measure of the associated frame sequences of the comparing schemes. Although *SeqThresh* perform slightly better in homogeneity, they have much lower completeness and hence lower V-measure. This is because the deterministic methods applied are easily affected by frame ambiguities and noises, causing the incomplete association in the sequences. *TimeSig* again has the worst performance due to the lack of inter-frame interval features. It is noted that Cappuccino outperforms Espresso by a limited margin in the figure. This is reasonable because both Cappuccino and Espresso are designed to tackle large-scale crowded scenarios – it is hard to capture ground truths under virtual MAC addresses in a large-scale shopping mall; only

TABLE 2
Computation time comparison on frame correlation estimator between Cappuccino and Espresso.

Method	Processor	Batch size	Computational time
Espresso	Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz	1	$1.8ms$
Cappuccino	Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz	1	$1.6ms$
Cappuccino	NVIDIA GeForce RTX 2080 Ti	100	$0.1ms$
Cappuccino	NVIDIA GeForce RTX 2080 Ti	1000	$7.9\mu s$
Cappuccino	NVIDIA GeForce RTX 2080 Ti	2000	$7.5\mu s$
Cappuccino	NVIDIA GeForce RTX 2080 Ti	3000	$7.2\mu s$

six smartphones are used to capture ground truths. Nevertheless, Cappuccino shows balanced homogeneity and completeness in the figure, which is consistent with the experiments with physical MAC addresses in Figure 20.

We finally compare the computational time of the frame correlation estimator (to compute one correlation probability) between Cappuccino and Espresso in Table 2. Since the frame correlation estimator of Cappuccino is based on deep learning, we can leverage parallel computing on GPU to reduce its computational time. We use an off-the-shelf GPU for computation; we repeat each experiment for 50 times and average the results. As shown in the table, both approaches' frame correlation estimator use $\sim 2ms$ on CPU. However, Cappuccino is much more computationally efficient than Espresso when it is computed in batch on GPU. Overall, Cappuccino's frame correlation estimator is ~ 270 times faster than that of Epresso.

7 Conclusion

To protect user privacy, MAC address randomization has been deployed on modern devices, where randomly generated virtual MAC addresses are used in probe requests. As the MAC address from a single device changes randomly, many identity-oblivious statistical analytic approaches such as people counting, crowd flow estimation, and trajectory inference are defeated. In this paper, we present ***Cappuccino*** to establish the association between probe requests under MAC address randomization. Cappuccino works on existing Wi-Fi infrastructure without specially designed hardware, external localization systems, or offline device calibration/training. It is able to self-adapt to new environments. Our scheme consists of two important modules: frame correlation estimator and multi-frame association. The correlation estimator leverages contrastive learning to estimate frame correlation by jointly considering the multiple modalities of frame attributes, including information element, sequence number, and signal strength. On top of that, Cappuccino models the multi-frame association as a minimum-cost network flow problem, where the nodes represent the frames to be associated, and the edge weight is a decreasing function of the frame correlation. We have implemented Cappuccino and conducted extensive experiments to verify its performance. Our results show that Cappuccino achieves remarkable performance in terms of discrimination accuracy ($> 80\%$) and V-measure scores (> 0.85), which outperforms state-of-the-art by 27% discrimination error reduction and 4% completeness improvement on average.

References

[1] S. He and S.-H. G. Chan, "Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[2] C. J. Bernardos, J. C. Zúñiga, and P. O'Hanlon, "Wi-Fi internet connectivity and privacy: Hiding your tracks on the wireless Internet," in *Proceedings of 2015 IEEE Conference on Standards for Communications and Networking*. Tokyo, Japan: IEEE, Oct. 2015, pp. 193–198.

[3] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A Study of MAC Address Randomization in Mobile Devices and When it Fails," in *Proceedings of 2017 Privacy Enhancing Technologies Symposium*, vol. 2017, Oct. 2017, pp. 365–383.

[4] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, "Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS '16. Xi'an, China: ACM, 2016, pp. 413–424.

[5] Google, "Privacy: MAC Randomization," https://source.android.com/devices/tech/connect/wifi-mac-randomization, Jan. 2020.

[6] I. Grgurević, K. Juršić, and V. Rajič, "Overview of wi-fi-based automatic passenger counting solutions in public urban transport," in *Sustainable Management of Manufacturing Systems in Industry 4.0*, L. Knapcikova, D. Peraković, M. Perisa, and M. Balog, Eds. Cham: Springer International Publishing, 2022, pp. 181–196.

[7] T. Zang, Y. Zhu, Y. Xu, and J. Yu, "Jointly modeling spatio–temporal dependencies and daily flow correlations for crowd flow prediction," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 4, mar 2021. [Online]. Available: https://doi.org/10.1145/3439346

[8] A. Di Luzio, A. Mei, and J. Stefa, "Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications*. San Francisco, CA, USA: IEEE, Apr. 2016, pp. 1–9.

[9] J. Weppner, B. Bischke, and P. Lukowicz, "Monitoring crowd condition in public spaces by tracking mobile consumer devices with wifi interface," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp '16. Heidelberg, Germany: ACM, Sep. 2016, pp. 1363–1371.

[10] P. Robyns, B. Bonné, P. Quax, and W. Lamotte, "Noncooperative 802.11 MAC Layer Fingerprinting and Tracking of Mobile Devices," *Security and Communication Networks*, vol. 2017, pp. 1–22, May 2017.

[11] C. Matte, M. Cunche, F. Rousseau, and M. Vanhoef, "Defeating MAC Address Randomization Through Timing Attacks," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '16. Darmstadt, Germany: ACM, 2016, pp. 15–20.

[12] IEEE, "Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID (CID)," IEEE, Tech. Rep., Sep. 2017.

[13] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 User Fingerprinting," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, ser. MobiCom '07. Montréal, Québec, Canada: ACM, 2007, pp. 99–110.

[14] M. Cunche, M.-A. Kaafar, and R. Boreli, "Linking wireless devices using information contained in Wi-Fi probe requests," *Pervasive and Mobile Computing*, vol. 11, pp. 56–69, Apr. 2014.

[15] T. Kohno, A. Broido, and K. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, Apr. 2005.

[16] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz, "On the Reliability of Wireless Fingerprinting Using Clock Skews," in *Proceedings of the Third ACM Conference on Wireless Network Security*. Hoboken, NJ, USA: ACM, 2010, pp. 169–174.

[17] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Van Randwyk, and D. Sicker, "Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting," in *Proceedings of the 15th Conference on USENIX Security Symposium*, ser. USENIX-SS '06, vol. 15. Vancouver, BC, Canada: USENIX, Jul. 2006, pp. 1–12.

[18] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral Fingerprinting of IoT Devices," in *Proceedings of 2018 Workshop on Attacks and Solutions in Hardware Security*, ser. ASHES '18. Toronto, Canada: ACM, 2018, pp. 41–50.

[19] F. Guo and T.-c. Chiueh, "Sequence Number-Based MAC Address Spoof Detection," in *Proceedings of 2005 International Workshop on Recent Advances in Intrusion Detection*, A. Valdes and D. Zamboni, Eds. Berlin, Heidelberg: Springer, 2005, pp. 309–329.

[20] G. Chandrasekaran, J.-A. Francisco, V. Ganapathy, M. Gruteser, and W. Trappe, "Detecting Identity Spoofs in IEEE 802.11e Wireless Networks," in *Proceedings of 2009 IEEE Global Telecommunications Conference*. Honolulu, HI, USA: IEEE, Nov. 2009, pp. 1–6.

[21] J. Martin, E. Rye, and R. Beverly, "Decomposition of MAC Address Structure for Granular Device Inference," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, ser. ACSAC '16. Los Angeles, CA, USA: ACM, 2016, pp. 78–88.

[22] Subhash Challa, Mark R. Morelande, and Robin J. Evans, *Fundamentals of Object Tracking*. Cambridge, UK: Cambridge University Press, 2011.

[23] M. Bredereck, X. Jiang, M. Körner, and J. Denzler, "Data association for multi-object Tracking-by-Detection in multi-camera networks," in *Proceedings of the Sixth International Conference on Distributed Smart Cameras*. Hong Kong, China: IEEE, Oct. 2012, pp. 1–6.

[24] L. Fan, Z. Wang, B. Cail, C. Tao, Z. Zhang, Y. Wang, S. Li, F. Huang, S. Fu, and F. Zhang, "A Survey on Multiple Object Tracking Algorithm," in *Proceedings Ot 2016 IEEE International Conference on Information and Automation*. Ningbo, China: IEEE, Aug. 2016, pp. 1855–1862.

[25] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, Jan. 2013.

[26] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, Jan. 2015.

[27] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proceedings of 2017 IEEE International Conference on Robotics and Automation*. Singapore: IEEE, May 2017, pp. 1722–1729.

[28] R. Vezzani, D. Baltieri, and R. Cucchiara, "People reidentification in surveillance and forensics: A survey," *ACM Computing Surveys*, vol. 46, no. 2, pp. 29:1–29:37, Dec. 2013.

[29] Q. Leng, M. Ye, and Q. Tian, "A Survey of Open-World Person Re-Identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 4, pp. 1092–1108, Feb. 2019.

[30] J. Tan and S.-H. Gary Chan, "Efficient Association of Wi-Fi Probe Requests under MAC Address Randomization," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, May 2021, pp. 1–10.

[31] IEEE Standards Association, "802.11-2016 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2016.

[32] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[34] I. S. Association *et al.*, "Ieee 802.11 ac-2013-ieee standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 11: wireless lan medium access control (mac) and physical layer (phy) specifications—amendment 4: enhancements for very high throughput for operation in bands below 6 ghz."

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[36] W. H. Cunningham, "A network simplex method," *Mathematical Programming*, vol. 11, no. 1, pp. 105–116, Dec. 1976.

[37] J. B. Orlin, "A polynomial time primal network simplex algorithm for minimum cost flows," *Mathematical Programming*, vol. 78, no. 2, pp. 109–129, Aug. 1997.

[38] A. V. Goldberg and R. E. Tarjan, "Finding minimum-cost circulations by canceling negative cycles," *Journal of the ACM*, vol. 36, no. 4, pp. 873–886, Oct. 1989.

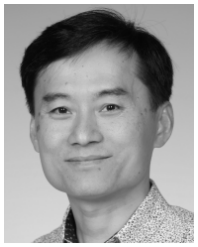
[39] Google, "OR-Tools," https://developers.google.com/optimization/flow/mincostflow, Oct. 2018.

[40] A. Rosenberg and J. Hirschberg, "V-Measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 410–420.

[41] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD '96. Portland, OR, USA: AAAI, 1996, pp. 226–231.

Tianlang He received his bachelor of engineering degree (with honor) from Donghua University, Shanghai, China, in 2018. He obtained his master of science degree from The Hong Kong University of Science and Technology (HKUST), Hong Kong, China, in 2019. He is working towards his PhD degree in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. His research interest includes wireless computing, edge computing, and multi-agent system.

Jiajie Tan received the bachelor of engineering degree from Zhejiang University, Hangzhou, Zhejiang, China, in 2012. He is working towards his PhD degree in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong, China. His research interest includes wireless location sensing, Internet of Things (IoT), and mobile computing.



S.-H. Gary Chan is currently Professor in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong. He is also Affiliate Professor in Innovation, Policy and Entrepreneurship Thrust Area of HKUST(GZ), Chair of the Committee on Entrepreneurship Education Program at HKUST, and Board Director of Hong Kong Logistics and Supply Chain MultiTech R&D Center (LSCM). He received MSE and PhD degrees in Electrical Engineering with a Minor in Business Administration from Stanford University (Stanford, CA). He obtained his B.S.E. degree (highest honor) in Electrical Engineering from Princeton University (Princeton, NJ), with certificates in Applied and Computational Mathematics, Engineering Physics, and Engineering and Management Systems. His research interest includes smart sensing and IoT, cloud and fog/edge computing, indoor localization and mobile computing, video/location/user/data analytics, and IT entrepreneurship.

Professor Chan has been an Associate Editor of IEEE Transactions on Multimedia, and a Vice-Chair of Peer-to-Peer Networking and Communications Technical Sub-Committee of IEEE Comsoc Emerging Technologies Committee. He has been Guest Editor of ACM Transactions on Multimedia Computing, Communications and Applications, IEEE Transactions on Multimedia, IEEE Signal Processing Magazine, IEEE Communication Magazine, etc. He is a steering committee member and was the TPC chair of IEEE Consumer Communications and Networking Conference (IEEE CCNC), and area chair of the multimedia symposium of IEEE Globecom and IEEE ICC.

Professor Chan has co-founded and transferred his research results to several startups. Due to their innovations and commercial impacts, his startups and research projects have received local and international accolades. Notably, he received Hong Kong Chief Executive's Commendation for Community Service for "outstanding contribution to the fight against COVID-19" in 2020. He is the recipient of Google Mobile 2014 Award and Silver Award of Boeing Research and Technology. He was a visiting professor and researcher in Microsoft Research, Princeton University, Stanford University, and University of California at Davis. At HKUST, he was Director of Entrepreneurship Center, Director of Sino Software Research Institute, Co-director of Risk Management and Business Intelligence program, and Director of Computer Engineering Program. He was a William and Leila Fellow at Stanford University, and the recipient of the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence at Princeton. He is a member of honor societies Tau Beta Pi, Sigma Xi and Phi Beta Kappa, and a Chartered Fellow of The Chartered Institute of Logistics and Transport (FCILT).