vContact: Private WiFi-Based IoT Contact Tracing With Virus Lifespan

Guanyao Li, Siyan Hu, Shuhan Zhong, Wai Lun Tsui, and S.-H. Gary Chan, *Senior Member, IEEE*

***Abstract*—Covid-19 is primarily spread through contact with the virus, which may survive on surfaces with a lifespan of hours or even days if not sanitized. To curb its spread, it is hence of vital importance to detect those who have been in contact with the virus for a sustained period of time, the so-called *close contacts*. Most of the existing digital approaches for contact tracing focus only on direct face-to-face contacts. There has been little work on detecting indirect environmental contact, which is to detect people coming into a contaminated area with the live virus, i.e., an area last visited by an infected person within the virus lifespan. In this work, we study automatic Internet of Things (IoT) contact tracing when the virus has a lifespan, which may depend on the disinfection frequency at a location. Leveraging the ubiquity of WiFi signals, we propose vContact, a novel, private, pervasive, and fully distributed WiFi-based IoT contact tracing approach. Users carrying an IoT device (phone, wearable, dongle, etc.) continuously scan WiFi access points (APs) and store their hashed IDs. Given a confirmed case, the signals are then uploaded to a server for other users to match in their local IoT devices for virus exposure notification. vContact is not based on device pairing, and no information of other users is stored locally. The confirmed case does not need to have the device for it to work properly. As WiFi data are sampled sporadically and asynchronously, vContact uses novel and effective signal processing approaches and a similarity metric to align and match signals at any time. We conduct extensive indoor and outdoor experiments to validate vContact performance. Our results demonstrate that vContact is effective and accurate for contact detection. The precision, recall, and F1-score of contact detection are high (up to 90%) for close contact proximity (2 m). Its performance is robust against AP numbers, AP changes, and phone heterogeneity. Having implemented vContact as an Android software development kit and installed it on phones and smart watches, we present a case study to demonstrate the validity and implementability of our design in notifying its users about their exposure to the virus with a specific lifespan.**

***Index Terms*—Contact tracing, COVID-19, data management and analytics, exposure notification, social impacts.**

I. Introduction

THE OUTBREAK of COVID-19 has had a profound impact on our lives and global economy. COVID-19, like many other infectious diseases, is primarily spread through viral contact. Recent studies have shown that the virus has a lifespan: in airborne droplets, it can last more than 10 min, and on surfaces, it can survive for hours to days if not properly disinfected (in low temperatures, it may last even longer) [1], [2]. The health of any person coming into contact with the live virus for a sustained period of time, say 15–30 min, may be at risk [3]. In order to effectively contain the spread of the disease, tracing and quarantining these *close contacts* as soon as possible is of paramount importance.

Close contacts could be manually traced based on personal interviews with infected people by medical officers. However, such an approach is labor intensive and slow. Due to mismemory, the contact information may be incomplete or error prone. To address the issues, automatic digital contact tracing approaches have been proposed in recent works. Some use GPS [4] and cellular signals [5]. While effective, these approaches do not work well in indoor environments due to signal blockage. Because they are also based on explicit user geolocations, and such locations may be computed or stored in other's networks, the systems raise concerns regarding location privacy. Due to these reasons, they have not become mainstream. Some privacy-preserving approaches based on phone-to-phone pairing using Bluetooth low energy (BLE) have attracted much attention and been implemented recently [3], [6], [7]. However, they work for only direct face-to-face contact tracing, and are not applicable for infection through environmental contact, i.e., the case with nonzero virus lifespan.

To overcome the above limitations, we propose vContact, a novel, private, and digital contact tracing solution using Internet of Things (IoT) with possibly location-dependent virus lifespan. Anyone in contact with the living virus is considered at risk. This includes those simultaneously located with the patient, and those sharing the same environment which the patient has left. vContact leverages ubiquitous WiFi signals to achieve pervasive, fully distributed, and automated contact tracing. Note that although for concreteness our discussion will focus on WiFi signals, vContact can be straightforwardly extended to other radio-frequency (RF) signals, such as Bluetooth and their combinations. To the best of our knowledge, this is the first decentralized work using RF signals for both direct and indirect private IoT contact tracing with a virus lifespan.

We illustrate the process of vContact in Fig. 1. A user carries a WiFi-enabled IoT device (phone, wearable, dongle, etc.). For concreteness and ease of illustration, we use a phone as the example in the figure. With an installed app, it periodically scans for WiFi, with each scan collecting a *signal*

Manuscript received December 27, 2020; revised June 25, 2021; accepted July 12, 2021. Date of publication July 26, 2021; date of current version February 21, 2022. This work was supported in part by the Hong Kong General Research Fund under Grant 16200120. *(Corresponding author: Guanyao Li.)*

The authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China (e-mail: gliaw@cse.ust.hk; siyanhu@ust.hk; szhongaj@cse.ust.hk; ptsuiwl@ust.hk; gchan@cse.ust.hk).

Digital Object Identifier 10.1109/JIOT.2021.3100276

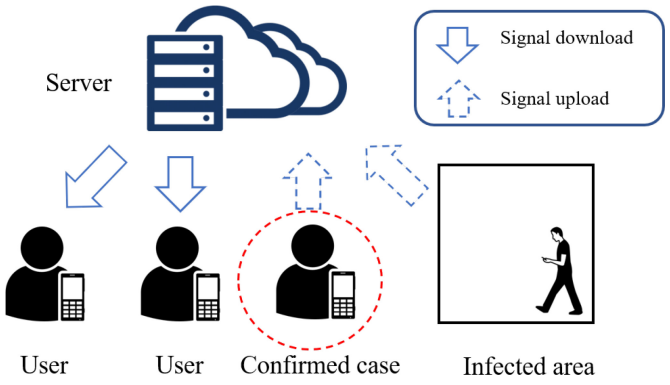


Fig. 1. Process of vContact contact tracing using WiFi.

vector consisting of two elements: 1) the signal IDs, which are the hashed (and optionally encrypted) values of the MAC addresses of the WiFi access points (APs) and 2) the corresponding received signal strength indicators (RSSIs) of the signal IDs. Each signal vector is associated with a timestamp, which is the scanning/collection time of the signals. Over time, the device collects and stores a time series of the signal vectors, termed the *signal profile*. The signal profile may be kept for a certain duration corresponding to the virus incubation period, usually 14–28 days for Covid-19.

Upon positive confirmation in hospital, the patient has the following two possibilities.

1) *With the Installed App:* With the consent of the patient, the health officer may access his/her signal profile. (Before sharing with the officer, the patient may blank out some time spans of the signal profile for personal reasons.) Because the signal IDs are hashed (and possibly encrypted) from AP MACs, the officer does not know the patient's geolocations, but only clusters of anonymized IDs and their collection times. Based on that, the officer works with the patient to identify the physical venues presenting potential health risks to the public. The corresponding anonymized IDs are extracted and labeled with their projected virus lifespan at the location at that time (depending on disinfection frequency). The resultant signal profile is then uploaded to a secure server for other IoT users to download and match with their own local profile in a distributed manner. Upon detecting a close contact, the user is alerted at once in private to check their health condition and seek medical advice.
2) *Without the App:* In this case, the confirmed case has to rely on his/her memory to recall the major venues and visit time as in the manual case. Then, some staff will go to these venues (the infected areas) to collect offline their WiFi information and label them with the visit time and viral lifespan at that time. These manually labeled data are then uploaded after being processed, and matched by the other IoT users the same way as in the previous case.

vContact complements the existing approaches for automatic digital contact tracing and may be integrated with them (such as [3] and [7]). Compared with prior arts, vContact has the following strengths and unique features.

1) *Contact Detection With Virus Lifespan:* vContact captures the realistic scenario of virus lifespan, which may be location dependent and temporally varied depending on the disinfection operation. It comprehensively covers, in a single framework, those in direct face-to-face contact *and* indirect environmental exposure in the areas previously visited by an infected person. The lifespan of the virus, set at the time of signal upload, may be heterogeneous and customized depending on the frequency of disinfection operation in the venue.
2) *No Device-to-Device Pairing and Communication:* Prior contact tracing proposals based on Bluetooth require device pairing, which means both devices, including the infected one, have to be installed with the app or software in order to work properly. To achieve tracing effectiveness, they hence demand a high adoption rate (in the range of reportedly 40%–70%). Moreover, such a device pairing approach may suffer from replay/relay attack [8] and raise privacy and security concerns [9]. In contrast to such pairing, each vContact device operates independently without any pairing or communication, and does not require the confirmed case to have already had the IoT device. This greatly relaxes the adoption barrier and provides a graceful adoption path. Furthermore, users do not store any information of or exchange any messages with other users; it hence offers much better protection of user anonymity, privacy, and attacks.
3) *Privacy by Design:* vContact is privacy by design. First, it does not require a user registration process, and hence, accesses no personal information such as names, phone numbers, IDs, contact lists, images/videos, etc. Second, the collected data never leave the local storage without the explicit consent of the owner, and even so (i.e., the case of a confirmed case) no personally identifiable data are uploaded, and the data remain anonymous at the server. Finally, vContact is fully decentralized. The collected data are exclusively stored in one's own device, and the contact is computed and detected locally on the device in a scalable manner without any other centralized entity (party or server) having full information. As no user data are stored anywhere beyond one's device, a user may exit the system at any time by device removal or app uninstallation without leaving his/her data behind. Upon detection of close contact, vContact conveys the message to its users in private. It is clear that such data fragmentation and minimization protect data privacy, and prevent data repurposing, abuse, and misuse. Due to its distributed and, hence, scalable nature, it is deployable from small local communities to across a country.
4) *No GPS-Based Geolocation:* vContact is not based on GPS. It is based on the hashed values of WiFi MAC addresses (namely, signal IDs) without storing the user's physical geolocation. This leads to much stronger location confidentiality than other GPS-based approaches, because the association of signal IDs to their physical

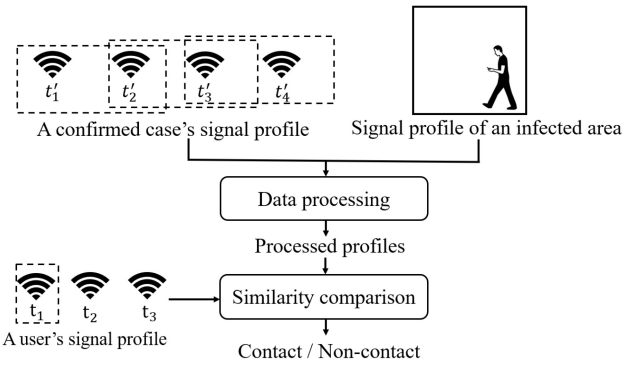


Fig. 2. Overview of contact detection using WiFi in vContact.

locations takes an enormously and prohibitively large amount of manual work (that is, to visit every indoor and outdoor spot of the city and logging down the locations of all the MAC addresses encountered). Furthermore, unlike other GPS approaches, vContact can detect indoor contacts and hence, is more pervasive.

Detecting close contact using WiFi data is a challenging problem. It is because signal vectors are sampled sporadically at random discrete times. Such independence and asynchrony among IoT devices result in difficulties detecting contact at any arbitrary time. Furthermore, signals may be sparsely sampled in the space (once every minute or so) Therefore, the scanned IDs and their RSSIs at a location at a distinct time may be different because of the change in the environment. Moreover, due to the device heterogeneity on antenna design and sensitivity, the collected signal IDs may also differ for different users.

vContact overcomes these problems by employing an efficient approach to represent the values between consecutive signal vectors and a novel similarity metric to match signal values for contact tracing. We present in Fig. 2 an overview of vContact. It first processes the discrete signal profile from a confirmed case or infected area by transforming it into a continuous profile (the processed profile). Using that given the signal vector of a user at time *t*, if *t* falls in the time range of the virus lifespan of a processed vector, vContact compares their level of matching using a novel signal similarity metric. If the similarity is larger than a given threshold, the user is said to be in contact with the virus at *t*. A user is identified as a close contact if the contact time exceeds a certain sustained period of time as specified by health officials.

vContact is simple, and we have implemented it as a software development kit (SDK) and Android app. We conduct extensive indoor and outdoor experiments with a diverse and representative set of IoT devices, such as smart watches and phone brands in the market (Samsung, Honor, Huawei Nova, Huawei Mate30, Xiaomi, and OPPO). Our results show that it achieves high precision, recall, and F1-score (up to 90% for the contact proximity of 2 m), and its performance is robust against AP numbers, AP changes, and devices of different brands. vContact can achieve good accuracy even when the AP number is low (as few as five APs in our experiments), meaning that it is widely applicable to city or suburban areas.

TABLE I
MAJOR SYMBOLS EMPLOYED IN THIS ARTICLE

Symbol	Explanation
A	Signal vector
$\hat{A}$	Processed vector
a	Signal ID
s	Signal strength
W	Signal profile
$\hat{W}$	Processed profile
t	Time
τ	Virus lifespan
O	Overlap ratio
D	Average RSSI difference
P	Signal similarity
α	Proximity threshold

TABLE II
MAJOR ACRONYMS USED IN THIS ARTICLE

Acronym	Explanation
AP	Access point
RSSI	Received signal strength indicator
BLE	Bluetooth Low Energy
RF	Radio frequency
RFID	Radio frequency identification
GPS	Global Position System
GPRS	General Packer Radio Service
INS	Inertial navigation system
SDK	Software development kit
AMD	Average Manhattan distance
AED	Average Euclidean distance
iOS	iPhone operating system

We summarize some major symbols employed in this article in Table I, and some major acronyms in Table II. The remainder of this article is organized as follows. We introduce related works in Section II. In Section III we present the approach of vContact. We have implemented vContact as an SDK, and discuss the experiment setting and illustrative results in Section IV. With the SDK, we have installed it in IoT smart watches and built an app, and present its implementation details and measurement in Section V-A. We conclude with future works in Section VI.

II. RELATED WORKS

Automatic contact tracing has attracted much attention in both academia and industry due to its importance in containing the spread of the Covid-19 pandemic [8], [10], [11]. In this section, we present the prior arts in the area.

Some studies have used signals, which reveal user geolocation, such as GPS, cellular data, and radio-frequency identification (RFID). GPS signal provides a user's exact location for contact tracing [12]–[15], but it is usually weak and noisy in indoor environments, limiting its contact coverage. Cellular data can be used to infer a user's public transportation trips [16], [17], which is crucial for contact tracing. Given the data, one can detect users taking the same bus, train, or subway with a confirmed case. However, this approach often has high location errors, because the coverage of the cell tower is large, and close proximity is difficult to detect. Some researchers have also proposed using RFID to understand contact [18], [19]. Nevertheless, special devices have to be deployed for data collection. When using WiFi service,

user devices would be associated with an AP. Based on the AP association log message of user devices in the server, WiFiTrace [20] and WiFiMon [21] reconstruct the locations visited by a user for contact detection. In these works, WiFi data are collected by passive sensing (i.e., association log), in which user devices are required to associate with APs and the data are stored and analyzed in a third-party server. Instead, our proposed solution uses active WiFi sensing for data collection. User devices scan surrounding APs once the devices are WiFi-on, without the requirement of WiFi association. Moreover, all user data are stored and analyzed locally in our solution.

Meanwhile, some geolocation-based contact tracing systems have been deployed around the world, such as Corowarner in Turkey [22], Aarogya Setu in India [23], and Cotrack in Argentina [24]. Corowarner and Aarogya Setu use GPS data, while Cotrack fuses signals of RFID, GPRS, GPS, and telecommunication technologies. To motivate users to contribute their sensitive data (e.g., location information), CovidCrowd [25] treats the process of collecting contact tracing data from a crowdsourcing perspective, in which users are offered rewards if they upload their contact tracing data. To this end, it computes the optimal reward value, which could maximize the utility of the system. All the above works may be extended to contact tracing with the virus lifespan. However, they may raise privacy concerns as they are based on user's physical geolocation and a third-party server for data analytics. In contrast, vContact offers much better location confidentiality, achieves better location accuracy, and is pervasive and easy to use.

Location privacy is a major concern for contact tracing [26]. To better protect it, some works propose using a magnetometer [27]. However, geomagnetism suffers from location ambiguity, which may lead to unsatisfactory proximity detection in practice. There has also been much work based on device-to-device message exchange using Bluetooth [6], [28]–[30]. User devices broadcast their ID using Bluetooth and scan the nearby IDs. Based on the scanned IDs, one can know if he/she has had close contact with an infected case [31]. Among the Bluetooth approaches, centralized solutions rely on a third-party server for contact tracing. Among these works, BlueTrace [32] and ROBERT [33] are the two most representative protocols. They use a decentralized framework to collect data, but a centralized system to analyze the exposure risks. Bluetooth data are collected via device-to-device communication and are stored locally. Once a user is infected, he/she can upload his/her scanned data to a security server for analysis. Users who are at risk will then be identified by the centralized system. The major difference of the two protocols is the way that people know their risks. In BlueTrace, the health authority would proactively contact the individuals who have a high likelihood of virus exposure, while users of ROBERT have to periodically probe the server for their risk score of exposure. Based on the BlueTrace protocol, the automatic contact tracing app TraceTogether [34] has been deployed in Singapore, which is the first national deployment of the Bluetooth-based contact tracing system. Based on a similar concept to that of TraceTogether, another system called COVIDSafe has been deployed in Australia to slow the spread of COVID-19 [35]. Furthermore, DESIRE [36] is an extension of the ROBERT protocol, which is based on the same architecture of ROBERT with some major privacy improvements. BeepTrace [37] also uses a decentralized framework to store user data and a centralized approach to detect close contact. In the BeepTrace, user devices upload collected GPS/WiFi/cellular tower data to the blockchain network, and a third-party solver is responsible for contact detection.

Since a third-party server may raise the concern of possible data abuse, other works advocate a fully distributed approach, where the exposure detection and notifications are processed on an individual device. Representative works include PACT-UW [7], DP-3T [3], PACT-MIT [38], and Pronto-C2 [39] (Note that both PACT-UW and PACT-MIT are termed as PACT in their origin papers.). In these decentralized systems, users collect the encrypted IDs of their nearby users and store them locally. When someone is confirmed as being infected, he/she can upload his/her encrypted ID for other users to download for contact tracing. Compared with centralized solutions, only the encrypted IDs of infected cases are uploaded for the decentralized solutions, and contact information is distributed on user devices for storage.

Based on the concept of decentralized systems, Google and Apple provide a toolkit for privacy-preserving contact tracing using Bluetooth [40]. Some Bluetooth-based decentralized systems have also been deployed in some countries, such Covid Watch in the U.S. [41] and SwissCovid [42] in the Switzerland. All these schemes are independently designed and very similar, apart from some minor variations in implementation and efficiency. All the above works focus on detecting face-to-face close contact, and they cannot be extended to the case with virus lifespan. We propose a private WiFi-based approach to detect close contacts with *virus lifespan*. To the best of our knowledge, this is the first decentralized work considering a virus lifespan for both direct and indirect private contact tracing using WiFi. Moreover, no IoT device pairing or communication are needed in our proposed scheme, and hence, no minimal adoption rate.

In summary, we categorize the above representative works according to their main distinctive features in Table III. These works are categorized in terms of signals, data collection method, detection method, virus lifespan consideration, and geolocation privacy. Compared with existing works, vContact uses WiFi data for contact detection. It employs decentralized frameworks for both data collection and contact detection, i.e., it stores all user data in one's own device exclusively (in case someone is confirmed infection and is consent to share her/his data), and the data matching process is completed locally. Furthermore, vContact does not rely on the user's physical geolocation, leading to much stronger location privacy than other geolocation-based approaches.

III. vCONTACT DETAILS

We present the details of vContact in this section. We first discuss its data processing approaches to construct the processed profile from the raw signal profile, for the patient

TABLE III
COMPARISON OF RELATED WORKS

Schema name	Signal	Decentralized data collection	Decentralized contact detection	Virus lifespan	Geo-location privacy
MCT [27]	Magnetometer	×	×	×	√
CovidCrowd [25]	GPS	×	×	√	×
Corowarner [22]	GPS	×	×	√	×
Aarogya Setu [23]	GPS	×	×	√	×
Cotrack [24]	RFID, GPRS, and GPS	×	×	√	×
WiFiTrace [20]	WiFi	×	×	√	×
WiFiMon [21]	WiFi	×	×	√	×
BeepTrace [37]	GPS, Bluetooth, Cellular and WiFi	√	×	√	×
TracingTogether [34]	Bluetooth	√	×	×	√
COVIDSafe [35]	Bluetooth	√	×	×	√
DESIRE [36]	Bluetooth	√	×	×	√
PACT-UW [7]	Bluetooth	√	√	×	√
PACT-MIT [38]	Bluetooth	√	√	×	√
DP-3T [3]	Bluetooth	√	√	×	√
Pronto-C2 [39]	Bluetooth	√	√	×	√
Covid Watch [41]	Bluetooth	√	√	×	√
SwissCovid [42]	Bluetooth	√	√	×	√
Exposure notification [40]	Bluetooth	√	√	×	√
vContact	WiFi	√	√	√	√

with and without the installed app on their IoT devices in Sections III-A and III-B, respectively. We then introduce in Section III-C an efficient and novel signal similarity metric to measure signal similarity, given a user's signal vector and a processed vector. We summarize vContact and outline its contact detection algorithm in Section III-D.

We define signal vector and signal profile as follows.

Definition 1 (Signal Vector): A signal vector A is represented as $\{(a_1, s_1), (a_2, s_2), \ldots, (a_i, s_i), \ldots, (a_n, s_n)\}$, where a_i is the signal ID (hashed and possibly encrypted AP MAC address) and s_i is its RSSI.

Definition 2 (Signal Profile): A user's signal profile is defined as a sequence of signal vectors over time: $W = \{(A_1, t_1), (A_2, t_2), \ldots, (A_i, t_i), \ldots, (A_n, t_n)\}$, where A_i is the signal vector scanned at time t_i.

In other words, a signal vector represents the signals and RSSIs scanned by an IoT device at a certain time, while a signal profile is a collection of the signal vectors over time. The contact tracing is then stated as follows. Given a user's signal profile $W = \{(A_1, t_1), (A_2, t_2), \ldots, (A_i, t_i), \ldots, (A_n, t_n)\}$, detect if the user has contact with the virus at each t_i by comparing the similarity of the signal vector at that time with the signal profile of a confirmed case or an infected area.

A. Profile Processing for Patient With the App

Signals are not sampled continuously but at sporadic and random intervals. Consequently, signal data are not continuously observable, leading to difficulty in comparing signal similarity at any arbitrary time. We propose here a data processing approach to construct continuous profiles from raw signal profiles for patients with our installed software.

We show an example to illustrate the signal profile processing in Fig. 3. A confirmed case's signal profile $\{(A_1, t_1), (A_2, t_2), (A_3, t_3), (A_4, t_4)\}$ consists of some signal vectors at discrete times. However, the signal vectors of locations where the confirmed case was located between two consecutive timestamps are not observable due to the sporadic

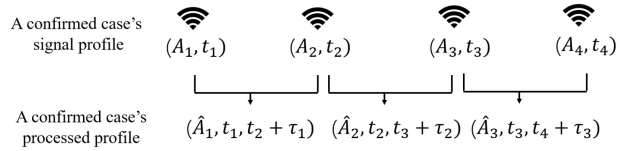


Fig. 3. Signal profile processing for a confirmed case with app.

data sampling. To address this issue, we generate a processed vector $\hat{A}_i$ from any two consecutive signal vectors A_i and A_{i+1} to denote the signals of the area where the confirmed case visited between t_i and t_{i+1}. Furthermore, we use $(t_i, t_{i+1} + \tau_i)$ to indicate that anyone coming into the area between t_i and $t_{i+1} + \tau_i$ would be at risk. Note that the virus lifespan τ_i may vary with time.

The formal definition of the processed vector is defined as follows.

Definition 3 (Processed Vector): A processed vector $\hat{A}$ is denoted as $\{(a_1, s_1^{\min}, s_1^{\max}), (a_2, s_2^{\min}, s_2^{\max}), \ldots, (a_i, s_i^{\min}, s_i^{\max}), \ldots, (a_n, s_n^{\min}, s_n^{\max})\}$, where $(a_i, s_i^{\min}, s_i^{\max})$ denotes that the RSSI range of a signal a_i is from $s_i^{\min}$ to $s_i^{\max}$.

The signal strength in a processed vector is represented as a range instead of an exact value in a signal vector. Given two consecutive signal vectors $A_i = \{(a_1^i, s_1^i), \ldots, (a_j^i, s_j^i), \ldots, (a_n^i, s_n^i)\}$ at t_i and $A_{i+1} = \{(a_1^{i+1}, s_1^{i+1}), \ldots, (a_k^{i+1}, s_k^{i+1}), \ldots, (a_m^{i+1}, s_m^{i+1})\}$ at t_{i+1}, the processed vector in the time range from t_i to t_{i+1} is denoted as $\hat{A}_i = \{(a_\ell, s_\ell^{\min}, s_\ell^{\max}) | \ell = 1, 2, \ldots, |A_i.a \cup A_{i+1}.a|\}$, where $a_\ell \in A_i.a \cup A_{i+1}.a$ and $(s_\ell^{\min}, s_\ell^{\max})$ is the signal strength range. There are three cases for $a_\ell \in A_i.a \cup A_{i+1}.a$: 1) $a_\ell \in A_i.a \cap A_{i+1}.a$; 2) $a_\ell \in A_i.a$ but $a_\ell \notin A_{i+1}.a$; and 3) $a_\ell \in A_{i+1}.a$ but $a_\ell \notin A_i.a$. Therefore, $(s_\ell^{\min}, s_\ell^{\max})$ is calculated as

$$\begin{cases} (\min(s_\ell^i, s_\ell^{i+1}), \max(s_\ell^i, s_\ell^{i+1})), & \text{for } a_\ell \in A_i.a \cap A_{i+1}.a \\ (\gamma, s_\ell^i), & \text{for } a_\ell \in A_i.a, a_\ell \notin A_{i+1}.a \\ (\gamma, s_\ell^{i+1}), & \text{for } a_\ell \in A_{i+1}.a, a_\ell \notin A_i.a. \end{cases} \quad (1)$$

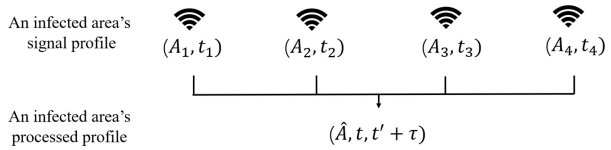


Fig. 4. Signal profile processing for an infected area.

Here, γ is a value indicating a weak signal strength. The signal strength of WiFi received in mobile devices usually ranges from -30 (strong) to -100 dBm (weak) [43], [44]. Thus, we set $\gamma = -100$ as a weak signal strength in our experiments.

Then, we construct a continuous processed profile from a confirmed case's signal profile considering the virus lifespan. We present a formal definition of a processed profile.

Definition 4 (Processed Profile): A processed profile contains a sequence of processed vectors over time: $\hat{W} = \{(\hat{A}_1, t_1^{\text{start}}, t_1^{\text{end}}), (\hat{A}_2, t_2^{\text{start}}, t_2^{\text{end}}), \ldots, (\hat{A}_i, t_i^{\text{start}}, t_i^{\text{end}}), \ldots, (\hat{A}_m, t_m^{\text{start}}, t_m^{\text{end}})\}$, where $\hat{A}_i$ is a processed vector for the time slot from t_i^{start} to t_i^{end}, and $(t_i^{\text{start}}, t_i^{\text{end}})$ indicates the time slot of the virus lifespan.

Given a confirmed case's signal profile $W = \{(A_1, t_1), (A_2, t_2), \ldots, (A_i, t_i), \ldots, (A_n, t_n)\}$, the processed profile is represented as $\hat{W} = \{(\hat{A}_1, t_1, t_2 + \tau_1), (\hat{A}_2, t_2, t_3 + \tau_2), \ldots, (\hat{A}_i, t_i, t_{i+1} + \tau_i), \ldots, (\hat{A}_{n-1}, t_{n-1}, t_n + \tau_{n-1})\}$, where $\hat{A}_i$ is constructed from A_i and A_{i+1} and τ_i is the virus lifespan for the time slot from t_i to t_{i+1}. Note that τ_i is given by the health officer, and it can vary for different time slots depending on the frequency of disinfection operation in the venues.

B. Signal Profile Processing for Infected Areas

For the case where the patient has not installed the app, we need to extract the signals in the infected areas through a survey (signal collection process). We can evaluate if a user has been in contact with an infected area by measuring the similarity of her/his signal vector and signal vectors of each position in the area. However, collecting WiFi data for every position in the infected area is inefficient. We propose an efficient approach to construct the processed profile for an infected area using just some sampled signal data in the area.

Instead of collecting signal data at every position, staff walk around the area with a WiFi-on device such as a phone or a Raspberry Pi. The collected signal profile is some signal vectors over time. To generate a representative processed profile for the area, we aggregate all signals and their RSSIs in the signal profile. As shown in Fig. 4, we merge the signal vectors in the signal profile $\{(A_1, t_1), (A_2, t_2), (A_3, t_3), (A_4, t_4)\}$, which are collected in the infected area. We also consider the time range $[t, t']$ when a confirmed case stays in the area and the virus lifespan τ to construct the processed profile for the infected area.

The processed profile of an area is represented as $\hat{W} = (\hat{A}, t^{\text{start}}, t^{\text{end}})$, where $\hat{A}$ is a processed vector and $[t^{\text{start}}, t^{\text{end}}]$ is the time range of the virus lifespan. Given the signal profile collected in the infected area $W = \{(A_1, t_1), (A_2, t_2), \ldots, (A_i, t_i), \ldots, (A_n, t_n)\}$, the time range of a confirmed case staying in the area $[t, t']$, and the virus lifespan τ, the processed profile $\hat{W} = (\hat{A}, t^{\text{start}}, t^{\text{end}})$ is constructed as follows: $\hat{A} = \{(a_j, s_j^{\min}, s_j^{\max}) | j = 1, 2, \ldots, |\cup_i^n A_i.a|\}$ where a_j is a scanned signal in W (i.e., $a_j \in \cup_i^n A_i.a$), and $s_j^{\min}$ is the minimum signal strength of a_j in W while $s_j^{\max}$ is the maximum signal strength of a_j in W; the surviving time of the virus in the infected area is from t to $t + \tau$.

TABLE IV
AVERAGE NUMBER OF SIGNALS IN A SIGNAL VECTOR FOR VARIOUS MOBILE PHONES

Phone	Average number of signals
Honor	75.00
Huawei Mate 30	128.12
OPPO	180.16
Huawei Nova	92.87
Xiaomi	102.09

C. Signal Similarity Metric

We propose a signal similarity metric to compare the similarity of a signal vector with a processed vector for exposure detection. The metric considers the signal IDs overlap ratio and the RSSI difference.

Intuitively, the closer a user is to the location of the virus, the more signal IDs are shared between the user's signal vectors and the vectors in the processed profile. Thus, we could use the overlap ratio of two vectors' signal IDs to indicate their proximity. Given a user signal vector A at time t and a processed vector $\hat{A}$, the overlap ratio is calculated as

$$O = \frac{\left|A.a \cap \hat{A}.a\right|}{\min\left(|A.a|, \left|\hat{A}.a\right|\right)} \tag{2}$$

where $A.a$ is the Signal IDs in A, $\hat{A}.a$ is the signal IDs in $\hat{A}$, and $|\cdot|$ denotes the number of signal IDs.

The proposed metric (2) has the same numerator as the well-known Jaccard's index but different denominator. The reason of using $\min(|A.a|, |\hat{A}.a|)$ instead of $|A.a \cup \hat{A}.a|$ as the denominator is to mitigate the impact of the dynamic environment and device heterogeneity. An IoT device such as a phone or smart watch may scan different numbers of WiFi APs at a location at different times. Moreover, different IoT devices may have different abilities to scan signals, resulting in two co-located devices possibly scanning different numbers of signals. Table IV shows the average numbers of signals in a signal vector of various co-located phones in a shopping mall. The average number of signals is heterogeneous for different phones. The difference could be significant for some phones. Therefore, using $|A.a \cup \hat{A}.a|$ as the denominator will introduce more variance while $\min(|A.a|, |\hat{A}.a|)$ is a more proper measure.

A signal could cover a large area, so it is possible that two vectors with a large proportion of common signals are not in close proximity. Thus, we also consider the RSSI difference to denote the proximity. If a user stays close with the virus, the RSSI difference of the same signal in two vectors should be small. Given a user signal vector $A = \{(a_1, t_1), (a_2, t_2), \ldots, (a_i, t_i), \ldots, (a_n, t_n)\}$ and a processed

Algorithm 1: Contact Detection

1 **Input**: A user's signal profile W_1 ;
A confirmed case's or an infected area's signal profile W_2;
Virus lifespan $\{\tau_i | i = 1, 2, ..., |W_2| - 1\}$;
A proximity threshold α.
2 **Output**: results of contact detection at different timestamps.
3 Initialize S to empty;
4 Construct the processed profile $\hat{W}$ from W_2 and $\{\tau_i | i = 1, 2, ..., |W_2| - 1\}$;
5 **foreach** $(A_i, t_i) \in W_1$ **do**
6 contact = False;
7 **foreach** $(\hat{A}_j, t_j^{start}, t_j^{end}) \in \hat{W}$ **do**
8 **if** $t_j^{start} \leq t_i \leq t_j^{end}$ **then**
9 $s = P(A_i, \hat{A}_j)$;
10 **if** $s \geq \alpha$ **then**
11 contact = True;
12 break;
13 **end**
14 **end**
15 **end**
16 **if** contact == True **then**
17 Add $(True, t_i)$ to S;
18 **else**
19 Add $(False, t_i)$ to S;
20
21 **end**
22 **return** S

vector $\hat{A} = \{(a_1, s_1^{\min}, s_2^{\max}), (a_2, s_2^{\min}, s_2^{\max}), \ldots, (a_j, s_j^{\min}, s_j^{\max}), \ldots, (a_m, s_m^{\min}, s_m^{\max})\}$, for $a_k \in A.a \cap \hat{A}.a$, its RSSI difference is calculated as

$$d(a_k) = \begin{cases} s_j^{\min} - s_i, & s_i < s_j^{\min} \\ s_i - s_j^{\max}, & s_i > s_j^{\max} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The average RSSI difference at a timestamp is defined as

$$D = \frac{\sum_{a_k \in \left(A.a \cap \hat{A}.a\right)} d(a_k)}{\left|A.a \cap \hat{A}.a\right|} \tag{4}$$

where $|\cdot|$ denotes the number of signal IDs.

When a user has contact with the virus, the overlap score O [(2)] should be large, while the RSSI difference D [(4)] should be small. Therefore, we define the signal similarity of A and $\hat{A}$ as

$$P\left(A, \hat{A}\right) = \frac{O}{D + 1} \tag{5}$$

where $0 \leq P(A, \hat{A}) \leq 1$. A larger $P(A, \hat{A})$ indicates closer proximity.

D. vContact Algorithm and Computational Complexity

Anyone having contact with the surviving virus may be at risk. Given a user's signal vector A_i at t_i, if the timestamp t_i is within the virus lifespan, and the similarity of A_i and the processed profile of a confirmed case or an infected area are larger than a threshold, the user will be detected as having contact with the virus at t_i. The algorithm is presented in Algorithm 1.

Given a user's signal profile W_1, the signal profile of a confirmed case or an infected area W_2, the virus lifespan $\{\tau_i | i = 1, 2, \ldots, |W_2| - 1\}$, and a proximity threshold α, we first construct the processed profile from W_2 and $\{\tau_i | i = 1, 2, \ldots, |W_2| - 1\}$ (line 4). Then, for each signal vector A_i at time t_i in W_1, if t_i falls in the time slot of a processed vector in the processed profile, we calculate the signal similarity [using (5)] at t_i (lines 7–9). If the similarity at t_i is larger than the given threshold α, the user is identified as having contact with the virus at t_i (line 11). The algorithm evaluates the similarity of each signal vector in W_1 and $\hat{W}$, and returns a list of detection results. The threshold α depends on how we define the contact proximity for close contact. We will discuss the relationship between the signal similarity and physical proximity, and the determination for the proximity threshold α in the following section.

As presented in Algorithm 1, given a user's signal profile W_1, and the signal profile of a confirmed case or an infected area W_2, the computational complexity of detecting contact is $O(|W_1| \times |W_2|)$, where $|W_1|$ and $|W_2|$ are the number of signal vectors in W_1 and W_2, respectively.

IV. Illustrative Experimental Results

We have implemented and packaged vContact as an SDK. In this section, we present illustrative experimental results on the SDK, using phones as IoT devices. We first introduce the experiment settings in Section IV-A. Then, we study how to set the threshold α in Section IV-B. We present the performance of vContact for patients with app and infected areas in different sites in Sections IV-C and IV-D, respectively. Then, we compare vContact with other state-of-the-art approaches in Section IV-E. The studies on the impacts of different AP numbers, dynamic environment, and heterogeneous devices are covered in Sections IV-F–IV-H, respectively. Finally, we discuss the impact of data sampling rate in Section IV-I.

A. Experimental Settings

We collect WiFi data using five mobile phones in three different sites. The brands and models of phones are different, and include Honor, Huawei Nova, Huawei Mate30, Xiaomi, and OPPO. According to some latest reports, these brands are representative in the market. The three experimental sites are an office, a bus station, and a store in a shopping mall. The size of the office is around 10 m $\times$12 m. The bus station is an outdoor area, the size of which is around 2 m $\times$15 m. The area in the shopping mall for experiments is a large store with a size of 20 m $\times$25 m. The total signal numbers are 32 in the office, 109 in the bus station, and 301 in the shopping mall. The average number of signals (i.e., scanned APs) in signal vectors of the office, bus station, and shopping mall is 19.02, 24.0, and 46.29, respectively.

To evaluate the detection performance for the case where the signal profiles of confirmed cases are available, we first put the five mobile devices at a location ℓ_0 for 10 min to collect the WiFi data in each site. The WiFi signals with RSSIs scanned by a device are collected. Then, we put the devices at a location ℓ_i for 10 min for data collection, where $i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$, and the distance between ℓ_0 and ℓ_i is i meters. The data sampling rate is set as 5 s per record, so we have around 120 records of data for a device in each distance setting for each site.

To evaluate the detection performance for the case where a confirmed case's signal profile is unavailable, we walk in the experimental sites to collect WiFi data using a mobile phone to construct the processed profiles for each site. Then, we wander around and outside the area with five mobile phones collecting WiFi data for testing. The time when we were in and outside the area is recorded during the experiments.

Given the data D collected by a user's device, we use D_a to denote the data that are collected when the user has contact with the virus (i.e., within the contact proximity with a confirmed case or in an infected area), and use D_b to denote the data that are detected as having contact with the virus. D_a is the ground-truth data while D_b is the detection result. Precision, recall, and F1-score are used as metrics to evaluate the contact detection results. The precision is defined as

$$\text{precision} = \frac{|D_a \cap D_b|}{|D_b|} \tag{6}$$

where $|\cdot|$ represents the data size. Similarly, recall is defined as

$$\text{recall} = \frac{|D_a \cap D_b|}{|D_a|}. \tag{7}$$

Based on the definition of precision and recall, F1-score is defined as

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{8}$$

We compare vContact with some other state-of-the-art approaches, which are introduced as follows.

1) *Bluetooth:* It is widely used for digital contact tracing, such as schemes [3], [7], [38]. To collect Bluetooth data, two mobile devices are put at a distance of k meters for 10 min in the three experimental sites, where k is set to be $\{1, 2, \ldots, 10\}$. We use one device as the broadcaster and another as the scanner. The scanner can scan the Bluetooth signal from the broadcaster, and the RSSI is recorded over time. For each contact proximity k meters, a threshold is selected for contact detection. If a received signal strength is larger than the threshold, they are detected as having contact.
2) *Jaccard Similarity:* It is used to evaluate the similarity of two sets, and it is defined as the size of the intersection divided by the size of the union of two sets. If the Jaccard similarity of two signal vectors is larger than a threshold, they are identified as within the contact proximity. It is also used in a relevant work for proximity estimation [45].
3) *Average Manhattan Distance (AMD):* It is used in previous works [9], [45], which is defined as
$$\text{AMD} = \frac{\sum_i \left|\text{RSSI}_{A,i} - \text{RSSI}_{B,i}\right|}{N} \tag{9}$$
where $\text{RSSI}_{A,i}$ is the received signal strength of AP i measured by user A, and N is the total number of overlapping APs. If the AMD of two signal vectors is less than a threshold, they are identified as within the contact proximity.
4) *Average Euclidean Distance (AED):* It is also used in the previous work [9], which is defined as
$$\text{AED} = \frac{\sqrt{\sum_i \left(\text{RSSI}_{A,i} - \text{RSSI}_{B,i}\right)^2}}{N} \tag{10}$$
where $RSSI_{A,i}$ is the received signal strength of AP i measured by user A, and N is the total number of overlapping APs. If the AED of two signal vectors is less than a threshold, they are identified as within the contact proximity.

For the baseline approaches AMD and AED, given two signal vectors A and B, if a signal is scanned in A but not in B, the signal strength is set as -100 in B for calculation, and *vice versa*.

B. Threshold α

As mentioned in Section III, the contact detection algorithm relies on a threshold α to identify contacts. In this section, we discuss the selection of α. Given the contact proximity km, if the distance of a user and the virus is less than km, she/he should be detected as having contact with the virus. Intuitively, α is relevant to the contact proximity and it should be different for different contact proximities. We use the data collected at ℓ_0 in a site as the data from confirmed cases, and detect contacts for data, which are collected at ℓ_i ($i > 0$) in the same site. When k meters is set as the contact proximity, D_a contains the data collected at ℓ_i where $i \leq k$.

Precision and recall are used as metrics, and the results of α versus precision and recall for $k = 1$ m, $k = 2$ m, and $k = 4$ m are presented in Fig. 5. As the threshold α increases, the precision increases while the recall declines. The reason is that a larger threshold indicates closer proximity. Thus, increasing the threshold would lead to high precision. However, if the threshold is set to be too large, some of the data distance of which is less than km will not be detected, resulting in a drop in recall.

The threshold can be selected according to the requirements of precision and recall for close contact detection. To balance the precision and recall, we select the intersection points, the precision and recall of which are equal for our following discussion. In Fig. 5(a), the precision and recall for $k = 1$ m are low when α is set as 0.25, which indicates that identifying contact within 1 m is difficult. As shown in Fig. 5(b), the precision and recall for $k = 2$ m have a significant improvement when the threshold is around 0.20. The precision and recall in Fig. 5(c) for $k = 4$ m are high (around 70%) if the threshold is around 0.17. We use the same strategy to select thresholds for other contact proximities.

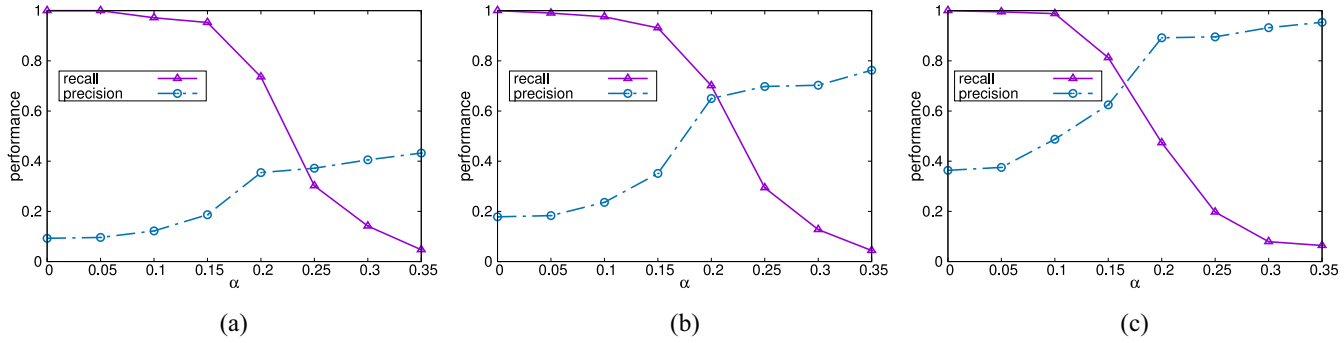


Fig. 5. Precision and recall for different contact proximity K. (a) $k = 1$ m. (b) $k = 2$ m. (c) $k = 4$ m.

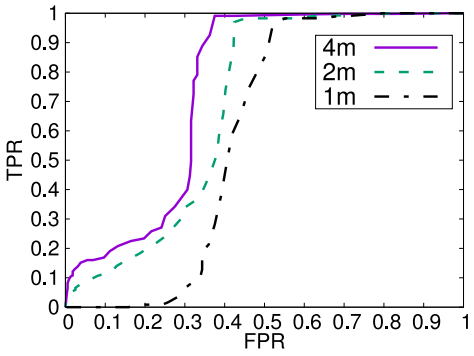


Fig. 6. ROC curve.

Furthermore, we use ROC to evaluate the performance of vContact for $k = 1$ m, $k = 2$ m, and $k = 4$ m. The ROC curves are created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold α. We present the results in Fig. 6. The results demonstrate the effectiveness of vContact. Moreover, as k increases, the performance of vContact becomes better because the signal difference is more significant when the distance between locations increases. The threshold α can also be selected based on the ROC curve to balance the TPR and FPR.

C. Site Study

We present the performance of contact detection in different sites in this section. We use different distances ($k = 1$, 2, 3, 4, 5) m to denote the contact proximity, and the threshold is set according to the discussion in Section IV-B. The results of precision, recall, and F1-score versus contact proximity are shown in Fig. 7.

In Fig. 7(a), as the contact proximity increases, the precision in the three sites increases, indicating that it is easier to detect contacts within a greater proximity. The precision for $k = 1$ m is low in all sites. The result shows the difficulties of identifying whether the contact happens in 1 m because the WiFi signals within a 1-m range are usually similar. However, the precision has significant improvements for larger contact proximity. The precision is high (50%–70%) when the proximity is 2 m. The precision indoors (office and shopping mall) is better than the precision outdoors because WiFi signals are more stable indoors. The improvement is more significant in the office scenario compared with the shopping mall scenario. The recall shown in Fig. 7(b) indicates the good performance of vContact to detect those who have close contact. We present the F1-score result in Fig. 7(c), indicating the satisfactory overall performance of vContact.

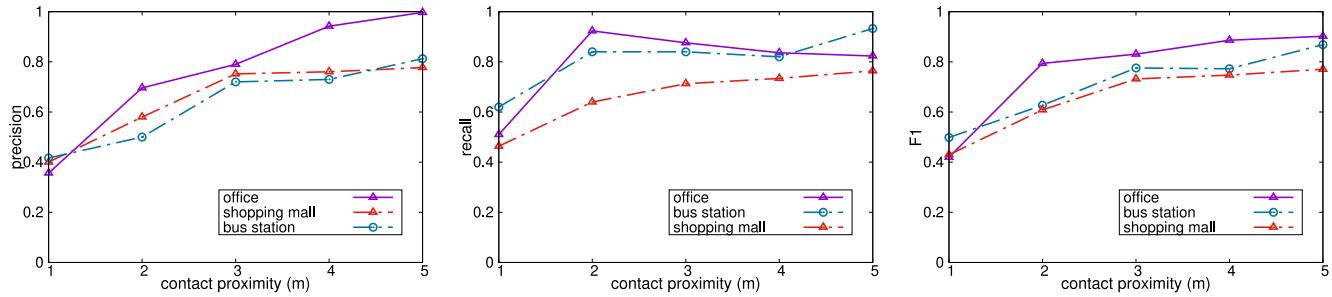
D. In–Out Detection of Infected Areas

Contact detection for confirmed cases without the app is to detect whether a user has been in or outside an infected area. We construct processed profiles for the office, bus station, and a store in a shopping mall using the collected WiFi data. Then, we compare the similarity between the processed profile of the area and the data collected in and outside the area. If the similarity is larger than the threshold α, the data are identified as being collected in the area and having contact with the virus. α is set as 0.2 in the experiment. Precision and recall are used as the metrics for evaluation. The results are shown in Fig. 8. The detection in all the sites achieves good performances. The precision and recall are high for the three sites, illustrating that vContact is very efficient for in–out detection of infected areas.

E. Comparison With Other Approaches

As the baseline approaches rely on a selected threshold to detect contact, for a given contact proximity, we use the same strategy to select thresholds as discussed in Section IV-B. Precision, recall, and F1-score are used as metrics for performance comparison.

The results of precision, recall, and F1-score versus proximity on the three data sets are presented in Figs. 9 (the office), 10 (the bus station), and 11 (the shopping mall). In Fig. 9, the precision, recall, and F-1 score of different approaches increase as the contact proximity increases. vContact always outperforms other baseline approaches on the metrics of precision and F-1 score. vContact has higher recall than others when contact proximity is less than 5 m and has similar performance to Bluetooth when the contact proximity is 5 m. The curves of precision, recall, and F1-score on the other data sets have a similar trend to that on the office data set. As shown in Fig. 10(a), the precision of Bluetooth is slightly higher than vContact on the bus station data set. But vContact has better performance than Bluetooth and other approaches with respect to recall and F1-score. As for the performance on the shopping mall data set, vContact has similar precision



(a) (b) (c)

Fig. 7. Performance in different sites. (a) Precision. (b) Recall. (c) F1-score.

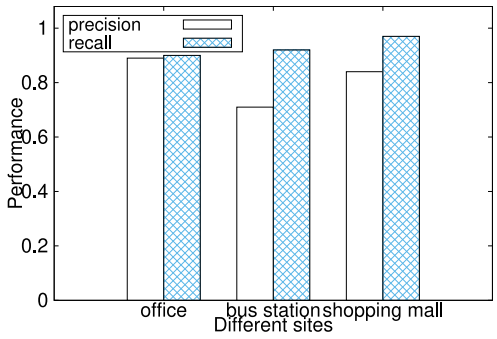


Fig. 8. Precision and recall of in–out detection.

to Bluetooth when contact proximity is 1 and 2 m, but has a significant improvement on precision when contact proximity is 3 and 4 m. In Fig. 11(b), vContact has similar recall to Bluetooth and AMD. vContact always outperforms other approaches, which use WiFi data for detection. Overall, vContact has a higher F-1 score than other approaches in all data sets, indicating that it is more efficient for contact detection. We can also learn from the figures that vContact and other approaches have better performance in the indoor scenario, and the improvement of vContact is more significant compared with the outdoor site.

F. AP Number

In this part, we evaluate the impact of AP number on the performance when the contact proximity is set as $k = 2$ m. . We randomly filter $\sigma\%$ signals from the signal vectors for each site, and compare the signal similarity of two devices for contact detection. The filtering rate $\sigma\%$ is set to be 10%–90%. The precision and recall versus the average signal number are presented in Fig. 12.

In Fig. 12, as the average signal number increases, the precision increases slightly. The precision is still acceptable when the average signal number is small. Even removing 90% of the signals, the precision does not drop significantly for the office and shopping mall sites. The precision outdoors (the bus station) are more stable than others. The recall shown in Fig. 12 does not have obvious change as the signal number changes, demonstrating the robustness of our approach.

G. Environmental Dynamicity

APs in a site may change at different times, for example, some APs may shut down or the RSSIs may be different. We study the impact of the difference of APs and RSSIs on the performance of vContact. Following the previous experiments, two phones are put at a distance of 2 m for data collection.

To study the impact of the difference of APs, we filter out $\sigma\%$ signal IDs from the signal profile of a phone while another remain the same. The filtering rate $\sigma\%$ is set to be 10%–90%. The precision, recall, and F1 score versus the filtering rates are presented in Fig. 13. As shown in Fig. 13(a), when more signals are filtered (i.e., $\sigma\%$ becomes larger), the precision, recall, and F1 score all decline. However, even with 50% of the signals filtered, vContact still achieves good performance in the three sites, which illustrates the robustness of vContact w.r.t the difference of APs.

Furthermore, to evaluate the impact of the difference of RSSIs, we add a Gaussian noise to the RSSIs in one phone's signal profile as follows:

$$s_i = s_i + d, d \sim \text{Gaussian}(0, \eta) \tag{11}$$

where s_i is the raw RSSI and d is the Gaussian noise. η is set to be 1–8. The precision, recall, and F1 score versus the filtering rates are presented in Fig. 14. The precision of vContact at the bus stop and shopping mall increases slightly when the noise becomes larger. The reason is that when the noise becomes large, the signal similarity becomes smaller and false positive declines. However, the recall and F1 score drop with the increase in the noise. Overall, the performance of vContact remains good when the noise is small (less than 3) but it drops significantly when the noise is large.

H. Heterogeneous Devices

Different devices have different abilities to scan WiFi signals. Two co-located devices may scan different signals and RSSIs. We evaluate the performance of different devices. For each device, we compare its data at ℓ_0 with other devices' data at ℓ_i ($i > 0$) in the same site. We set the contact proximity as 1–5 m, and set the threshold following the discussion in Section IV-B. Precision, recall, and F1-score are used as metrics.

The precision versus contact proximity for different devices in the office site is presented in Fig. 15(a). Given the contact

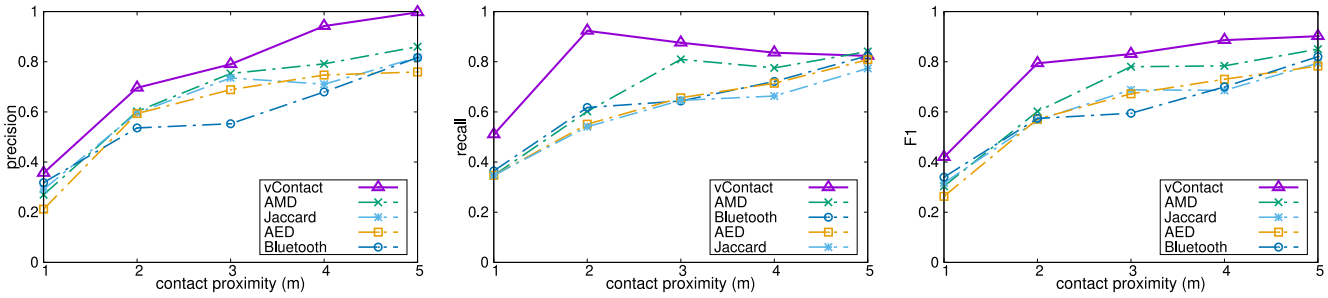


Fig. 9. Comparison with baseline approaches on the office data set. (a) Precision. (b) Recall. (c) F-1 score.

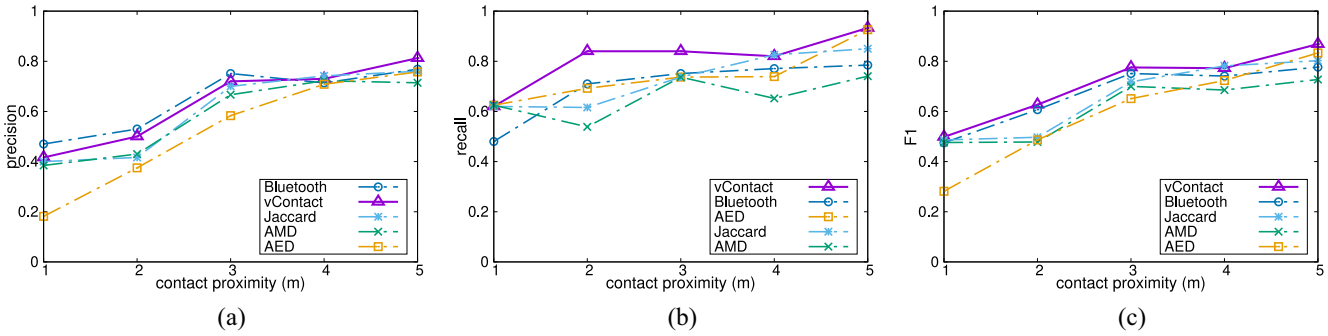


Fig. 10. Comparison with baseline approaches on the bus station data set. (a) Precision. (b) Recall. (c) F-1 score.

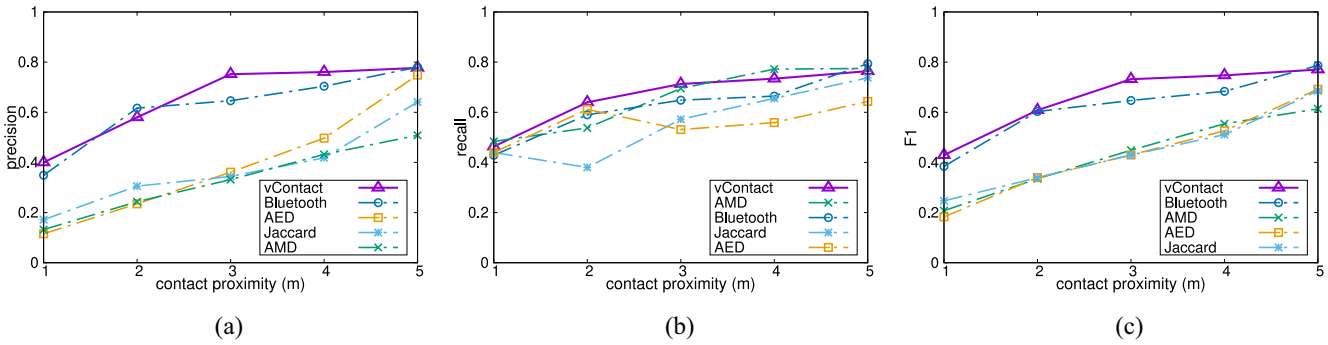


Fig. 11. Comparison with baseline approaches on the shopping mall data set. (a) Precision. (b) Recall. (c) F-1 score.

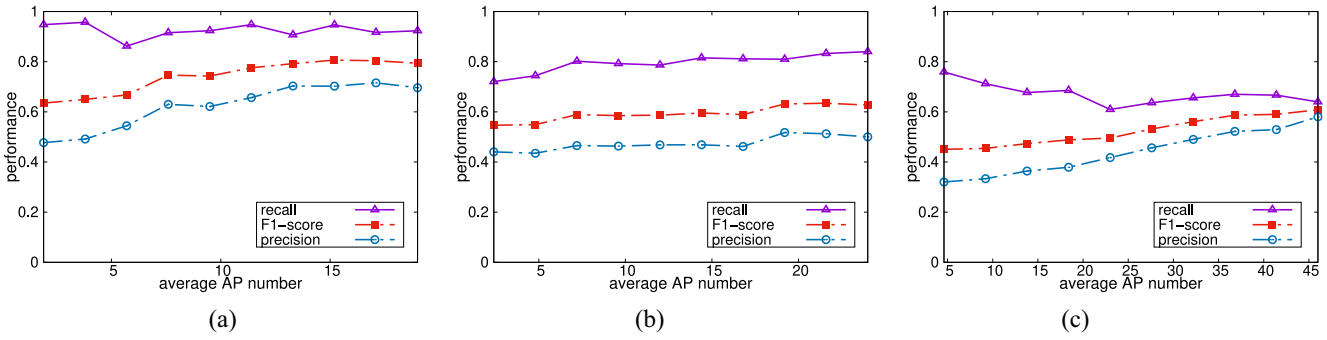


Fig. 12. Impact of signal numbers (AP numbers) on the performance of contact detection. (a) Office. (b) Bus station. (c) Shopping mall.

proximity, the precision is different for distinct devices, which is consistent with our discussion. As the contact proximity increases, the precision of all devices increases. The precision of all devices significantly increases when $k \geq 2$ m. The recall versus contact proximity for different devices in the office is presented in Fig. 15(b). Similar to the result of precision, the performance of all devices has a large improvement in recall when $k = 2$ m. All devices achieve high recall when $k \geq 2$ m,

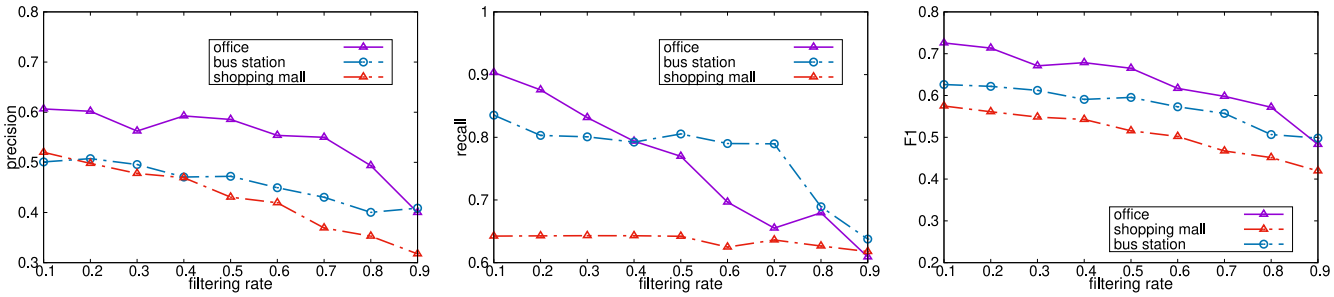


Fig. 13. Impact of different filtering rates on the performance of contact detection. (a) Precision. (b) Recall. (c) F1-score.

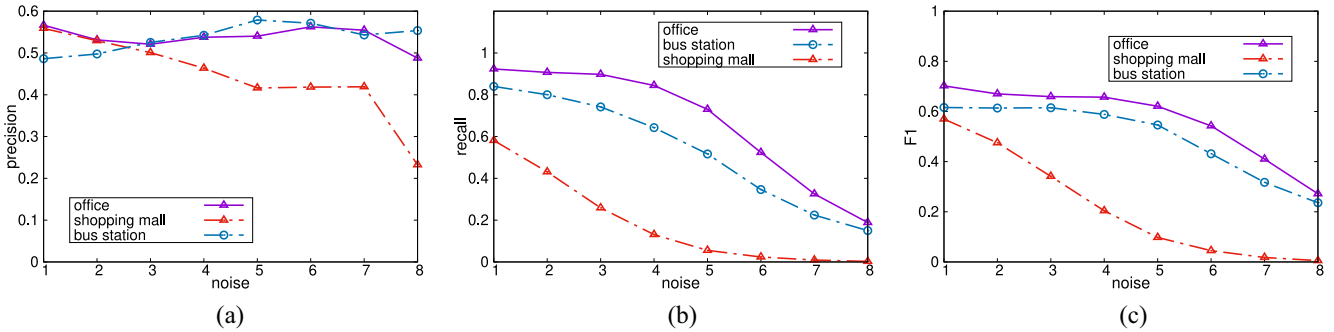


Fig. 14. Impact of different noise levels on the performance of contact detection. (a) Precision. (b) Recall. (c) F1-score.

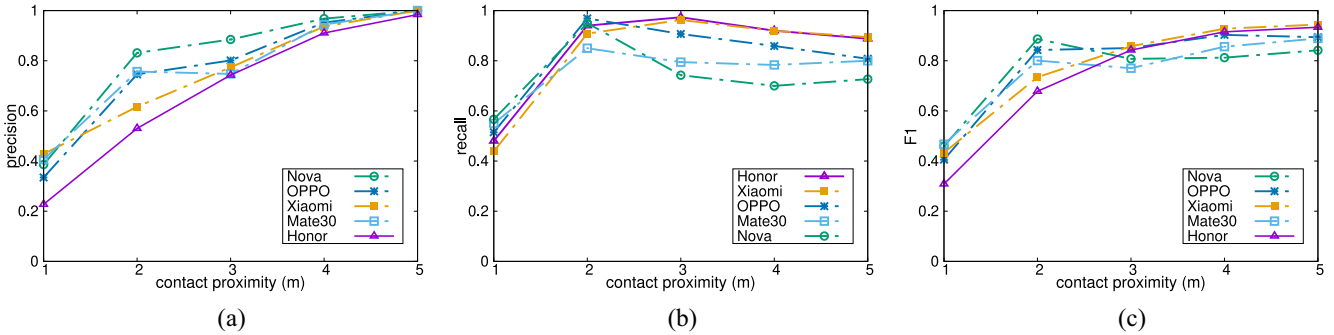


Fig. 15. Performance of different devices. (a) Precision. (b) Recall. (c) F1-score.

indicating the good performance of our approach on recall. The F1-score result is shown in Fig. 15(c), which demonstrates the good overall performance of all tested devices. The results demonstrate that our approach is effective and can be applied to phones of different brands.

I. Data Sampling Rate

Since the APs and their RSSIs of a site do not change in a short time, the impact of data sampling rates is not obvious when users are stationary. Consequently, we discuss its impact for the scenario when users are moving.

Some users are walking in groups in the campus with their mobile phone to collect WiFi data. The time interval of data is set as 10–80 s in the experiment. For the data of the same group, we use recall as the metric to evaluate the performance of contact detection. The result of recall is presented in Fig. 16. As the time interval becomes larger, the recall of the detection declines. The reason is that users may scan WiFi data at two locations where the similarity of the WiFi is significantly different when users are moving and the time interval is larger. As a result, contact is more difficult to detect.

V. IoT Implementation as Case Study

With the vContact SDK, we have installed it into Android smart watches whose data are synced to one's phone. Through an app, the user is notified of his/her exposure duration to the virus. We report the smart watch implementation details and user interface in Section V-A. Besides smart watches, we have also installed the SDK on Android phones. We validate its design and performance in Section V-B.

A. Smart Watch Implementation

Our SDK can be run independently on Android phones for data collection and exposure detection. It can also run

TABLE V
RESULT OF EXPOSURE NOTIFICATION FOR A SEPARATION OF 2 M

Confirmed Case \ User	Honor	Mate 30	OPPO	Huawei Nova	Vivo
Honor	–	√	√	√	√
Mate30	×	–	√	√	√
OPPO	√	√	–	√	√
Huawei Nova	×	√	√	–	√
Xiaomi	×	√	√	√	–

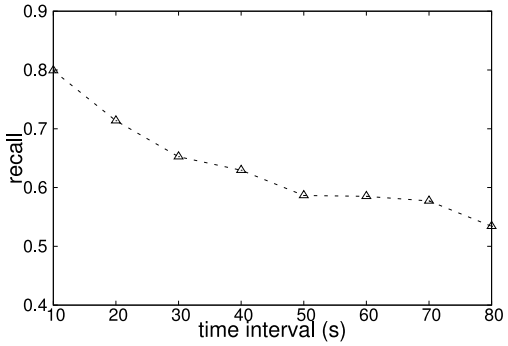


Fig. 16. Impact of different time intervals.

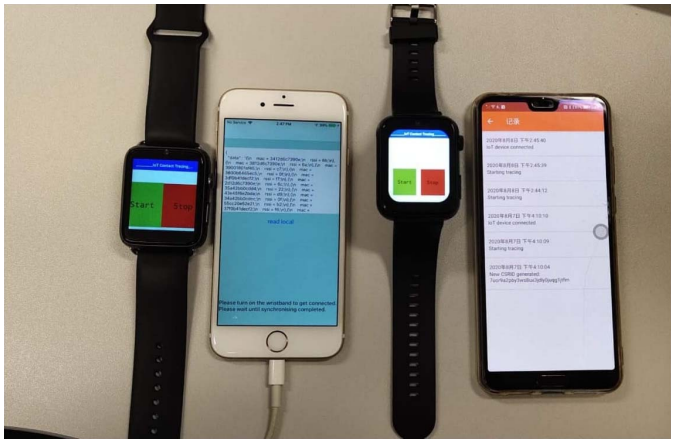


Fig. 17. Data transition from smart watches to Android and iOS phones.

on Android IoT devices to collect data and transit the data to an Android/iOS phone for exposure detection. We show in Fig. 17 an IoT smart watch system we have built based on vContact. The smart watches pair with phones through an Android or iOS app, as illustrated here with an Android phone and iPhone. Scanned data of the smart watches are synced with their phone apps for exposure computation and notification.

The user interface of the phone app is shown in Fig. 18. As shown in Fig. 18(a), once a user turns on the button of "Exposure data collection," the app will start to scan nearby WiFi and store the data locally every 1 min. Users may turn off data collection anytime and anywhere for personal reasons. The signal IDs (i.e., the AP MAC addresses) are encrypted when the data are stored. If a user is confirmed as being infected, she/he could upload her/his signal profile to the server [Fig. 18(b)], so that others could download the data for matching. If a user has close contact with a confirmed case, she/he will receive a notification, showing when the close contact happened and how long the contact duration was [Fig. 18(c)]. In the app, data are downloaded and matched automatically every day. For the purpose of testing, we also have a testing mode as shown in Fig. 18(d), by which we can download the data, and trigger the detection manually during the testing.

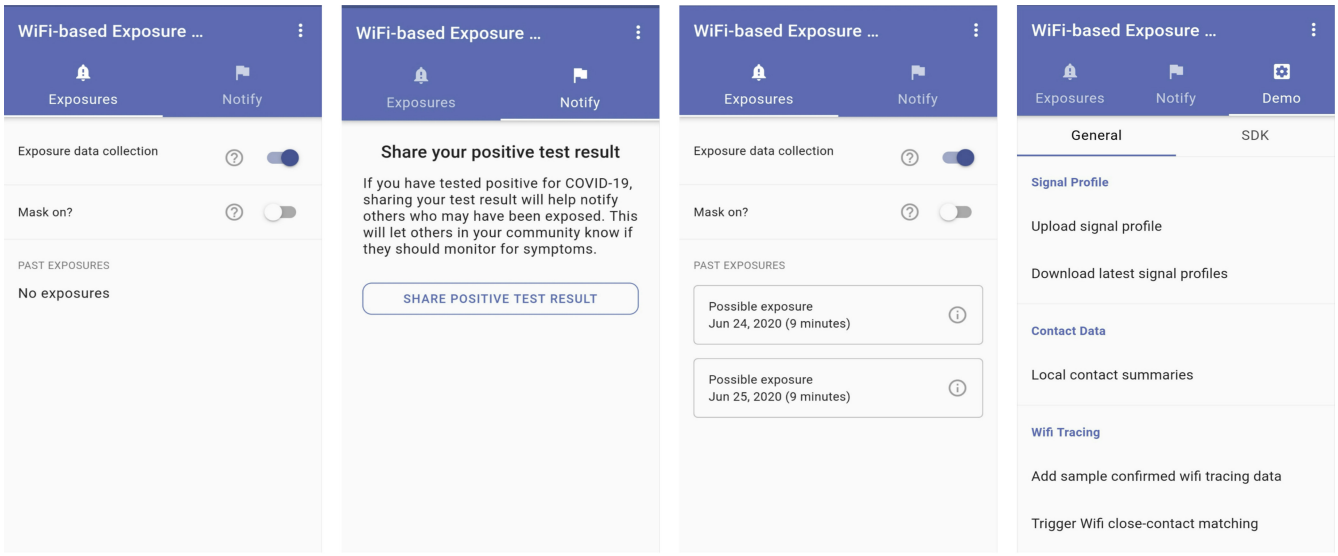
B. Testing and Validation for Phones

Besides smart watches, we have installed the SDK on five Android phones, namely, Honor, Huawei Nova, Huawei Mate30, Xiaomi, and OPPO. We present here a case study on these phones. We set the contact proximity as 2 m for testing. The app collects WiFi data every 1 min. Hence, the detection approach introduced in Section III will report a detection result (i.e., true or false) for the data at each minute. In our testing, if a user stays with the virus within 2 m for more than 5 min in a 10-min sliding time window, she/he will receive a possible exposure notification. The virus lifespan is set to be 30 min. Note that the contact duration, the length of the sliding time window, and the virus lifespan are parameters for the app, which can be changed according to the advice of the health officer.

We test the app in an office using the five phones. The procedures are as follows. One of the phones is selected as the confirmed case, and other phones are put at a location, which is 2 m away from the confirmed case. The button exposure data collection is turned on for 15 min. Then, the confirmed case uploads its signal profile, and the other phones download the signal profile for matching. After that, we put other phones at a location, which is 4-m away from the confirmed case and repeat the testing. Each phone is selected as the confirmed case in turn. The ideal result is that a phone only receives a notification when it is 2-m away from the confirmed case but there is no notification for 4 m. The testing results are presented in Tables V and VI. √ represents that a phone receives a notification, while × means it does not receive a notification.

Table V shows the results of exposure notification for 2 m. It illustrates the good performance of our app for exposure notification. The performance of the Honor phone is not as good as that of other phones, indicating the different ability of phones to scan WiFi signals.

We show the results of exposure notification for 4 m in Table VI. Compared with the results in Table V, more phones are detected as having nonclose contact, which is consistent with our expectation. Performance is distinct for different phones, but the overall performance is good.



(a) (b) (c) (d)

Fig. 18. User interface of an app to notify users of viral exposure duration. (a) Exposure data collection. (b) Share positive test result. (c) Possible exposure notification. (d) Testing mode.

TABLE VI
RESULT OF EXPOSURE NOTIFICATION FOR A SEPARATION OF 4 M

Confirmed Case \ User	Honor	Mate 30	OPPO	Huawei Nova	Vivo
Honor	–	×	×	√	√
Mate30	×	–	√	×	×
OPPO	×	√	–	√	√
Huawei Nova	×	×	×	–	×
Xiaomi	×	√	√	√	–

VI. CONCLUSION AND FUTURE WORKS

We have proposed vContact, a novel WiFi-based private IoT contact tracing scheme with virus lifespan, which may be spatial-temporally different due to the sanitization process. By detecting close contact based on the similarity of WiFi data, vContact captures both direct face-to-face and indirect environmental contact. To the best of our knowledge, this is the first decentralized work to consider the virus lifespan for both direct and indirect private contact tracing using WiFi. We proposed and studied data processing approaches and a signal similarity metric for close contact detection. Due to the ubiquity of WiFi signals, vContact can be pervasively deployed for contact tracing.

We conducted extensive experiments on vContact by implementing it as an SDK. Our experimental results show that vContact achieves high precision, recall, and F1-score (up to 90% when the contact proximity is 2 m) for different experimental sites, and its performance is robust against AP numbers, and devices of different brands. Even with a small number of signals (5), vContact still achieves good performance. This mean it is widely applicable to city or suburban areas. It is also robust against environmental changes to detect indirect contact, even if a substantial fraction (50%) of the APs has been changed. We have installed vContact SDK into IoT devices of smart phones and watches, and validate the simplicity, implementability, and efficiency of our design.

We discussed below the possible future directions of the work. One is to extend vContact so that it can be integrated with other non-RF signals, such as INS and geomagnetism to strengthen its contact tracing capability, especially in areas where WiFi or Bluetooth are not available. Moreover, the theoretical analysis of the proposed technique under hypothesis testing context is crucial and useful, which has been used in WSN-based data fusion [46], [47]. Another direction is to identify those dynamic or ephemeral APs (e.g., hotspots of smartphone) from their MAC address, so that they could be filtered out in contact tracing. To this end, we can build a dynamic, scalable, and crowdsourced reference database for those permanent MACs to improve further the robustness of vContact. Yet another direction is to strike a balance between power conservation and WiFi scanning frequency. To preserve battery without compromising on tracing accuracy, we may use a lower data sampling rate when users are stationary and a higher one when moving. To achieve this, we need to devise a dynamic data sampling algorithm by estimating user activity using INS or other signals.

REFERENCES

[1] W. Aylin and G. Shayanne. (2020). *One Chart Shows How Long the Coronavirus Lives on Surfaces Like Cardboard, Plastic, Wood, and Steel*. [Online]. Available: https://www.businessinsider.com/coronavirus-lifespan-on-surfaces-graphic-2020-3

[2] R. Gray. (2020). *COVID-19: How Long Does the Coronavirus Last on Surfaces?* [Online]. Available: https://www.bbc.com/future/article/20200317-covid-19-how-long-does-the-coronavirus-last-on-surfaces

[3] C. Troncoso *et al.*, "Decentralized privacy-preserving proximity tracing," document DP-3T, Github, San Francisco, CA, USA, 2020.

[4] A. Berke, M. Bakker, P. Vepakomma, R. Raskar, K. Larson, and A. Pentland, "Assessing disease exposure risk with location histories and protecting privacy: A cryptographic approach in response to a global pandemic," 2020. [Online]. Available: arXiv:2003.14412.

[5] A. Wesolowski, C. O. Buckee, L. Bengtsson, E. Wetter, X. Lu, and A. J. Tatem, "Commentary: Containing the ebola outbreak-the potential and challenge of mobile network data," *PLoS Currents*, vol. 6, Sep. 2014.

[6] J. Bell, D. Butler, C. Hicks, and J. Crowcroft, "TraceSecure: Towards privacy preserving contact tracing," 2020. [Online]. Available: arXiv:2004.04059.

[7] J. Chan *et al.*, "PACT: Privacy sensitive protocols and mechanisms for mobile contact tracing," 2020. [Online]. Available: arXiv:2004.03544.

[8] N. Ahmed *et al.*, "A survey of COVID-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134577–134601, 2020.

[9] P. Sapiezynski, A. Stopczynski, D. K. Wind, J. Leskovec, and S. Lehmann, "Inferring person-to-person proximity using WiFi signals," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–20, 2017.

[10] J. Li and X. Guo, "COVID-19 contact-tracing apps: A survey on the global deployment and challenges," 2020. [Online]. Available: arXiv:2005.03599.

[11] L. Reichert, S. Brack, and B. Scheuermann, "A survey of automatic contact tracing approaches," in *Proc. Cryptol. ePrint Archive*, 2020, p. 672.

[12] F. Qi and F. Du, "Tracking and visualization of space-time activities for a micro-scale FLU transmission study," *Int. J. Health Geograph.*, vol. 12, no. 1, pp. 1–16, 2013.

[13] J. K. Fitzsimons, A. Mantri, R. Pisarczyk, T. Rainforth, and Z. Zhao, "A note on blind contact tracing at scale with applications to the COVID-19 pandemic," 2020. [Online]. Available: arXiv:2004.05116.

[14] L. C. Klopfenstein, S. Delpriori, G. M. Di Francesco, R. Maldini, B. D. Paolini, and A. Bogliolo, "Digital AriadNE: Citizen empowerment for epidemic control," 2020. [Online]. Available: arXiv:2004.07717.

[15] L. Reichert, S. Brack, and B. Scheuermann, "Privacy-preserving contact tracing of COVID-19 patients," *IACR Cryptol. ePrint Arch.*, vol. 2020, pp. 375–276, 2020.

[16] G. Li, C.-J. Chen, W.-C. Peng, and C.-W. Yi, "Estimating crowd flow and crowd density from cellular data for mass rapid transit," in *Proc. 6th Int. Workshop Urban Comput.*, Halifax, NS, Canada, 2017, pp. 18–30.

[17] G. Li *et al.*, "Public transportation mode detection from cellular data," in *Proc. ACM Conf. Inf. Knowl. Manag.*, 2017, pp. 2499–2502.

[18] L. Isella *et al.*, "Close encounters in a pediatric ward: Measuring face-to-face proximity and mixing patterns with wearable sensors," *PLoS ONE*, vol. 6, no. 2, 2011, Art. no. e17144.

[19] M. Salathé, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones, "A high-resolution human contact network for infectious disease transmission," *Proc. Nat. Acad. Sci. USA*, vol. 107, no. 51, pp. 22020–22025, 2010.

[20] A. Trivedi, C. Zakaria, R. Balan, A. Becker, G. Corey, and P. Shenoy, "WiFiTrace: Network-based contact tracing for infectious diseases using passive wifi sensing," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 1–26, 2021.

[21] E. Cecchet, A. Acharya, T. Molom-Ochir, A. Trivedi, and P. Shenoy, "WiFiMON: A mobility analytics platform for building occupancy monitoring and contact tracing using WiFi sensing," in *Proc. 18th Conf. Embedded Netw. Sensor Syst.*, 2020, pp. 792–793.

[22] Turkey. (2020). *Corowarner*. [Online]. Available: https://www.aa.com.tr/en/latest-on-coronavirus-outbreak/turkey-to-use-contact-tracing-app-to-detect-coronavirus/1804425

[23] India. (2020). *Aarogya Setu Mobile App*. [Online]. Available: https://www.mygov.in/aarogya-setu-app/

[24] Argentia. (2020). *Cotrack*. [Online]. Available: https://www.cotecna.com/en/services/government/cargo-tracking-solution-cotrack

[25] P. Wang, C. Lin, M. S. Obaidat, Z. Yu, Z. Wei, and Q. Zhang, "Contact tracing incentive for COVID-19 and other pandemic diseases from a crowdsourcing perspective," *IEEE Internet Things J.*, early access, Jan. 4, 2021, doi: 10.1109/JIOT.2020.3049024.

[26] H. Cho, D. Ippolito, and Y. W. Yu, "Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs," 2020. [Online]. Available: arXiv:2003.11511.

[27] S. Jeong, S. Kuk, and H. Kim, "A smartphone magnetometer-based diagnostic test for automatic contact tracing in infectious disease epidemics," *IEEE Access*, vol. 7, pp. 20734–20747, 2019.

[28] A. Hekmati, G. Ramachandran, and B. Krishnamachari, "CONTAIN: privacy-oriented contact tracing protocols for epidemics," 2020. [Online]. Available: arXiv:2004.05251.

[29] F. Sattler *et al.*, "Risk estimation of SARS-CoV-2 transmission from Bluetooth low energy measurements," 2020. [Online]. Available: arXiv:2004.11841.

[30] Y. Xia and G. Lee, "How to return to normalcy: Fast and comprehensive contact tracing of COVID-19 through proximity sensing using mobile devices," 2020. [Online]. Available: arXiv:2004.12576.

[31] C. Günther, M. Günther, and D. Günther, "Tracing contacts to control the COVID-19 pandemic," 2020. [Online]. Available: arXiv:2004.00517.

[32] J. Bay, J. Kek, A. Tan, C. S. Hau, L. Yongquan, J. Tan, and T. A. Quy. (2020). *BlueTrace: A Privacy-Preserving Protocol for Community-Driven Contact Tracing Across Borders*. [Online]. Available: https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf

[33] C. Castelluccia *et al.*, "ROBERT: Robust and privacy-preserving proximity tracing," PRIVATICS Team, Inria, France Collaboration Fraunhofer AISEC, Garching, Germany, Working Paper, May 2020. [Online]. Available: https://hal.inria.fr/hal-02611265

[34] Singapore. (2020). *Tracetogether*. [Online]. Available: https://www.tracetogether.gov.sg

[35] Australian. (2020). *Covidsafe*. [Online]. Available: https://www.health.gov.au/resources/apps-and-tools/covidsafe-app

[36] C. Castelluccia *et al.*, "DESIRE: A third way for a european exposure notification system leveraging the best of centralized and decentralized systems," 2020. [Online]. Available: arXiv:2008.01621.

[37] H. Xu, L. Zhang, O. Onireti, Y. Fang, W. J. Buchanan, and M. A. Imran, "BeepTrace: Blockchain-enabled privacy-preserving contact tracing for COVID-19 pandemic and beyond," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3915–3929, Mar. 2021.

[38] R. L. Rivest *et al.* (2020). *PACT: Private Automated Contact Tracing*. [Online]. Available: https://pact.mit.edu/

[39] G. Avitabile, V. Botta, V. Iovino, and I. Visconti, "Towards defeating mass surveillance and sars-cov-2: The pronto-c2 fully decentralized automatic contact tracing system," IACR, Bellevue, WA, USA, Rep. 2020/493, 2020. [Online]. Available: https://eprint.iacr.org/2020/493

[40] Google and Apple. (2020). *Privacy-Preserving Contact Tracing*. [Online]. Available: https://www.apple.com/covid19/contacttracing

[41] A. Sydney *et al.* (2020). *Slowing the Spread of Infectious Diseases Using Crowdsourced Data*. [Online]. Available: https://www.covid-watch.org/

[42] Switzerland. (2020). *SwissCovid*. [Online]. Available: https://en.wikipedia.org/wiki/SwissCovid

[43] M. K. Hoang, J. Schmalenstroeer, C. Drueke, D. T. Vu, and R. Haeb-Umbach, "A hidden Markov model for indoor user tracking based on WiFi fingerprinting and step detection," in *Proc. IEEE 21st Eur. Signal Process. Conf. (EUSIPCO)*, 2013, pp. 1–5.

[44] W. Qian, F. Lauri, and F. Gechter, "Convolutional mixture density recurrent neural network for predicting user location with WiFi fingerprints," 2019. [Online]. Available: arXiv:1911.09344.

[45] M. Dmitrienko, A. Singh, P. Erichsen, and R. Raskar, "Proximity inference with WiFi-colocation during the COVID-19 pandemic," 2020. [Online]. Available: arXiv:2009.12699.

[46] D. Ciuonzo and P. Salvo Rossi, "DECHADE: Detecting slight changes with hard decisions in wireless sensor networks," *Int. J. Gen. Syst.*, vol. 47, no. 5, pp. 535–548, 2018.

[47] D. Ciuonzo, P. S. Rossi, and P. K. Varshney, "Distributed detection in wireless sensor networks under multiplicative fading via generalized score tests," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9059–9071, Jun. 2021.

Guanyao Li received the Bachelor of Engineering degree from Zhejiang University, Hangzhou, China, in 2015, and the Master of Engineering degree from National Chiao Tung University, Hsinchu, Taiwan, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong.

His research interests include spatial–temporal data mining, urban computing, and graph mining.

Siyan Hu received the Master of Science degree from the Hong Kong Polytechnic University, Hong Kong, in 2014.

He is currently working as a Research Assistant with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong. Her research interests include mobile computing and indoor positioning.

Shuhan Zhong received the bachelor's and master's degrees from Wuhan University, Wuhan, China, in 2015 and 2018, respectively, the Master of Science degree from The Hong Kong University of Science and Technology, Hong Kong, in 2020, where he is currently pursuing the Ph.D. degree in computer science and engineering with the Department of Computer Science and Engineering.

His research interests include multimodal deep learning, spatiotemporal data mining, and urban computing.

Wai Lun Tsui received the Bachelor of Engineering degree from The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2020.

He is currently working as a Research Assistant with the Department of Computer Science and Engineering, HKUST. His research interests include indoor positioning and mobile computing.

S.-H. Gary Chan (Senior Member, IEEE) received the B.S.E. degree (Highest Hons.) in electrical engineering from Princeton University, Princeton, NJ, USA, in 1993, with certificates in Applied and Computational Mathematics, Engineering Physics, and Engineering and Management Systems, and the M.S.E. and Ph.D. degrees in electrical engineering with a minor in business administration from Stanford University, Stanford, CA, USA, in 1994 and 1999, respectively.

He is currently a Professor with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong, where he is also an Affiliate Professor of Innovation, Policy and Entrepreneurship Thrust Area of HKUST(GZ), a Chair of the Committee on Entrepreneurship Education Program, and a Board Director of Hong Kong Logistics and Supply Chain MultiTech Research and Development Center. He was a Visiting Professor and a Researcher of Microsoft Research with Princeton University, Stanford University, and the University of California at Davis, Davis, CA, USA. At HKUST, he was a Director of Entrepreneurship Center, Sino Software Research Institute, and Computer Engineering Program, and a Co-Director of Risk Management and Business Intelligence Program. His research interests include smart sensing and IoT, cloud and fog/edge computing, indoor localization and mobile computing, video/location/user/data analytics, and IT entrepreneurship.

Prof. Chan was a recipient of the Google Mobile 2014 Award and Silver Award of Boeing Research and Technology. He was a recipient of the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence at Princeton. He has been an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA, and a Vice-Chair of Peer-to-Peer Networking and Communications Technical SubCommittee of IEEE Comsoc Emerging Technologies Committee. He has been a Guest Editor of *ACM Transactions on Multimedia Computing, Communications and Applications*, IEEE TRANSACTIONS ON MULTIMEDIA, *IEEE Signal Processing Magazine*, and *IEEE Communication Magazine*. He is a Steering Committee Member and was the TPC Chair of IEEE Consumer Communications and Networking Conference, and a Area Chair of the Multimedia Symposium of IEEE Globecom and IEEE ICC. He 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 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. He was a William and a Leila Fellow at Stanford University.