


# COVID-19 and Your Smartphone: BLE-Based Smart Contact Tracing

Pai Chet Ng , *Student Member, IEEE*, Petros Spachos , *Senior Member, IEEE*, and Konstantinos N. Plataniotis , *Fellow, IEEE*

**Abstract**—While contact tracing is of paramount importance in preventing the spreading of infectious diseases, manual contact tracing is inefficient and time consuming as those in close contact with infected individuals are informed hours, if not days, later. This article proposes a smart contact tracing (SCT) system utilizing the smartphone's Bluetooth low energy signals and machine learning classifiers to automatically detect those possible contacts to infectious individuals. SCT's contribution is two-fold: a) classification of the user's contact as high/low-risk using precise proximity sensing, and b) user anonymity using a privacy-preserving communication protocol. To protect the user's privacy, both broadcasted and observed signatures are stored in the user's smartphone locally and only disseminate the stored signatures through a secure database when a user is confirmed by public health authorities to be infected. Using received signal strength each smartphone estimates its distance from other user's phones and issues real-time alerts when social distancing rules are violated. Extensive experimentation utilizing real-life smartphone positions and a comparative evaluation of five machine learning classifiers indicate that a decision tree classifier outperforms other state-of-the-art classification methods with an accuracy of about 90% when two users carry their smartphone in a similar manner. Finally, to facilitate research in this area while contributing to the timely development, the dataset of six experiments with about 123 000 data points is made publicly available.

**Index Terms**—Bluetooth low energy, COVID-19, contact tracing, physical distancing, proximity, smartphone.

## I. INTRODUCTION

CONTACT tracing is an important step in containing a disease outbreak [1], [2]. Many efforts have been devoted to tracing a list of contacts when a person is diagnosed with a highly infectious disease, such as COVID-19. The current contact tracing method, which requires a collaborative effort from several authorized personnel, is labor-intensive and time-consuming [3]. Since it takes time to trace the contact, the group of users who have been in contact with an infected individual might spread the disease to another group of people before they

get informed. It is critical to have an effective contact tracing method that not only can automatically inform the potential users immediately but also reducing the required amount of labor force [4].

To this end, a smart contact tracing (SCT) system is introduced by exploiting the Bluetooth low energy (BLE) signals on smartphones. BLE is readily available on many smartphones making it ideal for the introduced system [5]. On the other hand, smartphones have become an intimate device in our everyday life. While we might leave the smartphone away from us when we are in our private space (e.g., home, private office, etc.), we always carry the smartphone when we do the grocery shopping, commute on public transport, walk along the open street, etc. In this way, smartphones are the best choice for contact tracing, in which the tracing is only performed when the user is in the public space. At any time, no location or any other information regarding the users is collected or transmitted. The application uses only BLE signals and no information exchange. The system has three main objectives: preserve-privacy, provide accurate contact tracing, and provide real-time proximity alerts.

1) *Preserve Privacy*: We leverage the beaconing feature in BLE wireless technology to broadcast an encrypted packet periodically [6]. This encrypted packet is broadcast on the nonconnectable advertising channels, preventing unauthorized access to the user's smartphone. The packet encrypts a piece of signature information based on the ambient environmental features, which is unique and is almost impossible to be duplicated by another device on another occasion. All the broadcast signatures and observed signatures will be stored in the local storage. The user is only required to upload their own broadcast signatures to the signature database when the user is confirmed to be infected with the contagious disease. Otherwise, the signature store in the local storage will be deleted automatically when it is expired. We define the signature expiration according to the disease spreading time window, as suggested by the health authorities. By comparing the signature of each smartphone, a list of possible contacts can be retrieved without explicitly revealing the sensitive information of the infected user.

2) *Accurate Contact Tracing*: The smartphone application identifies contacts in proximity, over time. It records the estimated distance and the duration of interaction between individuals. It will identify when someone has been too close to an infected individual for too long (the too close for too long (TC4TL) problem). For instance, when people hug each other they are too close for a short duration, while inside the cabin of a flight, people can be in ten meters distance for too long, breathing the same air. At the same time, a distance of two meters in a classroom might be safe while three meters in a subway train might trigger an alert. We have to rely heavily on the virologists

Manuscript received May 11, 2020; revised October 26, 2020 and December 16, 2020; accepted January 26, 2021. Date of publication March 9, 2021; date of current version December 9, 2021. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under an Alliance COVID-19 grant. (Corresponding author: Petros Spachos.)

Pai Chet Ng is with the Department of Electronics and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: pcng@connect.ust.hk).

Petros Spachos is with the School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada (e-mail: petros@uoguelph.ca).

Konstantinos N. Plataniotis is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S, Canada (e-mail: kostas@ece.utoronto.ca).

Digital Object Identifier 10.1109/JSYST.2021.3055675

and epidemiologists to identify the healthy distance in different environments, and through this application, we can give them access to these crucial details.

3) *Real-Time Proximity Alerts*: The application will provide a real-time alert to the user if the physical distancing between any two users is not maintained. This will be achieved by detecting the proximity between any users in a given location, including the grocery store, public transit, etc. This proximity information can be retrieved by inspecting the RSS patterns from the user's smartphone [7]. A smartphone can measure the RSS value upon seeing the packet broadcast by the nearby smartphone. Since RSS is inversely proportional to the square of the distance [8], we can use it to estimate the distance between any two smartphones and then classify the proximity based on the recommended physical distancing rule.

Our idea of SCT is inspired by an international trip (from Asia to North America) by the end of February. Borrowing our rich experiences on BLE research, we understand that using BLE on a smartphone is feasible but challenging. Specifically, RSS is subject to severe fluctuation especially the body shadowing effect since the smartphone is carried by users [9]. We examine five machine learning-based classifiers: decision tree (DT), linear discriminant analysis (LDA), naive Bayes (NB),  $k$ -nearest neighbors (kNN), and support vector machine (SVM), over six different smartphone positions: Hand-to-hand (HH), hand-to-pocket (HP), hand-to-backpack (HB), pocket-to-backpack (PB), pocket-to-pocket (PP), and backpack-to-backpack (BB).

In summary, this article has the following contributions.

- 1) *Proximity sensing and real-time physical distance alert*: we classify the proximity of a user to any user based on the RSS values measured by each smartphone, while a notification is pushed to alert the users when anyone violates the physical distancing rule. After approximately 10 s of interaction between smartphones, the system is able to provide a reliable estimation. DT is the most accurate classifier with an accuracy of up to 90%.
- 2) *Privacy-preserving signature protocol*: our SCT system provides a secure contact tracing by using the nonconnectable advertising channels and an encrypted packet containing unique signature information based on the ambient environmental features observed by a smartphone.
- 3) *Smartphone implementation and effects of smartphone's position*: we prototyped our system design and implemented the application in modern smartphones to demonstrate the feasibility of our proposed SCT. The energy requirements of the application are negligible. We compared the classifiers in terms of their estimation accuracy, while we examine six different positioning sets of smartphones. When the users have their smartphones in similar positions, the classifiers can improve accuracy.
- 4) *Extensive experiments*: we performed extensive experiments in a real-world setting to verify the effectiveness of our SCT. All the collected data are available in the IEEE Dataport [10] and Github [11]. The overall dataset contains the measurement data obtained from six experimental sets, amounting to a total of 123 718 data points. We believe that the dataset will serve as an invaluable resource for researchers in this field, accelerating the development of contact tracing applications.

The rest of this article is organized as follows. Section II provides a succinct overview of contact tracing and BLE technology. Section III presents the architecture of our proposed SCT.

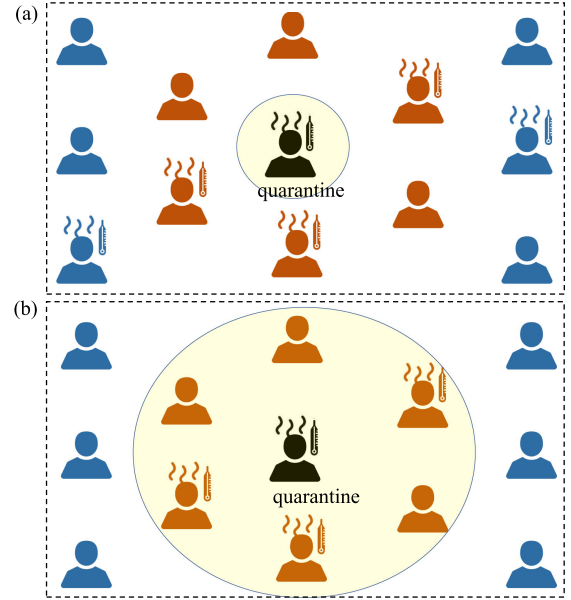


Fig. 1. Spread of an infectious disease: (a) only quarantine the infected individual and (b) quarantine the infected individual and all the users who have been in contact with the infected individual.

Section IV presents our proposed privacy-preserving signature protocol. Section V describes the risk classification based on the proximity information. Section VI discusses the experiments conducted in a real-world setting. Finally, Section VII concludes this article.

## II. BACKGROUND, MOTIVATION, AND COMPONENTS

Contact tracing aims to track down a group of users who have encountered an infected individual. The goal is to inform this group of users regarding the potential risk that they might face so that they can take appropriate actions as recommended by the local health authority.

### A. Current Development in Contact Tracing

An illustration of the differences in having a contact tracing system is shown in Fig. 1. In practice, a person will be quarantined immediately when they are confirmed to be infected with the disease. However, those people who have been in close contact with the infected individual are still free to move without realizing that they may become a virus carrier. With contact tracing, we can inform most of the potential close contact so that they can take appropriate action to isolate themselves from the crowd.

Recognizing the importance of contact tracing, many countries have put efforts to develop a smartphone-based contact tracing system.

- 1) China: A close contact detector based on QR codes and country-wide surveillance system is implemented [12]. The system can push an alert to users if they have been in close contact with the infected individual.
- 2) South Korea: A tracker application uses the GPS data from the user's smartphone to detect the distance to the infected individual. The application will also push a notification

that contains the personal details of the infected individuals [13].

- 3) Singapore: A privacy-preserving approach is adopted by the government's initiated TraceTogether application [14]. The application exploits the BLE signal on the smartphone to detect the proximity between any two individuals and broadcasts an encrypted packet generated by a secret key distributed by the Ministry of Health.
- 4) Switzerland: The SwissCovid application is developed based on a decentralized privacy-preserving proximity tracing protocol. It utilizes the BLE signal on the smartphone to detect the proximity while applying an encryption method to protect user identity.
- 5) Australia: By adopting the same protocol provided by the Singapore government, the Australian government launches COVIDSafe [15] to protect its people from the outbreak of coronavirus.
- 6) Italy: Immuni [16] is a contact tracing application for people in Italy. Similarly, Immuni is a privacy-preserving application based on BLE technology.
- 7) Canada: An exposure notification application, called COVID Alert [17], is launched to keep track of users' proximity to each other while pushing a notification to alert the user of possible coronavirus exposure.
- 8) United Kingdom: After struggling through the initial trial with Google and Apple's exposure notification framework, a more stable NHS COVID-19 [18] is launched to store the data locally instead of a centralized database.

The first two approaches might compromise user's privacy since their applications require monitoring the user's mobility and locations. The third and onward approaches, on the other hand, although it preserves privacy by tracking only the proximity between users without explicit location information, the encryption process can be compromised.

Besides the national-level effort, there are collaborations in industry and academia in delivering an effective contact tracing solution while preserving user privacy [19], [20]. Rather than using location data, many of these initiatives focus on the use of BLE signals for proximity detection. For instance, Pan European Privacy-Preserving Proximity Tracing detects the proximity based on the broadcast BLE packet containing a full anonymous ID [21]. COVID-19 Watch, on the other hand, can automatically alert the user when they are in contact with the infected individual [22]. Similarly, the privacy-preserving automated contact tracing exploits the BLE signals in combination with secure encryption to detect possible contacts while protecting privacy [23]. While Montreal AI lab exploits machine learning method to predict the risk of possible encountered contacts [24], the work assumes ideal RSS measurements from the smartphone. However, this might not be the case for BLE signals transmitted through a noisy channel. To ensure the continuity of business, OkDetect [25] is a contact tracing solution designed specifically to protect workers from the food supply chain from possible infection.

To fasten the application development for contact tracing purposes, Google and Apple launch a privacy-preserving contact tracing framework containing a set of software development kits that enables the developer to directly plug and play the functional library to develop the contact tracing application that suits their need [26]. The framework contains a list of libraries for exposure notification based on BLE and cryptographic techniques. Most of the above-mentioned initiatives assume that the BLE signals

Advertising Packet			
1 byte	4 bytes	8 – 39 bytes	3 bytes
Preamble	Access Address	Protocol Data Unit (PDU)	CRC
2 byte	6 bytes	0 – 31 bytes	
Header	MAC Address	Payload	

Fig. 2. Advertising packet can take up to 47 bytes however, the available bytes for the actual payload is only 31 bytes.

will work for proximity detection without considering the smartphone positioning effects on it. To the best of our knowledge, there is a lack of a comprehensive study on the accuracy of BLE signals for contact tracing proximity sensing. Furthermore, most of the encryption methods are based on information provided by the user, which might be subject to possible information leaks if the encryption method is compromised. To bridge the gap, this article studies the proximity sensing with the BLE signals broadcast from the smartphones carried by the user while designing a privacy-preserving signature protocol that uses the environmental feature instead of user information for packet broadcasting.

## B. BLE Technology

BLE is readily available in most modern smartphones regardless of the operating system. There are two advertising modes available with BLE: 1) nonconnectable advertising and 2) connectable advertising [27]. The latter advertising mode allows another device to request a secure connection establishment. In this article, we focus on the nonconnectable advertising mode, in which the device cannot accept any incoming connection requests. This feature is useful for our SCT system in ensuring no neighboring devices can access the smartphone to retrieve sensitive information. For contact tracing purposes, we configure the smartphone to periodically broadcast the advertising packet via the nonconnectable advertising mode. Nearby smartphones can measure the received signal strength (RSS) upon receiving the packet.

However, there are two major challenges: 1) the length of the advertising packet is only up to 47 bytes, and 2) the RSS values are subject to severe fluctuation.

1) *Advertising Packet*: In the nonconnectable advertising mode, the smartphone will broadcast the advertising packet periodically according to the advertising interval  $T_a$ , which defines how frequent a packet is broadcast. If  $T_a = 100$  ms, we shall expect to see at least 10 packets per second. The advertising packet can take up to 47 bytes, as shown in Fig. 2. Note that 16 bytes are used for preamble (1 B), access address (4 bytes), header (2 bytes), MAC address (6 bytes), and CRC (3 bytes). Hence, there are only 31 bytes left to put in the information related to the environmental signature. This poses a question on how to construct a unique yet useful signature that can be encapsulated into this 31 bytes payload.

2) *Received Signal Strength*: Following the inverse square law [8], the RSS is inversely proportional to the square of the distance. Let  $P_r$  denotes the signal strength in dBm, then:  $P_r \propto \frac{1}{d^n}$ , where  $d$  is the distance between any two devices and  $n$  is the path loss exponent, and its value is subject to the environmental setting when the measurement is taken. As shown in Fig. 3, different environments have different effects



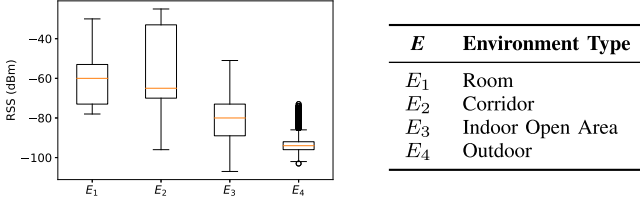


Fig. 3. Variation of RSS values in different environments.

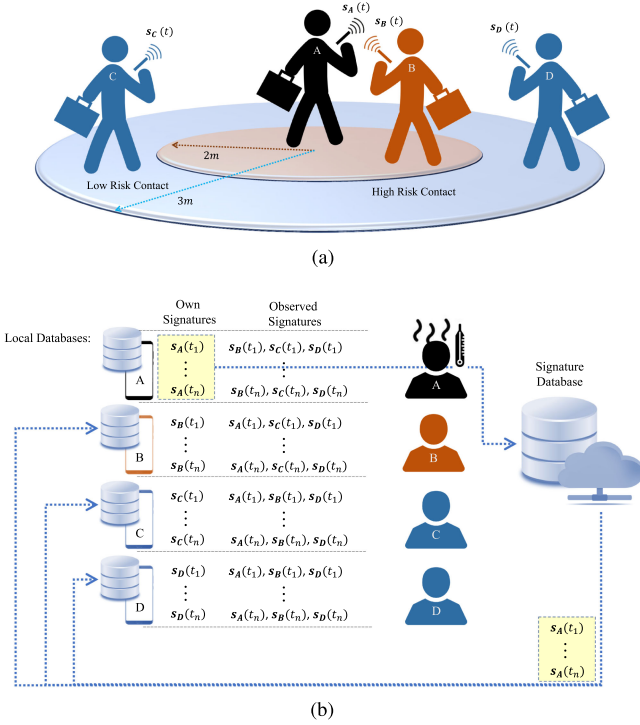


Fig. 4. Our proposed SCT system consists of 2 phases: (a) interaction phase to log the broadcast and observed signature; and (b) tracing phase to perform the signature matching when a user is diagnosed with the infected disease.

on the RSS variation even though the distance between any two devices in these environments are the same. Hence, we need to take the environmental factor into consideration when applying the path loss model to estimate the distance given the RSS.

### III. PROPOSED SMART CONTACT TRACING SYSTEM

There are two major phases with our SCT system, as shown in Fig. 4: the interaction phase and the tracing phase. The interaction phase focuses on the following two main components: 1) privacy-preserving signature protocol, and 2) precise proximity sensing; whereas the tracing phase aims to provide an efficient signature matching.

#### A. Interaction Phase

The interaction phase involves day to day activities in public locations, such as workplaces, public transports, grocery stores, outdoor parks, etc. The contact tracing application starts automatically when it detects the user is in a public location. The application executes the following functions.

- 1) *Signature generation*: The smartphone scans for the ambient environmental features. These features are selectively processed to generate a unique signature that can be fit into the 31 bytes advertising payload.
- 2) *Signature broadcasting*: The smartphone broadcasts the advertising packet containing the unique signature periodically according to  $T_a$ . The packet is broadcasted through the nonconnectable advertising channels.
- 3) *Signatures observation*: The smartphone scans the three advertising channels to listen for the advertising packet broadcast by the neighboring smartphones. The scanning is performed between the broadcasting event.
- 4) *Proximity sensing*: The smartphone measures the RSS values and uses them to estimate how close it is to the neighboring smartphones. It is assumed to be in proximity when the distance is less than 2 m.
- 5) *Physical distancing alert*: The smartphone triggers a real-time alert through push notification when it detects any physical distancing violation.

#### B. Tracing Phase

All the generated signatures and observed signatures will be stored inside the user's local storage, as shown in Fig. 4. Since the signature does not contain any information about the owner, there is no way for the user to trace or identify the original owner of the observed signatures. The signatures are deleted from the local storage permanently once it is expired. We define the expiration period for each signature based on the virus spreading timeframe recommended by the health authorities. For instance, for COVID-19 the expiration period should be 14 days from the day the signature was recorded. After 14 days, the corresponding signature will be deleted.

If a user is diagnosed with an infectious disease, they can upload all the signatures to the signature database. In Fig. 4, user A uploaded all his signatures after he became an infected individual. The database will distribute these signatures to all the users' smartphones. Signature matching is computed on each smartphone and a local alert is triggered when there is a match. The local alert means that the alert is triggered by the smartphone's program itself, not the centralized alert sent by the server. The server is only used to distribute the data. No program/code is executed on the server to find the close contact. In this way, we can protect the user from revealing their identity and ensure that none of the match cases can be eavesdropped on by malicious hackers. Besides signature matching, the application also performs the classification to classify the potential risk of a user according to the time and distance the user spent with the infected individual.

### IV. PRIVACY-PRESERVING SIGNATURE PROTOCOL

While the smartphone can use the nonconnectable advertising mode to refuse any incoming connection request attempt, we also need to ensure that the packet broadcasting will not reveal one's identity. Several methods have been suggested to protect the user's identity by using an encrypted packet [23], [28]. However, these methods require a randomly generated secret key that can be compromised. In this article, we propose to use the ambient environmental signal to construct a signature vector that can fit into the advertising packet.

### A. Ambient Environmental Features for Signature Generation

When the application starts the contact tracing, it first generates a signature that can fit into the 31 bytes advertising packet. The signature is a transformed vector containing the ambient environmental features. When the smartphone scans for the packet broadcast by the nearby smartphones, it may also see other BLE devices. For example, in a grocery store, the smartphone might see the BLE beacon attached to the promotional item, the BLE signal from a smartwatch, smart lighting control, etc. The signal strength of these devices is always changing subject to the location of users' smartphones. Furthermore, some of these devices (i.e., smartwatch, Apple pencil) might not always remain at the same location. Let  $P_r(d)$  be a function that returns the time average RSS value measured from a BLE device located at a certain distance from smartphone and  $B = \{b_1, b_2, \dots, b_m\}$  be a set of BLE devices excluding the smartphone used for contact tracing, then the observed vector can be expressed as follows:

$$\mathbf{o}_u(t) = (P_e(d_{b_1}), P_e(d_{b_2}), \dots, P_e(d_{b_n}))^T \quad (1)$$

where  $\mathbf{o}_u(t) \in \mathbb{R}^m$  is an  $m$ -dimensional vector observed by a smartphone of user  $u$  at time  $t$ . The length of the vector  $m$  is dependent on the size of  $B$ .

Rather than truncating the vector when  $m > 31$  or filling the vector with some arbitrary values when  $m < 31$ , we define a dictionary  $\Psi \in \mathbb{R}^{31 \times m}$  to transform the  $m$ -dimensional vector to a 31-dimensional vector. This dictionary is also known as the secret transformation key to the observed vector. The dictionary is generated randomly according to some transform matrices. For example, one can apply the discrete cosine transform (DCT) or wavelet transform to generate the matrix. In this article, we apply DCT matrix, which can be easily obtained by calling a *dctmtx* function in MATLAB [29]. In general, the dictionary can be defined as follows:

$$\Psi = \begin{pmatrix} \psi_{1,1} & \psi_{1,2} & \dots & \psi_{1,m} \\ \psi_{2,1} & \psi_{2,2} & \dots & \psi_{2,m} \\ \vdots & \vdots & \dots & \vdots \\ \psi_{31,1} & \psi_{31,2} & \dots & \psi_{31,m} \end{pmatrix}. \quad (2)$$

By multiplying the dictionary with the observed vector, we obtain our unique signature vector.

**Definition 1 (Signature vector):** A signature vector  $\mathbf{s}(t) \in \mathbb{R}^{31}$  can be obtained by a linear combination of column vectors from  $\Psi$  multiplying with the time average RSS value from each  $b_j$  in  $B$ , i.e.,

$$\mathbf{s}(t) = \sum_{b_j \in B} \Psi_j P_e(d_{b_j}) \quad (3)$$

where  $\Psi_j = (\psi_{1,j}, \psi_{2,j}, \dots, \psi_{31,j})^T \in \mathbb{R}^{31}$  is the  $j$ th column vector from the dictionary.

The dictionary for each smartphone is different. If any two smartphones observe the same ambient features, i.e., similar time average RSS values from a similar set of BLE devices, the generated signature vector is still different. Note that the above-mentioned case is very rare because even though both persons appear at the same location at the same time, the RSS values from the same set of BLE devices might be different due to the receiver sensitivity of the smartphone, the smartphone's antenna, the position and orientation of the smartphone, etc.

### B. Scanning and Advertising Activities

When the signature is generated, the smartphone encapsulates this signature information into its advertising packet. The smartphone can only see the packet when its scanning activity overlaps with the advertising activity. The timing diagram for the advertising, scanning, and signature generation activities is shown in Fig. 5. Each activity is triggered periodically according to their interval, i.e., generation interval  $T_g$ , advertising interval  $T_a$ , and scanning interval  $T_s$ . Given  $T_s$ , the smartphone will only stay active to listen for the incoming packet for a duration defined by the scanning window  $T_w$ .

As shown in Fig. 5, smartphone A fails to receive  $\mathbf{s}_B(t_1)$  from smartphone B, for the first two times since there is no scanning activity in smartphone A when  $\mathbf{s}_B(t_1)$  arrived. However, smartphone A manages to receive  $\mathbf{s}_B(t_1)$  when smartphone B broadcasts the same packet the third time. According to [30], the likelihood to see the advertising packet broadcast by neighboring smartphones is high as long as  $T_a < T_s$ . Intuitively, when the broadcasting frequency is higher than the scanning frequency given the scanning window is sufficiently long, then it is likely for one of the broadcast packets from A meets the scanning windows of B. We can use a continuous scanning (i.e., set  $T_w = T_s$ ) to increase the packet receiving rate. However, such a scanning approach can greatly affect the energy consumption of a smartphone and eventually create an adverse effect on a user's experience. This article mainly focuses on the privacy and preciseness of contact tracing, balancing the energy consumption, and packet receiving rate is a possible direction for future work.

The smartphone logs the generated signatures and the observed signatures in its own local storage, as shown in Fig. 5. A timestamp  $\tau$  is logged when the smartphone saves a signature into the local database. We could use either a SQL or NoSQL approach to construct this database. The logged timestamp is useful to examine the total time two persons spend in close proximity to each other. Note that for the observed signature, we also log the RSS value upon receiving the packet. This RSS value provides useful information for proximity sensing.

## V. PROXIMITY SENSING AND CLASSIFICATION

Proximity sensing has been employed in many scenarios, including identification of the user's proximity to museum collection [31], and gallery art pieces [32]. There are also works study proximity detection in dense environments [33], or proximity accuracy with filtering technique [34]. Most of these works study the proximity detection between a human and an object attached to a BLE beacon [35], but not the proximity sensing between the devices carried by humans.

### A. Proximity Sensing With Precise Distance Estimation

We use RSS to infer the distance between any two smartphones [36]. Given the RSS value, the distance of a smartphone to another smartphone that broadcasts the packet can be estimated as follows:

$$d = \exp \left( \frac{1}{n} \ln \left( \frac{1}{P_r - c} \right) \right) \quad (4)$$

where  $n$  is the path loss exponent and  $c$  is the constant coefficient, obtained through least square fitting. Given the distance, then we can determine if the user follows the safe physical distance as recommended by the health authorities.

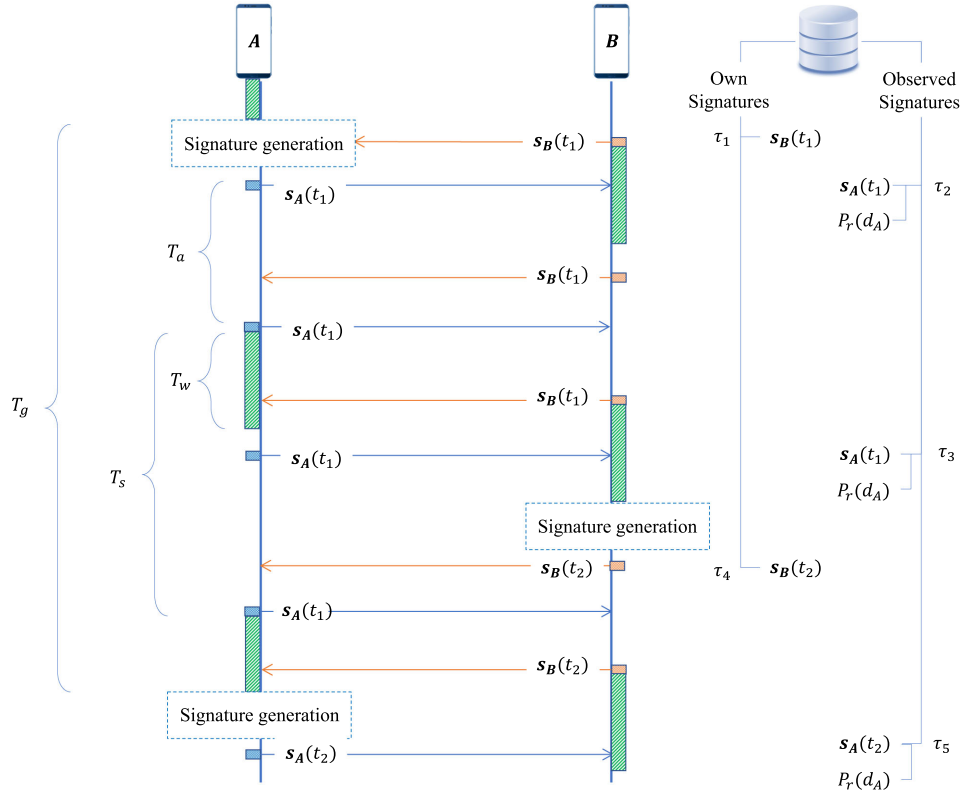


Fig. 5. Timing diagram for the advertising, scanning, and signature generation activities. All the generated and observed signatures will be logged in the local database, together with a timestamp  $\tau$ .

### B. Risk Classification

The problem of classifying the risk of a potential contact can be modeled as a binary hypothesis test. Consider a risk mapping function  $R : (d) \rightarrow \{+1, -1\}$ , where  $+1$  indicates high-risk and  $-1$  low-risk, then there are three hypotheses including the null hypothesis, since we need to consider also the case of false-positives and false-negatives. A false-positive is also known as a false alarm, in which a user actually belongs to the low-risk group, but the system wrongly classifies them to the high-risk group. A false-negative, on the other hand, wrongly classifies the user as the low-risk group, but they are actually in close proximity to the infected individual.

Let  $H_+$  denote the hypothesis that the user belongs to the high-risk ( $+1$ ) group and  $H_-$  the low-risk ( $-1$ ), then the possible hypotheses are

$$\begin{aligned} H_+ : R(d) &= +1 \\ H_- : R(d) &= -1. \end{aligned} \quad (5)$$

Miss detection is undesirable because the user might be at risk but the system considers the user is safe. The false-negative misclassified the high-risk user to low-risk, but in comparison to miss detection, it can at least detect the user. However, this may give a wrong impression to the user that the possibility for them to get infected is low, but actually, the possibility could be high. The false-positive misclassified the low-risk user to high-risk. Even though it is a bit conservative to alarm the user that they are most likely to get infected while they may not, this is relatively safer than miss detection and false-negative.

## VI. IMPLEMENTATION AND EXPERIMENTS

We have developed a smartphone application to demonstrate our proposed SCT. The application has the following two functions: 1) contact tracing based on the privacy-preserving signature protocol and 2) physical distancing alert based on precise proximity sensing. We describe our experimental setup and then discuss our experimental results by comparing the performance with another five classifiers, i.e., DT, LDA, NB, kNN, and SVM.

### A. Application Development

We built an Android application to demonstrate the functionalities of our proposed SCT. First, the application generates a signature packet according to the privacy-preserving signature protocol. Then, it pushes an alert notification when the user violates the physical distancing rule. All the generated signatures, observed signatures, and their corresponding signal strengths are all stored in the smartphone's storage. We installed the application into Android smartphones, including Nokia 8.1 with Android 10, HTC M9 with Android 7.0 Nougat. At least API 23 is required for the BLE to operate. According to Google, at least 95% of smartphones support API 23. If a user has a lower API version, then they can still use the application, however, only to receive BLE signals. Other IoT devices, such as BLE beacons can be used by these users to enable them to transmit signals [37]. When running the application the power requirement is less than 0.24 W, which is negligible. For experimental purposes, we created another version of the application that allows us to log the ground truth during the experiment, in order to be able and compare the estimation with the real data.

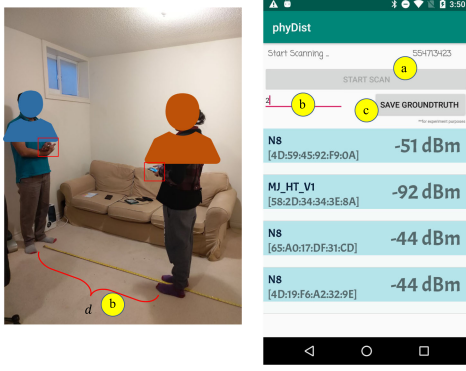


Fig. 6. For the experimental purpose, we created another version of the application by (a) adding a manual button to control the start and the end of the experiment, (b) an input field to key in the truth distance measured through the measuring tape, and (c) a save button that save the measurement data.

TABLE I  
ADVERTISING INTERVAL PROVIDED BY ANDROID BLE API

Mode	$T_a$
ADVERTISE_MODE_LOW_LATENCY	100 ms
ADVERTISE_MODE_BALANCED	250 ms
ADVERTISE_MODE_LOW_POWER	1000 ms

### B. Experimental Setting

For experimental purposes, the application that allows us to log the ground truth was used. We use the ground truth to evaluate the classification performance. The following information is logged: the truth distance, name of smartphone, MAC address of BLE chipset, the packet payload, RSS values, time elapsed, and timestamp. The time elapsed indicates the time difference between the previous broadcast packet and the current broadcast packet, whereas the timestamp is the exact time when the smartphone received the packet. The true distance is measured with a measuring tape during the experiment, as shown in Fig. 6.

In this experiment, both users are required to hold the smartphone in their hands while doing the measurement. The position of the hand is not fixed, and the user can randomly hold the smartphone according to their own comfort. The Android BLE API only provides three possible advertising interval settings, as shown in Table I. In our experiment, the application is configured to advertise at a “ADVERTISE\_MODE\_LOW\_LATENCY.” The application is initiated to start the scan when the user presses the scan button. The scan will continue until the user press the button again. We repeated the experiment for distance from 0.2 to 2.0 m (with 0.2 m increment each step), and 3 to 5 m (with 1 m increment each step). A total of 13 distance points where the measurement is conducted. For each distance, the application was running for at least 60 s. All the measurement data are saved into a “comma-separated values” (.csv) file format and exported to MATLAB for further experiments.

### C. Data Preprocessing

There are a total of 19903 data points collected. The statistical descriptions of our experimental data are shown in Table II. We can see that the variance is high at some distance points. This is mostly due to the multipath effects, in which the signal

TABLE II  
STATISTICAL DESCRIPTIONS OF EXPERIMENTAL DATA

Distance (m)	Count	mean( $P_r$ )	var( $P_r$ )
0.2	1548	-58.9994	48.8203
0.4	1203	-62.9967	9.8685
0.6	934	-70.3084	10.7666
0.8	1080	-74.3167	16.6930
1.0	1631	-79.3476	14.3153
1.2	1573	-74.7788	12.7322
1.4	3986	-80.6468	41.5620
1.6	1282	-89.6599	11.8577
1.8	1344	-79.4903	4.8413
2.0	1101	-80.1835	15.0263
3.0	886	-82.1704	16.0150
4.0	1220	-88.5475	10.7254
5.0	2115	-90.4591	38.0261

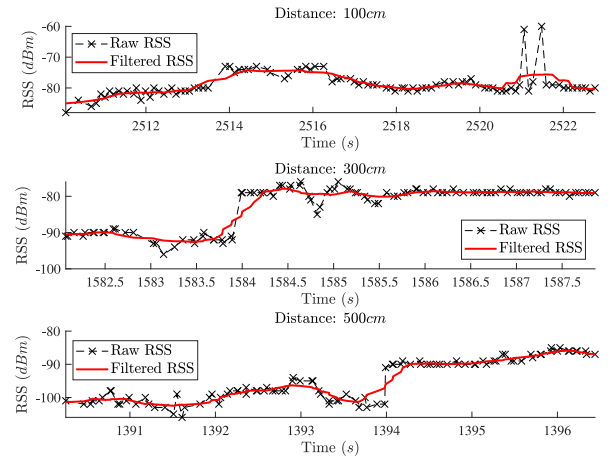


Fig. 7. Raw RSS data versus the filtered RSS data computed through moving average.

takes multiple paths to reach the receiver. At the receiving end, the signals might be added up constructively or destructively, resulting in many different RSS measurements even the distance is the same. Furthermore, the reflection by moving objects in practical environments might cause undesirable outliers.

To mitigate the possible outlier, we apply a moving average filter to the raw data. A comparison between the raw RSS data with the filtered data, for  $d = \{0.2, 1, 2, 3, 5\}$  m is shown in Fig. 7. It is clear that the filtered data provides a much smoother signal. However, there are still variations in signal strength even though the distance is the same. In practice, it is hard to obtain only line-of-sight (LOS) signal in indoor environments due to the reflection and diffraction of the signal. Extracting multipath profile features to achieve better distance estimation could be a possible future direction.

### D. Experimental Results

We applied a nonlinear least-square fitting to determine the value for coefficients  $n$  and  $c$ , in (4). We used the mean RSS described in Table II as the dependent variable and distance as the independent variable. Using this fitted path loss model, we can then estimate the distance by feeding the RSS value measured at each time step into the model. The ultimate goal is to classify if the user belongs to the high-risk or low-risk group assuming the other user is infected with the disease. Based on



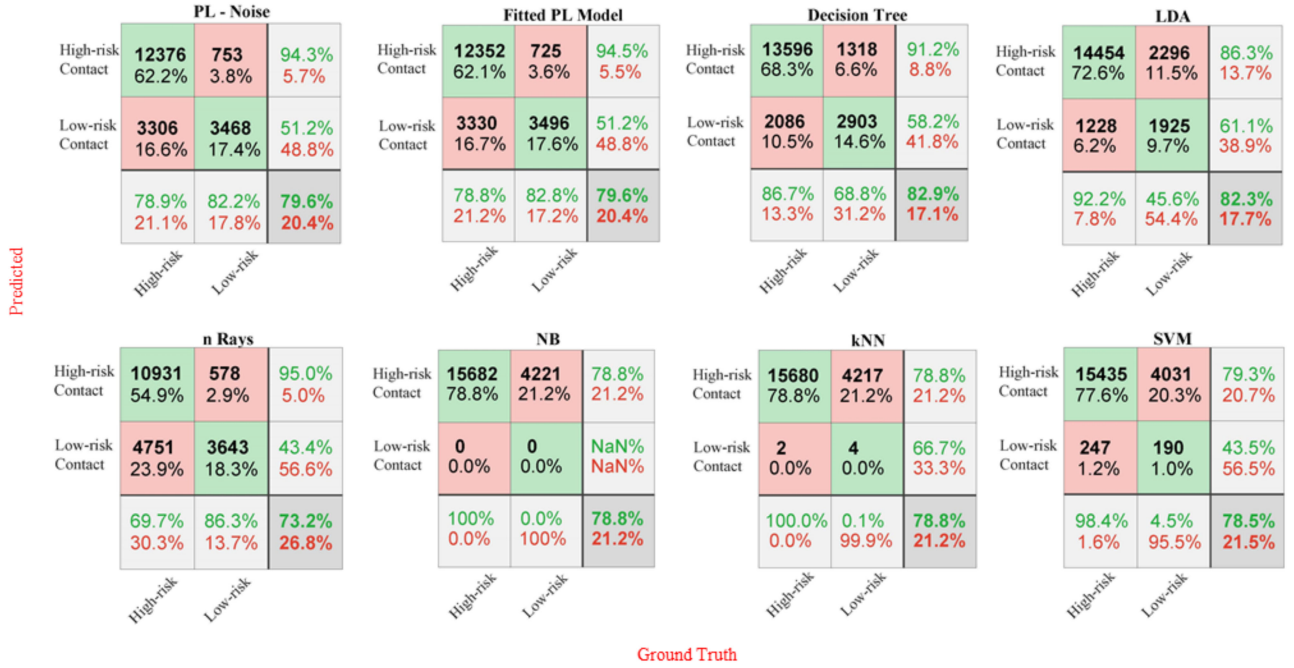


Fig. 8. Classification performance of each method is evaluated via a confusion matrix. The x-axis represents the ground truth, whereas the y-axis is the predicted output. The accuracy can be computed by averaging the diagonal elements of the matrix.

the physical distancing rule recommended by the Canada Health authority [38], we classify the user as high-risk if the estimated distance is  $\leq 2$  m, and low-risk if the estimated distance is  $> 2$  m.

We compared the fitted path loss model with five machine learning-based classifiers: DT, LDA, NB, kNN, and SVM. We also compared with the more realistic PL models by considering the noise and reflection from multipath. In particular, the PL-Noise is defined by feeding artificial Gaussian white noise to the fitted path loss model. The n-Rays model, in which n-1 refers to the number of reflected rays besides the LOS ray, is defined based on the generic two-ray model. Rather than assuming the antenna gains, heights, and reflective constant, we applied a nonlinear curve fitting method to obtain the value for all these parameters. We separated the measurement data into 80% training data and 20% testing data. The input RSS was encoded into an 8-b binary feature.

We used a confusion matrix to evaluate the performance of each classifier, as shown in Fig. 8. It is surprising to see that the n-Rays model has worse performance than the general PL model. This can be explained by the oscillating RSS values captured by the reflected rays is actually confusing the distance estimation. Even though the n-Rays model is more realistic and can capture the RSS fluctuation at different distances, but capturing fluctuation is indeed not useful for distance estimation. On the other hand, we need a model that is less sensitive to these fluctuations.

Overall, the DT method achieves the highest accuracy, i.e., 82.9%. If we examined the matrix, we can see that DT also produces a very high false-negative rate, i.e., it incorrectly classifies the high-risk group as low-risk for 10.5%. This is not a desirable result for contact tracing purposes because those in the high-risk group are those people that are very likely to get infected but DT method classified them as low-risk. On the other hand, both NB and kNN have a higher false-positive rate, i.e., 21.2%. This is relatively acceptable as it is rather to be more conservative than to be ignored. While the overall results are acceptable, there is still

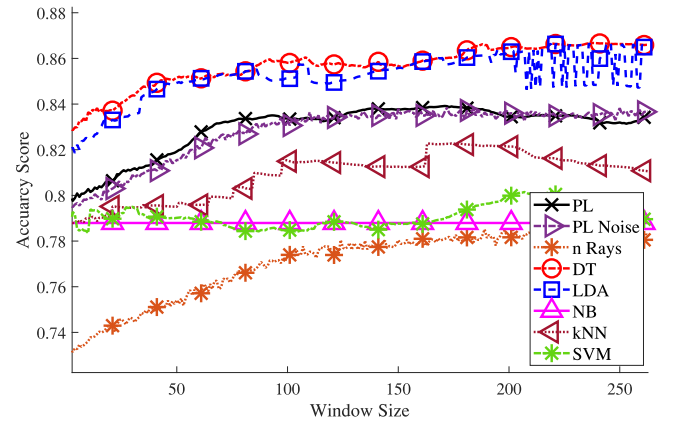


Fig. 9. Accuracy score for different window size.

room for improvement. Instead of using the raw measurement, we compared the results with preprocess data. Furthermore, we would like to understand how the distancing threshold affects accuracy.

1) *Implications of Filtered Window:* As discussed, we can mitigate the possible outliers by preprocessing the data. We further examine the effect of window size on the risk classification performance, and the results are shown in Fig. 9. It is clear that DT achieves the best performance in comparison to the other approaches when the window size increases. However, NB does not show any performance gain with increased window size. LDA, on the other hand, starts to show fluctuation when the window size increases. This could be due to overfitting during the training process. Overall, we see that the performance starts to saturate when the window size is more than 100. We can see that PL is the one that has benefited from the filtered RSS, in which it has a higher performance gain than the rest. In particular, we see that the accuracy obtained via PL is increased from 79.6%



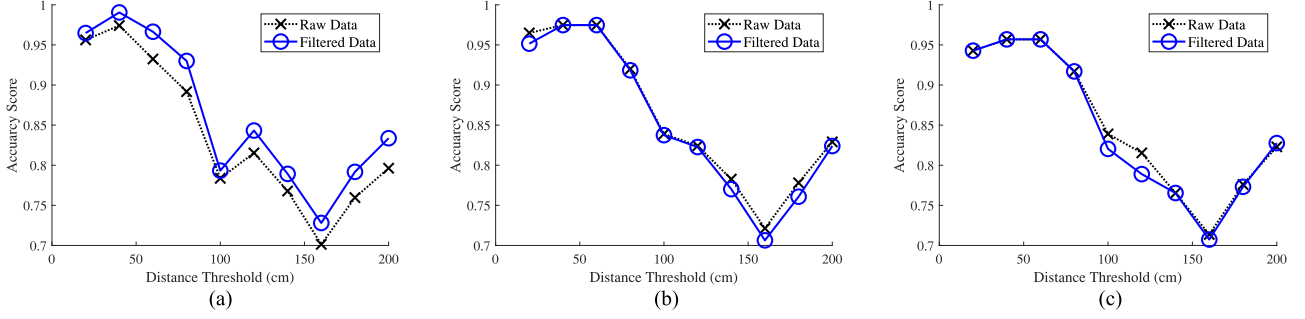


Fig. 10. Effect of distance thresholds (i.e., the distance rule used to classify the high-risk and low-risk contact) on the accuracy. (a) PL. (b) DT. (c) LDA.

to 83.36%, which is corresponding to 4.68% performance gain. However, not all the methods are benefited from the filtered RSS. Some methods show a performance drop when the window size increases, for example, SVM. Even though DT and LDA can achieve better performance than PL model, both of these methods require extensive training and the accuracy may drop when there are not sufficient data for training.

2) *Implications of Physical Distancing Threshold:* As discussed previously, our SCT system classifies the contact as high-risk or low-risk according to the physical distancing rule recommended by the health authority. While we might incorrectly classify the high-risk contact to low-risk due to the fluctuation of the RSS value, we observed that the classification accuracy, in fact, increases when the distancing threshold is smaller. This is preferable as it correctly classifies the user as high-risk when the user is very close to the infected individual. We plotted the accuracy score for different distance thresholds for PL, DT, and LDA. These three methods are selected because they show a good performance in previous experiments. We compared the raw data with the filtered data, by setting the window size to 100. This window size is selected based on the window effect on the accuracy score discussed previously. The accuracy is high when the distancing threshold is less than 1 m, as shown in Fig. 10. This result ensures that the system might produce some false-negatives, but this has mostly happened to the group of users with distances in between 1 to 2 m from the infected individual. Interestingly, the accuracy increases when the distance threshold is increased from 1.5 to 2 m. This indicates that the system is a bit conservative, in which it tends to classify the user as high-risk when the distancing threshold is more than 1.5 m. Overall, it is safer to have a high false-positive than a high false-negative especially if the virus is very contagious.

#### E. Effect of Smartphone's Positions on the Body

We extended the experiment to investigate the proximity sensing performance in connection to the positions of a smartphone on the body. The reason is that the user might not carry the smartphone in their hand most of the time. When they are walking on the street or doing grocery shopping, the user might either carry the phone in their hands, put their phone in their pockets or backpacks. As shown in Fig. 11, we consider additional five cases on top of the “HH” case discussed previously. All the measurement data collected from all these six cases can be found in IEEE Dataport [10] and Github [11]. There are a total of 123718 data points, in which HH contributes 19903 data points. The additional five cases and their total data points are listed as follows.

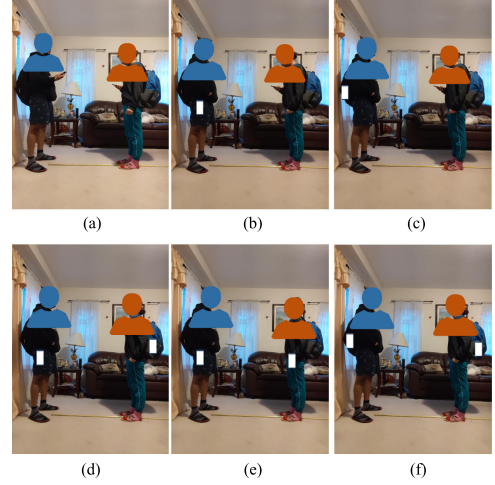


Fig. 11. Six combinations of smartphone positions. (a) HH. (b) HP. (c) HB. (d) PB. (e) PP. (f) BB.

- 1) HP: one user carries the phone in his/her hands and another keeps the phone in their pocket. A total of 16 081 data points were collected for this case.
- 2) HB: one user carries the phone in his/her hand and another user keeps the phone inside their backpack. There are a total of 10 330 data points collected for this case.
- 3) PB: one user keeps the phone in the pocket and another user keeps the phone inside their backpack. There are a total of 19 161 data points collected for this case.
- 4) PP: both users keep their phones in their pockets. There are a total of 24 151 data points collected for this case.
- 5) BB: both users keep their phones in their backpacks. There are a total of 34 092 data points collected for this case.

Previously, we have verified that the accuracy in distance estimation did affect the classification performance. Hence, we examine the distance estimation performance for all the five cases mentioned above. In particular, we used the mean absolute error to compute the error between the estimated distance and the ground truth distance.

The CDF of the distance estimation errors for all the six cases is plotted, as shown in Fig. 12. It is obvious that the filtered data achieves a better performance. The window size 100 is selected based on the justification provided in Fig. 9. We can see that, for 80% of the time, the error is less than 1.27 m for HH case, 0.76 m HP case, 3.93 m HB case, 14.94 m PB case, 1.79 m PP case, and 1.48 m BB case. We observe that PB has the worst performance. This can be explained by the fact that the signals

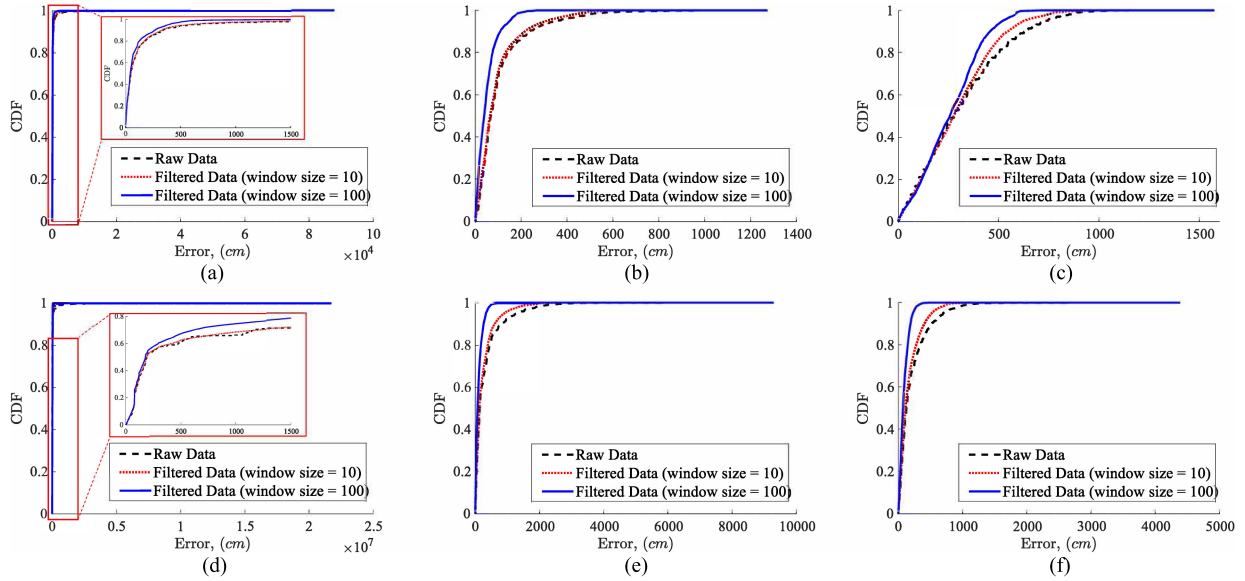


Fig. 12. CDF of the distance estimation errors for all the six cases. (a) HH. (b) HP. (c) HB. (d) PB. (e) pp. (f) BB.

from both smartphones were suffered through different paths of attenuation. Hence, even though we tried to calibrate the model based on the environmental factor, the model is unable to capture such variations. For HH, PP, and BB cases, the signals from both sides might suffer a similar path of attenuation. Take the BB case, for example, the smartphone observed a signal blocked by a human body since the smartphone on the other side was located inside the backpack. Similarly, the smartphone on the other side also observed a similar signal blocked by another human body. Hence, as long as both smartphones measure the signal from a similar position, it is most likely to produce a good estimation.

We also examine the classification performances for all these six cases. Table III shows the classification accuracy obtained using the raw data and the filtered data. From the table, we can see that DT achieves the best performance with more than 81% accuracy for all the cases. Compared to the classification based on distance estimation, the machine learning based methods are more robust to the signal variations caused by body shadowing. Rather than estimating the distance, these classifiers tried to memorize the output given the labeled input during the training process. Hence, the amount of data and the validity of data during the training is very important to train a good classifier. In general, the machine learning approach can be adopted when there are sufficient training data available, otherwise, the PL model is the best choice for instant proximity sensing.

The accuracy of the PL model increases when the time duration users spent in contact increases, as shown in Fig. 13. In general, when the duration increases, the smartphone will be able to observe more signals that help to produce better distance estimation and increase the classification accuracy. However, the accuracy starts to saturate after 10 s for most of the cases except the HB case. This result indicates that the smartphone has already observed sufficient RSS data for making the most accurate distance estimation when the time duration is at least 10 s. The accuracy of the HB case, on the other hand, drops when the time duration increases. Since the signals arrive at two smartphones from different attenuation paths, the more signals the smartphone observes, the more confusing the smartphone in making a correct estimation. Note that both HB and PB cases converge to the same accuracy score when the time duration

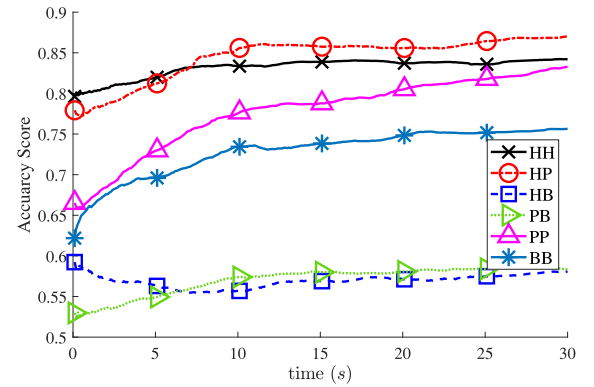


Fig. 13. Accuracy obtained over time.

increases. It is clear that the varying attenuation paths due to the position of smartphones on different body positions can severely affect accuracy. Future work can be conducted to study the effect of attenuation paths from both sides and come up with an adaptive path loss model that can cater to such diversity in attenuation paths.

#### F. Discussion

While the majority of the public might have concerns about the contact tracing solution with the smartphone, contact tracing is deemed essential in preventing further spread of the virus. Hence, continuing research efforts are deemed necessary to address any possible concerns and challenges that may arise. The traditional manual contact tracing is still essential in identifying the individual that is very likely to contract the virus. Currently, smart contact tracing is not meant to be a substitution to the traditional approach; but rather a complement to the traditional contact tracing so that it can release the burden and provide a more efficient way to contact tracing. Imagine if only 10% of the population is willing to use the smart contact tracing approach; it greatly reduces the workload of related healthy personnel in tracking down the contract and shorten the time,

TABLE III  
CLASSIFICATION ACCURACY FOR ALL THE SIX CASES

Case	Method	Raw Data		Filtered Data	
		Mean	95% Confidence Interval	Mean	95% Confidence Interval
HH	DT	82.90%	(82.77%, 85.64%)	<b>85.82%</b>	(83.00%, 85.83%)
	LDA	82.12%	(82.02%, 85.01%)	85.13%	(82.23%, 85.21%)
	NB	78.79%	(78.68%, 78.79%)	78.89%	(78.89%, 78.89%)
	kNN	78.81%	(78.66%, 81.35%)	81.53%	(78.96%, 81.63%)
	SVM	78.46%	(78.45%, 78.51%)	78.55%	(78.64%, 78.66%)
	PL	79.62%	(79.63%, 83.38%)	83.79%	(79.63%, 83.38%)
	PL-Noise	79.48%	(79.46%, 83.20%)	83.22%	(79.50%, 83.24%)
	n Rays	73.32%	(73.30%, 77.30%)	77.33%	(73.35%, 77.35%)
HP	DT	82.80%	(82.69%, 90.67%)	<b>90.75%</b>	(82.90%, 90.84%)
	LDA	82.17%	(82.04%, 90.16%)	90.19%	(82.30%, 90.33%)
	NB	80.27%	(80.14%, 81.71%)	81.81%	(80.40%, 81.98%)
	kNN	81.84%	(81.70%, 86.95%)	87.15%	(81.97%, 87.18%)
	SVM	18.13%	(18.00%, 79.14%)	79.27%	(18.26%, 79.39%)
	PL	77.92%	(77.92%, 85.34%)	85.34%	(77.92%, 85.34%)
	PL-Noise	76.78%	(76.76%, 85.05%)	85.07%	(76.81%, 85.10%)
	n Rays	77.32%	(77.31%, 88.51%)	88.54%	(77.35%, 88.56%)
HB	DT	77.04%	(76.87%, 81.29%)	<b>81.44%</b>	(77.20%, 81.60%)
	LDA	76.99%	(76.82%, 78.68%)	78.85%	(77.16%, 79.02%)
	NB	76.70%	(76.55%, 76.55%)	76.70%	(76.86%, 76.86%)
	kNN	76.92%	(76.75%, 77.82%)	77.99%	(77.10%, 78.16%)
	SVM	23.22%	(23.06%, 23.06%)	23.22%	(23.38%, 23.38%)
	PL	23.31%	(23.31%, 23.33%)	23.33%	(23.31%, 23.33%)
	PL-Noise	23.34%	(23.33%, 23.40%)	23.30%	(23.30%, 23.34%)
	n Rays	57.17%	(57.07%, 57.98%)	57.01%	(57.00%, 57.27%)
PB	DT	87.01%	(87.02%, 87.45%)	<b>87.51%</b>	(87.16%, 87.58%)
	LDA	87.01%	(87.02%, 87.02%)	87.01%	(87.18%, 87.18%)
	NB	87.12%	(87.05%, 87.05%)	87.13%	(87.21%, 87.21%)
	kNN	87.03%	(86.96%, 86.96%)	87.03%	(87.10%, 87.10%)
	SVM	86.62%	(86.54%, 83.45%)	83.53%	(86.69%, 83.61%)
	PL	56.38%	(56.38%, 60.37%)	60.37%	(56.38%, 60.37%)
	PL-Noise	57.93%	(57.91%, 60.68%)	60.71%	(50.95%, 60.73%)
	n Rays	68.19%	(68.15%, 68.24%)	68.28%	(68.22%, 68.32%)
PP	DT	73.35%	(73.12%, 87.18%)	<b>87.26%</b>	(73.47%, 87.34%)
	LDA	72.94%	(72.82%, 87.13%)	87.23%	(73.47%, 87.34%)
	NB	71.34%	(71.22%, 73.77%)	73.90%	(73.07%, 87.33%)
	kNN	71.41%	(71.29%, 73.79%)	73.91%	(71.47%, 74.02%)
	SVM	28.60%	(28.48%, 29.54%)	28.60%	(28.72%, 29.54%)
	PL	66.52%	(66.52%, 77.34%)	77.34%	(66.52%, 77.34%)
	PL-Noise	65.21%	(65.18%, 77.15%)	77.18%	(65.23%, 77.21%)
	n Rays	59.04%	(59.00%, 73.59%)	73.64%	(59.08%, 73.67%)
BB	DT	77.36%	(77.26%, 90.78%)	<b>90.85%</b>	(77.45%, 90.91%)
	LDA	76.90%	(76.81%, 85.21%)	85.28%	(76.99%, 85.36%)
	NB	76.93%	(76.85%, 85.18%)	85.25%	(77.02%, 85.32%)
	kNN	77.16%	(77.07%, 78.00%)	78.09%	(77.26%, 78.19%)
	SVM	22.94%	(22.85%, 69.34%)	69.44%	(23.02%, 69.54%)
	PL	62.18%	(62.18%, 73.28%)	73.28%	(62.18%, 73.28%)
	PL-Noise	62.96%	(62.94%, 72.67%)	72.70%	(62.98%, 72.72%)
	n Rays	60.99%	(60.96%, 77.99%)	78.02%	(61.02%, 78.05%)

which is critical to prevent the spreading of the virus and save lives.

So far, all our experiment setups were carefully designed to ensure reproducibility provided there are sufficient data to train the machine learning models. However, the possible threats to the validity are as follows.

- 1) *Construct validity*: one threat to the construct validity is RSS measurements in different environments. Different environmental settings might produce a different set of RSS measurements. The only way to eliminate the threat is to calibrate and retrain the existing model with the new set of data that reflects the change of environments.

- 2) *Internal validity*: one threat to the internal validity is the choice of devices that use to measure the RSS value. This problem arises due to the different BLE chipset and antenna design of the smartphone. Further work can be conducted to study the effect of smartphones on the RSS measurements so that the model can be calibrated based on the measurement device.
- 3) *External validity*: one threat to the external validity is the generalization of the trained model. This is the common pitfall with the machine learning approach, in which the trained model might fail to generalize to other scenarios. We can eliminate this threat by considering all possible



scenarios when collecting the data, as well as applying cross-validation during model training.

- 4) *Conclusive validity*: one threat to the conclusive validity is that the statistical distribution of RSS might not always follow a certain distribution pattern. Rather than assuming a certain distribution, it is more practical to validate the empirical distribution from the collected data.

In summary, the threats to the validity are all related to the uncertainty of RSS values owing to the environmental dynamics, measurement devices, and the assumed statistical distribution. Since our classification models are data driven, one approach to mitigate all these threats is by repeating the data collection and model training process from time to time.

## VII. CONCLUSION

Our SCT system offers tangible results of using RSS values for proximity sensing between two human beings. From the experimental results, we verified that a BLE-based system for contact tracing is a prominent solution for epidemic control and prevention. We have also shared the dataset in an open-source repository to encourage further research.

Currently, there are no works discuss contact tracing in private and public locations. While most users might willingly participate in contact tracing in public locations with a hope to flatten the disease spreading curve, they might feel a bit uncomfortable letting the contact tracing application running when they are having their private time in the private location (e.g., home, sleeping room, car, etc.). Future work can consider the embedded sensors on the smartphone to check if the user is in a private or a public location. Then, we can use this information to turn ON and turn OFF the contact tracing application accordingly.

## REFERENCES

- [1] H. Chen, B. Yang, H. Pei, and J. Liu, "Next generation technology for epidemic prevention and control: Data-driven contact tracking," *IEEE Access*, vol. 7, pp. 2633–2642, 2019.
- [2] K. T. Eames and M. J. Keeling, "Contact tracing and disease control," *Proc. Roy. Soc. London, Series B, Biol. Sci.*, vol. 270, no. 1533, pp. 2565–2571, 2003.
- [3] MIT Technology Review, Five things we need to do to make contact tracing really work, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.technologyreview.com/2020/04/28/1000714/five-things-to-make-contact-tracing-work-covid-pandemic-apple-google/>
- [4] L. Ferretti *et al.*, "Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing," *Science*, vol. 368, no. 6491, pp. 1–9, 2020.
- [5] M. Radhakrishnan, A. Misra, R. K. Balan, and Y. Lee, "Smartphones and BLE services: Empirical insights," in *Proc. IEEE 12th Int. Conf. Mobile Ad Hoc Sensor Syst.*, Oct. 2015, pp. 226–234.
- [6] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, "BLE beacons for Internet of Things applications: Survey, challenges, and opportunities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 811–828, Apr. 2018.
- [7] C. H. Lam, P. C. Ng, and J. She, "Improved distance estimation with BLE beacon using Kalman filter and SVM," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–6.
- [8] R. Sari and H. Zayyani, "RSS localization using unknown statistical path loss exponent model," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1830–1833, Sep. 2018.
- [9] M. Ayadi and A. B. Zineb, "Body shadowing and furniture effects for accuracy improvement of indoor wave propagation models," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 5999–6006, Nov. 2014.
- [10] Rssdata\_humanhuman, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://ieee-dataport.org/documents/rssdatahumanhuman>
- [11] rss\_humanhuman, 2020, Accessed: Feb. 12, 2021. [Online]. Available: [https://github.com/pc-ng/rss\\_HumanHuman](https://github.com/pc-ng/rss_HumanHuman)
- [12] China Launches Coronavirus 'close Contact Detector' App, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.bbc.com/news/technology-51439401>
- [13] Coronavirus mobile apps are surging in popularity in South Korea, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.cnn.com/2020/02/28/tech/korea-coronavirus-tracking-apps/index.html>
- [14] Tracetogether, safer together, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.tracetogether.gov.sg/>
- [15] Covidsafe, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.health.gov.au/resources/apps-and-tools/covidsafe-app>
- [16] Immuni, the contact tracing app to travel safely in Italy, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <http://www.italia.it/en/useful-info/immuni-the-contact-tracing-app-to-travel-safely-in-italy.html>
- [17] Covid alert, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.canada.ca/en/public-health/services/diseases/coronavirus-disease-covid-19/covid-alert.html>
- [18] Protect your loved ones, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://covid19.nhs.uk/>
- [19] M. Shukla *et al.*, "Privacy guidelines for contact tracing applications," *CoRR*, vol. abs/2004.13328, 2020. [Online]. Available: <https://arxiv.org/abs/2004.04059>
- [20] J. Bell, D. Butler, C. Hicks, and J. Crowcroft, "Tracesecure: Towards privacy preserving contact tracing," *CoRR*, vol. abs/2004.04059, 2020. [Online]. Available: <https://arxiv.org/abs/2004.04059>
- [21] Pan-European privacy-preserving proximity tracing. 2020. [Online]. Available: <https://www.pepp-pt.org/>
- [22] We put the power to reduce the spread of Covid-19 in the palm of your hand, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.covid-watch.org/>
- [23] Pact: Private automated contact tracing, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://pact.mit.edu/>
- [24] Montreal Ai Lab Develops 'privacy-First' contact tracing app to track Covid-19 cases, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.cbc.ca/radio/spark/montreal-ai-lab-develops-privacy-first-contact-tracing-app-to-track-covid-19-cases-1.5552857>
- [25] Smarter employee Covid19 health management, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.okdetect.com/>
- [26] Privacy-preserving contact tracing, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://covid19.apple.com/contacttracing>
- [27] S. R. Hussain, S. Mehnaz, S. Nirjon, and E. Bertino, "Secure seamless Bluetooth low energy connection migration for unmodified IoT devices," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 927–944, Apr. 2018.
- [28] T. Altuwaiyan, M. Hadian, and X. Liang, "Epic: Efficient privacy-preserving contact tracing for infection detection," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–6.
- [29] Discrete cosine transform with MATLAB, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.mathworks.com/help/images/ref/dctmtx.html>
- [30] W. S. Jeon, M. H. Dwijaksara, and D. G. Jeong, "Performance analysis of neighbor discovery process in bluetooth low-energy networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1865–1871, Feb. 2017.
- [31] P. Spachos and K. N. Plataniotis, "BLE beacons for indoor positioning at an interactive IoT-based smart museum," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3483–3493, Sep. 2020.
- [32] P. C. Ng, J. She, and S. Park, "Notify-and-interact: A beacon-smartphone interaction for user engagement in galleries," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2017, pp. 1069–1074.
- [33] P. C. Ng, J. She, and S. Park, "High resolution beacon-based proximity detection for dense deployment," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1369–1382, Jun. 2018.
- [34] A. Mackey, P. Spachos, L. Song, and K. N. Plataniotis, "Improving ble beacon proximity estimation accuracy through Bayesian filtering," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3160–3169, Apr. 2020.
- [35] P. C. Ng, J. She, and R. Ran, "A compressive sensing approach to detect the proximity between smartphones and BLE beacons," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7162–7174, Aug. 2019.
- [36] S. Liu, Y. Jiang, and A. Striegel, "Face-to-face proximity estimation using bluetooth on smartphones," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 811–823, Apr. 2014.
- [37] P. Spachos and K. Plataniotis, "BLE beacons in the smart city: Applications, challenges, and research opportunities," *IEEE Internet Things Mag.*, vol. 3, no. 1, pp. 14–18, Mar. 2020.
- [38] Physical distancing: How to slow the spread of Covid-19, 2020, Accessed: Feb. 12, 2021. [Online]. Available: <https://www.canada.ca/en/public-health/services/publications/diseases-conditions/social-distancing.html>