A Reliable Smart Interaction with Physical Thing attached with BLE Beacon

Pai Chet Ng, *Student Member, IEEE,* James She, *Member, IEEE,* and Rong Ran, *Member, IEEE*

***Abstract*—Bluetooth Low Energy (BLE) beacon is a key-enabler for smart interaction between the user device and the physical thing, in which the physical thing can actively engage users for interaction via its advertising packet. However, reliability is always an issue for beacon-based interaction since the beacon employs an unreliable broadcasting approach which provides no way to check if the user device has received the correct packet. We define sparse observation to describe the phenomenon where the number of packets received by the user device within an arbitrarily small time duration is less than the number of deployed beacons. This paper studies the sparse observation causing by the following two factors: 1) the unpredictable environmental variations, and 2) the uncontrollable operating conditions of a beacon. An analysis is provided to investigate the interaction reliability in connection with the above two factors. Motivated by the above challenges, a novel solution, which exploits the ambient RF fingerprinting to address the sparse observation issues, is proposed to enhance the interaction reliability. Our proposed solution is validated with extensive experiments consisting of real data collected from both indoor and outdoor environments. Lastly, the feasibility of our proposed solution is demonstrated with a proof-of-concept prototype implemented over multiple physical things.**

***Index Terms*—BLE beacon, Reliability, Smart Interaction**

I. Introduction

WHILE there is a constant growth in the Internet of Things (IoT) related research, the current smart thing still fail to have a good interaction with a human. Over the past few decades, various technologies (e.g., QR code [1], RFID [2], [3], [4], etc) were adopted to enable the physical thing to communicate with user devices (e.g., smartphone, smart wearable, etc). However, most of these technologies require the users to initiate the interaction. Take the example of a sculpture attached with an RFID tag, the users need to be told explicitly that they need to scan the RFID tag with their smartphone prior to further interaction. Chances are users might miss the opportunity to interact with the sculpture of their interest if they were unaware of the possible interaction modalities. A smart interaction, on the other hand, should not just wait for the user to take the initiative. In this paper, we explore the promising feature of Bluetooth Low Energy (BLE) beacon towards the realization of a smart interaction and discuss the reliability issue with beacon-based interaction before presenting our novel solution.

Pai Chet Ng and James She are with the Department of Electronics and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong. E-mail: {pcng, eejames}@ust.hk

Rong Ran is with School of Electrical and Computer Engineering, Ajou University, South Korea. E-mail: sunnyran@ajou.ac.kr

A. Background and Motivation

The emergence of BLE beacon has created a new paradigm of interaction with the physical thing [5] [6]. Specifically, the active advertising feature of BLE beacon is capable of notifying the user devices as users approaching. Such a promising feature has witnessed an exponential adoption of BLE beacon for a wide variety of applications especially in promoting the proximity marketing [7]. By attaching a beacon to a physical thing, the thing can broadcast their presence to the surrounding via the advertising packet. Upon receiving the advertising packet, the user device can access the associated content by matching the identifier encapsulated inside the packet with the list of identifiers in the online lookup table. The user device might unable to interact with the physical thing if it could not observe the packet broadcast by the attached beacon.

The broadcasting approach of BLE beacon is unreliable as there is no way to check if the advertising packet had reached the user device. Many applications simply shorten the advertising interval such that the beacon can broadcast as many packets as possible with a hope that one of the packets can somehow reach the user device within a very short time. However, these applications only consider if their own beacon can successfully deliver the packet and did not take the real-world scenarios, which involve many beacons, into consideration. In this paper, we define the sparse observation as a phenomenon where the number of unique packets observed by the user device within an arbitrarily small time duration is less than the number of beacons in a given space. So far, there is no work study the possible factors causing the sparse observation. This paper provides a reliability analysis by investigating the packet loss owing to (1) the environmental variations and (2) the operating condition of a beacon.

B. Contributions

This paper introduces the notion of thing of interest (ToI) to define the physical thing attached with a beacon. Fig. 1 provides four examples of ToI: (a) the ToI carries some information about the promotional items which might of interest to the shoppers[1]; (b) the ToI tells the visitors some interesting stories regarding the sculpture[2]; (c) the ToI provides

[1]Tesco launches biggest iBeacon trial with Magnum campaign, "https://www.marketingweek.com/2015/06/12/tesco-launches-biggest-ibeacon-trial-with-magnum-campaign/"

[2]TED Baker Brings Mannequins To Life with Beacon Technology, "http://www.ibeacontrends.com/ted-baker-beacon-mannequins/"

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974230, IEEE Internet of Things Journal

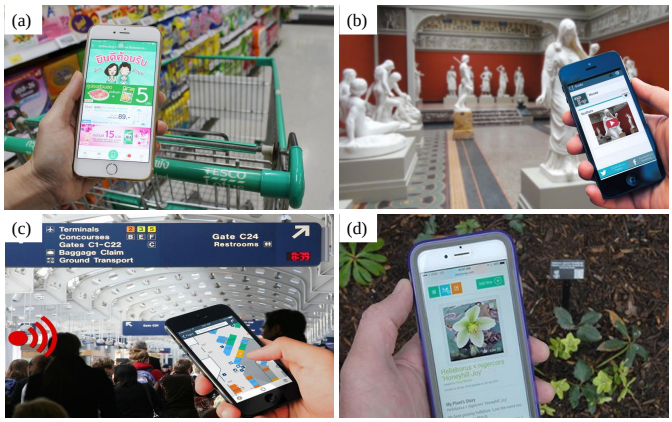

Fig. 1. BLE beacons have been widely adopted by many industries: (a) retail store, (b) art exhibit, (c) airport, and (d) garden.

the route information to direct the travelers inside an airport[3]; and (d) the ToI educates the curious people about the species of the plantation[4]. All the applications above can be realized by simply associating the tiny beacon without modifying the existing infrastructure.

Even though there are a number of works provide the reliability model based on the packet loss rate [8] [9], there is no work study the reliability of beacon-based interaction which has a stringent requirement on accuracy and latency. This paper describes a reliable interaction as the ability of the ToI in delivering their advertising packet within a stringent time requirement such that the user device can trigger a correct interaction at the right time. However, it is almost impossible for all the ToIs to successfully deliver their advertising packet, resulting in a sparse observation at the user device. Hence, it creates an open challenge, i.e., can the user device still trigger a correct interaction under such a sparse observation?

Motivated by these challenges, this paper proposes a novel solution, which exploits the ambient RF fingerprinting as the ToI's identifier and designs a sparsify differential evolution (sDE) to search for the target ToI. The main contributions of this paper are summarized as follows:

- **Reliability analysis:** we provide an analysis to study the effect of (1) the environmental variations and (2) the operating condition of a beacon, on delivering a reliable smart interaction.
- **Novel fingerprint solution:** we exploit ambient RF sources to fingerprinting the ToI. Such a fingerprinting approach allows the user device to correctly identify the ToI even under a sparse observation.
- **Extensive Evaluation:** the proposed solution is evaluated with real data collected from both indoor and outdoor environment, to better reflect the environment variations.
- **Proof-of-concept Prototype:** a prototype is built to demonstrate the reliability of our proposed solution in delivering a smart interaction with multiple ToIs, in which each ToI is attached with a beacon of different operating conditions.

[3]Travellers Want More Self-Service Tech In The Airport, "http://www.webintravel.com/travellers-want-more-self-service-tech-in-the-airport/"

[4]Internet Of Things Usability Testing At Kew Gardens, "http://www.bunnyfoot.com/about/clients/internet-of-things-usability-testing-at-kew-gardens2016"

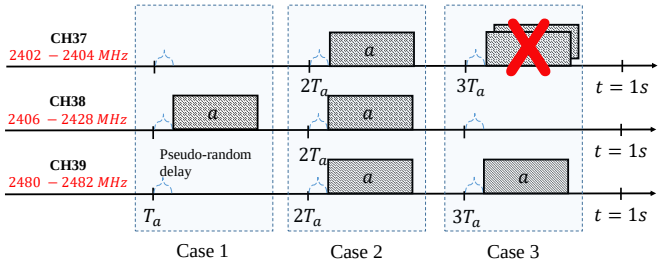

Fig. 2. Given the scarce advertising channels, the advertising packets can access the channels successfully when there is no more than two packets access the same channel the same time.

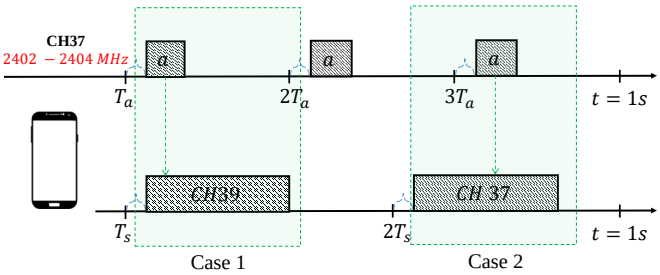

Fig. 3. The user device can only receive the incoming advertising packet when the user device scans on the same channel taken by the corresponding beacon.

The rest of the paper is organized as follows. Section II reviews the related works. Section III presents the interaction model and provides analysis related to interaction reliability Section IV formulates the problem, and then presents our proposed sDE. Section V evaluates the performance of our proposed solution in comparison to the baseline algorithms. Section VI demonstrates the feasibility of our proposed solution with a prototype. Section VII concludes the paper.

II. Related Works

This section succinctly reviews the current technology related BLE beacon, before discussing the works that employ beacons technology for interaction.

A. Bluetooth Low Energy Beacon

BLE beacon is a low power device that broadcast its advertising packet periodically according to the pre-defined advertising interval [6]. The shorter the advertising interval, the more the packet can be broadcast in $1s$. However, simply increasing the advertising interval cannot guarantee if the packet can successfully reach the user device. On the other hand, it might create unnecessary traffic and congested the scarce advertising channels, as discussed by [10]. More specifically, referring to BLE standard, there are only three channels (i.e., Channel 37, 38, and 39) allocated to handle the advertising events. As illustrated in Fig. 2, when there is only one beacon broadcast the packet at a given time, the packet can be transmitted

successfully through any channel successfully. Case 2 and 3 provide further illustration with 3 beacons broadcast at the same time. In case 2, all the packets can be transmitted successfully through the channels if all the 3 beacons broadcast on 3 different channels. In case 3, when there are 2 beacons attempted to broadcast on the same channel, both packets collide. In practice, it is very unlikely for any 2 beacons to broadcast at the same time when there are only a few beacons in the vicinity. However, the chances for any 2 beacons to broadcast at the same time on the same channels increases when the number of deployed beacons getting denser [11]. To mitigate this problem, Bluetooth standard imposes a pseudo-random delay ranging from 0 to $10\mu s$ on each advertising interval. Hence, the previous 2 collided packets might be able to get through the channel in the next advertising interval, assuming both of them still transmitted on the same channel.

There are a lot of works exploit beacon for localization problem [12]. For example, [13] uses radio fingerprinting to locate the position of a smartphone; whereas [14] constructs a robust 3D fingerprint by exploiting the autoencoder. Similarly, none of these works consider the reliability issue involving the interaction between the ToI and the user device. Another critical parameter affecting the packet reception at the user device is the scanning duration by the user device [15] [16]. The user device is able to listen to any incoming packet during the scanning, which implies that the longer the scanning, the more packets the user device could receive. However, the number of packets a user device can receive is very much subject to the scanning behavior. Bluetooth standard imposes a sequential scanning behavior. More precisely, the user device will scan on channel 37 for a certain duration within the pre-defined scanning interval T_s, and then repeat the same scanning on channel 38 at the next scanning interval, i.e. scan on channel 38 at $2T_s$, 39 at $3T_s$, 37 at $4T_s$, and so on.

Assuming there is a beacon broadcast the advertising packet periodically on channel 37, then the user device cannot receive the packet if its scanning duration falls in between the two subsequent advertising intervals. If the scanning duration overlaps with the time when the advertising packet is transmitted, then we have the following two cases, as illustrated in Fig. 3. In case 1, the user device cannot receive the packet because it scans on the different channel then the channel taken by the beacon. In case 2, the user device can successfully receive the packet. Even though there are a number of works investigate the relationship between the scanning duration and the latency in discovering the neighboring beacons [17] [18], none of these works disclose the latency issue related to interaction which is far more complicated than merely neighbor discovery. Furthermore, there is no work addresses the sparse observation issue owing to the short scanning duration. Besides considering the various operating conditions of a beacon, this paper also considers the worst-case scenario when the beacon stops working.

B. Beacon-based Interaction

The advertising feature of BLE beacon is a key-enabler to many novel interaction applications, especially in enabling the ToI to actively engage users for interaction. In general, many beacon-based interaction applications employ the proximity approach to trigger the interaction [19]. For example, [20] provides a beacon-based framework to invite the users to interact with the art pieces through direct notification; whereas [21] exploits the BLE technology for human-robot interaction. The proximity approaches employed by the above applications simply use the raw RSS values to trigger the interaction. Specifically, they use a peak detection (PD) algorithm to identify the interaction subject (i.e., the art pieces for the application presented by [20] and the robot for [21]). While PD is advantageous for its simple computation, it fails to correctly identify the target ToI and eventually lead to a wrong interaction when the user device encounters a sparse observation.

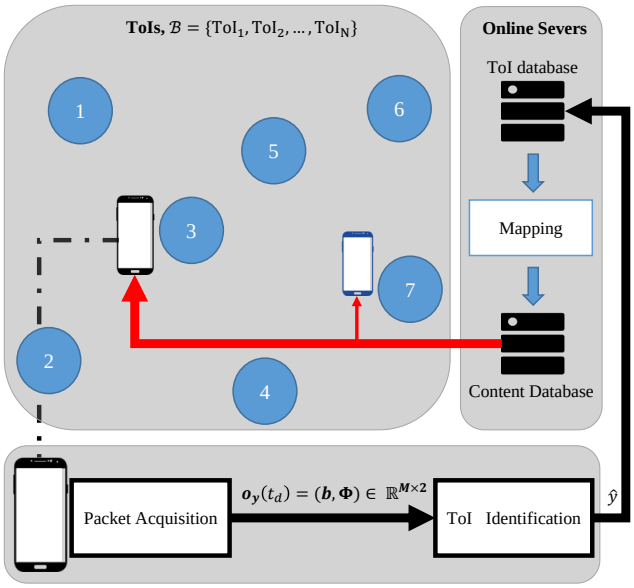

Fig. 4. The interaction model consists of multiple ToIs and user devices. User devices can access the content upon identifying the ToI.

While there exist more complicated algorithms to enhance the accuracy in proximity detection, such as kNN by [22] and CoSaMP by [23], they mainly focus on providing an accurate detection without considering the issue with latency. Lately, many works have started to exploit machine learning approaches to tackle the issues related to RSS, such as preprocessing the raw RSS data to learn a better fingerprint representation [24], or using a reinforcement learning approach to localize the user [25]. However, none of these works study the reliability of using RF fingerprint for retrieving the interactive service. Since many real-world applications have massively deployed beacons for interactive applications [26] [5], it is deemed important to study the reliability issue involving the interaction between the ToI and the user device.

III. Interaction Model and Analysis

The beacon-based interaction model describes the input-output relationship between multiple ToIs and user devices, as illustrated in Fig. 4. As discussed, a ToI is capable of broadcasting its advertising packet to the pre-defined advertising interval T_a via its attached beacon. This packet contains the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974230, IEEE Internet of Things Journal

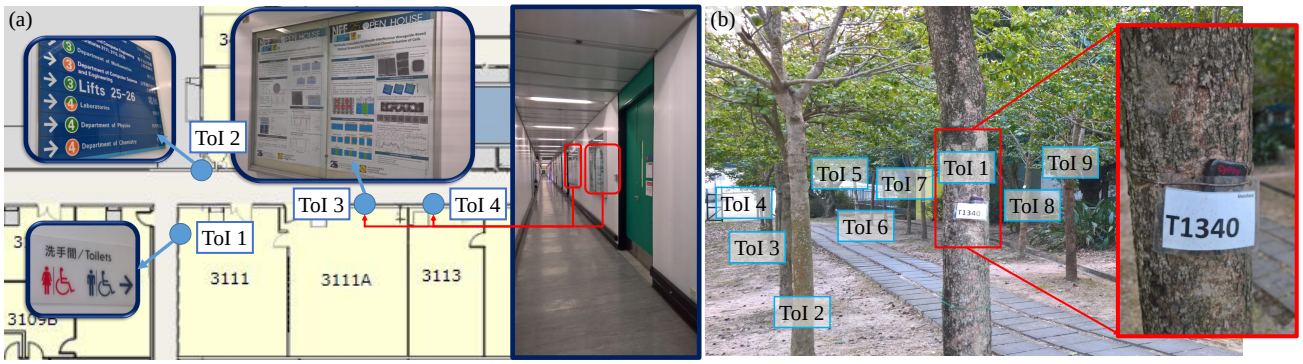

Fig. 5. The data was collected from two different environments: (a) Indoor , and (b) Outdoor environments.

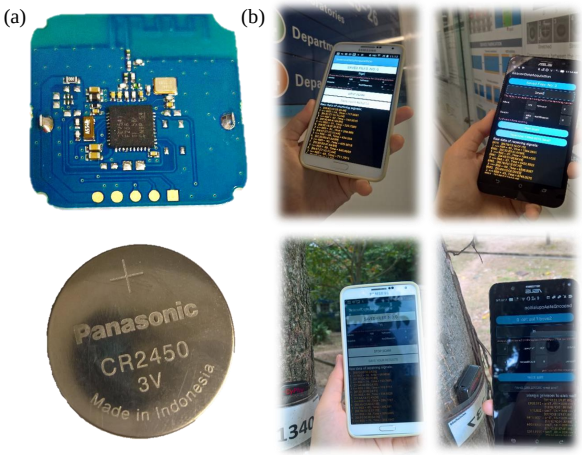

Fig. 6. (a) the beacon is built with the CC2541 chipset from Texas Instrument; (b) Samsung and Asus smartphone are installed with our customized App for data collection.

identifier of the ToI which can be used to access the content stored in the server. Let set $\mathcal{B}$ be a list of ToIs, we have

$$\mathcal{B} = \{\text{ToI}_1, \text{ToI}_2, ..., \text{ToI}_i, ..., \text{ToI}_N\} \tag{1}$$

where $N = |\mathcal{B}|$ denotes the total number of ToIs.

User device performs packet acquisition via a background scanning operation. The list of packets received during a particular scanning interval T_s is stored in a temporary buffer. We define an observation vector $\mathbf{o}_y(t_d) = (\mathbf{id}, \mathbf{\Phi}) \in \mathbb{R}^{M\times 2}$ to hold the unique list of packets observed during an arbitrarily small time duration $t_d \leq T_s$ given the user device is in proximity with ToI_y. By unique list, we mean that $\mathbf{o}_y(t_d)$ consists of only one distinct packet from each ToI. The column vector $\mathbf{id} \in \mathbb{R}^M$ contains a list of ToI's identifiers; whereas the column vector $\mathbf{\Phi} \in \mathbb{R}^M$ indicates the list of average RSS values. Note that M is usually smaller than or equal to N, in which $M = N$ indicates a complete observation. Let $R_r^{(i)}(t)$ be the instantaneous RSS measured at time t for the ith ToI, the average RSS $\phi_i \in \mathbf{\Phi}$ can be computed as follows:

$$\phi_i = \frac{1}{a}\sum_{t\leq t_d} R_r^{(i)}(t), \quad \forall i \in \mathcal{B} \tag{1}$$

where a is the total number of packets from ToI_i.

A user device can automatically identify the target ToI by inspecting the vector $\mathbf{\Phi}$. Identification is important to an interactive application, that is, a smart interaction should be able to automatically identify the target ToI rather than having the user to indicate the target ToI he/she would like to interact with. Mathematically, the identification process can be described using a black box function $f(\cdot)$,

$$\hat{y} = f(\mathbf{o}_y(t_d)) = f(\mathbf{\Phi}_y) \tag{2}$$

where $\hat{y}$ is the index pointing to the target ToI. Note that $\mathbf{o}_y(t_d)$ is reduced to $\mathbf{\Phi}_y$ for brevity. There are various methods to deal with Eq. (2). The simple and intuitive method is by searching the element which returns the strongest average RSS. This method has been adopted by many commercial applications despite its unreliability in connection with the sparse observation, especially when M is far less than N.

A. Data Collection

This section presents the data collection methodology in different environments before examining the effects of sparse observation on the interaction reliability in Section III-B. Fig. 5 (a) illustrates the indoor environment consisting of 4 ToIs: two bulletin boards, one directory signboard, and one toilet signboard. Fig. 5 (b) illustrates the outdoor environment consisting of 9 ToIs: 9 trees lined up along a pathway. These ToIs were chosen for their potential in triggering a serendipitous interaction.

The attached beacon is a tiny device built with the CC2541 chipset from Texas Instruments [27], as shown in 6(a). Initially, we configured the beacon to operate with $T_a = 100ms$. Fig. 6(b) shows the 2 different Smartphones (i.e., *Asus Zenfone 2 Deluxe ZE551ML* and *Samsung Galaxy Note 3*) used to collect the data. Both phones were installed with our data collection App, which can automatically log the following information: Mac Address, device name, advertising packet payload, RSS value, and the packet receiving time. The logged data were saved inside the local storage as a .csv file.

Two smartphones were placed at different locations close to the target ToI before starting the App to collect the data. The App is configured to scan for the incoming packet continuously until at least 1000 data were collected. The same collection steps were repeated for all the 13 ToIs, and for $T_a = 100ms$ to $T_a = 1000ms$ with $200ms$ increment each time. For each repeated process, we placed the smartphones randomly selected locations as long as it is still in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974230, IEEE Internet of Things Journal

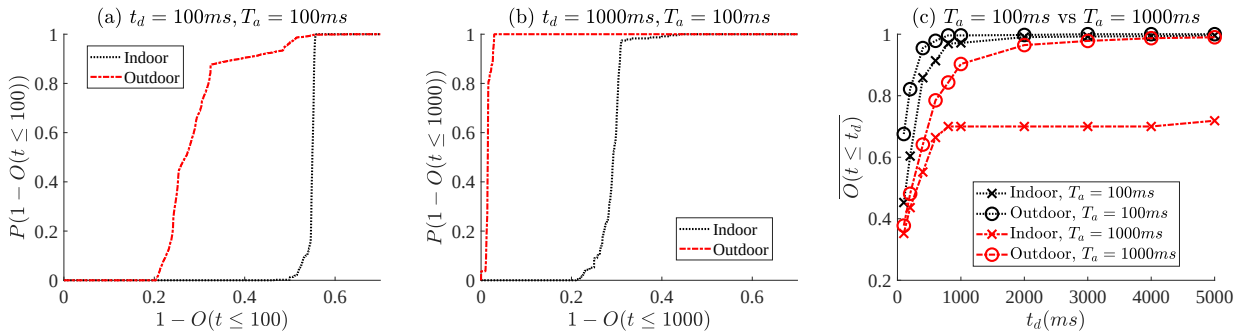

Fig. 7. The distribution of the observation rate for both indoor and outdoor environments when (a) $t_d = 100ms$, and (b) $t_d = 1000ms$ for $T_a = 100ms$. (c) The average observation rates for t_d from $100ms$ to $5s$.

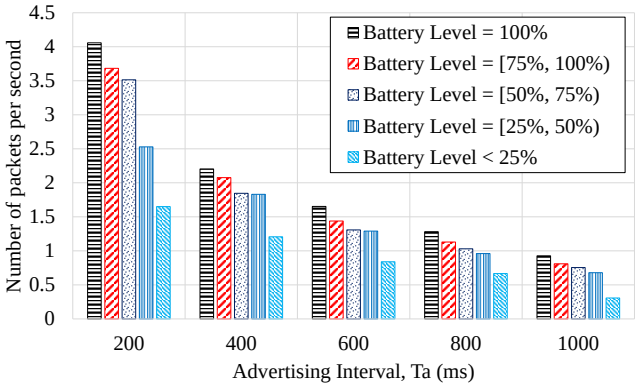

Fig. 8. The number of same packets recorded by the user device in $1s$ for $T_a = 200ms$ to $1000ms$.

proximity with the target ToI. Moreover, the data collection was performed in an uncontrollable environment where there were always people roaming around. About $130,000$ data were collected in total. The data were then exported to the Matlab for further analysis.

B. Reliability Analysis

Most of the related works [8] [9] merely investigate the reliability in terms of packet loss rate and focus on the applications related to wireless communication. However, this paper focuses on the interactive applications which are mostly latency-sensitive. Having said that, an interaction is also considered unreliable if the packet managed to reach to the user device at a much later time. Accordingly, the interaction reliability is defined as the ability to trigger the correct interaction within a stringent time requirement. Sparse observation defines the phenomenon when the size of the current observation vector $\mathbf{o}_y(t_d)$ is far smaller than the total number of ToIs in set $|\mathcal{B}|$, i.e., $N < M$. Note that sparse observation might not necessarily cause an incorrect interaction, but might be resulting in a high possibility of incorrect interaction. Let $O(t \leq t_d)$ be the observation rate. Then we define the accuracy of the identifying, (denoted as P_A), as the probability of successfully triggering a correct interaction upon identifying the target ToI $\hat{y}$ conditioned on the probability distribution of the observation rate $P(O(t \leq t_d))$, i.e.,

$$P_A = P(Y = \hat{y} | O(t \leq t_d)) \tag{3}$$

where Y stands for the identifying result. Given Eq. (3), the reliability can be expressed as follows:

$$R = P(1 - P_A < \varepsilon_{max}) \tag{4}$$

where ε_{max} denotes the maximum tolerable error. Intuitively, Eq. (4) implies the ability of the user device to produce a high probability of accuracy despite the observation rate. The observation rate $O(t \leq t_d)$ can be computed by dividing the height of $\mathbf{o}_y(t)$ (more specifically, the column's size of $\mathbf{id}_y$ or $\mathbf{\Phi}_y$) with the size of set $\mathcal{B}$, i.e.,

$$O(t) = \frac{\|\mathbf{\Phi}_y(t)\|_0}{|\mathcal{B}|} \tag{5}$$

Both $\|\cdot\|$ and $|\cdot|$ return the cardinality of a vector and a set, respectively.

Eq. (4) indicates that the interaction reliability is dependent on the observation by the user device at time t. Obviously, a correct interaction can be triggered with a high probability when the user device has a complete observation, i.e., $M = N$. However, the complete observation rarely happens in practical scenarios owing to (1) the unpredictable environmental variations, and (2) the uncontrollable operating conditions of a beacon.

1) Environmental Variations: Many works have concluded that indoor environments exhibit severe environmental variations compared to outdoor environments owing to the shadowing and multipath effects [28] [29]. Hence, we compare the observation rate between the indoor and outdoor environments using the collected dataset. We varied t_d from $100ms$ to $5s$ and computed $O(t \leq t_d)$ using Eq. (5). Fig. 7(a) and (b) indicate the corresponding probability distribution when $t_d = 100ms$ and $1s$, for $T_a = 100ms$. It is clear that the user device was having a lower observation rate (i.e., higher packet loss) in the indoor environment than the outdoor environment. The packet loss rate in the outdoor environment is less than 0.39 for at least 90% of the time, whereas the packet loss rate in the indoor environment is up to 0.55, as shown in Fig. 7(a). The packet loss rate reduces to 0.03 and 0.31 for both outdoor and indoor environments when t_d is increased to $1s$, as shown in Fig. 7(b). Fig. 7(c) summarizes and compares the average observation rate $\overline{O(t \leq t_d)}$ for $T_a = 100ms$ and $T_a = 1000ms$.

In general, $\overline{O(t \leq t_d)}$ increases when t_d increases. Moreover, the user device can obtain a better observation when

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974230, IEEE Internet of Things Journal

the beacon is configured to broadcast its packet every $100ms$ as compared to $1s$. Based on the above analysis, it is obvious that the user device is able to trigger a correct interaction with high probability in outdoor environments, which is subjected to less environmental variations. Note that even though there is a high guarantee on observation rate when $t_d = 5s$, the long t_d, on the other hand, compromises the latency requirement of an interaction.

2) Operating Conditions of BLE Beacons: Operating conditions of a beacon are generally related to its energy level of a beacon. We used the battery level to monitor the energy level of a beacon. For all the beacons used in the experiment, their battery levels were logged and group according to their percentage. For every $1s$, we counted the number of packets received from the same ToI. That is, when the beacon is configured to broadcast its packet with $T_a = 200ms$, the user device should be expecting a total of $5\,packets/s$ from that beacon. More precisely, the frequency of the received packets per ToI with respect to its advertising interval T_a, i.e.,

$$f_r^{(i)} = \frac{1}{T_a^{(i)}}, \quad \forall i \in \mathcal{B} \tag{6}$$

Note that f_r is different from $O(t \leq t_d)$, yet f_r has certain level of influence on the $O(t \leq t_d)$. Specifically, when f_r from a beacon is low, say less than $0.2packets/second$, then the chances for the user device to observe the packet from that beacon would be very low and eventually decreasing the overall $O(t \leq t_d)$.

As illustrated in Fig. 8, when the battery level is full, the number of same packets recorded by the user device per second is closed to the expected f_r. Meanwhile, f_r drops when the battery level of the beacon decreases. Specifically, a severe drop is observed when the battery level is less than 25%. In practical cases, it is impossible to have all the beacons operate with a nearly full battery level. On the other hand, each beacon will suffer a certain level of battery drops from time to time, and it is impractical to keep monitoring the battery level to ensure the interaction reliability. However, being able to provide a reliable interaction is of critical important considering the exponential adoption of beacons for all kinds of emerging interactive applications.

IV. ToI's Identifier with Proposed RF Fingerprint

This paper proposes an ambient RF fingerprinting to label the ToI, rather than merely using the hardcoded identifier encapsulated inside the advertising packet. In contrast to the conventional RF fingerprinting which only consider the homogeneous RF signal from a single source [30] [31], the ToI is fingerprinted with the RSS harnessed from the attached beacon as well as the beacons in its vicinity. Moreover, our ToI fingerprint is dynamic and not confined to a specific physical space as those location fingerprints presented by [13].

A. Fingerprint Construction

Let $\mathbf{\Omega} \in \mathbb{R}^{N \times N}$ be the fingerprint matrix, in which each column represent a ToI's fingerprint, i.e,

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{\Phi}_1 & \mathbf{\Phi}_2 & \cdots & \mathbf{\Phi}_i & \cdots & \mathbf{\Phi}_N \end{pmatrix} \tag{7}$$

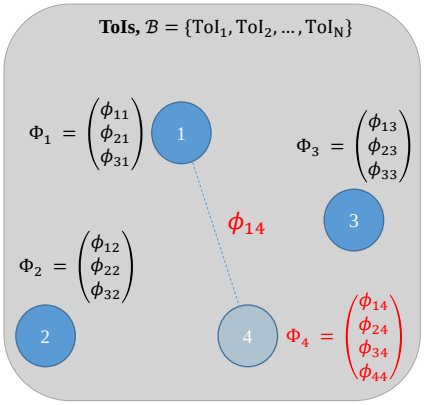

Fig. 9. When a new ToI, say ToI_4 is added, we only need to perform the fingerprint registration on ToI_4. Then we can use the ϕ_{14} measured at ToI_4 to update ϕ_{41} for ToI_1.

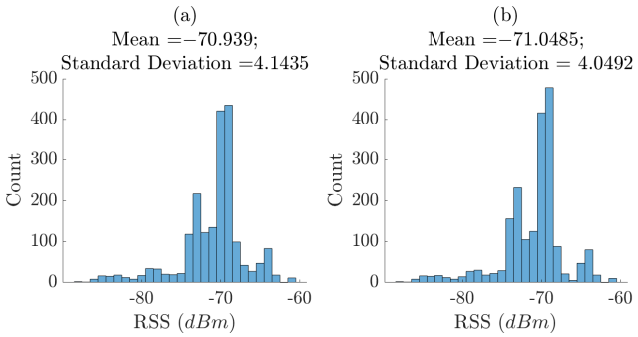

Fig. 10. (a) The distributions of ϕ_{14} when registering the fingerprint at newly added ToI_4. (b) The distributions of ϕ_{41} when repeating the fingerprint registration at ToI_1. It is clear that the KL divergence between 2 distribution is very small.

where $\mathbf{\Phi}_i \in \mathbb{R}^N$ is the fingerprint vector consisting of the RSS values measured from all N ToIs by ith ToI, i.e., $\mathbf{\Phi}_i = \begin{pmatrix} \phi_{1i} & \phi_{2i} & \ldots & \phi_{ji} & \ldots & \phi_{Ni} \end{pmatrix}^T$. Each element ϕ_{ji} can be computed using Eq. (1), and ϕ_{ji} is maximum when $j = i$. For the purpose of fingerprint registration, it is always desirable to repeat the scanning several times at several different locations close to the ToI to obtain a better fingerprint. As a rule of thumb, the time taken to register a fingerprint should be at least $5s$, but less than $1min$.

Now, suppose that a new ToI is added, then $\mathcal{B} = \{\text{ToI}_1, \text{ToI}_2, ..., \text{ToI}_i, ..., \text{ToI}_N, \text{ToI}_{N+1}\}$, which indicates that the size of $\mathbf{\Omega}$ should be $N+1 \times N+1$. Definitely, the new ToI would has a fingerprint vector of size $N+1$, but this is not the case for the rest of N fingerprints in the database. Considering the example illustrated in Fig. 9, we note that the newly added ToI_4 has a 4-dimensional fingerprint vector. However, the existing fingerprint vectors registered in the database have 3 dimensions. It is relatively challenging, if possible, to append the newly registered fingerprint vector into the existing $\mathbf{\Omega}$. One possible solution is to repeat the fingerprint registration for all the N ToIs, which can be labor intensive especially when N is large.

Thank for the reciprocity theorem of a wireless communication link that is, if in a wireless communication link, the role of the receive and transmit devices are functionally interchanged, the transfer characteristics of the wireless channel remain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974230, IEEE Internet of Things Journal

unchanged. The theorem is further confirmed by observing the distributions of ϕ_{14} and ϕ_{41}, as illustrated in Fig. 9, which behave very similarly and the similar results can also be observed by other ToIs. Therefore, we argue that the value of ϕ_{14} approximates that of $\phi_{41)}$ from an average sense. By leveraging the symmetrical property of the measured RSSs from the above argument, the existing fingerprint matrix $\mathbf{\Omega} \in \mathbb{R}^{3\times3}$ can be updated to $\mathbf{\Omega} \in \mathbb{R}^{4\times4}$ based on the newly fingerprint vector $\mathbf{\Phi}_4$ only without repeating the whole fingerprint registration process for all the existing ToIs. More generally, the fingerprint matrix in Eq. (7) can be updated to the following:

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{\Phi}_1 & \cdots & \mathbf{\Phi}_i & \cdots & \mathbf{\Phi}_N & |\mathbf{\Phi}_{N+1} \end{pmatrix} = \begin{pmatrix} [ccccc|c]\phi_{11} & \cdots & \phi_{1i} & \cdots & \phi_{1N} & \phi_{1(N+1)} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ \phi_{i1} & \cdots & \phi_{ii} & \cdots & \phi_{iN} & \phi_{i(N+1)} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ \phi_{N1} & \cdots & \phi_{Ni} & \cdots & \phi_{NN} & \phi_{N(N+1)} \\ \phi_{(N+1)1} & \cdots & \phi_{(N+1)i} & \cdots & \phi_{(N+1)N} & \phi_{(N+1)(N+1)} \end{pmatrix} \tag{8}$$

where $\phi_{1(N+1)} \approx \phi_{(N+1)1}$, $\phi_{i(N+1)} \approx \phi_{(N+1)i}$, and so on. Moreover, we implemented a customize beacon firmware which allows the beacon to listen to the incoming packets and update its fingerprint vector from time to time without human intervention. The fingerprint registration and updating process, in general, would not affect the on-going interaction events taken place in the adjacent ToIs.

B. k-Sparse Problem Formulation

The relationship between $\mathbf{\Phi}_y$ and fingerprint matrix $\mathbf{\Omega}$ can be described as follows:

$$\mathbf{\Phi}_y = \mathbf{\Omega}\mathbf{b} + \mathbf{\Delta} \tag{9}$$

where $\mathbf{b} \in \mathbb{R}^N$ is the indication vector and $\mathbf{\Delta} \in \mathbb{R}^N$ is the measurement noise. The indication vector $\mathbf{b}$ should be a 1-sparse vector with all elements equal to 0 except the element indicating the target ToI. However, it is almost impossible to obtain a clean 1-sparse solution in practical scenarios. For example, when the user device is in proximity with ToI_{700}, the indication vector should contain all elements equal to zero except at the element equal to 700. Ideally, the 700th element should equal to 1; however, we might obtain an indication vector with k non-zero elements due to the presence of noise and interferences. In this case, the 700th element should contain the value higher than the rest of $k-1$ elements. Hence, the problem is reduced to finding a k-sparse vector $\mathbf{b}$ containing at least k non-zero elements, i.e.,

$$\widetilde{\mathbf{b}} = \arg\min \|\mathbf{b}\|_0 \tag{10}$$

These k non-zero elements should sum up to 1; the element with the highest value indicates the target ToI, and the rest of $k-1$ elements indicate the influence from adjacent ToIs. Note that Eq. (10) introduces a NP-hard problem, which can be relaxed to the following objective function:

$$\begin{aligned} \widetilde{\mathbf{b}} &= \arg\min \|\mathbf{b}\|_0 \\ \text{s.t.} \quad & \|\mathbf{\Phi}_y - \mathbf{\Omega}\mathbf{b}\|_2^2 < \epsilon \\ & |\mathbf{b}| \leq 1 \end{aligned} \tag{11}$$

The above objective function can be solved by any optimization algorithms, for example, the differential evolution (DE) [32]. Nonetheless, Eq. (9)-(11) are only valid when the size of $\mathbf{\Phi}_y$ equals to the height of $\mathbf{\Omega}$, i.e., $M = N$. This might not be the case consider the issue with sparse observation discussed in Section III-B. More specifically, the problem is undertermined when $M << N$. To address this problem, sDE extends the existing DE by imposing a sparsify basis to transform the $N \times N$ matrix to $M \times N$ matrix.

C. Sparsify Differential Evolution

As illustrated in Fig. 11, DE uses three processes (i.e., mutation, crossover, and selection) to iteratively search for the optimum solution. In contrast to the conventional matching pursuit algorithms, DE is capable of avoiding being trapped in the local optimum via constant evolution [33] [34]. Such a property is critical to an interactive application in which the user might shift from interacting with current ToI to another ToI. The random shifting from one ToI to another ToI creates multiple optimum points within a short time interval. However, DE could not be applied directly to Eq. (11) when the observation vector $\mathbf{\Phi}_y$ is sparse. For this reason, this paper proposes sDE, a novel extension to the existing DE, by imposing a sparsify basis during the selection process. As illustrated in Fig. 12, sDE relates the sparse observation $\mathbf{\Phi}_y \in \mathbb{R}^M$ with $\mathbf{\Omega}\mathbf{b} \in \mathbb{R}^N$ via a sparsify basis $\Psi \in \mathbb{R}^{M\times N}$. Each element ψ_{ij} indicates if the identifier of jth observation (i.e., $\mathbf{id}_y^{(i)}$) matches with the identifier pointed by $\text{ToI}_j \in \mathcal{B}$.

$$\psi_{ij} = K(\mathbf{id}_y^{(i)}, \text{ToI}_j) \tag{12}$$

where function $K(\cdot)$ returns 1 if the identifier matches. The common function used by $K(\cdot)$ is by simply taking the modulus between both variables. More sophisticated cryptographic techniques can be adopted to obtain a more secure $K(\cdot)$.

Given the sparsify basis Ψ, the selection process of sDE is defined as follows:

$$\mathbf{b}_\rho^{(G+1)} = \underset{u=\{\alpha,\vartheta\}}{\arg\min}(\|\mathbf{\Phi}_y - \Psi\mathbf{\Omega}\mathbf{b}_u\|_2^2) \tag{13}$$

where $\mathbf{b}_\alpha$ and $\mathbf{b}_\vartheta$ are the possible solutions obtained through the evolution processes; whereas $\mathbf{b}_\rho^{(G+1)}$ is the best solution which can be proceeded to the next generation G. Initially, three vectors (i.e., $\mathbf{b}_\alpha$, $\mathbf{b}_\beta$ and $\mathbf{b}_\gamma$) are randomly initialized and go through the mutation and crossover processes to produce $\mathbf{b}_\vartheta$. During the mutation process, both vector $\mathbf{b}_\beta$ and $\mathbf{b}_\gamma$ are subtracted and multiplied with a mutation rate $\mathcal{M}$, before adding with $\mathbf{b}_\alpha$ to produce a donor vector $\mathbf{b}_\zeta$, i.e.,

$$\mathbf{b}_\zeta^{(G)} = \mathbf{b}_\alpha^{(G)} + \mathcal{M}(\mathbf{b}_\beta^{(G)} - \mathbf{b}_\gamma^{(G)}) \tag{14}$$

The crossover process exchange the elements between $\mathbf{b}_\alpha$ and $\mathbf{b}_\zeta$ to produce the evolved vector $\mathbf{b}_\vartheta$. The exchange is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974230, IEEE Internet of Things Journal

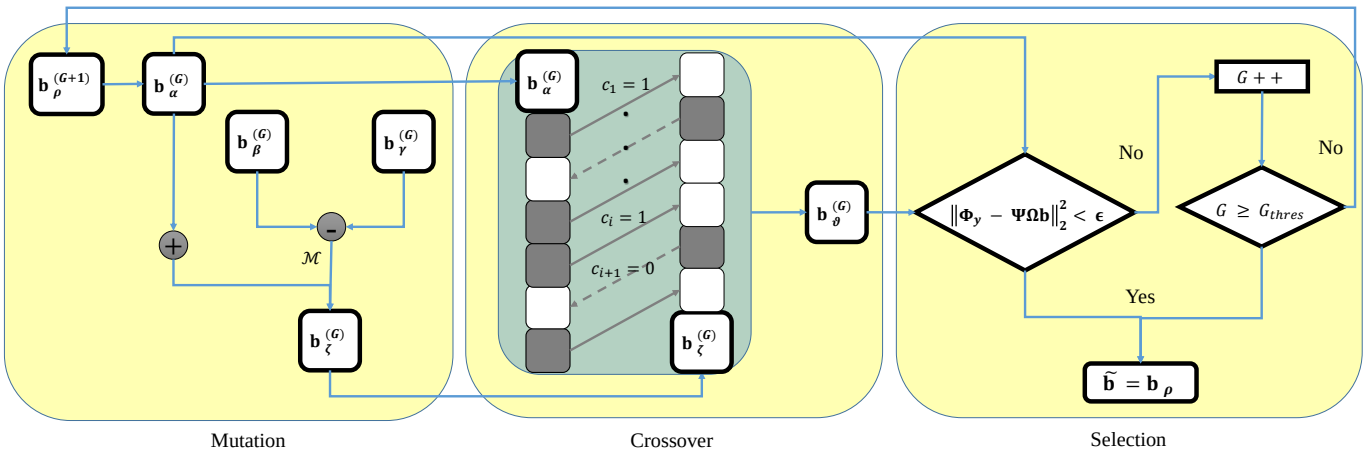

Fig. 11. sDE imposes a sparsify basis during the selection process in determining the current optimum solution from a pool of possible solutions produced through the mutation and crossover processes.

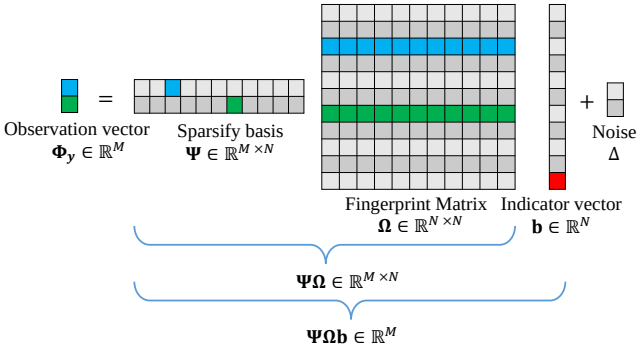

Fig. 12. The sparsify basis transform the $N \times N$ fingerprint matrix to $M \times N$ to deal with the sparse observation.

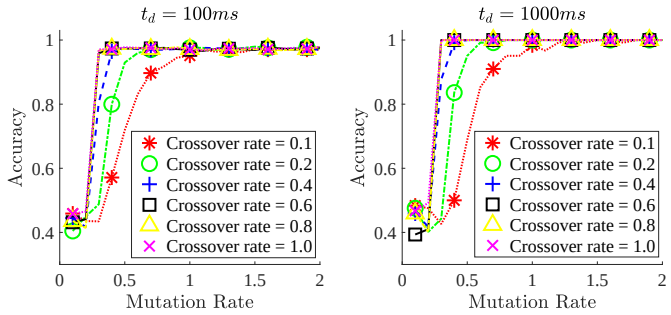

Fig. 13. The accuracy given different combinations of mutation and crossover rate.

governed by a crossover vector $\mathbf{c}$, in which each element in $\mathbf{c}$ is either 1 or 0 according to the $\mathcal{C}$ is the crossover rate, i.e.,

$$c_i = \begin{cases} 1, & R_i(n) \geq \mathcal{C} \\ 0, & R_i(n) < \mathcal{C} \end{cases}, \quad \forall c_i \in \mathbf{c} \tag{15}$$

where $R(\cdot)$ is the random number generator. Intuitively, $\mathbf{b}_\vartheta$ inherits the element from $\mathbf{b}_\zeta$ when $c_i = 1$; otherwise, $\mathbf{b}_\vartheta$ inherits the element from $\mathbf{b}_\alpha$. Mathematically, $\mathbf{b}_\vartheta$ obtained at current generation G can be expressed as follows:

$$\mathbf{b}_\vartheta^{(G)} = \mathbf{c}^T \mathbf{b}_\zeta^{(G)} + \mathbf{c}'^T \mathbf{b}_\alpha^{(G)} \tag{16}$$

where $\mathbf{c}'$ is the complement to $\mathbf{c}$.

The same selection process is repeated by comparing $\mathbf{b}_\vartheta^{(G)}$ with the best solution obtained from previous generation $\mathbf{b}_\alpha^{(G)} = \mathbf{b}_\rho^{(G-1)}$. The process of sDE would halt when either one of the following conditions, i.e., $\mathbf{b}_\rho < \epsilon$ or $G > G_{thres}$, are met. Then, $\tilde{\mathbf{b}}$ would be the the current best $\mathbf{b}_\rho$. Based on $\tilde{\mathbf{b}}$, the target ToI can be selected as well as retrieving the corresponding identifier for interaction.

V. Experiments and Results

In this section, we first validate the optimum mutation rate $\mathcal{M}$ and the crossover rate $\mathcal{C}$ required by sDE before further with extensive experiments to evaluate the performance of sDE in comparison to three carefully selected algorithms. The experiments were performed with the same dataset collected from both indoor and outdoor environments. Section V-B presents the three selected algorithms, the corresponding results are discussed in Section V-C.

A. Mutation and Crossover Rates

Both $\mathcal{M}$ and $\mathcal{C}$ are the two design parameters required by sDE during the mutation and crossover process, as described in Eq. (14)-Eq. (16). In general, it is recommended to have $\mathcal{M} = (0, 2]$ and $\mathcal{C} = (0, 1]$. Detail exploration regarding the various configurations of $\mathcal{M}$ and $\mathcal{C}$ over various benchmark functions are available in [32]. In this paper, we validate the configurations of $\mathcal{M}$ and $\mathcal{C}$ using our real dataset. We ran the experiment by setting $t_d = 100ms$ and $1000ms$. From Fig. 13, it was observed that the accuracy increases when $\mathcal{M}$ and $\mathcal{C}$ increase. However, it might not be a wise choice to use high $\mathcal{M}$ and $\mathcal{C}$, which might easily lead to overfitting and fail to generalize for other related problems. Furthermore, a high $\mathcal{M}$ can cause a frequent mutation which increases the algorithm runtime. On the other hand, high $\mathcal{C}$ means that the evolve vector would always inherit the element from the current best solution which might cause the solution to trap at the local optimum. Hence, we opt to set $\mathcal{M} = 0.5$ and $\mathcal{C} = 0.6$ in the light of the results obtained in Fig. 13.

B. Baseline Algorithms

Three algorithms are selected as the baseline, i.e., PD, kNN, and CoSaMP. Among them, PD has been widely adopted by the industrials, whereas kNN and CoSaMP have been exploited by related academia research.

- PD is a simple algorithm provided by most beacon manufacturers to escalate the development process [35]. Usually, PD selects the target ToI by selecting the element in $\mathbf{\Phi}_y$ which returns the strongest RSS measurement.

$$\tilde{y} = \arg\max_{i \in \Phi_y} \Phi_y \tag{17}$$

  where $\tilde{y}$ is the pointer to the target ToI.
- kNN has been used by many fingerprinting related works. In [22], an artificial RSS value (typically $-100dBm$) is introduced to fill in the missing entries, i.e., $\phi_{i,y} = \{-100 : i \in b \vee \phi_{i,y} = \emptyset\}$, before applying kNN to select the target ToI. In general, kNN computes the Euclidean distance to locate the fingerprint which has the smallest distance to Φ_y, i.e.,

$$\tilde{y} = \arg\min_{\forall \Phi_i \in \mathbf{\Omega}} \|\Phi_y - \Phi_i\|_2^2 \tag{18}$$

- CoSaMP is widely used in dealing with compressive sensing problem [36], and it also has been used to tackle the sparse issue related to proximity fingerprint [37]. Mathematically, CoSaMP transforms $\Psi\Phi$ into an orthonormal basis $Q = \text{orth}(\Psi\Phi)^T$, and then use a signal pre-processing operator $T = Q(\Psi\Phi)^\dagger$ to transform Φ_y to $\Phi'_y = T\Phi_y$. before estimating the ToI indication vector $\widehat{\mathbf{b}}$.

$$\widehat{\mathbf{b}} = \arg\min_{\mathbf{b}} (\|\Phi'_y - Q\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|_1) \tag{19}$$

  Based on the resultant $\widehat{\mathbf{b}}$, $\tilde{y}$ can be found by taking the entry which contributes the highest value.

C. Results

Two metrics are used to evaluate the performance of the above three algorithms in comparison to our proposed sDE. Accuracy is computed by taking the ratio of the number of correct interaction to the total number of interactions which have been triggered; whereas, the reliability is evaluated using Eq. (4), which implies that the reliability is the ability to trigger the correct interactions within an arbitrarily short time duration t_d under a certain tolerable error.

1) Accuracy and Reliability: Fig. 14 shows the accuracies achieved by all the four algorithms with respect to t_d, for $T_a = 100ms$, $400ms$, $600ms$, and $1000ms$. Given $T_a = 100ms$, PD and kNN have a very low accuracy (below 40%) when $t_d = 10ms$; whereas CoSaMP achieves 43.90% and sDE 57.35%. Even though the chances to observe a packet in such a short t_d (i.e., $10ms$) is relatively low, our proposed sDE is still able to achieve above average accuracy. In general, sDE achieves 81.91% accuracy when t_d increases to $40ms$, and 100% when $t_d = 400ms$. Note that most of the interactive applications can accept latency up to $400ms$, hence, the results indicate the capability of sDE in guaranteeing a real-time interaction with 100% accuracy.

However, a short T_a is generally not the preferable choice consider the battery lifespan of a beacon. Hence, we also verified the performance of all the four algorithms with different T_a. It is clear that all the four algorithms suffer performance degradation when T_a increases. Given $t_d = 400ms$, our sDE still can maintain a decent performance. In particular, sDE achieves 89.97% accuracy when $T_a = 200ms$, 86.12% when $T_a = 400ms$, 82.80% when $T_a = 600ms$, 75.91% when $T_a = 800ms$ and 74.65% when $T_a = 1000ms$. According to the specifications provided by the beacon's manufacturer, the beacon can last for at least 2 years when $T_a = 800ms$. From the results shown in Fig. 14, we are well-assured that our proposed solution can achieve more than 80% accuracy when $T_a = 600ms$, which also allow the beacon to last for more than 1 year.

Fig. 15 illustrates the relationship between the reliability and the accuracy, given $t_d = 100ms$, $400ms$, and $1000ms$. Obviously, the reliability decreases when t_d decreases, and increases when t_d increases to $1000ms$. Refer back to Fig. 7(c), it is clear that the observation rate increases when $t_d = 1000ms$. Given the high observation rate, most of the algorithms produce 0% error with high reliability. In other words, sDE, CoSaMP, PD and kNN are guaranteed to produce 100% accuracy for at least 70.32%, 59.43%, 42.70% and 35.92%, respectively, when $t_d = 1000ms$. However, to cater for the latency requirement of interaction, it is advisable to keep t_d to $400ms$ and below. When $t_d = 400ms$, the reliability of many algorithms drop, as shown in Fig. 15(b). Among these algorithms, our proposed sDE is reliable, in which sDE is able to guarantee at least 80% accuracy for 83.41% of the time when $t_d = 400ms$, i.e., $R = P(1 - P_A \leq 0.2 | t_d = 400ms) = 0.8341$.

2) Implications of Algorithm's Runtime: We measured the runtime in a normal computer with a single core processor Intel Core i5-8265U 1.6GHz and 12GB RAM. For quick verification, we implemented all the algorithms in MATLAB and used the collected data for experiment. The runtime measurement presented in Fig. 13 is based on the result obtained through MATLAB. For the practical demonstration, the algorithms were implemented directly on the smartphone in C (using the Xamarin framework for mobile app development). Nowadays the modern smartphones are powerful enough to execute all these algorithms. From the results illustrated in Fig. 14 and Fig. 15, it is obvious that sDE achieves a superior performance than the rest of the three algorithms. Nonetheless, the performance of CoSaMP is comparable with sDE especially when t_d and T_a increase, and thus increase the observation rate. However, CoSaMP is generally having a more complex algorithm which requires longer runtime than sDE; whereas PD is simple and straightforward. Note that we have been discussing that $t_d = 400ms$ is acceptable for latency-sensitive applications which involve frequent interactions; however, this $400ms$ simply indicates that the time used by the packet acquisitions, and does not take the algorithm runtime into consideration.

Fig. 16 indicates the runtime of all the 4 algorithms with respect to the data size. Two graphs are used to plot the runtime because both kNN and PD are definitely having a much lower

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974230, IEEE Internet of Things Journal

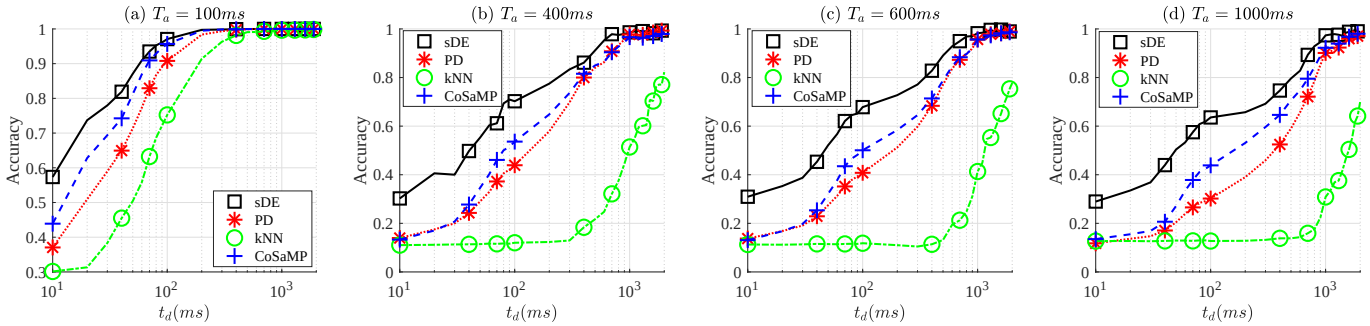

Fig. 14. Average accuracies achieved by the four algorithms with respect to t_d when (a) $T_a = 100ms$, (b) $T_a = 400ms$, (c) $T_a = 600ms$, and (d) $T_a = 1000ms$

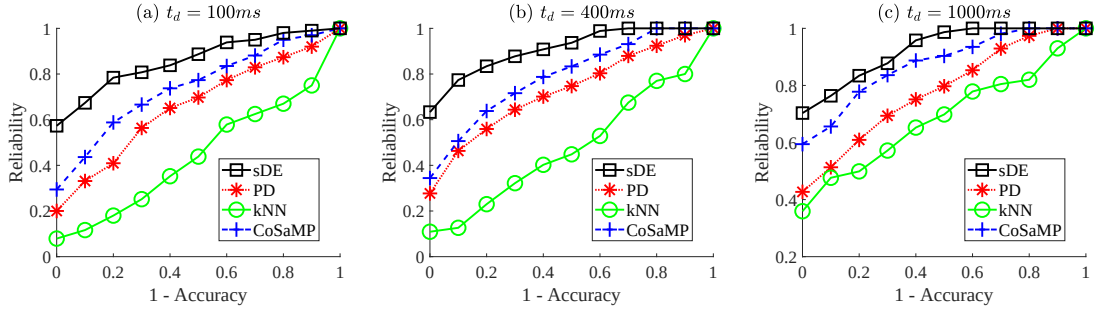

Fig. 15. The reliability of all the four approaches when (a) $t_d = 100ms$, (b) $t_d = 400ms$, and (c) $t_d = 1000ms$.

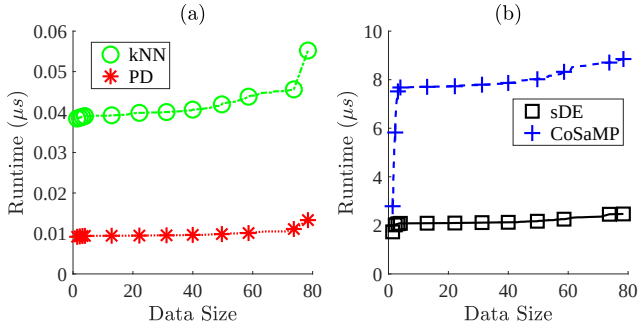

Fig. 16. The algorithms' runtime with respect to the data size.

runtime than CoSaMP and sDE, which might unclear when putting all the four algorithms in one graph. The data size simply means the total number of packets the user device received during t_d, which is different from the observation rate, as discussed in Section III-B2. Obviously, PD is having the fastest runtime compared to the rest, i.e., in average $9.64ps$. The runtime of CoSaMP increases abruptly when the data size grows from 0 to 10, whereas sDE has almost consistent runtime (i.e., about $2.148\mu s$ in average) despite the growing data size. Since the worst-case runtime is still in the unit of μs, it is obvious that none of the algorithm's runtime has a significant impact on the total latency. Hence, the impact of the algorithm's runtime is negligible for all the cases.

VI. Proof-of-Concept Prototype

A proof-of-concept prototype is built and tested in a practical testing to demonstrate the feasibility of our proposed sDE for real-world interactive applications. The prototype is a complete system consists of an online server, smartphones, and ToIs. Fig. 17(a) indicates the seven ToIs. Each ToI is attached with a working beacon with $T_a = 600ms$. An Android-based application which is capable of performing fingerprint registration and ToI identification is developed and installed into the smartphones. When a ToI is added, we can use the App to register its fingerprints and the registered fingerprint would be uploaded and stored in a fingerprint database. Fig. 17(b) shows that a unique ID would be assigned automatically upon a successful registration. Fig. 17(c) shows the list of registered fingerprints, these fingerprints would be exported to Matlab to construct the fingerprint matrix.

The user can interact with the ToI when the user device successfully retrieves its identifier, i.e., the fingerprint in our context. The packets obtained through the real-time scanning would be streamed to the Matlab through TCP/IP socket. All the four algorithms are implemented in the Matlab. The final results obtained by these four algorithms would then return to the smartphone through the same socket communication. Each ToI can speak automatically to the user through the smartphone when the user device approaches the ToI. We used the Android Text-to-Speech (TTS) API to achieve the above purpose. Using the smartphone microphone, the user can also speak to the ToI.

A. Implications of the Battery Conditions

In this experiment, we tested each ToI for at least 25 times. The interaction response was marked incorrect when (1) the user device fails to output the correct ToI, or (2) the user device only outputs the correct ToI by the time the user device has already moved away. In the first experiment, we inserted a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974230, IEEE Internet of Things Journal

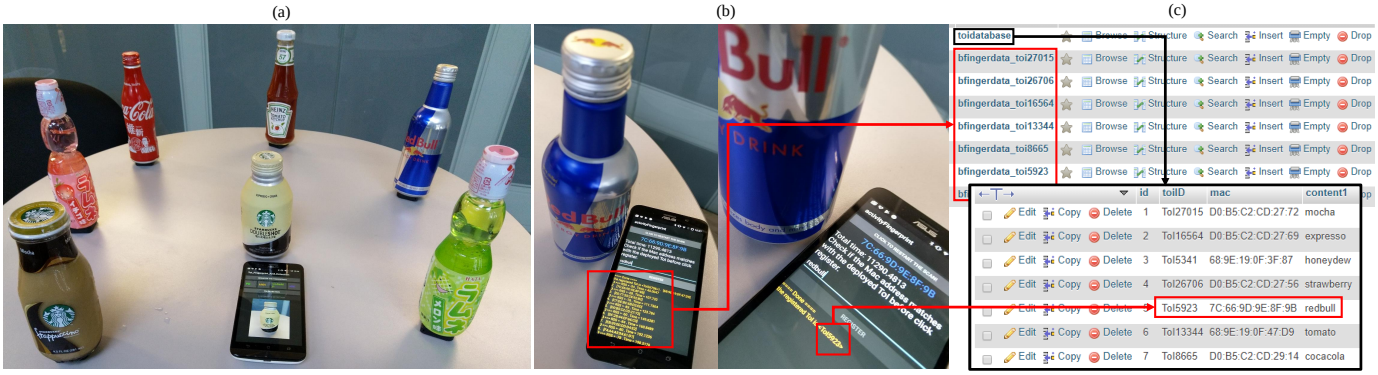

Fig. 17. (a) A total of 7 ToIs are implemented. (b) The process of fingerprint registration when a ToI is added. (c) The registered fingerprint is uploaded and stored in an online server, based on MySQL framework.

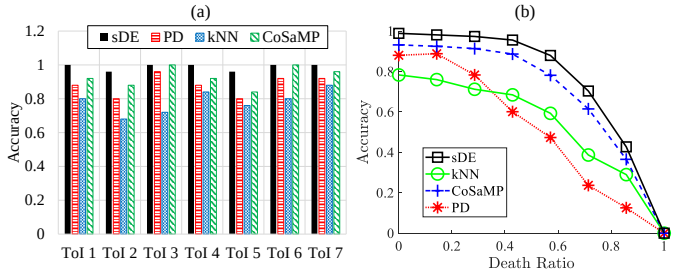

Fig. 18. The accuracy achieved by all the four algorithms when (a) all beacons were with fully working battery, (b) some batteries were dead.

TABLE I
AVERAGE RESPONSE TIME REQUIRED BY EACH IMPLEMENTED ALGORITHM TO RETURN A CORRECT IDENTIFICATION.

Algorithm	PD	kNN	CoSaMP	sDE
ToI 1	$1200ms$	$1550ms$	$800ms$	$402ms$
ToI 2	$1350ms$	$1670ms$	$870ms$	$400ms$
ToI 3	$1220ms$	$1770ms$	$990ms$	$403ms$
ToI 4	$1250ms$	$1650ms$	$950ms$	$400ms$
ToI 5	$1310ms$	$1670ms$	$930ms$	$400ms$
ToI 6	$1220ms$	$1720ms$	$890ms$	$403ms$
ToI 7	$1300ms$	$1750ms$	$900ms$	$403ms$
Average	**1264.3ms**	**1682.9ms**	**904.29ms**	**401.57ms**

new battery to each attached beacon and manually checked the working condition to ensure all the beacons are fully working. The result is shown in Fig. 18(a). In general, sDE achieves a decent performance with average accuracy of 98.85%, follows by CoSaMP 93.14%, PD 88% and kNN 78.28%. In the second experiment, we purposely took out the battery in some beacons to emulate a worst-case scenario with died battery. The death ratio is computed by dividing the number of beacons with a dead battery by the total number of ToIs. Similar steps were performed for each death ratio, and the result is illustrated in Fig. 18(b) Again, our proposed sDE outperforms the rest with a very reliable performance, i.e., sDE achieves more than 80% accuracy even though more than 50% of the beacons were with a dead battery.

B. Real-time Performance

Since the interaction involves a socket communication connecting the user device and the server, it is of critical importance to examine the end-to-end latency. In this experiment, we manually adjust t_d and wait patiently until the user device returns with correct interaction. By correct interaction, we mean the ability of user device to respond to the user with the necessary multimedia content (e.g., the speech or the image) when the user is in the proximity of the ToI. The App is programmed to output the time elapsed from scanning, communicating with the server to transmitting content. Hence, we can jot down the time precisely up to ms. If the algorithm returned a correct interaction response, we will reduce t_d; otherwise, we will increase t_d until it is able to return a correct response. Same procedures were repeated for ten times, and the results are presented in Table I. Again, sDE is able to return correct interaction response within $400ms$, which aligned to the results discussed in Section V-C1.

VII. CONCLUSIONS

While there is an exponential adoption of beacon for commercial interactive applications, there is no work, so far, investigate the issue related to interaction reliability. This paper studies the smart interaction between the user device and ToI, highlight the research challenges and presents our proposed solution to enhance the interaction reliability. This is the first work that provides a thorough analysis regarding the possible challenges in connection with interaction reliability. Most importantly, as the adoption of beacon continues to grow, this paper takes a step forward to provide a solution which can guarantee the reliability despite the environmental variations and the beacon's operating conditions. For future work, we can incorporate machine learning method to train a ToI classification method (to first classify the ToI according to varying environment) directly on smartphone (Note that the has been a lot of efficient machine learning algorithms for on-device training, for example, tensorflow lite) before applying the sDE for further identification.

ACKNOWLEDGMENT

This work was supported by HKUST-NIE Social Media Laboratory.

REFERENCES

[1] I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J. M. Gaudin, and C. Guichard, "Two-level qr code for private message sharing and document authentication," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 571–583, March 2016.

[2] L. Zhang, W. Xiang, and X. Tang, "An efficient bit-detecting protocol for continuous tag recognition in mobile rfid systems," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 503–516, March 2018.

[3] H. Wu, Y. Wang, and Y. Zeng, "Capture-aware Bayesian RFID tag estimate for large-scale identification," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 119–127, 2018.

[4] K. Ding and P. Y. Jiang, "Rfid-based production data analysis in an iot-enable smart job-shop," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 128–138, 2018.

[5] A. F. Harris III, V. Khanna, G. Tuncay, R. Want, and R. Kravets, "Bluetooth low energy in dense iot environments," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 30–36, 2016.

[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 of Things Journal*, vol. 5, no. 2, pp. 811–828, April 2018.

[7] M. Papandrea, S. Giordano, S. Vanini, and P. Cremonese, "Proximity marketing solution tailored to user needs," in *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*. IEEE, 2010, pp. 1–3.

[8] X. Ma, X. Yin, G. Butron, C. Penney, and K. S. Trivedi, "Packet delivery ratio in k-dimensional broadcast ad hoc networks," *IEEE Communications Letters*, vol. 17, no. 12, pp. 2252–2255, December 2013.

[9] L. Yi, L. Shiyao, and C. Deyuan, "The reliability of bluetooth data transmission in mobile medical information acquisition system," in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Oct 2016, pp. 498–505.

[10] R. Kravets, A. F. Harris, III, and R. Want, "Beacon trains: Blazing a trail through dense ble environments," in *Proceedings of the Eleventh ACM Workshop on Challenged Networks*, ser. CHANTS '16, 2016, pp. 69–74.

[11] P. C. Ng, J. She, and S. Park, "High resolution beacon-based proximity detection for dense deployment," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1369–1382, June 2018.

[12] W. He, P. Ho, and J. Tapolcai, "Beacon deployment for unambiguous positioning," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1370–1379, Oct 2017.

[13] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, Nov 2015.

[14] C. Xiao, D. Yang, Z. Chen, and G. Tan, "3-d ble indoor localization based on denoising autoencoder," *IEEE Access*, vol. 5, pp. 12 751–12 760, 2017.

[15] S. Sridhar, P. Misra, G. S. Gill, and J. Warrior, "Cheepsync: a time synchronization service for resource constrained bluetooth le advertisers," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 136–143, January 2016.

[16] K. Cho, G. Park, W. Cho, J. Seo, and K. Han, "Performance analysis of device discovery of bluetooth low energy (ble) networks," *Computer Communications*, vol. 81, pp. 72–85, 2016.

[17] J. Liu, C. Chen, and Y. Ma, "Modeling neighbor discovery in bluetooth low energy networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1439–1441, September 2012.

[18] P. H. Kindt, M. Saur, M. Balszun, and S. Chakraborty, "Neighbor discovery latency in ble-like protocols," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 617–631, March 2018.

[19] A. Ghose, C. Bhaumik, and T. Chakravarty, "Blueeye: A system for proximity detection using bluetooth on mobile phones," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 1135–1142.

[20] P. C. Ng, J. She, and S. Park, "Notify-and-interact: A beacon-smartphone interaction for user engagement in galleries," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 1069–1074.

[21] M. M. Scheunemann, K. Dautenhahn, M. Salem, and B. Robins, "Utilizing bluetooth low energy to recognize proximity, touch and humans," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug 2016, pp. 362–367.

[22] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based indoor localization with bluetooth low energy beacons," *Sensors*, vol. 16, no. 5, p. 596, 2016.

[23] P. C. Ng, L. Zhu, J. She, R. Ran, and S. Park, "Beacon-based proximity detection using compressive sensing for sparse deployment," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2017, pp. 1–6.

[24] P. C. Ng, J. She, and R. Ran, "Towards sub-room level occupancy detection with denoising-contractive autoencoder," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[25] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. Oh, "Semisupervised deep reinforcement learning in support of iot and smart city services," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624–635, April 2018.

[26] M. Wang and J. Brassil, "Managing large scale, ultra-dense beacon deployments in smart campuses," in *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*. IEEE, 2015, pp. 606–611.

[27] Cc2541 (active) bluetooth low energy and proprietary wireless mcu. [Online]. Available: http://www.ti.com/product/CC2541

[28] M. Leigsnering, F. Ahmad, M. G. Amin, and A. M. Zoubir, "Compressive sensing-based multipath exploitation for stationary and moving indoor target localization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 8, pp. 1469–1483, Dec 2015.

[29] M. Zhou, F. Qiu, Z. Tian, K. Xu, and Q. Jiang, "Positioning error vs. signal distribution: An analysis towards lower error bound in wlan fingerprint based indoor localization," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[30] S. Y. Park, S. Lim, D. Jeong, J. Lee, J. S. Yang, and H. Lee, "Pufsec: Device fingerprint-based security architecture for internet of things," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9.

[31] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Qin, X. Shen, and X. y. Li, "S2m: A lightweight acoustic fingerprints-based wireless device authentication protocol," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 88–100, Feb 2017.

[32] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[33] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb 2011.

[34] Xin Yao, Yong Liu, and Guangming Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, July 1999.

[35] F. Zafari, I. Papapanagiotou, M. Devetsikiotis, and T. Hacker, "Enhancing the accuracy of ibeacons for indoor proximity-based services," in *IEEE ICC*, 2017.

[36] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[37] P. C. Ng, J. She, and R. Ran, "A compressive sensing approach to detect the proximity between smartphones and ble beacons," *IEEE Internet of Things Journal*, pp. 1–1, 2019.