Denoising-Contractive Autoencoder for Robust Device-Free Occupancy Detection

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

***Abstract*—Device-free occupancy detection is very important for certain Internet of Things applications that do not require the user to carry a receiver. This paper achieves the device-free occupancy detection with RF fingerprinting, which labels each zone with a $2M$-dimensional fingerprint vector. Specifically, the fingerprint vector consists of received signal strength (RSS) values measured from M Bluetooth low energy (BLE) beacons and also their corresponding temporal RSS variations. However, the unreliable RSS values caused two common issues with the fingerprint vector: 1) noise and 2) sparsity. To this end, we propose denoising-contractive autoencoder (DCAE) to jointly deal with these two issues, by learning a robust fingerprint prior to device-free occupancy detection. We validate the performance of our proposed DCAE with large-scale real-world datasets. The experimental results indicate the substantial performance gain of our proposed DCAE in comparison with state-of-the-art autoencoders. In particular, the classifier trained using the fingerprints learned by our proposed DCAE is able to maintain at least 90% accuracy when the noise factor or sparsity ratio increases to 0.6 and 0.5, respectively.**

***Index Terms*—Autoencoder, Bluetooth low energy (BLE) beacon, occupancy detection.**

I. Introduction

DEVICE-FREE occupancy detection often manipulates wireless signals to infer the presence of an occupant without having the occupant carry any device. In this paper, we refer to any living subject including humans and animals as an occupant. Even though smartphones are ubiquitous nowadays, it is very unlikely for users to carry a smartphone with them all the time, especially when they are at home or using the washroom. While a camera-based approach is commonly used for occupancy detection, it is impossible to install cameras in every corner of home owing to their intrusive nature. For example, it is uncomfortable to have a camera installed in the sleeping room or the washroom. Furthermore, it is ridiculous to have a baby or a pet in the house carry with them a smartphone. RF fingerprinting has been widely exploited by localization applications to provide location-based service [1]. However, these localization applications generally consider a public scenario with many users moving around an indoor or outdoor environment [2]. Our occupancy detection, on the other hand, considers a more private use case involving less number of occupants. For example, it can help to find our pet inside our house, or notify us if our beloved family members fainted inside the washroom.

RF fingerprinting is a technique that labels a zone with a fingerprint vector $\mathbf{\Phi} \in \mathbb{R}^M$, where M denotes the number of ambient wireless transmitters. Note that the word "*zone*" refers to a smaller area that is a division of a large location. For example, we can divide our home into a few smaller zones, such as the reading zone, the dining zone, etc., subject to the functionality of each space. In this paper, we exploit the signals transmitted by M Bluetooth low energy (BLE) beacons to construct an M-dimensional vector $\mathbf{\Phi}$. Each element in $\mathbf{\Phi}$ indicates the time average received signal strength (taRSS) value measured from a particular beacon. Most RF fingerprinting systems are device-based in which each occupant is assumed to carry with them a receiver. Hence, it is relatively easy to observe the change of RF fingerprint when the user moves from one zone to the other since the receiver is moving with the user. However, such an approach also suffers the unreliability owing to the RSS fluctuation. Many robust RF fingerprint methods have been proposed to mitigate this RSS fluctuation issue. For example, Lin *et al.* [3] proposed a neighbor relative RSS (NR-RSS) in constructing the fingerprint; whereas He *et al.* [4] constructed the fingerprint by fusing the distance information. However, both methods above are only suitable for cases involving moving receivers.

While our previous work [5] achieves good detection performance with device-based occupancy detection inside a private room, it is relatively challenging to achieve similar performance with a device-free system. However, consider the infrequent use of smartphone inside a private location, it is deemed necessary to provide device-free occupancy detection. In general, the device-free system deployed the receiver as infrastructure in a fixed location to monitor the presence of an occupant. Having said that, the RSS values measured by the receiver is fairly consistent exclusive of the occupant's movement. Despite the above challenge, our preliminary experiment unveils that there exists a huge variation in the RSS values in connection with the presence of a human/animal body. In light of this, we compute the temporal difference between subsequent RSS (tdRSS) measurements and append this M-dimensional tdRSS to the existing fingerprint, resulting in a $2M$-dimensional fingerprint vector for each zone.

In practical scenarios, the RF fingerprint vector $\mathbf{\Phi}$ always suffers the following two issues: 1) noise and 2) sparsity. A fingerprint vector corrupted by noise always has changing and

Manuscript received March 10, 2019; revised June 30, 2019; accepted July 14, 2019. Date of publication July 19, 2019; date of current version December 11, 2019. This work was supported by the HKUST-NIE Social Media Laboratory. *(Corresponding author: Pai Chet Ng.)*

The authors are with the Department of Electronics and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: pcng@ust.hk; eejames@ust.hk).

Digital Object Identifier 10.1109/JIOT.2019.2929822

2327-4662 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

unpredictable RSS values even though the receiver remains stationary. On the other hand, a sparse fingerprint vector means that the vector contains many zeros elements, caused by frequent packet loss. To this end, this paper proposes denoising-contractive autoencoder (DCAE) to learn a robust fingerprint from the noisy and sparse fingerprint vector. To the best of our knowledge, this is the first work that jointly addresses the above two issues with a novel autoencoder. The main contributions of this paper are summarized as follows.

1) Device-free occupancy detection is achieved by unveiling the subtle change of the RF fingerprint with respect to the human/animal body. Furthermore, we construct the fingerprint vector by appending tdRSS to the conventional taRSS.
2) DCAE is capable of reconstructing the fingerprint vector from both noisy and sparse fingerprint vector. We validate the DCAE with different optimization methods before applying the learned fingerprints for classification training.
3) The performance of our proposed approach is validated with real-world datasets. The experimental results indicate that our proposed DCAE outperforms state-of-the-art autoencoders with different classifiers.

The rest of this paper is organized as follows. Section II reviews the related work. Section III provides an empirical analysis of the relationship between the RF signals and the body. Section IV presents our device-free occupancy detection system. Section V presents our proposed DCAE. Section VI describes our experimental testbed for collecting the dataset and discusses the experimental results. Section VIII concludes this paper.

II. Related Work

This section reviews the related works in occupancy detection, including both device-based and device-free occupancy detection. After that, we review the recent works that leverage machine learning techniques (both supervised and unsupervised) to enhance the detection performance.

A. Occupany Detection

Occupancy detection is a binary classification problem where value 1 indicates the presence of an occupant. Such occupancy information is beneficial for many applications, including emergency management [6], energy consumption monitoring [7], proximity marketing [8], etc. In general, occupancy detection can be broadly divided into the following two types.

1) Device-Based Occupancy Detection: There have been many research works on detecting the presence/position/location of a mobile device [9], [10] using wireless signals from BLE beacons. In particular, Wu *et al.* [9] jointly considered the RSS values from the beacon and the accelerometer measurements from the smartphone to deliver an accurate localization. On the other hand, Faragher and Harle [10] constructed the RF fingerprint with the RSS values from the deployed beacons. Both Wu *et al.* [9] and Faragher and Harle [10] assumed that the users always carry with them a smartphone. In other words, it is impossible to detect the presence of an occupant in the absence of a smartphone. Furthermore, none of the above works consider the packet loss issue. Note that the smartphone is unable to measure the RSS value from a particular beacon when it fails to observe a packet from that beacon.

2) Device-Free Occupancy Detection: It is very unlikely for users to always carry with them a smartphone, especially when they are in their own house. However, occupancy detection is deemed necessary for a home environment to ensure the safety of our family members and pets. There are a number of works that manipulate diverse sensor information (e.g., light, temperature, CO_2, etc.) [11] or power consumption information [7] to achieve device-free occupancy detection. However, none of these works can estimate the number of occupants in a given time frame. Recently, there has been active research in exploiting wireless signals to deliver nonintrusive device-free occupancy detection. In particular, Yang *et al.* [12] used the channel state information (CSI) from the WiFi device to infer the occupancy activities. Chang *et al.* [13], on the other hand, delivered fine-grained occupancy detection with RSS-based radio map. While Mager *et al.* [14] addressed the detection performance issue with respect to the changing environments, none of the above works consider performance degradation due to the packet loss.

B. Machine Learning for Occupancy Detection

Many machine learning techniques have been applied to provide better occupancy detection, either through a supervised or unsupervised approach.

1) Supervised Learning: Supervised learning trains a model to classify zones. Many different classification techniques have been adopted to deal with either occupancy or localization problems. Supervised neural network (NN) classifier has been used by Ekwevugbe *et al.* [15] for occupancy detection, and by Anzum *et al.* [16] for zone-based localization. Lei *et al.* [17] adopted logistic regression to improve the localization accuracy in changing environments, while Zhou *et al.* [18] used a support vector machine (SVM) classifier to enhance the performance in noisy environments. Note that all these works merely use the raw RSS values to construct the fingerprint vector; hence, they are able to achieve good performance only when the testing environments match the training environment. Recently, there has been significant interest in using unsupervised machine learning techniques, especially autoencoder, to enhance the detection performance by learning a robust fingerprint vector.

2) Unsupervised Learning: Autoencoder is an unsupervised NN that can learn a robust hidden representation via a reconstruction process [19], [20]. Most autoencoders, including stacked autoencoder (SAE) [21] and denoising autoencoder (DAE) [22], aim to learn a compressed representation with a discriminative approach. Xiao *et al.* [23] and Chen *et al.* [24] leveraged DAE to learn a robust fingerprint in noisy environments. Wu and Tseng [25] and He *et al.* [26], on the other hand, used SAE to learn a robust fingerprint

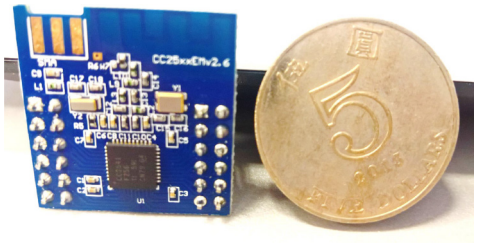

Fig. 1. BLE beacon is a tiny device with a similar size to a five dollar coin.

from WiFi devices. While all of the above works improve the detection performance by learning a robust fingerprint representation in noisy and always changing environments, none of them jointly address the issues related to noise and sparsity.

III. Bluetooth Low Energy Beacon

BLE is an emerging technology tailored for Internet of Things (IoT) development [27], [28]. Among all the BLE-enabled devices, BLE beacon has received a lot of interest for its promising features for IoT application. BLE beacon is a low power transmitter, which is designed to broadcast its packet according to the pre-defined advertising interval T_a. Previous work [10] has shown that BLE beacons are the potential infrastructure for RF fingerprint in comparison to the WiFi access points. Furthermore, the small form factor of BLE beacon, as shown in Fig. 1, allows the beacon to be deployed in a remote area that is not accessible by the commodity WiFi device.

As discussed, BLE beacon broadcast their packets periodically according to T_a. The RSS value can be measured by the receiver upon receiving the packet. RSS is a measurement in dBm scale [i.e., $P_{r,(\text{dBm})} = 10\log(P_{r,(\text{watt})}/1\text{ mW})$], which typically ranges from -20 to -90 dBm subject to the distance between the beacon and the receiver. Many existing works simply use the raw RSS measurements to construct the fingerprint vector [10], [29], [30]. However, RSS values have been proven to be unreliable as a consequence of shadowing effects and multipath fading [31], [32]. In this section, we analyze the RSS values in different environments using the real-world datasets collected by Lazik *et al.* [33]. We also show that the fingerprint vector is always sparse by analyzing the dataset provided by Mohammadi *et al.* [34]. Lastly, an empirical analysis is conducted to investigate the signal attenuation factor with respect to the human/animal body.

A. Always Changing RSS Values Due to Unpredictable Environmental Variations

We used the CMU localization dataset [33] to investigate the variations of RSS in different environments. The beacon data contains the following attributes: 1) timestamp; 2) major value; 3) minor value; 4) distance; 5) RSS value; and 6) proximity information. The data were collected from six different environments, and each beacon was mounted on a tripod, as depicted in Fig. 2. Fig. 3(a) shows the RSS values from ten beacons deployed in a lobby area, and Fig. 3(b) shows the RSS values from four beacons deployed along a corridor. Note that the receiver remained stationary during the data collection;

Fig. 2. CMU localization dataset [33] consists of the beacon data collected from ten beacons mounted on different tripods.

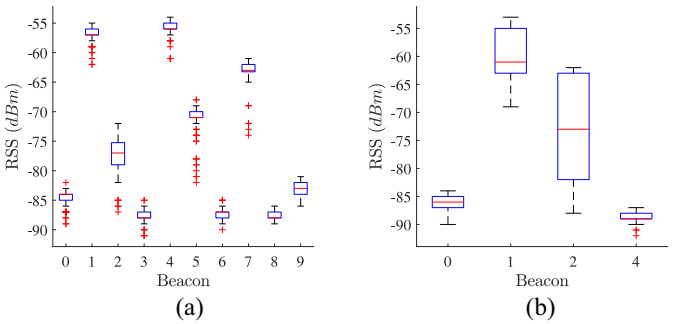

Fig. 3. RSS values suffer severe fluctuation in both (a) lobby and (b) corridor environment.

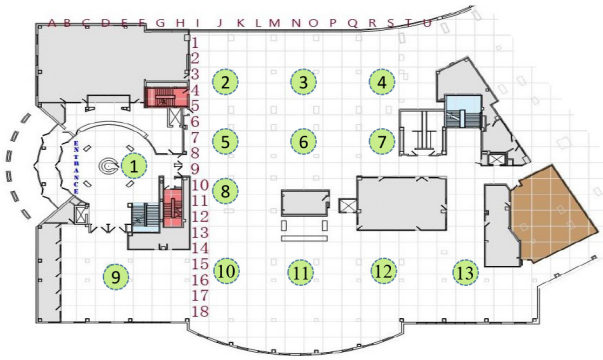

Fig. 4. BLE RSSI dataset [34] consists of RSS values measured from 13 beacons (indicated by the green dotted circle) deployed in Waldo Library.

however, from both Fig. 3(a) and (b), we observe that the RSS values suffer severe fluctuation, in which their values varied from -91 to -53 dBm. Such variations could be due to the environmental noise and multipath reflection. To address the RSS fluctuation issue, this paper adopts the Frobenius norm to train the autoencoder such that the encoded feature learned in the hidden layer is less sensitive to the input variations.

B. Sparse Fingerprint Vector Due to Frequent Packet Loss

The fingerprint vector acquired during real-time observation always suffers from a sparsity problem as a consequence of unpredictable packet loss. We used the BLE RSSI dataset [34] to verify the sparseness of the fingerprint vector. This dataset consists of RSS values measured from 13 beacons deployed in the Waldo Library, Western Michigan University. Fig. 4 depicts the blueprint of the library and the location of each beacon. In total, 1420 fingerprint vectors were collected at 105 different

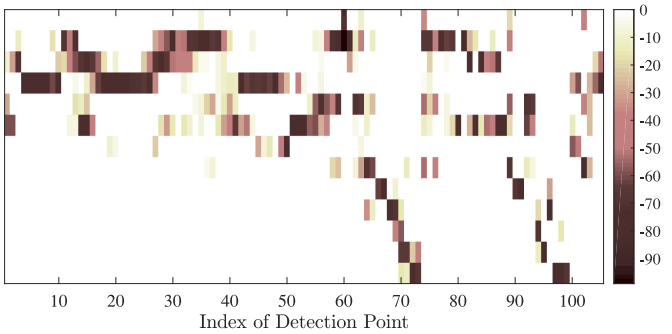

Fig. 5. Fingerprint matrix indicates the RSS values measured from the 13 beacons at all 105 detection points.

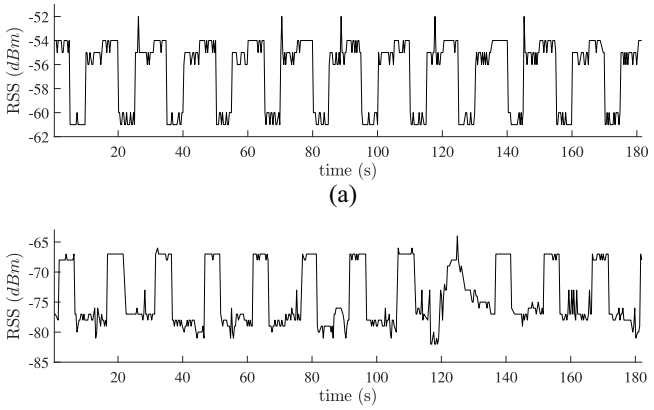

Fig. 6. Pattern of RSS values (a) when there is no human body in the environment and (b) when there is. The RSS values further suffer uncommon variations when there is a significant human movement.

detection points, where each point is defined by its column alphabet and row number. Fig. 5 illustrates the fingerprints by detection points matrix. Note that the white area with the value 0 indicates a missing element instead of 0 dBm. From Fig. 5, we can see that the fingerprint vectors for most of the detection points are, in fact, very sparse. In other words, some of the packets from certain beacons fail to arrive at the receiving end. To address the sparsity issue, we exploit the loss function introduced by the DAE to train the autoencoder.

C. Relationship Between Signal Attenuation and Body

To investigate the relationship between signal attenuation and body, we performed data collection in a controlled environment by making sure that there is no human/animal body presence during the data collection process. After this, we repeated the same data collection process with human presence. Fig. 6(a) and (b) shows the pattern of the RSS values from 0 to 3 min for the above two cases, respectively. It is clear that the presence of a human body causes severe attenuation on the RSS values. In particular, the mean RSS reported by the receiver increases from $\bar{r} = -56.69$ dBm to $\bar{r} = -74.36$ dBm when there is a human body in the environment, whereas the variance increases from $\text{var}(r) = 8.3524$ to $\text{var}(r) = 28.4019$. Note that there is a uncommon variation during 120–140 s, as indicated in Fig. 6(b). Such an uncommon variation was

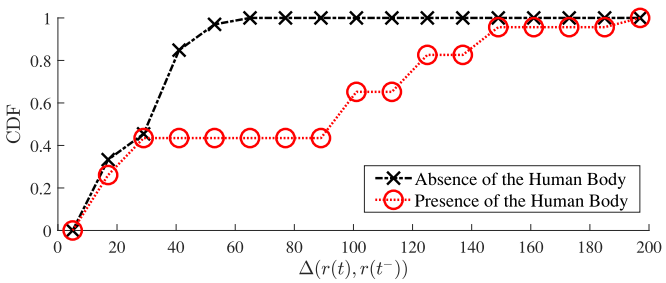

Fig. 7. CDF of dfRSS, i.e., Δ between subsequent RSS values.

TABLE I
SUMMARY OF MATHEMATICAL NOTATIONS

Notation	description
$\mathcal{B}$	a set of deployed beacon nodes
$\mathcal{Z}$	a set of zones
M	total number of beacons, $M = \lvert\mathcal{B}\rvert$
N	total number of zones, $N = \lvert\mathcal{Z}\rvert$
$\mathcal{R}$	a set of instantaneous RSS measurements
$\mathbf{\Omega}$	fingerprint matrix
$\mathbf{\Phi}^{(z_i)}$	fingerprint vector
r	received signal strength (RSS) value in dBm scale
Δ	temporal difference between subsequent RSS values
T_a	advertising interval
T_s	scanning duration
ϵ	noise vector

caused by the movement of the body. More precisely, the presence of a body causes signal attenuation, and the movement of the body further causes unpredictable variations.

Furthermore, there is a significant difference between subsequent RSS values with the presence of a human body. Fig. 6(a) and (b) shows that RSS values vary between $[-52, -62]$ dBm in the absence of a human body, and between $[-65, -85]$ dBm with the presence of a human body. We computed the tdRSS values, i.e., $\Delta = (r(t) - r(t^-))^2$, and plot the cumulative distribution function of Δ, as shown in Fig. 7. From Fig. 7, we can see that $\Delta < 60$ for at least 90% when there is no human body, whereas $\Delta < 160$ when there is a human body. Based on such observations, we can use Δ to construct the fingerprint vector other than merely rely on the raw RSS values.

IV. Device-Free Occupancy Detection

This section describes our device-free occupancy detection system based on the inspiration we obtained from the observations described in Section III-C. Fig. 8 illustrates the system, which consists of: 1) training and 2) detection phase. Both training and detection phase use the fingerprint vector obtained during the data acquisition for training and detection. The mathematical notations used throughout this paper are summarized in Table I.

A. System Model

The occupancy detection can be regarded as a binary classification problem with $\mathcal{Y} \in \{0, 1\}$, where 1 indicates the presence of an occupant and 0 the absence. Given N zones over a large location, the occupancy detection can be easily extended to multinomial classification. Let $\mathcal{Z} = \{z_i | 0 < i \leq N\}$

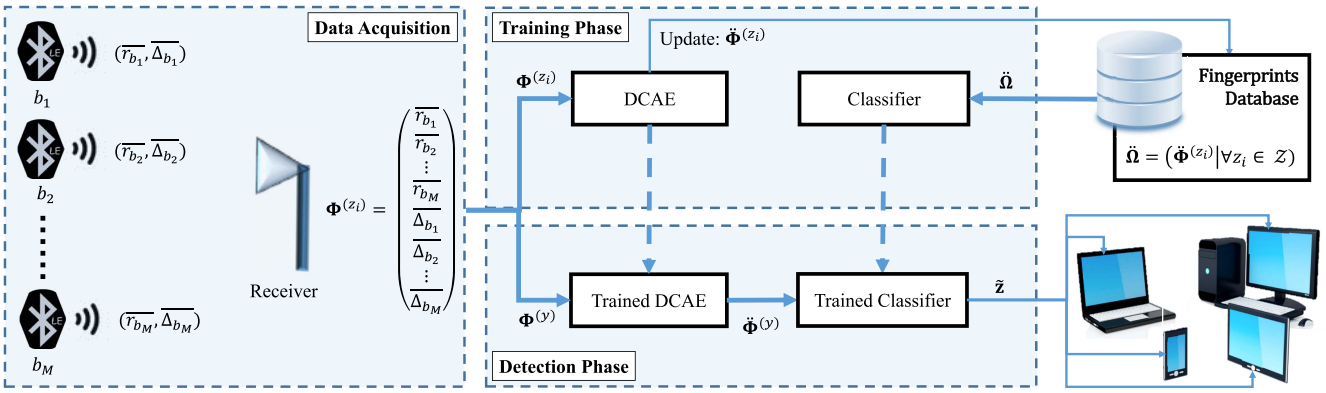

Fig. 8. Our device-free occupancy detection system consists of the following two major phases: a training phase for learning a robust fingerprint, and a detection phase for delivering real-time occupancy detection.

be a set of zones, the occupancy detection in a given zone is a mapping function that maps $\mathcal{Z}$ to $\mathcal{Y}$, more specifically, $g : \mathcal{Z} \rightarrow \mathcal{Y}$. The output is an N-dimensional occupancy vector $\mathbf{z}$, which can be described as follows:

$$\mathbf{z} = \begin{pmatrix} g(z_1) & g(z_2) & \cdots & g(z_i) & \cdots & g(z_{N-1}) & g(z_N) \end{pmatrix}^T. \tag{1}$$

When there is only one occupant in a location, $\mathbf{z}$ will consists of only one 1 and $N-1$ zeros because it is impossible for an occupant to appear in two different zones at the same time. Similarly, when there are n occupants in a location, $\mathbf{z}$ should have n ones and $N-n$ zeros. Suppose that the observed fingerprint vector $\mathbf{\Phi}^{(y)}$ is based on M beacons, the relationship between $\mathbf{z} \in \{0, 1\}^N$ and $\mathbf{\Phi}^{(y)} \in \mathbb{R}^M$ can then be modeled as

$$\mathbf{\Phi}^{(y)} = \Omega \mathbf{z} + \epsilon \tag{2}$$

where ϵ is the noise vector and $\Omega \in \mathbb{R}^{M \times N}$ is the fingerprint matrix, in which each column of Ω denotes the fingerprint vector $\mathbf{\Phi}^{(z_i)} \in \mathbb{R}^M$ for zone z_i. The objective here is to find $\mathbf{z}$ given $\mathbf{\Phi}^{(y)}$ and Ω.

However, as discussed in Sections III-A and III-B, solving $\mathbf{z}$ can be challenging especially when $\mathbf{\Phi}^{(z_i)}$ is: 1) *noisy* and 2) *sparse*. Furthermore, if we merely construct the fingerprint vector with the RSS values from M beacons, the system might be unable to capture the subtle change in the signal behavior for device-free occupancy detection. Hence, this paper exploits the RSS differences besides the RSS values in constructing the fingerprint vector. The next section first describes the general data acquisition process, before presenting our proposed fingerprint vector.

B. Data Acquisition

Suppose that the advertising interval T_a of a beacon is 100 ms, then a receiver with scanning duration $T_s = 1$ s should able to receive at least ten packets during the data acquisition process. Upon receiving the packet, the receiver can measure the RSS value and at the same time identify the beacon based on the MAC address and the device name encapsulated inside the beaconing packet. Being able to identify the beacon is very important in ensuring the fingerprint vector is constructed in proper order.

Let $\mathcal{B} = \{b_j | 0 < j \leq M\}$ be a set of beacons and each beacon has the same T_a, then at least $10 \times M$ packets should have arrived at the receiving end after 1 s, assuming no packet loss. Generally, the fingerprint vector is constructed by averaging the RSS values over T_s. Let $\mathcal{R} = \{r_{b_j}(t) | 0 < t \leq T_s, b_j \in \mathcal{B}\}$ be a set of instantaneous RSS values acquired over T_s at zone z_i, the taRSS would be $\overline{r_{b_j}}^{(z_i)} = (1/|\mathcal{R}_{b_j}|) \sum_{r_{b_j} \in \mathcal{R}_{b_j}} r_{b_j}(t)$, where $\mathcal{R}_{b_j} \subset \mathcal{R}$ contains a list of r from beacon b_j. By rearranging $\overline{r_{b_j}}^{(z_i)}, \forall b_j \in \mathcal{B}$ according to the order defined by $\mathcal{B}$, the fingerprint vector at zone i can be obtained

$$\mathbf{\Phi}^{(z_i)} = \begin{pmatrix} \overline{r_{b_1}}^{(z_i)} & \overline{r_{b_2}}^{(z_i)} & \ldots & \overline{r_{b_M}}^{(z_i)} \end{pmatrix}^T. \tag{3}$$

The dimensionality of $\mathbf{\Phi}^{(z_i)}$ should be the same as the cardinality of set $\mathcal{B}$ if only taRSS is considered. In this paper, besides taRSS we also exploit tdRSS (i.e., $\Delta = (r(t) - r(t^-))^2$) in constructing the fingerprint vector. Similarly, we compute the average of Δ given the set $\mathcal{R}$ recorded over T_s, i.e.,

$$\overline{\Delta_{b_j}}^{(z_i)} = \frac{1}{|\mathcal{R}_{b_j}| - 1} \sum_{r_{b_j} \in \mathcal{R}_{b_j}} \left(r_{b_j}(t) - r_{b_j}(t^-) \right)^2. \tag{4}$$

By appending (3) with vector $\overline{\Delta} \in \mathbb{R}^M$, we will have a $2M$-dimensional fingerprint vector as follows:

$$\mathbf{\Phi}^{(z_i)} = \begin{pmatrix} \overline{r_{b_1}}^{(z_i)} & \overline{r_{b_2}}^{(z_i)} & \ldots & \overline{r_{b_M}}^{(z_i)} & | & \overline{\Delta}^{(z_i)} \end{pmatrix}^T \tag{5}$$

where $\overline{\Delta}^{(z_i)} = \begin{pmatrix} \overline{\Delta_{b_1}}^{(z_i)} & \overline{\Delta_{b_2}}^{(z_i)} & \ldots & \overline{\Delta_{b_M}}^{(z_i)} \end{pmatrix}^T$.

Hence, the fingerprint matrix Ω should now be a $2M \times N$ matrix instead of $M \times N$ as described in (2). Similarly, the dimension of $\mathbf{\Phi}^{(y)}$ should be $2M$. During the training phase, the receiver is configured to have a much longer T_s such that the system can observe all the M beacons for fingerprint construction. On the other hand, T_s should be as short as possible for real-time detection purposes. Such a short T_s can be another problem leading to sparse observation. Furthermore, the observed fingerprint vector during the detection phase might be distorted by the environmental noise, and causes performance degradation if we simply match the observed fingerprint with a list of registered fingerprints in the database.

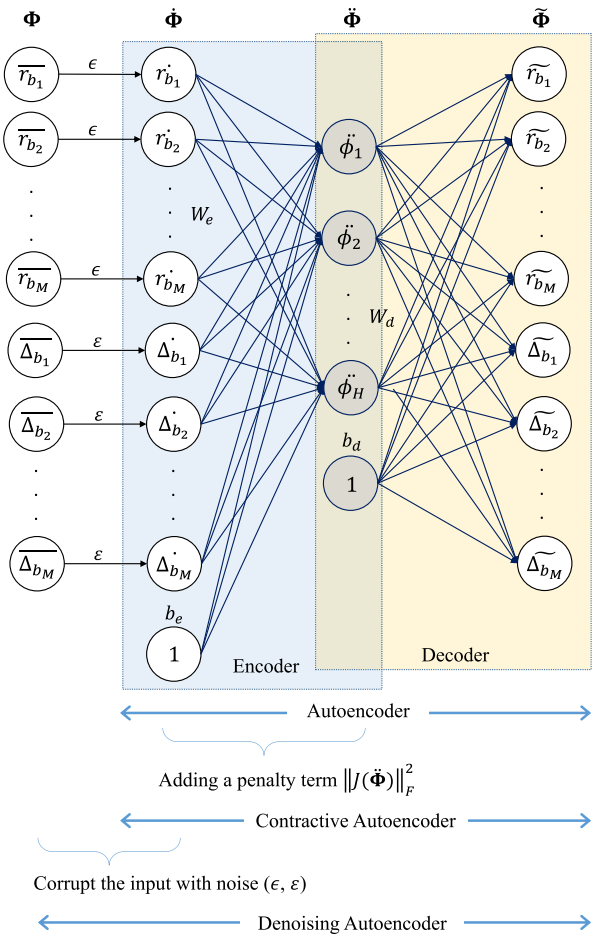

Fig. 9. Our proposed DCAE jointly addresses the noise and sparsity issues by performing denoising and contractive training on the same NN.

In the next section, we present our proposed DCAE which can learn a robust fingerprint representation from the noisy and sparse fingerprint inputs.

V. Denoising-Contractive Autoencoder

This section first describes the general autoencoder before presenting our proposed DCAE. Lastly, we describe the classification process to detect the zone occupied by an occupant based on the fingerprint vector reconstructed by our DCAE.

A. General Loss Function of Autoencoder

Autoencoder is an unsupervised NN, which learns to reconstruct the output given the input data. In our context, the input data is the fingerprint vector $\mathbf{\Phi} \in \mathbb{R}^{2M}$, and the autoencoder will learn to reconstruct $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{2M}$ through backpropagation. Such a reconstruction problem is nontrivial by imposing a bottleneck to its hidden layer such that the size of the hidden layer H is much smaller than $2M$.

As illustrated in Fig. 9, the autoencoder consists of two parts: 1) encoder and 2) decoder. The encoder learns a hidden representation given the input data, whereas the decoder reconstructs the output data from the hidden representation. Given the fingerprint vector $\mathbf{\Phi}$, the encoder and decoder can be described as follows:

$$\begin{aligned} \ddot{\mathbf{\Phi}} &= \sigma(W_e \mathbf{\Phi} + b_e) \\ \tilde{\mathbf{\Phi}} &= \sigma\left(W_d \ddot{\mathbf{\Phi}} + b_d\right) \end{aligned} \tag{6}$$

where $W_e \in \mathbb{R}^{H \times M}$ and $b_e \in \mathbb{R}^H$ is the weight and bias, respectively, learned by the encoder and $W_d \in \mathbb{R}^{M \times H}$ and $b_d \in \mathbb{R}^M$ is the weight and bias learned by the decoder. $\sigma(\cdot)$ is the nonlinear activation function, which can be either a sigmoid or tanh function. The dimension of the encoded vector $\ddot{\mathbf{\Phi}} \in \mathbb{R}^H$ is always smaller than the dimension of the fingerprint vector $\mathbf{\Phi} \in \mathbb{R}^M$. The loss function for a general autoencoder is defined as

$$\mathcal{L}\left(\mathbf{\Phi}, \tilde{\mathbf{\Phi}}\right) = \|\tilde{\mathbf{\Phi}} - \mathbf{\Phi}\|_2^2. \tag{7}$$

The general autoencoder is useful in learning a compressed representation. However, it might be unable to deal with noisy and sparse input. In other words, the reconstruction error increases when the input is noisy and sparse.

B. DCAE for Noisy and Sparse Input

Fig. 9 depicts the NN architecture of our proposed DCAE. Note that our DCAE uses the same encoder and decoder function as the general autoencoder described by (6). However, instead of directly using the input fingerprints $\mathbf{\Phi}$, we impose some noise $(\epsilon, \varepsilon) \sim \mathcal{N}(0, 1)$ to vector $\mathbf{\Phi}$. Hence, the encoder function learns the encoded vector based on $\dot{\mathbf{\Phi}} = \mathbf{\Phi} + (\epsilon, \varepsilon)$ instead of $\mathbf{\Phi}$. Such an approach is similar to the conventional DAE.

However, we added a penalty term to the loss function to ensure that the learned hidden representation (i.e., the encoded vector) is less sensitive to the input variations. The penalty term is the same as the one introduced by the conventional CAE. While our DCAE is based on a similar approach to DAE and CAE, it is not obvious to simply integrate them for dealing with the noisy and sparse input at the same time. The common approach to combine DAE and CAE is by training each model in a sequential manner; however, such a sequential approach might increase the training cost and at the same time fail to generalize to the input that has both noise and sparsity issues.

Instead, our DCAE jointly considers the noise and sparsity issues by defining the following loss function:

$$\begin{aligned} \mathcal{L}\left(\mathbf{\Phi}, \tilde{\mathbf{\Phi}}\right) &= \|\tilde{\mathbf{\Phi}} - \dot{\mathbf{\Phi}}\|_2^2 + \lambda \|J\left(\ddot{\mathbf{\Phi}}\right)\|_F^2 \\ &= \|\sigma\left(W_d \dot{\mathbf{\Phi}} + b_e\right) - \mathbf{\Phi}\|_2^2 + \lambda \|J\left(\tilde{\mathbf{\Phi}}^{(t)}\right)\|_F^2 \\ &= \|\sigma\left(W_d \sigma\left(W_e \dot{\mathbf{\Phi}} + b_e\right) + b_d\right) - \mathbf{\Phi}\|_2^2 + \lambda \|J\left(\ddot{\mathbf{\Phi}}\right)\|_F^2 \end{aligned} \tag{8}$$

where the decoder output $\tilde{\mathbf{\Phi}}$ is reconstructed with respect to the noisy sample $\dot{\mathbf{\Phi}}^{(t)}$ instead of $\mathbf{\Phi}$. The second penalty term is the Frobenius norm of the jacobian matrix $\|J(\ddot{\mathbf{\Phi}}^{(t)})\|_F^2$ with regulation parameter λ. $\|J(\tilde{\mathbf{\Phi}}^{(t)})\|_F^2$ can be calculated as follows:

$$\|J\left(\tilde{\mathbf{\Phi}}^{(t)}\right)\|_F^2 = \sum_{mh} \left(\frac{\partial \dot{\phi}_h}{\partial \tilde{\phi}_m^{(t)}} \right)^2. \tag{9}$$

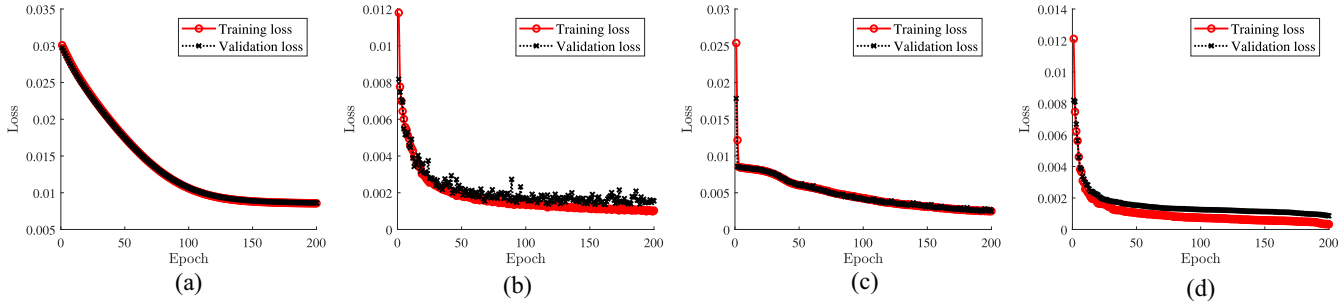

Fig. 10. Training and validation loss achieved by (a) SGD, (b) RMSprop, (c) Adadelta, and (d) L-BFGS optimizers.

Intuitively, we would like to make sure that the learned hidden input is less sensitive to the input variations. With our proposed DCAE, we can simply train the network with a single loss function, rather than stacking up the DAE and CAE.

C. Optimization Methods

Different optimization methods can be used to minimize the loss function described by (8). We used a small dataset, which consists of beacon data collected in a room environment, to validate the performance of each optimization method before adopting the method for extensive experiments. Four optimization methods were tested: 1) stochastic gradient descent (SGD); 2) root mean square propagation (RMSprop); 3) adaptive learning rate (Adadelta); and 4) limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS).

We divided the data into training and validation set, and Fig. 10 shows the training and validation loss achieved by all the four optimizers. L-BFGS has the shortest running time among the four optimizers. In general, L-BFGS is preferable when the computer has limited memory. Considering we will also run the training on a local computer equipped with a standard CPU, we opt to adopt L-BFGS for the training. However, it seems that the training and validation loss start to diverge after 20 epochs. Hence, we might have to tune the parameters or reduce the training epochs to avoid overfitting.

D. Classification

Given the reconstructed fingerprint, we can apply any classifier to solve the multinomial occupancy detection problem. The common approach is to add a softmax layer to the decoder output for classification purposes. However, softmax is not the only choice. As illustrated in Fig. 8, our ultimate goal is to train a classifier, which can be any legacy classifier, to classify whether or not a zone is occupied. Without loss of generality, the objective of the classifier can be described as follows:

$$y = \arg\max_{z_i \in \mathcal{Z}} P_c\left(\tilde{z} = z_i | \tilde{\mathbf{\Phi}}^{(y)}\right) \tag{10}$$

where $P_c(\cdot)$ returns the probability for each zone based on the black box classification function c. In Section VI, we present the performance evaluation of different classifiers including multilayer perceptron (MLP), k nearest neighbor (kNN), SVM, and naive Bayes (NB).

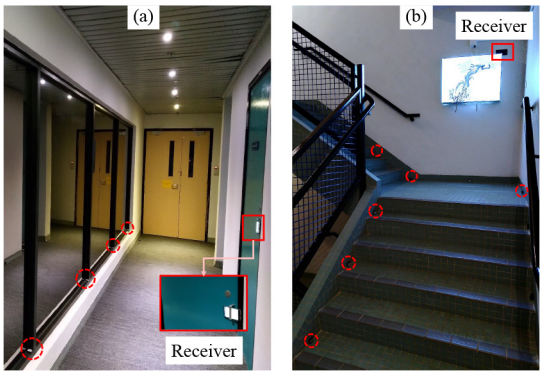

Fig. 11. Total of 16 beacons (marked with dotted line circles) were deployed in both of our experimental testbeds. (a) Hallway. (b) Staircase.

VI. Experimental Evaluations and Results

This section describes the setting of the experimental testbeds for data collection and then discusses the detection performance achieved by the proposed DCAE.

A. Data Collection

We collected the data from two different experimental testbeds: 1) the hallway and 2) the staircase, as shown in Fig. 11(a) and (b). There were a total of 16 beacons deployed in each testbed. The location of each beacon is indicated with the Bluetooth symbol, as shown in Fig. 12. We used an Android phone to act as a receiver since it is the commonly available receiver to collect the beacon data. A data collection App was developed and installed for the data collection purpose. During the collection process, the App logs the following information: beacon MAC address, RSS value, timestamp, and data payload. The collected data were saved in the local phone storage in ".csv" format. For each testbed, we divided the location into 32 different smaller zones. As depicted in Fig. 12, these zones were marked with the "x" marker. The user was required to indicate their presence in each zone before we start the data collection process remotely. Approximate 800+ set of fingerprints were collected in each zone, amounting to a total of 40k sets of fingerprints.

Note that we only required the effort from a user to notify us of his/her location during the remote data collection. However, the collected data can be manipulated to reflect the presence of multiple occupants in multiple zones. As illustrated in Fig. 13, the presence of the occupant at different zones will

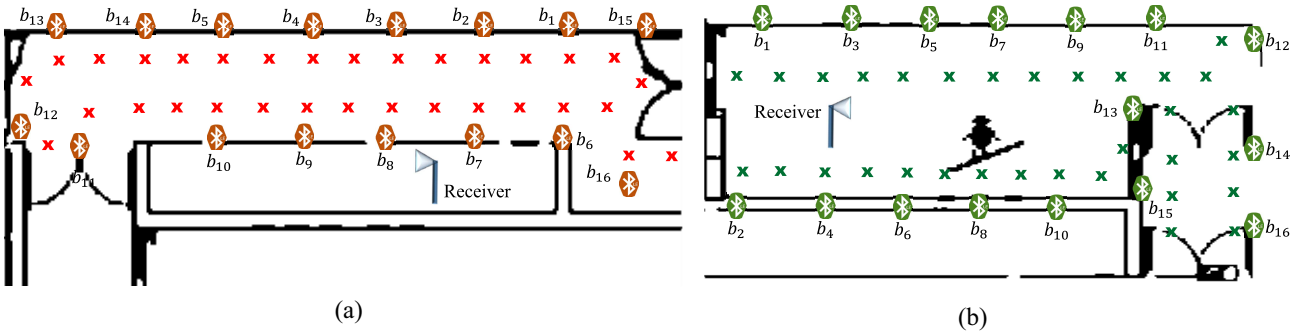

Fig. 12. Bluetooth symbols indicate the location of the deployed beacons, and the "x" markers indicate the zones. (a) Hallway. (b) Staircase.

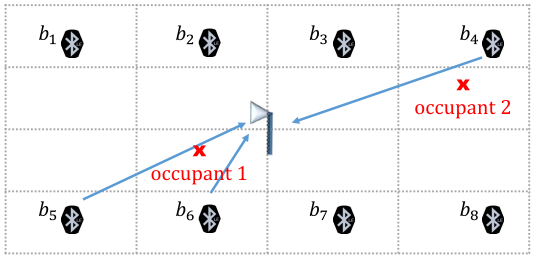

Fig. 13. Presence of occupant 1 will cause a change in the RSS values measured from b_5 and b_6; whereas the presence of occupant 2 will change the RSS value measured from b_4.

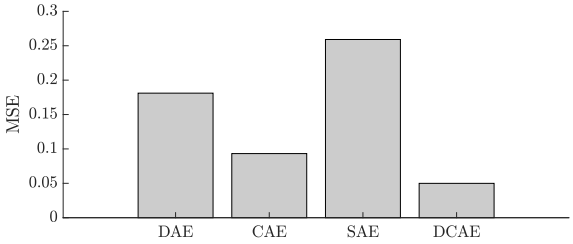

Fig. 14. Reconstruction error of our proposed DCAE in comparison to DAE, CAE, and SAE.

cause different changes to the existing fingerprint. By learning these subtle changes, we can then infer the current zone occupied by the user/animal. Similarly, when there are multiple occupants, we can directly infer their zones by observing the changes on the RSS fingerprint. Note that the layout of the deployed beacons shown in Fig. 13 is just for quick illustration, in practical scenario, the beacons can be randomly deployed at any location. One interesting observation is that different body composition might have different effects on the RSS measurements. Hence, one possible future work could be the identification between human and animal, or even between male and female.

B. Training and Testing

In total, there are 40k sets of fingerprints from a total of 64 zones. Given these sets of fingerprints, we divided them into 80% training and 20% testing set. We used the tensorflow framework to build our model, and the model was trained with both local computer and cloud computer. The local computer is equipped with a single core CPU (hyperthreaded Xeon Processors @2.3 GHz), whereas the cloud computer provides us with a GPU (1xTesla K80, 2496 CUDA cores, 12-GB GDDR5 VRAM) for the training process.

Furthermore, we build the model using state-of-the-art autoencoders, including DAE, CAE, and SAE. DAE has been used by Xiao *et al.* [23] for 3D localization, whereas SAE has been used by [35]. Mean squared error (MSE) was used to compute the reconstruction error, and the result achieved by each autoencoder is shown in Fig. 14. It is clear that our proposed DCAE suffers the least reconstruction error (i.e., around 0.047) as compared to the rest. Using the reconstructed output, we trained different classifiers for occupancy detection.

C. Results and Discussion

We evaluate the accuracy of each classifier by measuring the number of times the classifier returns the correct zone of occupants with the total number of tests. Besides evaluating the accuracy of each classifier with different autoencoders using the original 20% testing set, we added the noise factor and sparsity ratio to the testing set for further evaluation.

1) Detection Accuracy Versus Noise Factor: The noise factor is added by randomly drawing a value from the normal distribution $\mathcal{N}(0, 1)$. We varied the noise factor from 0.1 to 20, and the detection accuracy achieved by each classifier is shown in Fig. 15. The black dotted line indicates the performance of the classifier with the fingerprint vector constructed with raw RSS values. In general, the detection accuracy decreases when the noise factor increases. However, we can achieve a substantial performance improvement when applying the classifier with the fingerprint learned by the autoencoders.

Among all the autoencoders, our proposed DCAE achieves the best performance. Even though all the four classifiers have shown almost identical performance, kNN somehow outperforms the others with better detection accuracy. One possible reason is that the fingerprints learned by the 80% training data groups the fingerprints from the same zone into the same neighborhood boundary, in which, by casting the vote based on the top k neighbors, kNN is able to detect the zone correctly comparing to other classifiers.

2) Detection Accuracy Versus Sparsity Ratio: We evaluated the performance of each classifier by varying the sparsity

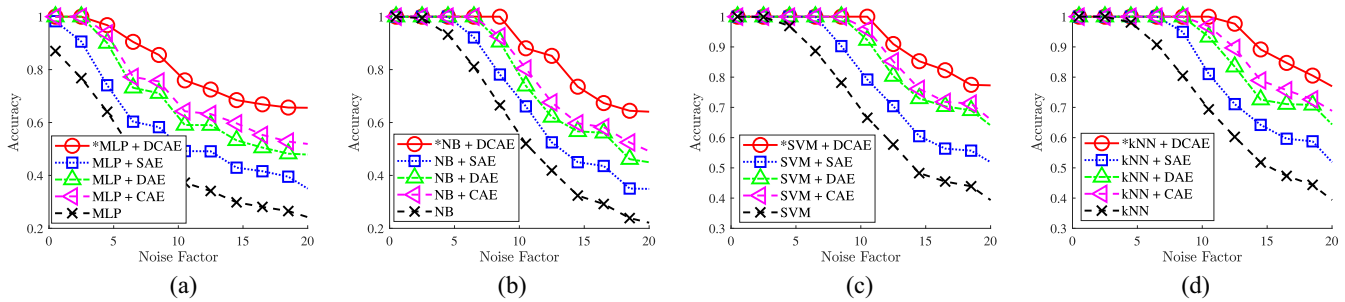

Fig. 15. Detection accuracy achieved by SAE, DAE, CAE, and our proposed DCAE with (a) MLP, (b) NB, (c) SVM, and (d) kNN classifiers, with respect to the noise factor.

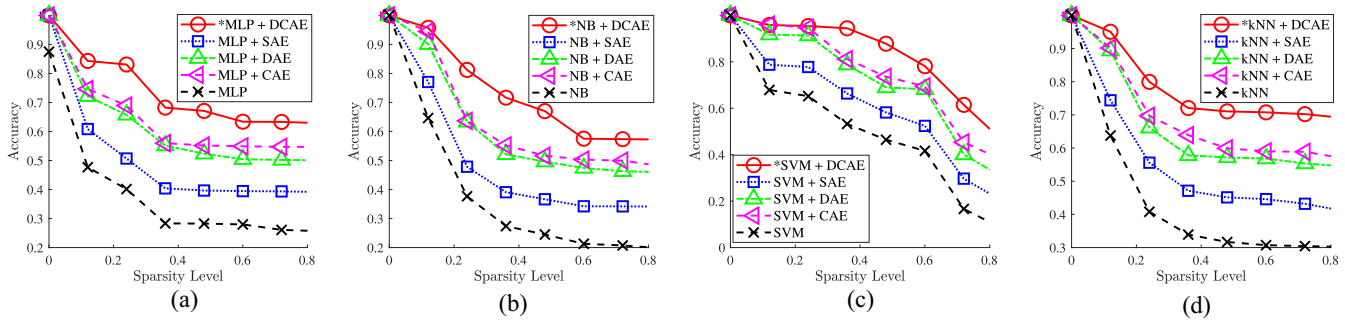

Fig. 16. Detection accuracy achieved by SAE, DAE, CAE, and our proposed DCAE with (a) MLP, (b) NB, (c) SVM, and (d) kNN classifiers, with respect to the sparsity ratio.

ratio. We define the sparsity ratio by taking the ratio between $\|\dim(\mathbf{\Phi})-\mathbf{\Phi}\|_0$ and $\dim(\mathbf{\Phi})$, where $\dim(\mathbf{\Phi})$ denotes the dimensionality of the fingerprint vector and $\|\mathbf{\Phi}\|_0$ returns the total number of nonzero elements of a given vector. Note that the sparsity ratio is bounded between $(0, 1]$. In other words, it is impossible to have a sparsity ratio equal to 1 because this simply means that the receiver observes nothing.

Hence, we varied the sparsity ratio from 0 to 0.8, and the results achieved by each classifier are shown in Fig. 16. Similarly, the detection performance decreases when the sparsity ratio increases. The performance can be enhanced by applying the autoencoder to reconstruct the sparse fingerprint vector. Again, our proposed DCAE outperforms the rest with good detection performance. SVM achieves better performance among all the classifiers. In general, the trained SVM can draw a better decision boundary with a sparse input vector in comparison to the rest.

VII. Implementation for Online Detection

To proceed with online detection, we developed a real-time database using Firebase platform to monitor the fingerprints at the two testbeds described above. More precisely, the on-site receiver will measure fingerprints and update to the online database; whereas the remote computer will retrieve these fingerprints to infer if there were any occupants. Fig. 17 shows the screenshot of the detection output on the remote computer when there is no one on any of the zones. Using the implemented system, we further performed a series of experiments for the following two cases: 1) single moving target and 2) multiple moving targets.

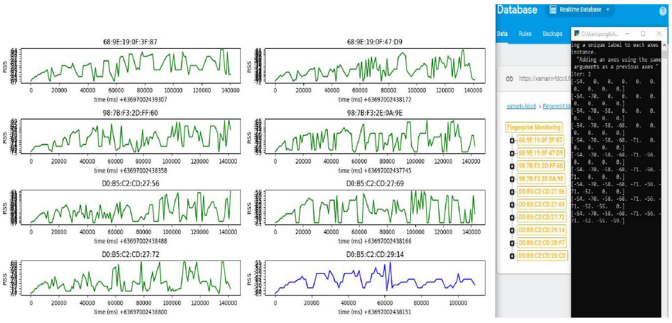

Fig. 17. Screenshot of the monitoring system by the remote computer: the left-hand side shows the retrieved RSS measurements from the database and the right-hand side shows the console output in computing the RSS measurements to identify the occupancy.

A. Case 1: Wandering Around the Hallway

We consider a scenario which always happens in a private home setting. That is, a user wandering around the hallway when they were puzzled. During the experiment, the user was required to randomly move around the zone every 1 min, for a total of 30 min. A form was provided for the user to jot down their current zone every 1 min. After that, we can measure the performance by comparing the information in that form with the detection output returned by the remote computer. We repeated the same experiment procedures with ten different users, and each user was required to repeat the experiment for two times. Fig. 18 shows one of the examples of the zone visited by the user at every minute. It is clear that our occupancy detection is able to output a correct zone for almost 99% of the time. However, the remote computer only able to

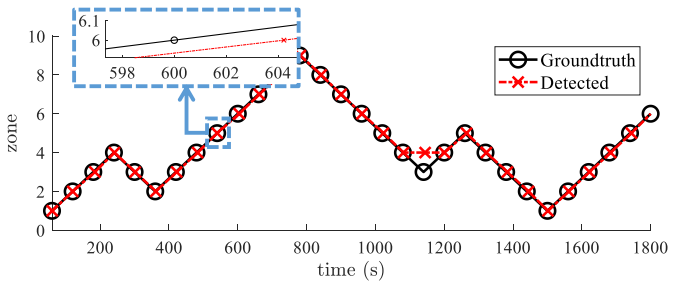

Fig. 18. Black curve indicates the zone jotted down by the user, whereas the red curve indicates the detection output observed from the remote computer.

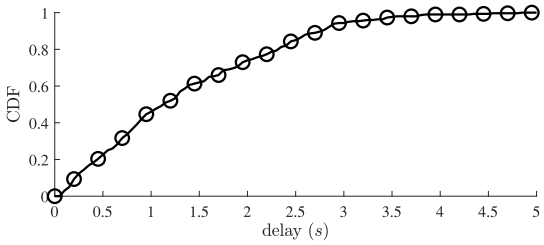

Fig. 19. CDF plot shows that the system took no more than 3 s (for at least 80% of the time) to return a correct zone.

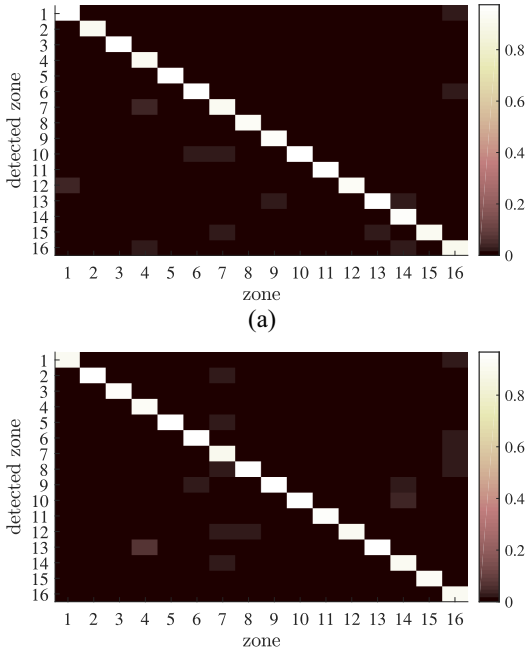

Fig. 20. (a) Upper confusion matrix shows the detection accuracy for the first user, and (b) the lower confusion matrix for the second user.

produce the detection output after a certain amount of delay. As we zoom into each point, we can see that it took about 1–3 s for the system to detect the zone. We computed the probability of delay by consolidating the delay from all the 20 experiment data (ten users × two times). Fig. 19 shows that for at least 80% of the time, the delay is generally less than 3 s.

B. Case 2: Hopping on the Staircase

Multiple moving targets are also a common scenario in many private home setting, for example, children like to play on the staircase. To emulate the above scenario, we required two users to randomly hop around the zone every 1 min. We used confusion matrices to evaluate the detection accuracy, as shown in Fig. 20. The x-axis of the confusion matrix indicates the zone, whereas the y-axis indicates the zone detected by the remote computer. The detection accuracy can be computed by averaging the diagonal elements of the confusion matrix. Note that the result can be generalized to the case with n targets. From the two confusion matrices, we can see that the detection accuracy drop a bit compared to the case of a single user. This can be explained by the fact that when the two users were in close proximity, the changes on the RSS fingerprint is not obvious, and hence, the system only able to detect the occupancy of a user and unable to detect the presence of the other user. Note that such an issue can be addressed by installing extra receiver at different locations such that this extra receiver can detect the changes on RSS fingerprint from another angle. Overall, the installation planning for the receivers is very much dependent on the size of the private location and the number of occupants we would like to monitor. For most private setting such as washroom, a single receiver should be sufficient to provide very good detection performance.

VIII. Conclusion

This paper presents device-free occupancy detection by learning a robust fingerprint vector with our proposed DCAE. The contributions of our device-free occupancy detection are twofold: first, we construct the fingerprint vector by appending taRSS with tdRSS and second, our proposed DCAE can jointly deal with the two common issues faced by most RF fingerprint approaches, i.e., noise and sparsity. To evaluate our device-free occupancy detection with BLE beacons, we collected a large-scale dataset from a total of 64 zones. About 800+ sets of fingerprints were collected, amounting to a total of 40k sets of fingerprints. Extensive experiments with different classifiers validate the superiority of our proposed DCAE in learning a robust representation. Lastly, we demonstrate the feasibility of our proposed approach with practical implementation. There are a few possible future works inspired by the practical experiment. For example, a beacon-receiver deployment strategy can be devised to optimized the detection performance for different use cases. Furthermore, we can also implement the training process on the receiver and then update the trained model to the cloud instead of relying on the cloud for training. The benefit of on-receiver training is that each receiver has a better picture of the fingerprint rather than the cloud, hence, on-receiver training can provide an updated model from time to time.

References

[1] S. He, B. Ji, and S.-H. G. Chan, "Chameleon: Survey-free updating of a fingerprint database for indoor localization," *IEEE Pervasive Comput.*, vol. 15, no. 4, pp. 66–75, Oct./Dec. 2016.

[2] S.-H. Fang, Y.-T. Hsu, and W.-H. Kuo, "Dynamic fingerprinting combination for improved mobile localization," *IEEE Trans. Wireless Commun.*, vol. 10, no. 12, pp. 4018–4022, Dec. 2011.

[3] K. Lin, M. Chen, J. Deng, M. M. Hassan, and G. Fortino, "Enhanced fingerprinting and trajectory prediction for IoT localization in smart buildings," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1294–1307, Jul. 2016.

[4] S. He, S.-H. G. Chan, L. Yu, and N. Liu, "Fusing noisy fingerprints with distance bounds for indoor localization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2506–2514.

[5] P. C. Ng, J. She, and R. Ran, "Towards sub-room level occupancy detection with denoising-contractive autoencoder," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, 2019, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8761294&isnumber=8761046. doi: 10.1109/ICC.2019.8761294.

[6] A. Filippoupolitis, W. Oliff, and G. Loukas, "Bluetooth low energy based occupancy detection for emergency management," in *Proc. 15th Int. Conf. Ubiquitous Comput. Commun. Int. Symp. Cyberspace Security (IUCC-CSS)*, Dec. 2016, pp. 31–38.

[7] M. Jin, R. Jia, and C. J. Spanos, "Virtual occupancy sensing: Using smart meters to indicate your presence," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3264–3277, Nov. 2017.

[8] M. Papandrea, S. Giordano, S. Vanini, and P. Cremonese, "Proximity marketing solution tailored to user needs," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2010, pp. 1–3.

[9] X. Wu, R. Shen, L. Fu, X. Tian, P. Liu, and X. Wang, "iBILL: Using iBeacon and inertial sensors for accurate indoor localization in large open areas," *IEEE Access*, vol. 5, pp. 14589–14599, 2017.

[10] R. Faragher and R. Harle, "Location fingerprinting with Bluetooth low energy beacons," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, Nov. 2015.

[11] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models," *Energy Build.*, vol. 112, pp. 28–39, Jan. 2016.

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

[13] L. Chang *et al.*, "FitLoc: Fine-grained and low-cost device-free localization for multiple targets over various areas," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 1994–2007, Aug. 2017.

[14] B. Mager, P. Lundrigan, and N. Patwari, "Fingerprint-based device-free localization performance in changing environments," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2429–2438, Nov. 2015.

[15] T. Ekwevugbe, N. Brown, V. Pakka, and D. Fan, "Real-time building occupancy sensing using neural-network based sensor network," in *Proc. 7th IEEE Int. Conf. Digit. Ecosyst. Technol. (DEST)*, Menlo Park, CA, USA, Jul. 2013, pp. 114–119.

[16] N. Anzum, S. F. Afroze, and A. Rahman, "Zone-based indoor localization using neural networks: A view from a real testbed," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.

[17] Q. Lei, H. Zhang, H. Sun, and L. Tang, "Fingerprint-based device-free localization in changing environments using enhanced channel selection and logistic regression," *IEEE Access*, vol. 6, pp. 2569–2577, 2018.

[18] R. Zhou, X. Lu, P. Zhao, and J. Chen, "Device-free presence detection and localization with SVM and CSI fingerprinting," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7990–7999, Dec. 2017.

[19] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proc. ICML Workshop Unsupervised Transf. Learn.*, 2012, pp. 37–49.

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.

[23] C. Xiao, D. Yang, Z. Chen, and G. Tan, "3-D BLE indoor localization based on denoising autoencoder," *IEEE Access*, vol. 5, pp. 12751–12760, 2017.

[24] X. Chen, C. Ma, M. Allegue, and X. Liu, "Taming the inconsistency of Wi-Fi fingerprints for device-free passive indoor localization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Atlanta, GA, USA, May 2017, pp. 1–9.

[25] G.-S. Wu and P.-H. Tseng, "A deep neural network-based indoor positioning method using channel state information," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Mar. 2018, pp. 290–294.

[26] S. He, S.-H. G. Chan, L. Yu, and N. Liu, "SLAC: Calibration-free pedometer-fingerprint fusion for indoor localization," *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 1176–1189, May 2018.

[27] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz, "Bluetooth 5: A concrete step forward toward the IoT," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 125–131, Jul. 2018.

[28] 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.

[29] N. Alsindi, Z. Chaloupka, N. AlKhanbashi, and J. Aweya, "An empirical evaluation of a probabilistic RF signature for WLAN location fingerprinting," *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 3257–3268, Jun. 2014.

[30] X. Tian, Z. Song, B. Jiang, Y. Zhang, T. Yu, and X. Wang, "HiQuadLoc: A RSS fingerprinting based indoor localization system for quadrotors," *IEEE Trans. Mobile Comput.*, vol. 16, no. 9, pp. 2545–2559, Sep. 2017.

[31] 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.

[32] J. Yang, X. Wang, S. I. Park, and H. M. Kim, "Optimal direct path detection for positioning with communication signals in indoor environments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, Jun. 2012, pp. 4798–4802.

[33] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe, "ALPS: A Bluetooth and ultrasound platform for mapping and localization," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2015, pp. 73–84.

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

[35] W. Zhang, K. Liu, W. Zhang, Y. Zhang, and J. Gu, "Deep neural networks for wireless localization in indoor and outdoor environments," *Neurocomputing*, vol. 194, pp. 279–287, Jun. 2016.