RF Sensing-enabled Interference Mitigation for Wi-Fi-based IoT Systems: A Deep Learning Approach

ChunChih Lin*, Gen Li*, Xiaonan Zhang†, and Linke Guo*
*The Holcombe Department of Electrical and Computer Engineering, Clemson University
†Department of Computer Science, Florida State University
Email: {chunchi, gen, linkeg}@clemson.edu, xzhang@cs.fsu.edu

2024 IEEE Conference on Communications and Network Security (CNS) | 979-8-3503-7596-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/CNS62487.2024.10735625

***Abstract*—The proliferation of the Internet of Things (IoT) over the 2.4GHz has transformed the way people interact with the world in their daily lives. For emerging applications requiring diverse types of data, using IoT devices from different protocols, such as Wi-Fi, Bluetooth, and ZigBee, for data collection will help fulfill designated objectives. Hence, the increasing use of these IoT devices will form a heterogeneous environment with inevitable wireless coexistence. Due to the dynamic IoT environment, the wireless transmission among heterogeneous IoT devices becomes unknown and notoriously hard to manage, where the network performance is always compromised. This paper leverages the RF sensing information to develop a deep learning-based framework for mitigating the cross-technology interference of the Wi-Fi-based IoT system. Each IoT device can provide detailed Channel State Information (CSI) of the transmission link, by which the gateway uses Generative Adversarial Networks (GAN) for building a detailed environmental RF map. Then, we propose a CSI-inspired gateway topology management strategy to find the optimal gateway location, anticipating reaching the highest network throughput. To enhance the efficiency of the proposed scheme, we also adopt the advanced differentiable augmentation approach to reduce the training data size for gateway deployment. Extensive experimental results have demonstrated both the feasibility and efficiency of our design.**

I. Introduction

Wide deployment of the Internet of Things (IoT) is reshaping the way we interact with the world, as evidenced by innovations such as wearable computing in healthcare, intelligent controls of home appliances, and real-time machinery health monitoring for smart manufacturing. For an application requiring diverse types of data, leveraging IoT devices with different wireless technologies to fulfill the objective becomes a must [1], whereby a heterogeneous wireless environment is a reality. Current mainstream IoT protocols, such as Wi-Fi, ZigBee, and Bluetooth/Bluetooth Low Energy (BLE), heavily overlap the Industrial, Scientific, and Medical (ISM) 2.4GHz bands. The prevalence of these wireless coexistence among heterogeneous IoT devices [2], [3] has congested the spectrum [4], mainly due to the infeasibility of perfectly handling the interferences.

Although the wireless coexistence has been extensively studied over a decade [5]–[8], the question of "*how to enhance the network performance with the presence of interference*" still remains. Taking Wi-Fi and ZigBee coexistence on 2.4GHz as an example, concurrent communication [5]–[8] achieves simultaneous data transmission by coordinating transmission power, time slots, and sub-channels for devices with different protocols. Unfortunately, this high-level coordination always requires precise synchronization among all devices, which is less practical and cannot perfectly handle the interference caused by unknown co-existence devices.

Recent works on ubiquitous RF sensing [9]–[11] have shown that the dynamics of the wireless channel can be captured by Channel State Information (CSI) and Received Signal Strength Indicator (RSSI) that are estimated from normal data transmissions. Instead of passively resolving system degradation, this paper will explore data-driven approaches to proactively enhance Wi-Fi-based IoT network performance by adjusting the network topology. Specifically, the existence of unknown interferences, e.g., cross-technology interferences (CTI), always deteriorates the network throughput [12]. We will exploit RF sensing to obtain channel status changes due to CTI, then seek the optimal network topology of gateway deployment for CTI mitigation. To find the best location where the channel status is unknown, we will explore Generative Adversarial Networks (GAN) for RF sensing data augmentation.

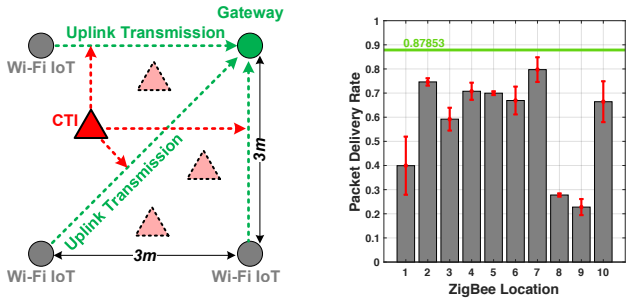
Our main contributions are listed as follows,

- To the best of our knowledge, we are the first to build an explicit mapping between the received CTI-contaminated CSI and the expected network throughput via network topology changes.
- Following the idea in the computer vision domain, we develop a CSI-inspired Deep Convolutional GAN for CSI data augmentation, anticipating generating a large-scale synthetic CSI sample used for determining the optimal network topology.
- To enhance the efficiency of the proposed GAN-based CSI data augmentation, we are the first to adopt the most advanced differentiable augmentation (DiffAugment) method for balancing the training costs and localization prediction accuracy.
- We collect a large dataset consisting of more than 2×10^8 CSI samples from 30 locations on a 3m×3m area, in which CSI samples from 12 different ZigBee interference locations have been incorporated. This real-world dataset has validated the feasibility of the proposed design.

II. Motivation

To verify how the CTI compromises the network work performance, we conduct an experiment consisting of three

Wi-Fi 802.11n IoT devices and a gateway, where Fig. 1a demonstrates the experiment layout (outdoor). We randomly select 10 positions to deploy a ZigBee device as a CTI source on the experiment plane and evaluate the Wi-Fi uplink throughputs from three Wi-Fi IoT devices (500 packets 5 times per device) while the ZigBee device continuously sends 802.15.4 packets. The average packet delivery rate can reach over 87% without the CTI. As a comparison in Fig. 1b, however, the existence of CTI compromises the network throughput at each of the deployed locations, indicating that deploying the gateway at the current location cannot serve these IoT devices with the maximum packet delivery rate.



(a) Exp. Layout (b) Wi-Fi Throughput

Fig. 1: **Network Performance w/ CTI**

Based on the theoretical analysis, the existence of CTI will not only alter the result of CSI estimation but also largely hamper the network throughput. Hence, the basic idea of this paper is to **analyze the CSI sequences to derive the optimal deployment location of the gateway**. Although this RF sensing result is highly location-dependent, there are three challenges ahead of finding the optimal location,

- **Challenge (i)**: The infeasibility of having all CSI sequences from all locations (not enough data);
- **Challenge (ii)**: The mapping between CTI-contaminated CSI and gateway location remains unknown;
- **Challenge (iii)**: The lack of an explicit relationship between Wi-Fi throughput and received CSI.

III. CSI-inspired Deep Learning-based CTI Mitigation

A. Design Overview

Our CTI mitigation scheme consists of three major building blocks as shown in Fig. 2. Although the CSI contains abundant information for gateway localization, it is impossible to collect CSI from all possible locations to find the optimal position. Hence, to tackle **Challenge (i)**, we construct a Deep Convolutional Generative Adversarial Network (DCGAN) data augmentation scheme for uplink CTI-contaminated CSI, anticipating to generate a set of synthetic CSI samples that satisfy the distribution of ground truth but cannot be retrieved in practice. Then, the synthetic CSI samples will be fed into two parallel modules, one for CSI-based gateway localization and one for deriving the highest achievable throughput. For **Challenge (ii)**, a CNN model will be trained to investigate the mapping between the ground truth CSI and the corresponding location. To tackle **Challenge (iii)**, we propose to use a one-class support vector machine (OCSVM) in order to evaluate whether the Wi-Fi packet via which the synthetic CSI sequences are estimated can be received. Finally, a grid-based location selection algorithm will be developed for the optimal gateway location, predicts the higher network throughput.

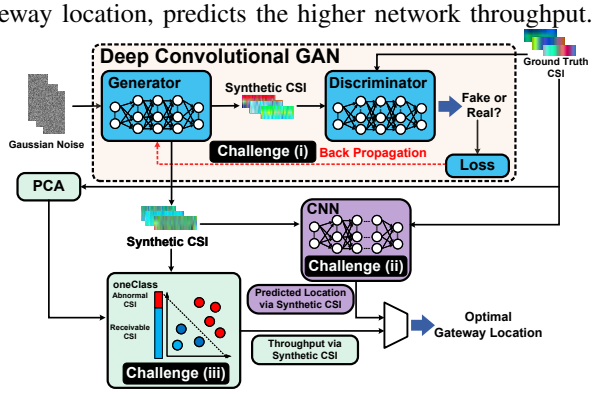


Fig. 2: **CSI-Inspired Deep Learning Architecture**

B. CSI-inspired GAN Design

First of all, to tackle **Challenge (i)**, a naïve way is to traverse all possible locations as in [13], and evaluate the overall throughput at each location, which is apparently infeasible for the implementation in large-scale IoT systems. Hence, we design a Generative Adversarial Network (GAN)-based data augmentation scheme for uplink CTI-contaminated CSI, anticipating generating a set of synthetic CSI samples that satisfy the distribution of ground truth but cannot be retrieved in practice. Following the design of GAN in the computer vision domain, the CSI sample data can also be seen as an "image" with a 2-dimensional matrix $\mathbf{C}$, where the element $C(n,k)$, where $n = 1, 2, \cdots, N$, and $k = 1, 2, \cdots, K$ are the n-th antenna and the k-th subcarrier. N is the number of the antenna, and K denotes the number of subcarriers at the Wi-Fi receiver. For our case, we have $N = 3$, $K = 30$.

To design the GAN, we choose the deep Convolutional Neural Network (CNN) as the generator and discriminator. Suppose the ground truth CSI data $\mathbf{h}$ follows the distribution $\mathbf{h} \sim p_{\text{data}}(\mathbf{h}) \in \mathbf{H}$, where $\mathbf{H}$ is the domain of CSI sequences. Meanwhile, we define the noise input $\mathbf{z} \sim \mathcal{N}(0,1) \in \mathbf{Z}$ follows Gaussian noise distribution. To build the Deep Convolutional Generative Adversarial Networks (DCGANs), a mapping function $G : \mathbf{Z} \rightarrow \mathbf{H}$ is required for the translation from Gaussian noise to CSI sequences. We also define the discriminator, D, and $D(\mathbf{h})$ denotes the probability that $\mathbf{h}$ comes from the ground truth CSI. Based on [14], we elaborate on the adversarial loss as

$$\begin{aligned}\mathcal{L}_{\text{DCGAN}}(G, D, \mathbf{H}, \mathbf{Z}) = \mathbb{E}_{\mathbf{h}\sim p_{\text{data}}(\mathbf{h})}[\log D(\mathbf{h})] \\ + \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,1)}[\log(1 - D(G(\mathbf{z}))]\end{aligned} \tag{1}$$

where the generator (G) tries to fool the discriminator (D) by generating synthetic CSI that looks like real CTI-contaminated CSI, while the discriminator will distinguish the synthetic CSI $G(\mathbf{z})$ and real $\mathbf{h}$, leading to optimizing $\min_G \max_D \mathcal{L}_{\text{DCGAN}}(G, D, \mathbf{H}, \mathbf{Z})$.

After playing this two-player minimax game for a number of iterations, a set of synthetic CTI-contaminated CSI, $\mathbf{h}^*$, will be generated across the experimental plane, featuring a network topology having a CTI source.

Our DCGAN network will be trained using Stochastic Adam Optimizer, which can help prevent overfitting and lead to convergence in a quick manner. It is also computationally efficient with little memory requirements. The following hyperparameters are used during the learning process. For both the generator and the discriminator networks, the learning rate for the Adam optimizer is 0.0002; the exponential decay rate for the 1-st moment estimates is 0.5; the momentum in batch normalization is 0.9, and the alpha in leaky ReLU is 0.2. For the discriminator, an additional dropout layer is added with a rate of 0.1. The activation used for the output dense layer is a sigmoid function.

C. CNN-based Localization Model

The synthetic CSI samples will be fed into two parallel modules, one for CSI-based localization, and one for deriving the highest achievable throughput. For the **Challenge (ii)**, a CNN model will be trained to investigate the mapping between the ground truth $\mathbf{h}_i$ and the corresponding location $(\bar{x}_i, \bar{y}_i)$, by which the final optimal location of the gateway can be deduced together with the throughput analysis. In particular, we use Mean Square Error (MSE) as the regression loss function,

$$\mathcal{L}_{\text{loc}} = \frac{1}{|\mathbf{h}|}\sum_{i=1}^{|\mathbf{h}|}[(\bar{x}_i - \bar{x}_i^p)^2 + (\bar{y}_i - \bar{y}_i^p)^2], \tag{2}$$

where the $(\bar{x}_i^p,\ \bar{y}_i^p)$ is the predicted location. Compared with existing CSI-based localization schemes adopting deep learning or machine learning [15]–[19], our developed CNN localization model specifically requires the interference pattern (in a form of noise) to be preserved in the CTI-contaminated CSI samples.

When developing an accurate CNN model, a common approach is to add a max pooling layer or batch normalization layer [20] to each convolution layer, by which noisy signals will be removed. Therefore, a tradeoff between noise effect and localization accuracy must be made. For our design, some max-pooling layers are selectively removed based on an empirical study in Sec V-D on the localization performance.

Parameter Selection. The CNN-based localization model is trained using the classical optimization algorithm Stochastic Gradient Descent (SGD) with a learning rate of 0.001 and a momentum of 0.9. The input to the CNN is the ground truth CSI samples with the size $3 \times 30 \times 3$ while the output is the 2-dimensional gateway's location. With this output, the activation function is changed from Sigmoid to Tanh. In addition, we take the Mean Squared Error (MSE) as the loss metric instead of the accuracy. Given the trained CNN network, we take the synthetic CSI samples in the first step as the input and further predict the corresponding location of the gateway.

D. Packet Reception Prediction

To tackle the **Challenge (iii)**, a few tools [21] calculate BER directly from received CSI samples via SNR estimation. However, these tools simply discard the packet if its preambles cannot be decoded, resulting in inaccurate network throughput. For our design, the major obstacle is that it remains unknown whether the synthetic CSI samples can be classified as it comes from a valid received Wi-Fi packet. Hence, we argue that all of the CSI obtained from real Wi-Fi packets will have designated characteristics, while some synthetic CSI samples may not have. On the other hand, it is unlikely to have negative examples because the ground truth CSI samples can only be obtained from received packets. Therefore, we first employ the Principal Component Analysis (PCA) to extract features from the ground truth $\mathbf{h}$ and evaluate the correlation with the synthetic CSI sample $\mathbf{h}_i^*$. Then, instead of building mathematical models, we use a One-Class Support Vector Machine (OCSVM) [22] to develop a packet reception prediction model on $\mathbf{h}_i^*$, in order to evaluate whether the synthetic CSI sequences can be estimated from a received Wi-Fi packet.

E. Grid-based PRR Calculation

From each synthetic CSI sample $\mathbf{h}_i^*$, $i = 1, 2, \cdots, M$, the predicted location information is denoted as $(\bar{x}_i^p,\ \bar{y}_i^p)$, and the associated packet reception result from OCSVM for each Wi-Fi sender $p_{ij}, j = 0, 1, 2$. To calculate the packet reception rate at the gateway, we divide the entire area into several square grids of the same size. We use the coordinates (x_g, y_g) to represent each grid, $g = 1, 2, \cdots, G$. A grid-based algorithm shown in Alg.1 is developed to calculate PRR for each grid. In the algorithm, we first use $\text{Pr}(x_g, y_g)$ to calculate the overall number of packets sent from all senders when the gateway's position is in the grid(x_g, y_g). Then, we use $\text{Grid}(x_g, y_g)$ to record the number of the predicted gateway's positions located in the grid. The $\text{PRR}(x_g, y_g)$ represents the PRR when the gateway is located in $\text{Grid}(x_g, y_g)$.

Algorithm 1 Grid-based PRR Calculation

Require:
One-class classification result vector, $\mathbf{p}_{ij}$.
Predicted location vector determined by CNN, $(\bar{x}_i^p,\ \bar{y}_i^p)$.

Initialize $\text{Grid}(x_g,\ y_g) = 0$, $\text{Pr}(x_g,\ y_g) = 0$ for $g = 1, 2, \cdots, G$.
for each i **do**
  if $\Sigma_j \mathbf{p}_{ij} > 0$ **then**
    $\text{Pr}(\bar{x}_i^p, \bar{y}_i^p) \leftarrow \text{Pr}(\bar{x}_i^p, \bar{y}_i^p) + \Sigma_j \mathbf{p}_{ij}$
  end if
  $\text{Grid}(\bar{x}_i^p, \bar{y}_i^p) \leftarrow \text{Grid}(\bar{x}_i^p, \bar{y}_i^p) + 1$
end for
for each x_g and y_g **do**
  $\text{PRR}(x_g, y_g) = \frac{\text{Pr}(x_g, y_g)}{3 \times \text{Grid}(x_g, y_g)}$
end for

IV. Advanced Scheme for Enhancing Training Efficiency

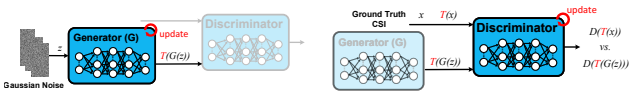
A. Motivation

The developed DCGAN requires the collection of a large number of ground truth CSI for each of the locations of the ZigBee CTI source. In our data collection process (ref. Sec. V-A2), the dimension of the data is approximately $8,370$. The offline training of the DCGAN model for each CTI location costs an average of 1,376.2s on a desktop (Intel-i7@3.8GHz and 32G RAM, 12 layers on the Generator and 8 layers on Discriminator including a total of $222,851$ parameters). On the other hand, the CNN-based localization model has $279,294$ parameters, for which the training process takes 93.5 minutes for all ZigBee locations. Apparently, the training process on both DCGAN and CNN largely compromises the efficiency for timely network deployment. Fortunately, the CNN-based localization model can be trained offline since it learns the location of CSI data no matter the location of the ZigBee interference source. However, regarding the DCGAN, once the CTI source moves or relative distances among IoT devices significantly change, the ground truth CSI has to be re-collected while the retraining of DCGAN is needed. Therefore, it is desirable to remedy the time consumption of training the DCGAN.

B. Differentiable Augmentation for CSI

To enhance the efficiency of the developed scheme, we plan to apply the differentiable augmentation scheme [23]–[25] to reduce the training data size for DCGAN. Although the numerical augmentation techniques have been widely adopted in the computer vision domain, no existing work explores its usage in augmenting the CSI samples. Each column of CSI values represents the amplitude and phase responses strictly to one subcarrier. The same value that happens on subcarrier 1 has a different meaning when it happens on subcarrier 2, indicating the orientations in CSI are not changeable. Due to the uniqueness of CSI samples, it is hard to apply flipping, modifying color space, cropping, rotation, noise injection, or data mixing to reduce the training data size.

In this advanced scheme design, we focus on using a cutout approach, which randomly masks a region of data during the GAN training process. As shown in Fig. 3, we introduce differentiable augmentation at three phases while updating the generator and the discriminator,



(a) **Generator-Aug.** (b) **Discriminator-Aug.**

Fig. 3: **GAN with DiffAugment.**

Phase 1. After CSI data was generated form generator;
Phase 2. While inputting ground truth data to discriminator;
Phase 3. When calculating the loss function for discriminator.

The loss function while updating the generator and discriminator can be written as,

$$\mathcal{L}_{\mathrm{G}_{Aug}}(G,\mathbf{Z}) = \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,1)}[\log(1-D(T(G(\mathbf{z})))] \quad (3)$$

$$\begin{aligned}\mathcal{L}_{\mathrm{D}_{Aug}}(D,\mathbf{H},\mathbf{Z}) = {} & \mathbb{E}_{\mathbf{h}\sim p_{\mathrm{data}(h)}}[\log(D(T(\mathbf{h}))] \\ & + \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,1)}[\log(1-D(T(G(\mathbf{z})))]\end{aligned} \quad (4)$$

where the $T(\cdot)$ operator denotes the augmentation that is taking place. We implement the Cutout method, which randomly suppresses 5 continuous sub-carriers CSI data to be zero. Note that the cutout region will be randomly chosen.

V. Performance Evaluation

A. Experimental Implementation

1) Device and Environment Setup: For our experimental study, Fig. 4a illustrates our implementation setup with 4 laptops, Lenovo ThinkPad T500 with Intel 5300 Wi-Fi NIC running the 802.11n protocol, performed as one gateway (collecting ground truth CSI) and three Wi-Fi senders. A ZigBee LaunchPad TI CC2652R1 [26] was placed in the area as the ZigBee interference source. Our experiment is carried out on the university campus, an outdoor environment to avoid multipath effects in signal propagation. Meanwhile, the chosen experimental environment does not have any commercial Wi-Fi coverage, and thus, no coexisting Wi-Fi signal will affect the performance of our experiments, especially for CSI estimation. Additionally, we show the specific locations that we use to collect ground truth CSI based on different ZigBee locations in Fig. 4b.

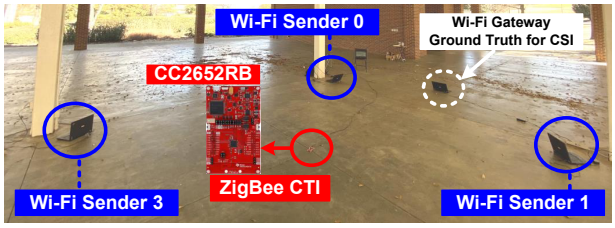
2) Datasets: **Data Collection.** On a 3m $\times$3m square open area, we place a ZigBee device in 12 different locations as shown in Fig. 4b. For each location, the ZigBee device continuously sends packets at the frequency of 2440 MHz with a power of 5 dBm. At the same time, the Wi-Fi sender at each of the 3 positions sends 500 random packets 5 times to the gateway placed at 31 different locations with a central frequency of 2442 MHz. The overlapped locations shown in Fig. 4b denotes that the same spot has been used for both the ZigBee device and Wi-Fi devices for collecting ground truth CSI with the presence of CTI. Since our floor plan is symmetric by nature, we omit the data collection for the bottom right part for the ZigBee device deployment.

Collected Dataset. Our dataset mainly consists of ground truth CSI samples given different CTI locations. The dimension of the collected CSI sample in each packet is 3×30, where 3 is the number of antennas at the gateway, and 30 is the number of subcarriers. With this setup, we will collect a total of $12 \times 31 \times 500 \times 3 \times 30 \times 3 \times 5 = 2.51 \times 10^8$ CSI samples. We split this dataset into multiple parts to appropriately train and test the performance of deep learning models for each of the experiments discussed in the following subsections.

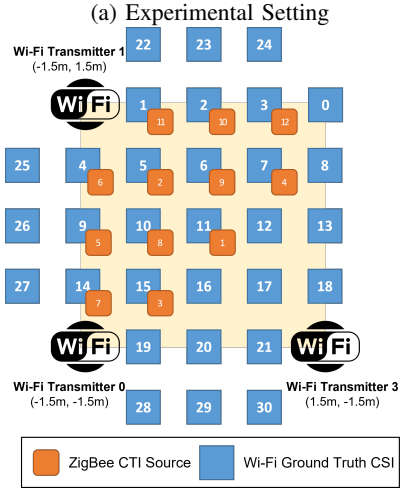
B. Gateway Throughput Analysis

First, we evaluate how the different locations of ZigBee affect the packet reception rate (PRR) of the gateway. The PRRs from 31 receiving locations where ZigBee is located at two

locations are shown in Fig. 5. The PRR at different locations varies significantly when the relative distance between the gateway and the ZigBee CTI sources changes. For example, as shown in Fig. 4b, the gateway deployed at positions 25 and 26 has better PRR compared with that from a closer distance to the CTI source, such as 14 and 19. In particular, the gateway located at the same location as the ZigBee device suffers from the most interference, where the PRR decreases drastically compared to the neighboring locations.

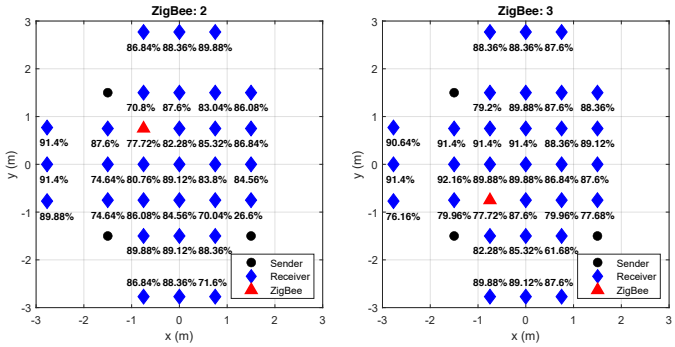


(a) Experimental Setting



(b) Specific Deployment Map

Fig. 4: **Experimental Implementation**



(a) ZigBee located at 2 (b) ZigBee located at 3

Fig. 5: **G.T. PRR from Different ZigBee Locations.**

C. Performance of DCGAN

The DCGAN training process adopts Stochastic Adam Optimizer [27] to help prevent over-fitting. To verify the correctness of GAN-based CSI data augmentation, we use Fréchet inception distance (FID) [28], a standard metric for assessing the quality of GAN, to compare the distribution of synthetic CSI samples and ground truth CSI on each subcarrier, which is calculated as follows,

$$\text{FID} = |\mu - \mu_w|^2 + \text{tr}(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}). \quad (5)$$

μ and μ_w refer to the mean values of the real and generated images, respectively, where Σ and Σ_w are the covariance matrix of the real and generated feature vectors, respectively, and $\text{tr}(\cdot)$ is the trace operator which sums up the elements on the main diagonal of $(\cdot)$. Fig. 6 demonstrates: (i) the increasing of ground truth data samples (e.g., CSI collected from 20 or 30 locations, detailed shown in Table. I) leads to synthetic CSI samples getting close to the ground truth distribution (lower FID value); and (ii) the developed GAN-based approach can generate highly accurate synthetic CSI samples used for optimal deployment prediction. The above result validates the feasibility of using GAN-based approaches for CSI data augmentation.

Number	Wi-Fi Sender's Locations
10	2, 9, 11, 13, 19, 20, 21, 23, 26, 29
20	besides 0, 4, 10, 14, 16, 22, 24, 25, 27, 28, 30
30	all of the locations besides 0

TABLE I: **Wi-Fi Sender's Ground Truth Locations**

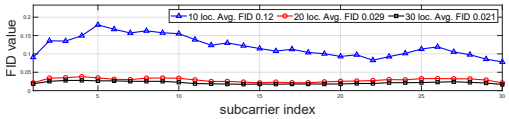


Fig. 6: **FID of Synthetic CSI from Different G.T. CSI.**

D. Performance of CNN-based Localization

Before we dive into the details of analyzing the PRR performance of the synthetic CSI, it is of great importance to first validate the performance of the CNN-based localization model. In practice, two major factors can affect its performance: model overfitting and training efficiency.

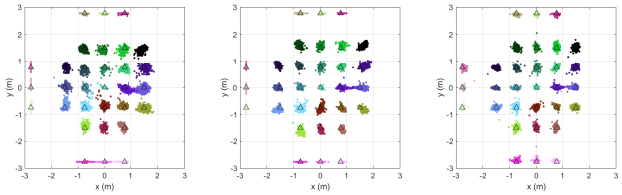
1) Comparison between images and CSI samples: To address these problems, multiple layers are used in the CNN model, e.g., using max pooling or batch normalization layers to accelerate the training process and/or using dropout to avoid overfitting the model. Although existing works [29], [30] implement several different layers into the model, such as adding batch normalization, dropout and max pooling layer after a few or every convolutional layer to avoid overfitting and reduce the computational cost, we found that those techniques are incompatible in our design. Specifically, conventional CNNs are used for image classification whose inputs are usually $512 \times 512 \times 3$ or even bigger vector images. Thus, layers like max pooling or dropout can efficiently reduce computational costs without losing too much information about the images themselves. In contrast, our input is a $3 \times 30 \times 3$ CSI images, which have a much smaller dimension and can not afford to lose any "pixel".

Model-1	Model-2	Model-3
Conv2D-32-3 Batch-Norm Dropout Conv2D-64-3 Batch-Norm Maxpooling2D Dropout Conv2D-128-3 Batch-Norm Maxpooling2D Dropout Conv2D-128-3 Maxpooling Flatten Dense-128 Dropout Dense-2	Conv2D-32-3 Batch-Norm Dropout Conv2D-64-3 Batch-Norm Dropout Conv2D-128-3 Batch-Norm Dropout Conv2D-128-3 Flatten Dense-128 Dropout Dense-2	Conv2D-32-3 Conv2D-42-3 Maxpooling2D Conv2D-128-3 Conv2D-128-3 Maxpooling2D Flatten Dense-20 Dropout Dense-2

TABLE II: **CNN Model Structure Comparison**

2) Evaluation Results: We have tested several different structures for the CNN model, where the detailed structures' information is listed in Table II.

Fig. 7 shows the predictions of different ground truth locations of CSI values by using different CNN structures. Points with the same color denote they are labeled as the same received location. In Fig. 7a, we implemented both batch normalization and max pooling layers right after every single convolutional layer. In this model, information is missing, and points located in interior regions are mixed with each other, which means the CNN can not discriminate the CSI values from different location sources. In Fig. 7b, we used only batch normalization layers after the first three convolutional layers. In this case, the points located in the interior region are slightly separated from each other but still mixed at some locations. For instance, CSI labeled at locations 12 and 13 are scrambled, and it is hard to decide on a clear boundary. On top of that, the predictions of points in the outer region still form a line shape. Finally, we adept max pooling layers only after every two convolutional layers and use a different way to dense the output. The result is shown in Fig. 7c, in which points in the interior region all have clear boundaries and the predictions in the outer region form like point sources, which makes more sense than the previously discussed models.

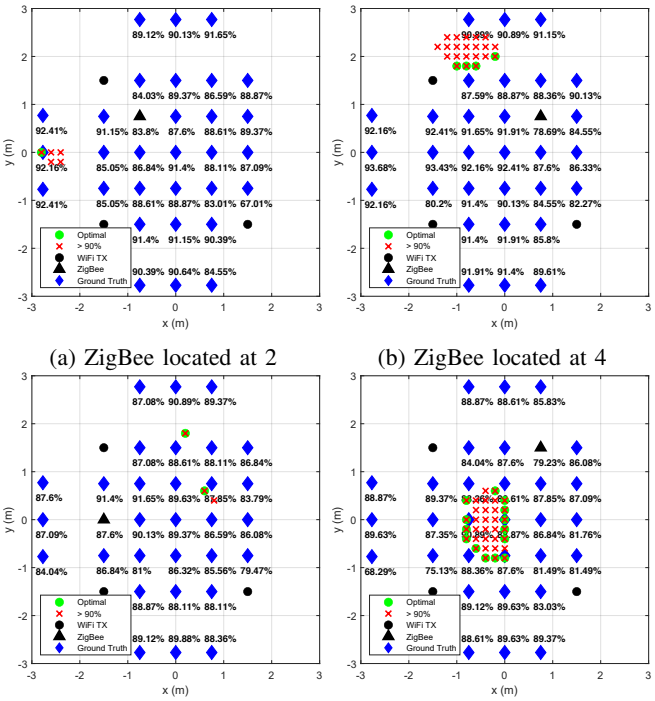


(a) Model 1 (b) Model 2 (c) Model 3

Fig. 7: **CSI Predictions of Different CNN Model Structures.**

E. PRR Calculation by PCA and OCSVM

We use PCA and OCSVM to decide whether the synthetic CSI can be received or not and then use Alg. 1 to calculate the packet reception rate of a location as the throughput. Therefore, given a specific CTI source, the optimal location of the gateway deployment can be found in Fig. 8. Apparently, the predicted region with the maximum throughput varies significantly with respect to the location of the CTI source.



(a) ZigBee located at 2 (b) ZigBee located at 4

(c) ZigBee located at 5 (d) ZigBee located at 12

Fig. 8: **Locations with Opt. Throughput by PCA.**

In Fig. 8b, both 90% and optimal predictions are located at the left side of the plane, which overlaps with ground truth locations with higher throughputs. In Fig. 8c and Fig. 8d, the prediction locations are not close to the ground truth with the highest packet reception rate, which is 91.65% in Fig. 8c and 89.64% in Fig. 8d, respectively. However, in both cases, packet reception rates are very close to all the ground truth locations. In those scenarios, it is hard to distinguish where the gateway is located due to their marginal difference.

For all 12 ZigBee locations, we compare the random selection (1 out of 30 ground truth locations) and our approach (optimal location) using the following equation,

$$\text{Improvement} = \frac{\mathbb{E}(P) - \mathbb{E}(R)}{\mathbb{E}(R)}, \tag{6}$$

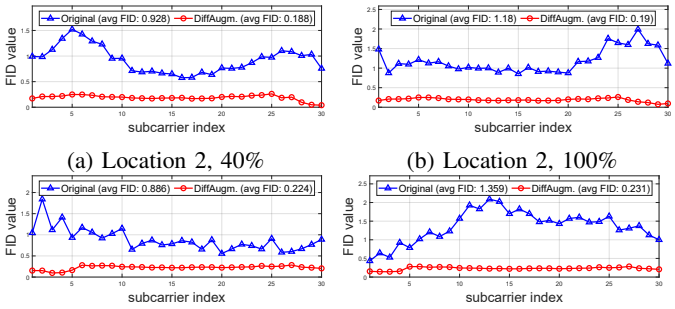
where P and R denote the predicted and ground truth locations matrix, respectively. $\mathbb{E}(\cdot)$ denotes the mean value of all locations in the input matrix. By using Eq. (6), we can conclude whether the PRR at the gateway has improved using our design. The improvement ratios are listed in Table. III. Together with the results in Fig. 8, the proposed scheme is able to reach more than 95% overall throughput and more than 10% average gain for all 12 ZigBee locations.

Zigbee Location	1	2	3	4	5	6
Improvement (%)	6.42	12.87	8.69	7.28	10.66	13.12
ZigBee Location	7	8	9	10	11	12
Improvement (%)	2.39	8.72	15.13	16.87	2.57	15.78

TABLE III: **PRR Improvement**

F. Evaluation of the Advanced Scheme

1) DiffAugment v.s. Basic Approach: **FID Comparison.** The major benefit brought by the differentiable augmentation is having a better GAN training result in terms of increased closeness between the ground truth and the synthetic data. Fig. 9 compares the FID values given different training data sizes. In particular, the FID value using 40% of the dataset with differentiable augmentation is smaller than using the original design, as shown in Fig. 10a. Meanwhile, the FID value of using the differentiable augmentation remains similar when using 40% and 100% of the dataset, which indicates we can use less dataset to achieve the same GAN training result.



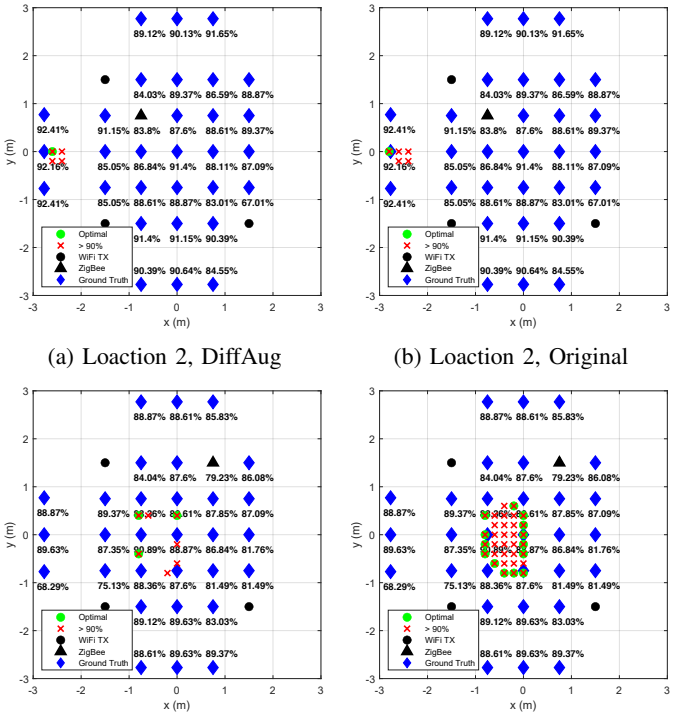
(a) Location 2, 40% (b) Location 2, 100%

(c) Location 5, 40% (d) Location 5, 100%

Fig. 9: **FID Comparison, Original v.s. DiffAugment.**

Location Prediction Comparison. We also evaluate how the DiffAugment will improve the final location prediction performances as shown in Fig. 10. Using the same amount of dataset (100% in both locations), the differentiable augmentation approach will help predict a more precise location as shown in Fig. 10a and Fig. 10c, both of which demonstrate the effectiveness of using the advanced scheme. On the other hand, the above two figures also show that the prediction results with differentiable augmentation are identical to the original scheme, rendering more flexibility in using fewer datasets for training to improve the scheme efficiency.

2) Improvement of PRR: Although using DiffAugment can significantly reduce the time consumption at the GAN training process, it remains unknown on how well it performs using the proposed design. The main reason, though, is that there is a lack of standard measurement on jointly evaluating the optimal localization performance and the expected PRR compared with that of the surrounding ground truth location. Therefore, we propose a new metric to evaluate the "optimality" of the



(a) Loaction 2, DiffAug (b) Loaction 2, Original

(c) Loaction 12, DiffAug (d) Loaction 12, Original

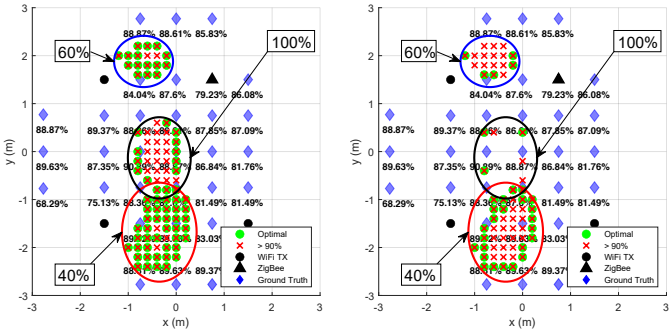
Fig. 10: **PRR Calculated by PCA in Original and Using DiffAugment.**

proposed design,

$$\text{score} = \frac{1}{I}\sum_{i=1}^{I}\frac{\prod_{m=1}^{M}\text{Td}(Op_i, G_{\min,m})\cdot\text{dis}(Op_i, G_{\min,m})}{\prod_{n=1}^{N}\text{Td}(Op_i, G_{\max,n})\cdot\text{dis}(Op_i, G_{\max,n})}, \tag{7}$$

where I is the total number of Synthetic CSI values with optimal PRR. Op_i denotes the i-th optimal points determined by PCA. We consider first N ground truth points which have the best throughput, and last M ground truth points which have the worst throughput, where $G_{\min,m}$ denotes the m-th ground truth point with minimum throughput and $G_{\min,n}$ denotes the n-th ground truth point with maximum throughput. $\text{Td}(a,b)$ is the operator which calculates the throughput difference point between a and b. $\text{dis}(a,b)$ is the operator that calculates the two-dimensional Euclidean distance between point a and b.

For this evaluation, we take ZigBee CTI at location 12 as an example. In Fig. 11, both original and differentiable augmentation results with 40%, 60% and 100% data for training GAN are shown. With the differentiable augmentation technique, the number of predicted optimal locations is reduced compared to the original scheme. The main reason for having a more accurate prediction is that using differentiable augmentation for GAN training can better learn the different characteristics of training data, leading to Synthetic CSI values being more distinctive if they represent CSI values from different locations. The above experimental results validate the theoretical analysis in Sec. V-F1, in which utilizing differentiable augmentation increases the accuracy of prediction. On



(a) 100%, Original (b) 100%, DiffAug

Fig. 11: **Original Scheme PRR Comparison Using DiffAugment with Subset.**

the other hand, the distribution of predicted optimal locations is not exactly the same with different sizes of training datasets. For example, the predicted optimal locations using 100% data stay in the middle of the plane, while that only using 60% will be more towards the upper part. This result demonstrates the optimal region with the highest PRR is not unique. Therefore, it is likely that multiple regions (locations) can be predicted, which highly depends on the distribution of the input synthetic CSI samples.

By using the metric in Eq. (7), the scores are listed in Table IV, in which two best PRRs and two worst PRRs have been taken into consideration ($M = N = 2$). When only using 40% of data for training, both original and differentiable augmentation schemes have scores of 11.1977 and 11.3384, respectively, because the training data are too few for GAN to learn effectively. While using 60% of data, the score of using differentiable augmentation is 71% higher than the original scheme and only 16% lower than using 100% data in the original scheme. In other words, the developed metric encourages the predicted location to be as close to the ground truth locations with higher throughput.

	Original Scheme	**Diff. Augmentation**
40%	11.1977	11.3384
60%	23.242	39.9245
100%	47.9958	10719.3411

TABLE IV: **Subset Size Comparison**

3) Training Efficiency Gain: The major efficiency gain brought by using differentiable augmentation is the time consumption on DCGAN training. Fig. 12 shows the comparison of FID scores along with the time consumption of DCGAN. Using differentiable augmentation will increase the time consumption for each given percentage of the used dataset due to the additional operations on the cutout. However, having the same FID score, say, 0.233, it took 21.7 minutes to train 100% data in the original scheme, while we only need 40% data and cost 10.75 minutes with differentiable augmentation. If we can further tolerate the FID, we can use 20% of data with a little bit more than 5 minutes to accomplish the DCGAN training.

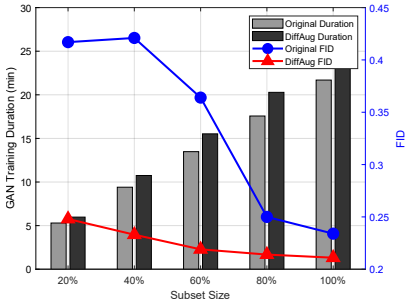


Fig. 12: **Accuracy and Efficiency Improvement**

To conclude, using differentiable augmentation can effectively reduce the time consumption and the size of data needed for training GAN while maintaining higher accuracy than the original scheme.

VI. Related Work

A. DL/ML-enabled Wireless Localization

Wi-Fi-based localization has been discussed for more than a decade, with extensive work from using the RSSI to CSI-based localization. In recent years, there have been some research works adopting DL/ML approaches to analyze the CSI sequences [31], [32]. Given the different application backgrounds, the use of contaminated CSI for localization has not been discussed in the above works. In [31], the authors frame their localization problem as an image translation problem for localizing the object and further enabling indoor navigation. Apparently, our work differs from the fact that the CSI samples in our case have been compromised by CTI, for which directly using the results from image translation research will not work.

B. Data Augmentations for GAN

Time consumption and overfitting problems are critical when training neural networks. There are numerous techniques focusing on addressing those problems. For example, [23]–[25] apply differentiable augmentation techniques such as cutout, rotation, adding noise, etc, while training neural network models to increase training efficiency and reduce overfitting. In [24], the author lowered the FID value by 40% and 46% by using only 20% and 10% data between applying and without applying differentiable augmentation, respectively. While all the previous works use images as input, the same model can not directly migrate to our model since CSI values are sensitive to subcarriers. We adopt the essence of the differentiable augmentation technique and implement a suitable augmented GAN model to improve efficiency.

VII. Conclusion

In this paper, we revisit a classic problem in wireless coexistence, i.e., CTI mitigation for throughput enhancement in a heterogeneous wireless system. Different from traditional approaches, we assume the gateway is rapidly deployable, which can alleviate the original CTI and enhance the network throughput. Our proposed scheme takes advantage of CSI to analyze the RF environment and further constructs a DCGAN

to augment the CSI sample data for unknown locations. The synthetic CSI sample will then be used to predict the optimal gateway position and determine the expected throughput via a CNN and PCA model. We conduct a thorough analysis based on real-world experiments, and the average throughput can be improved for 10.04% for 12 ZigBee locations. Furthermore, we apply the differentiable augmentation in our DCGAN to improve training efficiency. With differentiable augmentation, we only need 40% of synthetic CSI data for DCGAN training while lowering the training time to 50.46%.

ACKNOWLEDGEMENT

The work of L. Guo is sponsored by the Army Research Office and was accomplished under Grant Number W911NF-24-1-0044, NSF under grant CNS-2008049, CCF-2312616, CNS-2431440, and CCF-2427875. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The work of X. Zhang is partially by National Science Foundation under grant CCF-2312617, CNS-2431439, and CNS-2431553. This work also was partially supported by Clemson University's VIPR-GS Center, under Cooperative Agreement W56HZV-21-2-0001 with the US Army DEVCOM Ground Vehicle Systems Center (GVSC).

REFERENCES

[1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications surveys & Tutorials*, 17(4):2347–2376, 2015.

[2] Peng Duan, Yunjian Jia, Liang Liang, Jonathan Rodriguez, Kazi Mohammed Saidul Huq, and Guojun Li. Space-reserved cooperative caching in 5G heterogeneous networks for industrial IoT. *IEEE Transactions on Industrial Informatics (TII)*, 14(6):2715–2724, 2018.

[3] Shahid Mumtaz, Ai Bo, Anwer Al-Dulaimi, and Kim-Fung Tsang. Guest editorial 5G and beyond mobile technologies and applications for industrial IoT (IIoT). *IEEE Transactions on Industrial Informatics (TII)*, 14(6):2588–2591, 2018.

[4] Lin Zhang, Ying-Chang Liang, and Ming Xiao. Spectrum sharing for Internet of Things: A survey. *IEEE Wireless Communications*, 26(3):132–139, 2018.

[5] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving Wi-Fi interference in low power Zigbee networks. In *Proceedings of the 8th ACM conference on embedded networked sensor systems (SenSys)*, pages 309–322, 2010.

[6] Xinyu Zhang and Kang G Shin. Enabling coexistence of heterogeneous wireless systems: Case for zigbee and wifi. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 1–11, 2011.

[7] R. Chen and W. Gao. Enabling cross-technology coexistence for extremely weak wireless devices. In *Proceeding of the IEEE Conference on Computer Communications (INFOCOM)*, pages 253–261, 2019.

[8] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. Clearing the RF smog: making 802.11n robust to cross-technology interference. In *Proceedings of the ACM SIGCOMM*, pages 170–181, 2011.

[9] Jian Liu, Hongbo Liu, Yingying Chen, Yan Wang, and Chen Wang. Wireless sensing for human activity: A survey. *IEEE Communications Surveys & Tutorials*, 22(3):1629–1645, 2019.

[10] Chenning Li, Zhichao Cao, and Yunhao Liu. Deep ai enabled ubiquitous wireless sensing: A survey. *ACM Comput. Surv.*, 54(2), March 2021.

[11] KJ Ray Liu and Beibei Wang. *Wireless AI: Wireless Sensing, Positioning, IoT, and Communications*. Cambridge University Press, 2019.

[12] Daniele Croce, Domenico Garlisi, Fabrizio Giuliano, Nicola Inzerillo, and Ilenia Tinnirello. Learning from errors: Detecting cross-technology interference in WiFi networks. *IEEE Transactions on Cognitive Communications and Networking (TCCN)*, 4(2):347–356, 2018.

[13] Ashutosh Dhekne, Mahanth Gowda, Romit Roy Choudhury, and Srihari Nelakuditi. If wifi aps could move: A measurement study. *IEEE Transactions on Mobile Computing*, 17(10):2293–2306, 2018.

[14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[15] Cong Shi, Jian Liu, Hongbo Liu, and Yingying Chen. Smart user authentication through actuation of daily activities leveraging wifi-enabled iot. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 1–10, 2017.

[16] Xuyu Wang, Xiangyu Wang, and Shiwen Mao. Rf sensing in the internet of things: A general deep learning framework. *IEEE Communications Magazine*, 56(9):62–67, 2018.

[17] Chenshu Wu, Feng Zhang, Yusen Fan, and KJ Ray Liu. Rf-based inertial measurement. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 117–129. 2019.

[18] Zhipeng Zhou, Jihong Yu, Zheng Yang, and Wei Gong. Mobifi: Fast deep-learning based localization using mobile wifi. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[19] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 1–14, 2020.

[20] Daniel Graupe. *Deep learning neural networks: design and case studies*. World Scientific Publishing Company, 2016.

[21] linux-80211n-csitool-supplementary.

[22] Yang Zhang, Nirvana Meratnia, and Paul Havinga. Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks. In *2009 International Conference on Advanced Information Networking and Applications Workshops*, pages 990–995, 2009.

[23] Jin Zhang, Fuxiang Wu, Bo Wei, Qieshi Zhang, Hui Huang, Syed W. Shah, and Jun Cheng. Data augmentation and dense-lstm for human activity recognition using wifi signal. *IEEE Internet of Things Journal*, 8(6):4628–4641, 2021.

[24] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020.

[25] Connor Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019.

[26] TI. Launchxl-cc26x2r1 simplelink multi-standard cc26x2r wireless mcu launchpad development kit.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

[29] Xiangyu Wang, Xuyu Wang, and Shiwen Mao. Indoor fingerprinting with bimodal csi tensors: A deep residual sharing learning approach. *IEEE Internet of Things Journal*, 8(6):4498–4513, 2021.

[30] Xuyu Wang, Xiangyu Wang, and Shiwen Mao. Resloc: Deep residual sharing learning for indoor localization with csi tensors. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, 2017.

[31] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery.

[32] Xiaolong Zheng, Jiliang Wang, Longfei Shangguan, Zimu Zhou, and Yunhao Liu. Design and implementation of a csi-based ubiquitous smoking detection system. *IEEE/ACM Transactions on Networking*, 25(6):3781–3793, 2017.