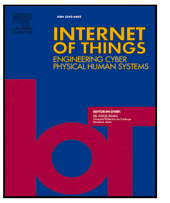
Internet of Things 21 (2023) 100646

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Research article

D-SCAN: Toward collaborative multi-radio coexistence in mobile devices via deep learning

Junhyun Park [a], Junyoung O. Park [b], Jaehyuk Choi [c,*], Ted Taekyong Kwon [a,*]

[a] Department of Computer Science and Engineering, Seoul National University, Seoul, Republic of Korea
[b] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, Los Angeles, CA 90095, United States
[c] School of Computing, Gachon University, 1342 Seongnam-daero, Seongnam, 13120, Korea

ARTICLE INFO

Keywords:
Bluetooth
Convolutional neural network
Deep learning
IoT
Scanning
Wi-Fi

ABSTRACT

As the demand for efficient and reliable wireless connectivity continues to increase, mobile Internet-of-Things devices equipped with multiple heterogeneous radios including Wi-Fi and Bluetooth have become prevalent. However, collocated Wi-Fi and Bluetooth operate in the same 2.4 GHz industrial, scientific, and medical band and interfere with each other internally and externally. Although current devices avoid internal cross-technology interference by enabling either Wi-Fi or Bluetooth to transmit packets at any specific time slot in a time-division multiplexing manner, the dissonance with external interference mitigation schemes can result in severe performance degradation. In this paper, we present *D-SCAN*, a novel collaborative coexistence mechanism. *D-SCAN* infers nearby Wi-Fi information efficiently using a collocated Bluetooth radio, thereby offsetting the overhead of key Wi-Fi functions and preventing collisions between Wi-Fi and Bluetooth. To this end, *D-SCAN* adopts a data-driven approach that captures the unique temporal and spectral features of Wi-Fi signals from Bluetooth spectrum measurements by leveraging deep neural networks. A *D-SCAN* prototype in real-world experiments reduces the latency and energy consumption of legacy Wi-Fi scanning by 23% and 45%, respectively. It also promotes the agile interference avoidance of Bluetooth that coexists with Wi-Fi on a single device. Thus, *D-SCAN* demonstrates efficiency in resource management and effectiveness in mitigating cross-technology interferences.

1. Introduction

Cross-technology interference (CTI) in the unlicensed industrial, scientific, and medical (ISM) bands has become a major issue as several different technologies, such as Wi-Fi, Bluetooth, and ZigBee, share the same frequency band. Many mechanisms have been proposed over the past several years to resolve the network-level external CTI problem. These include the detection of heterogeneous interferers and prevention of their collisions [1–5]. However, as the number of Internet-of-Things (IoT) devices exponentially increases and some IoT applications even rely on multiple heterogeneous radios in a single device, the internal CTI problem has also been aggravated.

To alleviate the problem of Wi-Fi and Bluetooth interfering with each other internally, time-division multiplex-based medium sharing approaches have been proposed [6–8]. That is, Wi-Fi and Bluetooth alternately access the 2.4 GHz ISM band. The most representative case would be *combo-module* solutions which integrate Wi-Fi and Bluetooth interfaces into a single system-on-chip

* Corresponding authors.
E-mail addresses: jhpark041@snu.ac.kr (J. Park), jop@ucla.edu (J.O. Park), jchoi@gachon.ac.kr (J. Choi), tkkwon@snu.ac.kr (T.T. Kwon).

https://doi.org/10.1016/j.iot.2022.100646
Received 25 August 2022; Received in revised form 23 October 2022; Accepted 15 November 2022
Available online 17 November 2022

(SoC) circuit. Most smartphones and many IoT devices (e.g., Samsung Galaxy, Apple iPhone, and Raspberry Pi) adopt these combo-modules owing to their cost-effectiveness in terms of form factor. However, the inherent nature of antenna sharing causes severe performance degradation, which has been reported repeatedly [9,10] for common application scenarios in which the two wireless interfaces are simultaneously used. The reason, which will further be elaborated on, is primarily that the external and internal CTI mitigation schemes may interfere with each other and lead to severe performance degradation.

In this paper, we exploit Wi-Fi and Bluetooth radios on the same device in a collaborative manner so that the advantages of one radio supplement the disadvantages of the other. We leverage the fact that a Bluetooth radio in a combo-module can sense Wi-Fi signals as Bluetooth channels and Wi-Fi channels overlap in the 2.4 GHz unlicensed band. With the low-power Bluetooth radio, we efficiently infer several useful pieces of Wi-Fi information, which include the identification of which Wi-Fi channels are used and the utilization of each Wi-Fi channel. Subsequently, we facilitate the reliable coexistence of Wi-Fi and Bluetooth within the same device by coordinating the heterogeneous radios to effectively share the same antenna and spectrum.

Many studies have attempted to utilize a low-power secondary radio to acquire Wi-Fi information. Wake-on-WLAN [11], S-WOW [12], and Esense [13] enable cross-technology communications between ZigBee and Wi-Fi using special codes to exchange their channel information. Others have suggested Wi-Fi and Bluetooth combo access points (APs) to embed Wi-Fi channel information into Bluetooth broadcast packets [14,15]. Zi-Fi [16], BlueScan [17], and C-SCAN [18] proposed hands-off listening techniques to detect Wi-Fi APs. For example, Zi-Fi searches for beacon frames by analyzing the periodicity of received signal strength indicator (RSSI) peaks. Choi et al. [19] and Jung et al. [18] proposed lightweight algorithms to detect Wi-Fi signals via the correlation analysis between RSSI samples. These studies contributed to collecting detailed information about Wi-Fi signals; however, a few limitations remain. The requirement for broad applicability demands a mechanism that (i) minimizes software or hardware modification, (ii) operates efficiently in terms of energy consumption, and (iii) adapts to dynamically changing environments.

To this end, we introduce a new data-driven approach, called *D-SCAN*, which models the unique temporal and spectral features of Wi-Fi signals from RSSIs measured by a Bluetooth radio via deep learning techniques. *D-SCAN* is widely applicable as it does not require hardware modifications, and is energy-efficient as the low-power Bluetooth radio offsets the overhead in Wi-Fi operations. Unlike previous heuristic approaches, it uses deep neural networks (DNNs) to reliably infer Wi-Fi information even under network dynamics. However, the implementation of *D-SCAN* may be complicated by the following challenges. First, measuring the RSSIs of Wi-Fi signals with a Bluetooth radio involves a wide search space. In addition, a large number of training data that cover complex signal and interference patterns from the real world is required. Finally, *D-SCAN* should capture Wi-Fi signal patterns within fluctuating RSSI samples. Note that the problem becomes particularly difficult as the density of APs increases.

The key idea of *D-SCAN* that enables the accurate analysis across 13 Wi-Fi channels[1] We overcome this challenge using a divide-and-conquer approach. *D-SCAN* divides the 2.4 GHz spectrum into tractable narrow frequency bands to render training and decision-making manageable. Subsequently, we extend two deep learning models to solve the scanning problem. We first apply the long short-term memory (LSTM) model to obtain sequential RSSI measurements of 20 MHz bands and to learn the temporal features of Wi-Fi signals. However, LSTM exhibits limited performance for short and sparse Wi-Fi signals owing to the low correlation between consecutive RSSI samples. We address this problem with a simple data representation method, termed *edge projection*, that accumulates the RSSI data for a target period. The edge projection preserves temporal and spectral correlations of Wi-Fi signals in 2-D matrices, and thus can effectively leverage the convolutional neural network (CNN) model.

The main contributions of this paper are summarized as follows:

- We propose the *D-SCAN* mechanism, which exploits a Bluetooth radio to efficiently obtain Wi-Fi channel information such as Wi-Fi channel usage, signal strength, and channel utilization.
- We introduce a new two-dimensional (2-D) RSSI data representation and effectively extend deep learning models to learn temporal and spectral features of Wi-Fi signals, achieving robust and adaptive estimation performance.
- We implemented a prototype of *D-SCAN* for three use cases, in which the obtained Wi-Fi information was used to improve the efficiency of both Wi-Fi scanning and handover and facilitate the coexistence of Bluetooth with Wi-Fi by promptly adapting to Wi-Fi interference.
- We generalize *D-SCAN*, which uses a low-power radio to learn the distinguished features of wideband and narrowband signals, and demonstrates its utility in IoT devices over different frequency bands.

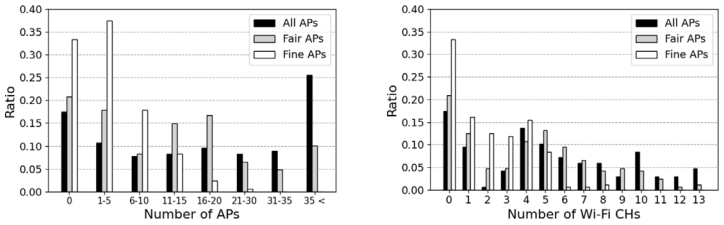
The remainder of this paper is organized as follows. Section 2 provides the preliminaries of our research. In Section 3, we explain the overview and design of *D-SCAN*. In Section 4, we describe use cases of *D-SCAN* and their implementations to improve Wi-Fi and Bluetooth performance. Section 5 presents the evaluation results that emphasize the applicability of *D-SCAN* in real-world scenarios. In Section 7, we conclude the paper and present potential future directions.

2. Preliminaries

2.1. Wi-Fi characteristics

To motivate the need for our solution *D-SCAN*, we first conducted experiments to collect data of actual Wi-Fi usage on various public locations such as parks, cafes, restaurants, and subway stations in Seoul, South Korea. Fig. 1(a) shows the ratio of the number

[1] Under the regulation of individual countries, the United States uses 11 Wi-Fi channels defined in the 2.4 GHz band, while most other countries use 13 Wi-Fi channels. are as follows. In general, the accurate classification by DNNs is not an easy task since there are a large number of possible outcomes that require high complexity.



(a) Adjacent APs (b) Active Wi-Fi channels

Fig. 1. The spectrum must be efficiently scanned and analyzed because the numbers of observed Wi-Fi APs and active channels with good signal quality tend to become limited as the numbers of APs and channels increase.

of places in which a given number of APs were observed to the total 168 visited places. Fig. 1(b) shows the ratio of the number of places in which a given number of Wi-Fi channels were actively used to the number of total visited places. Here, we counted the numbers of all the detected APs, fair APs with RSSIs above −80 dBm, and fine APs with RSSIs above −60 dBm, respectively, to differentiate the quality of Wi-Fi signals from the APs.

Many Wi-Fi APs, more than 35 in about 25% of the visited places, were deployed over multiple channels and formed complex interference patterns in the 2.4 GHz band. However, when we only considered fine APs, which are useful in actual communications, the number of detected APs was significantly low. In terms of the number of used channels, four or fewer active channels are the most common. If the Wi-Fi information, such as channel distribution of Wi-Fi APs, their signal strengths, and channel utilization, were obtained in advance using *D-SCAN*, Wi-Fi devices can reduce unnecessary scanning overhead by performing selective scans on channels that are confirmed to have active Wi-Fi APs. In addition, heterogeneous IoT devices or protocols that share the same frequency band can mitigate Wi-Fi interference by observing the Wi-Fi usage patterns.

2.2. Wi-Fi and Bluetooth coexistence in a combo-module

Wi-Fi and Bluetooth in a combo-module share a single antenna as well as the ISM band in a TDM manner, generating new interference patterns that result in performance degradation. More specifically, when Wi-Fi and Bluetooth are used simultaneously, both protocols fail to address nearby interferers and their collision rates mutually increase.

Bluetooth uses an adaptive frequency hopping (AFH) strategy to avoid interference. Each pair of Bluetooth devices holds an AFH channel map that simply identifies which channels are good to use based on their local channel assessments. However, the AFH, which requires considerably complex channel assessments and master–slave handshaking, cannot appropriately update the AFH map while the radio is being used by Wi-Fi. Thus, Bluetooth fails to adapt to wireless communication dynamics and wastes a significant portion of radio occupancy during collisions. Furthermore, impairing Wi-Fi frames using Bluetooth signals triggers the avalanche [9], in which Wi-Fi transmission rates are lowered by the rate adaptation mechanism.

For example, even users watching a video may require both Wi-Fi and Bluetooth: streaming down through Wi-Fi, and transmitting the audio to a headset via Bluetooth. Significant deterioration of video and audio quality in such use cases has been reported [10] with no clear solution other than turning off Bluetooth. We believe that *D-SCAN* can mitigate this coexistence problem.

3. D-SCAN

3.1. Overview

D-SCAN facilitates the collaborative coexistence of Wi-Fi and Bluetooth on a single device by harnessing the fact that a Bluetooth radio can measure the RSSI values of Wi-Fi signals over its Bluetooth channels, although it cannot decode the Wi-Fi signals. *D-SCAN* employs a data-driven approach that models the unique temporal and spectral features of Wi-Fi signals via deep learning techniques to extract useful Wi-Fi channel information such as Wi-Fi signal presence, signal strengths, and channel utilization.

With *D-SCAN*, we can improve the efficiency of Wi-Fi operations, such as Wi-Fi scanning and handover, and also enable the synergy of multi-radio coexistence. For instance, the channel information obtained by *D-SCAN* is used by the Wi-Fi radio to perform actual Wi-Fi scanning only on the channels in which the presence of APs is identified. It also aids the Wi-Fi radio (of the device) to associate with a better AP, by considering both Wi-Fi signal strengths and channel utilization data. Finally, *D-SCAN*, by informing Bluetooth of the used Wi-Fi channels and their utilization, helps reduce collisions between the two technologies. The use cases and their implementations are detailed in Section 4.

The overall flow of *D-SCAN* is illustrated in Fig. 2. It consists of four core components: Scheduler, Sampler, Data Formatter, and Channel Inspector. The Scheduler is invoked by the system or applications to reserve and initiate *D-SCAN* operations. The Sampler measures RSSIs over the 2.4 GHz band using a Bluetooth radio. These raw RSSIs are continuously fed to the Data Formatter to pre-process the input to the neural networks. The Channel Inspector, which includes pre-trained LSTM or CNN models, analyzes the input (by running its neural networks) and infers Wi-Fi information.

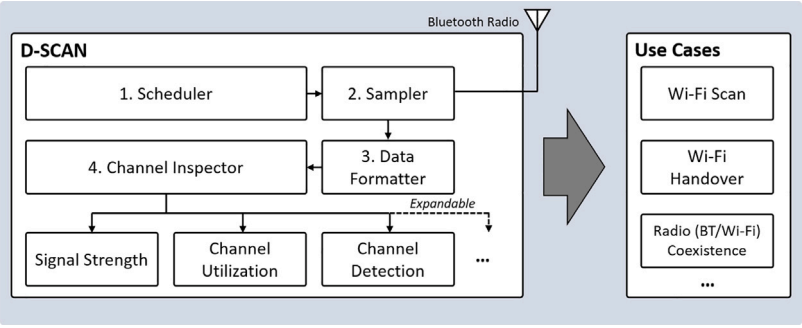


Fig. 2. *D-SCAN* architecture leverages Bluetooth to aid Wi-Fi to scan and analyze the Wi-Fi channel status efficiently.

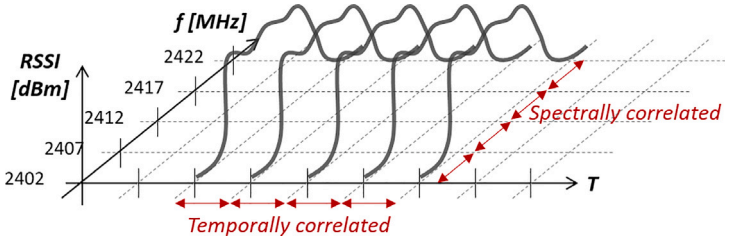


Fig. 3. Temporal and spectral correlations in Wi-Fi signals on CH1.

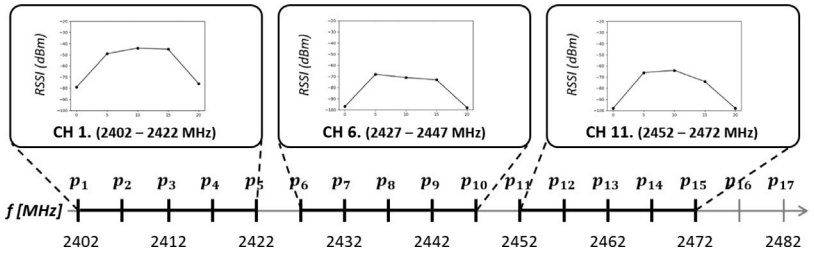


Fig. 4. Recurring pattern of Wi-Fi signals on different channels to reduce the problem complexity and facilitate efficient training.

3.2. Divide-and-conquer approach

To efficiently train a DNN for *D-SCAN* that encompasses various Wi-Fi environments, a large amount of training data and time are required. It involves numerous RSSI samples measured with Bluetooth radios to obtain comprehensive data across all Wi-Fi channels, i.e., 13 channels in our settings. To address this challenge, we characterize the waveforms of Wi-Fi signals on the 2.4 GHz band by collecting the RSSIs on the Bluetooth channels overlapping with the Wi-Fi ones. We can observe the recurring pattern of Wi-Fi signals on RSSI samples measured in the 20 MHz range corresponding to a Wi-Fi channel. For example, when we capture RSSIs of Wi-Fi signals when Wi-Fi APs exist on channels 1, 6, and 11, the signal characteristics in 2402–2422 MHz range are similar to those in 2427–2447 MHz and 2452–2472 MHz ranges (Fig. 4). This implies that the same method can be applied to infer the different Wi-Fi channels across the ISM frequency band.

This observation motivates us to employ a divide-and-conquer approach that simplifies the problem of training a DNN model for the entire Wi-Fi frequency band into smaller sub-problems of modeling a single Wi-Fi channel. In other words, we focus on developing a deep learning model for a 20 MHz rather than the entire 84 MHz (2.4–2.4835 GHz) in the ISM band and then apply the model multiple times to infer information on all the 13 Wi-Fi channels.

This divide-and-conquer approach has the following benefits: It significantly reduces the complexity of the deep learning problem and the amount of training data required. We can collect 13 20 MHz-wide training instances, which are partially overlapping but unique, from the 2.4 GHz band of each measurement site. We demonstrate that a divide-and-conquer approach can efficiently train neural networks to achieve high accuracy in various scenarios, including multi-AP signals.

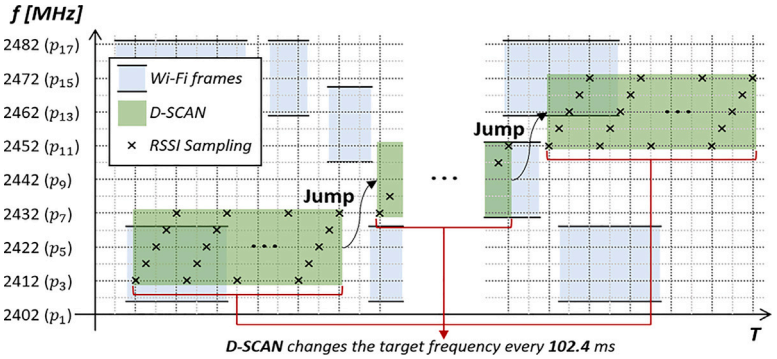


Fig. 5. Three sampling ranges of *D-SCAN* to cover the 2.4 GHz band.

3.3. Temporal and spectral features of Wi-Fi signals

To effectively design deep learning models for obtaining Wi-Fi information, we have to understand the unique features of Wi-Fi signals distinguished from those of other protocols or noise. This brings us to focus on the temporal and spectral correlations between measured RSSIs during the sampling procedure (Fig. 3).

From a receiver's perspective, a valid Wi-Fi signal occurs at an arbitrary time and lasts for a certain duration (i.e., airtime, T). Thereafter, the RSSIs measured consecutively for a period (T) at a frequency inside the 20 MHz range of the Wi-Fi signal have similar values and exhibit temporal correlation. However, the spectral feature across the bandwidth of a Wi-Fi signal should also be employed to filter out the signals of other narrowband protocols; the channel widths of 1–2 MHz for Bluetooth and 2 MHz for ZigBee are relatively small compared with 20 MHz for Wi-Fi. The RSSIs of Wi-Fi signals, measured over the 20 MHz channel bandwidth for the period T, exhibit spectral correlation, exhibiting a bell-shaped arrangement around the center frequency.

In *D-SCAN*, we reduce the search space from 84 to 17 frequencies, which we call *sampling points* (p_1, p_2, …, p_{17}), by spacing out the sampling frequencies apart by 5 MHz such that they are aligned with center frequencies of Wi-Fi channels as in Fig. 4. This coarse-grained sampling strategy with 5 MHz space significantly reduces the sampling cost and the RSSI measurement delay while preserving both the temporal and spectral features of Wi-Fi signals. It can also filter out the narrowband signals as they cannot affect two or more sampling points simultaneously. Note that using a fine-grained sampling, e.g., spaced 1 MHz apart instead of 5 MHz, in a limited amount of time may result in partial data redundancy in the spectrum domain and/or incomplete spectrum characterization in the time domain. If the temporal and spectral features are well preserved in sampled data regardless of fluctuations in RSSI values, the DNNs of *D-SCAN* can effectively capture specific patterns and adapt to network dynamics.

Based on the insights above, we design an RSSI sampling strategy for *D-SCAN* as follows. *D-SCAN* collects RSSI values by hopping five sampling points in a cyclical manner (e.g., from p_3 to p_7 and starting over from p_3 again as in Fig. 5) every τ. Here, τ is the sampling period, or slot time, which is ~160 us in the implementation using the CC2400 chipset. Once sampling begins, it lasts for a beacon interval (102.4 ms) to ensure beacon detection [20]. Wi-Fi APs can be sensed even without data exchange between APs and clients, as their beacon frames are periodically broadcasted. The upshot of this sampling domain and principle forms a solid foundation for DNN training and decision-making.

3.4. Sweeping the entire 2.4 GHz band

Legacy Wi-Fi scanning algorithms are inherently inefficient owing to a lack of prior knowledge on the APs in the vicinity and thus perform a full-channel search on 13 channels. As a result, legacy Wi-Fi scanning requires approximately 1300 ms for passive scanning and 520 ms for active scanning in the 2.4 GHz band [21].

As explained earlier in Section 3.2, *D-SCAN* employs a divide-and-conquer approach and uses the pre-trained deep learning models to infer the entire Wi-Fi channel information. The models are designed to obtain inputs from five sampling points to infer the Wi-Fi channel information over a 20 MHz range.

Here, it is worth noting that, in order to obtain information on all 13 Wi-Fi channels, *D-SCAN* requires to be executed only three times, instead of 13 times (Fig. 5), unlike the legacy Wi-Fi scanning algorithms. This is because when *D-SCAN* is targeted on a Wi-Fi channel, it can infer Wi-Fi information of five Wi-Fi channels simultaneously: one for the target channel and four for the adjacent overlapping channels. For example, by observing the five sampling points (p_3, p_4, p_5, p_6, p_7) targeted on Wi-Fi channel 3, we can also infer Wi-Fi information of channels 1, 2, 4, and 5. Out of these five sampling points, Wi-Fi signals on channel 1 affect the RSSI values of three sampling points, (p_3, p_4, p_5), Wi-Fi signals on channel 2 affect the RSSI values of four sampling points, (p_3, p_4, p_5, p_6), etc. Thus, *D-SCAN* can analyze these neighboring Wi-Fi channels, albeit with fewer hints in the input data.

Thus, we set the three sampling ranges, 2412–2432, 2432–2452, and 2452–2472 MHz and obtain information of Wi-Fi channels {1, 2, 3, 4, 5}, {5, 6, 7, 8, 9}, and {9, 10, 11, 12, 13}, respectively. Channels 5 and 9 benefit from additional verification through redundant analysis. In summary, *D-SCAN* is highly efficient in terms of both latency and energy, as the scan over the entire 2.4 GHz spectrum only lasts ~300 ms using a low-power Bluetooth radio. However, note that the training dataset consists of RSSI measurements from all 13 Wi-Fi channels.

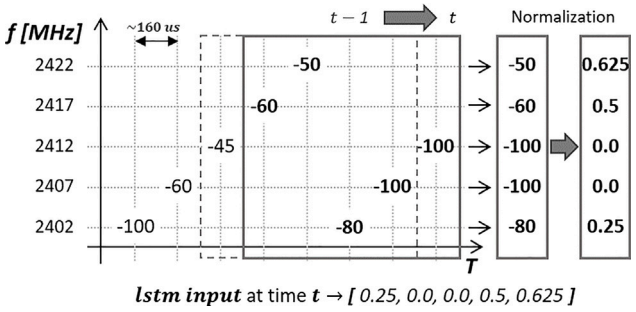


Fig. 6. Sequential RSSI measurements for the LSTM.

Table 1
RNN-LSTM architecture.

Stage	Layer description	Shape
	Sequence input w/ 5 dim	5
1	LSTM with 256 hidden units	256
2	LSTM with 256 hidden units	256
3	FC 32	32
	BN + ReLU	32
4	FC 5	5
5	Sigmoid	5
	Prediction output	5

3.5. Deep learning models

We introduce extended designs of two renowned deep learning models for *D-SCAN*: LSTM and CNN. The inputs are obtained from five sampling points in the 20 MHz range, but the format varies as each model is designed for specialized purpose. Instead, we set the output layer size of both neural networks to five to infer five-channel information by obtaining the input from the 20 MHz range. Here, the simultaneous analysis of multiple Wi-Fi channels barely affected the prediction accuracy in our experiments.

3.5.1. Time series RSSI measurements for LSTM

LSTM networks are designed to determine patterns in long time-series data [22–24]. Therefore, the ability to memorize characteristic Wi-Fi fragments in an RSSI sequence is important in the learning process. As an input to the LSTM model, we collect RSSI data for a beacon interval (102.4 ms) sequentially from five sampling points using a sliding window with a window size of 5 (Fig. 6). The resulting matrix of 5×640 is used for the input. The RSSIs are normalized with minimum and maximum boundaries of −100 and −20 dBm, respectively, before we feed the data to the LSTM. Generally, APs of RSSIs less than −90 dBm are considered unusable, and those greater than −30 dBm are considered prodigious.

3.5.2. LSTM architecture

The LSTM model of *D-SCAN* is designed to have five stages. Table 1 lists the parameters of our model. It has two LSTM layers with 256 hidden units, followed by two fully connected (FC) layers of sizes 32 and 5, respectively. We apply batch normalization (BN) and Rectified Linear Unit (ReLU) activation between the fully connected layers for standardized non-linear transformation on each element. In the final layer, a sigmoid activation function is applied to infer the Wi-Fi signal strength or channel utilization with a value between 0 and 1.

The LSTM model receives the RSSI sequence from five sampling points as the input. The first two LSTM layers locate characteristic Wi-Fi fragments and learn the long-term relationships between time series data. Two fully connected layers learn the nonlinear complex relationship between the hidden unit and final output. We expected the LSTM model to infer the Wi-Fi signal strength by recognizing the shape of the encoded Wi-Fi signal or infer the channel utilization by tracing the history of signal discovery.

In our experiments, the LSTM-based *D-SCAN* provides decent estimations on the overall Wi-Fi channel information, but its performance was surpassed by that of CNN. This is because the short and sporadic Wi-Fi signals leave only a few temporal patterns in RSSI samples for the LSTM to capture.

3.5.3. Edges and projections for CNN

The main goal of CNN is to analyze multi-dimensional images with spatial features [25–27]. We hypothesize that encoding temporal and spectral correlations between highly fluctuating (and, in some cases, seemingly random) RSSI samples as an image would form a solid pattern and enable us to discern meaningful Wi-Fi signals from noise. However, converting sequential 1-D RSSI data into discernible images poses a challenge.

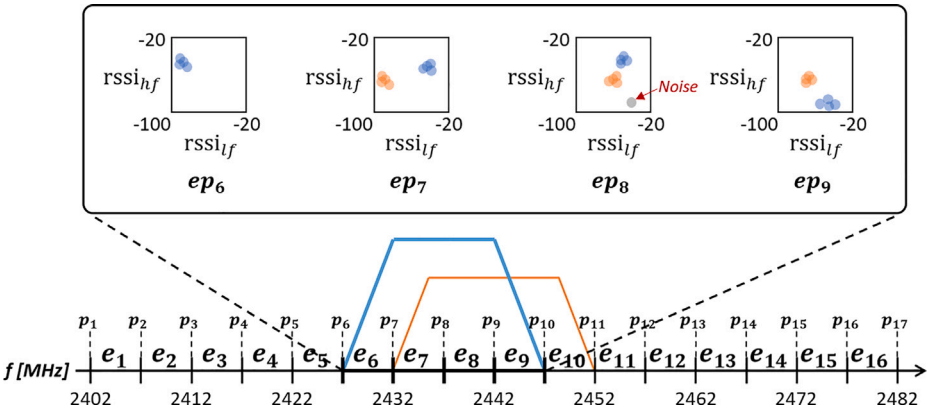


Fig. 7. Unique cluster pattern projected by Wi-Fi signals on corresponding four edge projections.

Table 2
CNN architecture.

Stage	Layer description	Shape
	Four-edge projection	4 × 80 × 80
1	Conv 3 × 3	16 × 80 × 80
	BN + ReLU + AvgPool(2,2)	16 × 40 × 40
2	Conv 3 × 3	16 × 40 × 40
	BN + ReLU + AvgPool(2,2)	32 × 20 × 20
3–1	Conv 3 × 11	128 × 20 × 20
3–2	Conv 11 × 3	128 × 20 × 20
	BN + ReLU + AvgPool(2,2)	256 × 10 × 10
4	Conv 3 × 3	256 × 10 × 10
	BN + ReLU + AvgPool(2,2)	256 × 5 × 5
5	FC 128	128
	BN + ReLU	128
6	FC 5	5
7	Sigmoid	5
	Prediction output	5

We address this problem by developing a novel data representation, termed *edge projection*, in which the focus is on the edges between two adjacent sampling points instead of the points themselves. There are 16 different edges (e_1, e_2, …, e_{16}) between 17 sampling points. An *edge* sample consists of two RSSI values measured successively from two adjacent sampling points. Accumulating all edge samples obtained during the sampling period into 2-D planes results in 16 edge projections (ep_1, ep_2, …, ep_{16}). More specifically, we can plot a point (corresponding to an edge sample) on a plane by using the RSSI value ($rssi_{lf}$) from a relatively low-frequency sampling point as the x coordinate and the RSSI value ($rssi_{hf}$) from a relatively high-frequency sampling point as the y coordinate. Here, we bin RSSI values between −100 and −20 dBm. Therefore, an edge projection is represented by an 80 × 80 matrix, whose elements are the count of edge occurrences.

Fig. 7 depicts an example of edge projections when two APs reside in different channels, 6 and 7. An edge (e_6) is constructed using two RSSI values from p_6 and p_7, which are measured consecutively and projected onto ep_6. The $rssi_{p_6}$ from the relatively low-frequency (p_6) and $rssi_{p_7}$ from the relatively high-frequency (p_7) are respectively used as the x and y coordinates of e_6 in ep_6. Thus, the aforementioned temporal and spectral correlations are preserved; two consecutively measured RSSI samples are integrated into a single point and their relative and absolute magnitudes are encoded as a position on a 2-D plane. The edge projection embodies the likelihood and consistency of a Wi-Fi signal by recording the number of signal occurrences onto a matrix corresponding to the edge samples' coordinates. In Fig. 7, the edges being accumulated are expressed as the colored points, which become darker as the points are repeatedly plotted at the same or adjacent locations.

Wi-Fi signals leave distinctive patterns in the edge projections. As shown in Fig. 7, when the signals from an AP in channel 6 are projected as edges, the weights are accumulated on ep_6, ep_7, ep_8, and ep_9. In an edge projection, edges from the same AP are plotted close to each other to form a unique cluster pattern, and the combination of these clusters across multiple edge projections characterizes that AP. Here, the distance of the clusters to the origin (−100, -100) represents the signal strength: the farther the distance, the stronger the Wi-Fi signals. Furthermore, the weights accumulated by edges imply the degree of medium utilization in the frequency range of that edge projection. Signals from protocols other than Wi-Fi would only affect a single edge projection, and noise would not form a cluster.

3.5.4. CNN architecture

Depending on the device, Wi-Fi signals exhibit various similar but different bell-shaped spectral patterns, and the CNN is specialized to distinguish them effectively. Our CNN consists of seven stages; four convolutional layers, two fully connected layers, and the output layer (Table 2). We apply BN and ReLU activation between the convolutional and fully connected layers. The 2×2 average pooling filters that follow the convolutional layers are applied with a stride of 2 to downsample the feature maps.

The four edge projections with a shape of 4×80x80 are provided as inputs to the CNN, with the second and fourth edge projections transposed to facilitate cluster combination search. The first two convolutional layers are designed to spot clusters in individual edge projections. The convolutional layers in stage 3 are designed to detect cluster combinations across edge projections by applying vertically and horizontally long filters. The final convolutional layer determines meaningful combinations among them. The subsequent fully connected layers learn relevance with the output layer to approximate the Wi-Fi channel information. We expected the CNN to infer (i) the strongest Wi-Fi RSSI by recognizing the outer-most cluster combinations of a valid Wi-Fi signal and (ii) the channel utilization by observing the accumulated edge occurrences.

3.5.5. Hyperparameter tuning

There are no strict rules for selecting network parameters when using neural networks. Therefore, we experimentally searched for the parameters that optimize the performance and remove unnecessary computations by adjusting network widths and depths [28,29]. In fully connected layers, we set the dropout rate to 0.5 to prevent overfitting and generalize the networks. Finally, the weight of networks was trained using the Adam optimizer with a learning rate of 2e-4. The prediction error through backpropagation was minimized by using the sum of squared error (SSE) to satisfy the following objective function:

$$\min_{\theta} Loss_{SSE}(\hat{y}, y) \tag{1}$$

$$Loss_{SSE}(\hat{y}, y) = \sum_{i} (\hat{y}[i] - y[i])^2 \tag{2}$$

where $\hat{y}$ is the set of predicted output on ith channel, and y denotes the set of ith channel's actual value. We implemented our deep learning architectures using Pytorch on a system with four NVIDIA CUDA-enabled Titan GPUs.

To account for the overhead put on the processors, we observe the number of floating-point operations (FLOPs), which refers to the number of multiplication-addition operations required for the network to output its inference. The number of FLOPs is a good indicator of the network complexity and computational overhead in that the value is not hardware-dependent and is determined by the network structure (i.e. the number of network parameters) and input size. VGG-16 [30] and ResNet [31], the traditional prominent CNNs for image analysis, process inputs of size 224x224 requiring about 15.3B and 3.6B FLOPs, respectively. MobileNet [28] has lightened the DNN so that the analysis can be performed on mobile devices with ∼0.6B FLOPs. Even compared to [28], CNN-based *D-SCAN* is lightweight as it processes a 4x80x80 input with 1.7M network parameters, requiring 0.18B FLOPs for each runtime. The exact time and energy required for *D-SCAN* depend on hardware specifications such as memory and processor. For each prediction on Wi-Fi information, our GPU system spends only 1.2 ms. *D-SCAN* is also applicable to low-spec devices, such as Raspberry Pi 3, and it requires 16 ms. The LSTM-based design has fewer parameters than the CNN. However, the CNN-based *D-SCAN* is faster by its design because data in the CNN can be processed in parallel, whereas data in the LSTM must be processed sequentially.

4. *D-SCAN* use cases

In this section, we present three use cases in which *D-SCAN* overcomes the limitations caused by the coexistence of Wi-Fi and Bluetooth on a single device. Specifically, we demonstrate the utility of *D-SCAN* in providing high performance in Wi-Fi operations and Bluetooth transmissions.

4.1. Wi-Fi scanning

Passive Scan: With passive Wi-Fi scanning, a device listens for beacon frames periodically broadcasted by APs in the vicinity on all channels. This method consumes considerable time and energy as the device must stay on each channel at least for a beacon interval to ensure that all APs are detected. However, there are often a small number of Wi-Fi channels with active APs, as we observed in Fig. 1(b), and operating an expensive Wi-Fi radio on channels in which no AP exists is wasteful.

D-SCAN, to suppress unnecessary Wi-Fi scanning on AP-free channels, triggers *selective* passive scanning. The information regarding AP presence on each Wi-Fi channel is acquired by *D-SCAN* quickly and efficiently because of the low-cost Bluetooth spectrum scanning.

Active Scan: With active Wi-Fi scanning, a device actively broadcasts probe requests on all channels to collect probe responses from nearby APs. This method has a considerably shorter latency compared with passive scanning as the device is not required to wait for beacon frames. However, when Wi-Fi APs are densely installed on certain channels (e.g. 1, 6, and 11) to avoid inter-channel interference, active scans may substantially deteriorate the quality of service (QoS) of Wi-Fi networks by overloading those channels with probe messages [32–35]. Moreover, active scans, which are periodically invoked from client devices to ensure reliable wireless connectivity, increase the Wi-Fi latency.

Table 3
PHY rate and Rx sensitivity (802.11n).

PHY rate (Mbps)	6.5	13	19.5	26	39	52	58.5	65
Rx sensitivity (dBm)	-94	−91.7	−89.2	−86.1	−82.5	−77.9	−76.3	−74.7

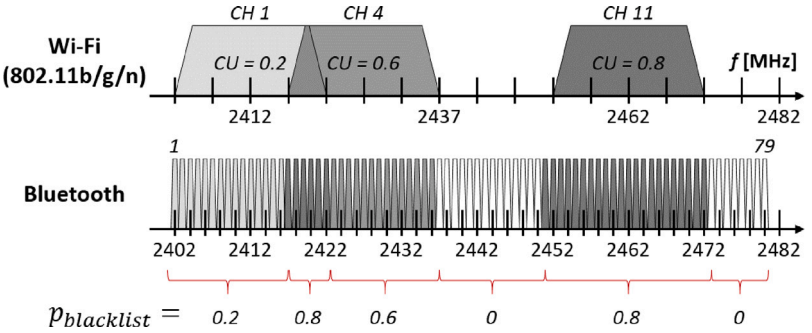


Fig. 8. Probabilities of Bluetooth channels to be blacklisted ($p_{blacklist}$) by *D-SCAN* based on Wi-Fi channel usage.

To reduce spectrum wastage and to provide shorter active scanning delay, *D-SCAN* limits the number of probe messages. *D-SCAN* utilizes the Wi-Fi signal strength and channel utilization to rank Wi-Fi channels and selectively scans a few of them, e.g., top-three channels, which are considered to have the most suitable APs to associate with. The assessment of Wi-Fi channels is based on the achievable throughput described in Section 4.2.

D-SCAN is invoked immediately before the scheduled Wi-Fi scans, both passive and active, to obtain the Wi-Fi information in advance. Subsequently, the reduced use of Wi-Fi radio owing to the selective scan significantly reduces scanning latency and energy consumption.

4.2. Wi-Fi handover

Wi-Fi handover is the process of a client device disconnecting from the currently associated AP and re-associating with another AP. In Wi-Fi, the handover mechanism primarily depends on signal strengths. However, an AP with a higher RSSI value does not always provide higher bandwidth when its channel is saturated with traffic from connected devices. Therefore, channel utilization must also be considered. We computed the achievable throughput (AT) as in BLEND [14]. Although BLEND required Wi-Fi/Bluetooth combo AP to broadcast channel utilization in the Bluetooth advertising packets, *D-SCAN* can acquire this information using simple and short sweeps over the spectrum. The achievable throughput is computed by multiplying the PR by the idle channel ratio:

$$AT = PR \cdot (1 - CU) \tag{3}$$

where PR is the achievable PHY rate approximated via measured RSSI and $(1 - CU)$ is the idle channel ratio, which is the opposite concept of channel utilization (CU). Table 3 lists the PHY rates of 802.11n Wi-Fi and their Rx sensitivities, which are the minimum RSSIs that satisfy the 90% packet delivery ratio (PDR). We consider the PR to be the maximum PHY rate that satisfies 90% of PDR given the estimated RSSI. When a device is in the disconnected state, *D-SCAN* connects it to the AP with the largest AT. During Wi-Fi handover in which a device is already connected to an AP, *D-SCAN* re-associates it to the AP with AT greater than AT of the current AP by Δ.

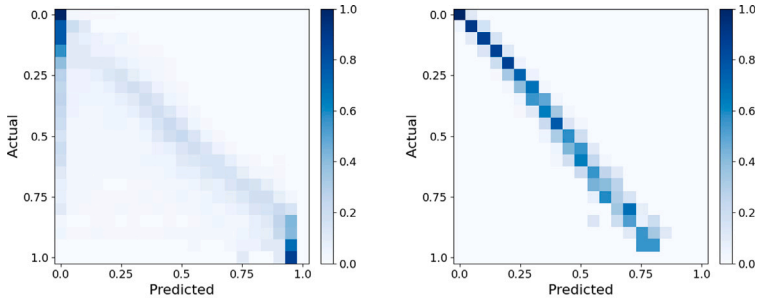
$$AT_{current} + \Delta < AT_{target} \tag{4}$$

$$\Delta = (1 - CU_{current}) \cdot \delta \tag{5}$$

The threshold (Δ) acts as hysteresis which prevents repetitive and rapid switching between two APs with similar ATs. That is, a Wi-Fi device attached to an AP maintains the connectivity until it finds another AP with AT large enough to trigger the handover. The value of δ is set to 6.5 Mbps in 802.11n scenarios as it is the minimum throughput difference between the PHY rates. Different PHY rate values can also be employed depending on the embedded Wi-Fi chipset.

4.3. Synergy for Wi-Fi and Bluetooth coexistence

Most modern devices equipped with combo-modules experience significant performance degradation in a congested environment in which Wi-Fi and Bluetooth are used together, as mentioned in Section 2.2. The main reason is that the increased number of retransmission attempts of Bluetooth radio owing to CTI can reduce the time that Wi-Fi radio uses the shared 2.4 GHz antennas.



(a) Signal strength (b) Channel utilization

Fig. 9. Normalized confusion matrices of the channel inspector.

Through spectrum analysis, *D-SCAN* acquires Wi-Fi channel information, which is a major factor impeding Bluetooth communication. Thus, we use *D-SCAN* to promptly update the AFH map and prevent Bluetooth collisions with the Wi-Fi signals. As shown in Fig. 8, the collision probabilities of each Bluetooth channel can be estimated using the information of corresponding Wi-Fi channels. Subsequently, *D-SCAN* blacklists each Bluetooth channel with the collision probability, dubbed $p_{blacklist}$. This strategy can be easily applied by using an existing command implemented in Bluetooth's host controller interface (HCI) [6]. We invoke *D-SCAN* to update the AFH map every 5 s in scenarios in which Wi-Fi and Bluetooth are used simultaneously.

5. Performance evaluations

We demonstrate the accuracy of *D-SCAN*'s prediction of signal strength and channel utilization. We then establish the performance gain of *D-SCAN* on three practical use cases: Wi-Fi scan, handover, and Bluetooth coexistence. Unless otherwise stated, the results are of CNN-based design, as it performed better than LSTM in many experimental scenarios. *D-SCAN* was implemented on an Ubuntu 18.04 laptop equipped with Qualcomm Atheros AR5B22 chipset. We modified the ath9k device driver of backports 5.6.8-1 [36] to operate *D-SCAN*. Ubertooth [37], an open-source Bluetooth platform equipped with a CC2400 transceiver, was attached to the laptop through a USB port for RSSI sampling via Bluetooth radio.

5.1. Training and testing data collection

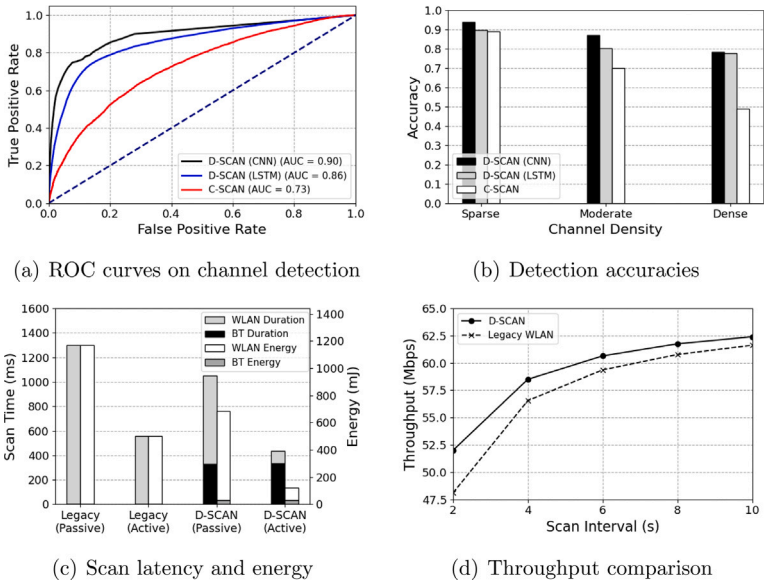
To demonstrate the feasibility of *D-SCAN*, we collected data and conducted experiments from real-world environments. We visited various places throughout Seoul to collect diverse data. Specifically, the data was obtained from 168 different spots of over 50 measurement sites, including cafes, restaurants, malls, parks, university buildings, subway stations, and apartments. To enrich the training data for each measurement site, we collected time-series RSSI from five sampling points for 100 ms on 13 different target frequency ranges that covered the entire 2.4 GHz spectrum. We labeled the samples with ground-truth Wi-Fi scan results, which included the list of APs residing in each channel and their RSSIs. In addition, we appended the channel utilization measured using multiple synchronized Ubertooth devices. The RSSI and channel utilization labels were normalized to values between 0 and 1. The size of the dataset reached about 100,000, and we performed five-fold cross-validation for parameter tuning and performance evaluations. Since the training dataset encompasses Wi-Fi signal characteristics under various environments, the trained DNN can handle Wi-Fi signals in any new location. Subsequent experiments evaluate the performance of *D-SCAN* using a separate test dataset. In the future, federated learning techniques, where a central server and edge devices collaborate to effectively form a rich dataset and train models, can be applied to further refine *D-SCAN*.

5.2. Signal strength and channel utilization

Fig. 9 depicts the accuracy of *D-SCAN* in estimating strongest signal strength (*SS*) and channel utilization (*CU*). We used 20×20 normalized confusion matrices to visually represent how close the predicted values are to the actual ones. *D-SCAN* accurately predicted the absence of a Wi-Fi signal. Weak signals with an RSSI less than 0.25 (−80 dBm) were considered absent. The SS prediction was effective when the Wi-Fi signal was very strong at approximately 1 (−20 dBm). Predictions of moderate signal strength deviated slightly but remained valid with the average loss of ∼0.07 (5.6 dBm). CU predictions clearly revealed a diagonal on the confusion matrix, implying a high degree of accuracy. It indicated loss values lower than 0.02 in most cases. Inferring the CU of a Wi-Fi channel is considered a relatively easy problem, as it is analogous to determining the proportion of RSSI samples affected by Wi-Fi signals out of the total samples. The overall performance of *D-SCAN* based on CNN and LSTM, which measures the average magnitude of the inference errors in terms of root mean square error (RMSE) and mean absolute error (MAE), is shown in Table 4. The hyperparameters of CNN and LSTM models were fine-tuned to take balance between the model size, the number of FLOPs, and accuracy as described in Section 3.5.5.

Table 4
Evaluation of signal strength and channel utilization predictions by CNN- and LSTM-based *D-SCAN*.

	CNN-SS	LSTM-SS	CNN-CU	LSTM-CU
RMSE	0.152	0.163	0.019	0.025
MAE	0.069	0.081	0.011	0.015



(a) ROC curves on channel detection

(b) Detection accuracies

(c) Scan latency and energy

(d) Throughput comparison

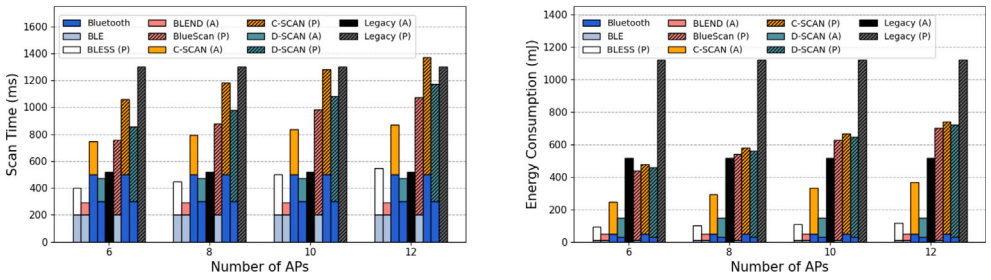
Fig. 10. *D-SCAN* accurately detects active Wi-Fi channels. Selective scanning on the identified channels significantly reduces scan latency and energy consumption of legacy Wi-Fi. The reduced impact of scanning on transmissions relieves the performance degradation in throughput.

5.3. Wi-Fi scanning

We evaluate the performance of *D-SCAN* in determining the AP presence of each channel. If it accurately identifies the active Wi-Fi channels, the performance of Wi-Fi scanning in terms of latency, energy, and throughput can be improved. We used the full scan result of legacy Wi-Fi as a ground-truth evaluation criterion. We compared the performance with that of the latest C-SCAN method [18], which also detects Wi-Fi APs using a Bluetooth radio. C-SCAN determines the AP presence based on a heuristically designed likelihood score; it records the length and number of occurrences of wideband signals to determine whether they are Wi-Fi signals.

D-SCAN determines that an AP exists in the corresponding channel when the estimated Wi-Fi signal strength exceeds a certain threshold. The receiver operating characteristic (ROC) curve is depicted in Fig. 10(a) to determine the threshold and to compare the diagnostic ability of each method. The false positive rate (FPR) and true positive rate (TPR) are plotted against the variation in the discrimination thresholds. A good classifier yields points closer to the upper left corner. *D-SCAN* (CNN) was the finest, followed by *D-SCAN* (LSTM) and C-SCAN, with area under the curves (AUCs) of 0.9, 0.86, and 0.73, respectively. We then compared the accuracy after each method adopted the threshold that exhibited the highest accuracy (Fig. 10(b)); each model used 0.025, 0.1, and 26 for the thresholds. The overall accuracies of the models were 0.9, 0.84, and 0.77, respectively. We classified the real-world test data into *sparse*, *moderate*, and *dense* scenarios according to the number of active channels. The sparse, moderate, and dense scenarios contained APs in 0–4, 5–9, and 10–13 channels, respectively. All methods had difficulties in determining the AP presence as the density increased. Among them, *D-SCAN* (CNN) exhibited the highest accuracy regardless of channel density and it was about 40% higher than that of C-SCAN in the worst-case scenario. *D-SCAN* (LSTM) exhibited good resistance to changing densities as a higher channel density provides more temporal patterns. C-SCAN, a fixed heuristic protocol, did not appear to adapt effectively to diverse signal patterns. *D-SCAN* (CNN) performed excellently, as it focuses more on spectral characteristics; the cluster shape in spatial coordinates facilitates the detection of Wi-Fi signals over noises.

We next observed scan latency and energy consumption in both passive and active scenarios (Fig. 10(c)). In the passive scan experiment, *D-SCAN* spent ~300 ms to perform a full scan and triggered Wi-Fi to selectively scan on active Wi-Fi channels. *D-SCAN* detected an average of 7.2 active channels, resulting in a total scan latency of about 1000 ms, which was a 23% time-saving. In particular, the performance gain of *D-SCAN* in terms of energy was even more pronounced owing to the employment of inexpensive Bluetooth instead of Wi-Fi. Bluetooth acquired channel information with small energy of about 20 mJ and reduced the energy consumption by about 45% compared with legacy Wi-Fi. In the active scan, the legacy Wi-Fi showed short scan latency with active probe exchanges. *D-SCAN* achieved a lower scan latency by exchanging probe messages only on up to three channels. By further restricting the use of the Wi-Fi radio, it consumed 86% less energy.



(a) Scan latency comparison. (b) Energy consumption comparison.

Fig. 11. Experimental results of different Wi-Fi scan methods.

Repetitive use of *D-SCAN* obtains fresh Wi-Fi channel information and contributes to sustained improvement of Wi-Fi connection. Because *D-SCAN* is lightweight in terms of latency and energy, these repeats of *D-SCAN* after the initial connection complements the periodic active scan. Fig. 10(d) presents the throughput measurements of *D-SCAN* and legacy Wi-Fi using different intervals for active scans. As *D-SCAN* eliminates unnecessary overheads in scan operations, an adverse impact of periodic scan on Wi-Fi communication (i.e., throughput) is reduced. The shorter the scan interval, the greater the throughput gain.

Additional Experiments: We further investigated several prior selective Wi-Fi scanning methods, BLESS [15], BLEND [14], BlueScan [17], and C-SCAN [18], to evaluate the performance of *D-SCAN* on acquiring Wi-Fi information. These methods commonly exploit Bluetooth radio to obtain Wi-Fi information and suppress unnecessary scanning attempts, offsetting the overhead in legacy Wi-Fi scanning.

BLESS and BLEND require Wi-Fi and Bluetooth combo functionality on both the AP and client sides. The combo APs embed Wi-Fi beacon information into Bluetooth low energy (BLE) advertising packets, which are periodically broadcasted to nearby stations. A station directly acquires nearby Wi-Fi AP information by collecting BLE advertising packets for 200 ms. In contrast, BlueScan, C-SCAN, and *D-SCAN* infer Wi-Fi information from the RSSI traces that are sequentially measured using Bluetooth radio. BlueScan, which attempts to spot beacon frames by observing periodicity in RSSI peaks, requires more than 2.5 s to fully scan 13 Wi-Fi channels in the 2.4 GHz band. C-SCAN and *D-SCAN*, because they analyze three and five Wi-Fi channels at once within 100 ms, requires 500 and 300 ms to scan 13 Wi-Fi channels, respectively.

Upon acquisition of Wi-Fi information, BLESS computes the optimal scanning sequence for beacon receptions and performs passive scans on the right channel at the right moment; otherwise, it stays in idle mode. BLEND estimates the achievable throughput of each AP with signal strength and channel utilization and performs an active scan on a single channel with the best AP. BlueScan, C-SCAN, and *D-SCAN* simply perform either passive or active scans on channels, which are identified to have active Wi-Fi APs. As an exception in the active scan mode, *D-SCAN* only scans top-three Wi-Fi channels considered to have good link quality as it can also rank Wi-Fi channels by inferred signal strength and channel utilization (similar to BLEND).

Fig. 11(a) and Fig. 11(b) depict the simulation results on scan latency and energy consumption, respectively, as the number of APs varies from 6 to 12. Here, the average numbers of occupied Wi-Fi channels are 4.96, 6.16, 7.16, and 8.03. In the experiment, as BlueScan required a significant amount of time to obtain Wi-Fi information (more than x2.5 that of legacy), we assumed that BlueScan simply represents the selective passive scanning approaches and, similar to BLESS and BLEND, exploits the BLE advertising packets. Here, note that the scan time of BLESS depends on the number of APs, whereas the scan time of BlueScan, C-SCAN, and passive *D-SCAN* depends on the number of occupied Wi-Fi channels. In contrast, the scan time of BLEND and active *D-SCAN* is constant as they scan on a fixed number of channels.

Both passive (P) and active (A) scanning methods shorten the duration of Wi-Fi scanning, which saves a significant amount of energy. The scan latency of BLESS, which is a passive scanning method, is very short as for other active (A) scanning methods, BLEND, C-SCAN (A), *D-SCAN* (A), and legacy (A), as it optimizes the scanning sequence from the beacon timing information. However, the scan latency of BLESS increases proportionally to the number of APs to discover all the APs. The scan latency of selective passive scanning methods, BlueScan (P), C-SCAN (P), and *D-SCAN* (P), depends on the number of occupied Wi-Fi channels, and it can be inferred from Fig. 11(a) that unnecessary scan operations on the empty Wi-Fi channels are effectively suppressed. In terms of energy, BLESS (P), BLEND (A), and BlueScan (P) utilize an efficient BLE chipset. However, because C-SCAN and *D-SCAN* can also employ BLE radio, the performance gap can be overcome.

In summary, each scanning method has advantages and disadvantages that should be further investigated in future research. The selective scanning methods that employ combo APs, such as BLESS and BLEND, are rapid and energy-efficient. However, they are not practical as widely deploying APs with combo functionality is difficult. C-SCAN and BlueScan require no additional infrastructure, but either the scanning requires an excessive amount of time or the scan results are not sufficiently accurate. Meanwhile, *D-SCAN* is a promisingly practical and reliable solution as it balances the deployment cost and performance.

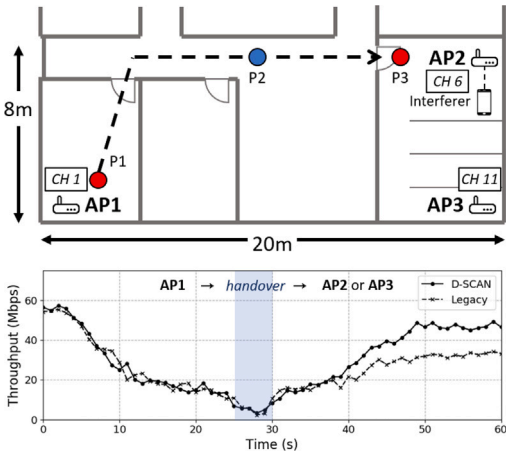


Fig. 12. *D-SCAN* selects an AP with the best achievable throughput during handover by taking both SS and CU into account.

5.4. Wi-Fi handover

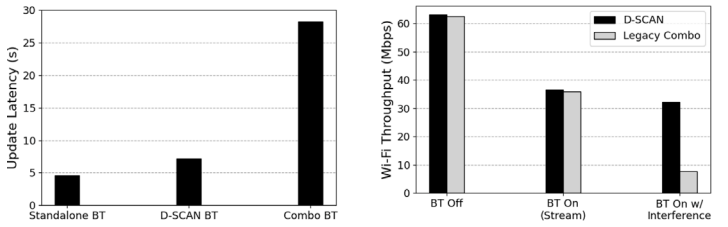
We examined the performance improvement of using *D-SCAN* during Wi-Fi handover in a controlled office environment, where three APs were configured to use non-overlapping channels (Fig. 12). A target Wi-Fi device was moved from P1 to P3, and we observed the throughput changes as it initially connected to AP1 and re-associated to either AP2 or AP3 at P2. We adjusted the transmit power and locations of AP2 and AP3 to observe respective RSSIs of −50 and −80 dBm during the handover. In addition, we deployed an interferer that occupied the medium of AP2, which had a relatively better signal quality. Then, we adjusted the medium occupancy, CU, of the interferer to 0.1, 0.2, 0.3, . . . , and 0.9 in each scenario to observe the impact of CU on Wi-Fi channel selection during the handover. Because the legacy Wi-Fi handover method selects an AP based on RSSI regardless of channel conditions, the presence of the interferer did not affect the selection. As a result, it always selected AP2 and the throughput was limited to 32.7 Mbps on average. *D-SCAN* selected AP2 when the interferer CU on AP2 was less than 0.3, and it selected AP3 afterward, achieving an average throughput of 46.8 Mbps. *D-SCAN* exhibited 43% higher throughput compared with the legacy as it used the Wi-Fi channel information obtained to estimate the achievable throughput and select a better AP.

5.5. Synergy for Wi-Fi and Bluetooth coexistence

A unique advantage of *D-SCAN* is its ability to synergistically improve Bluetooth (BT) performance in combo devices. We first compared how standalone Bluetooth, combo-module Bluetooth, and combo-module Bluetooth aided by *D-SCAN* respond to interference. For the experiment, we generated a jamming signal over a 20 MHz-wide Wi-Fi channel to impede Bluetooth and measure the latency from the beginning of the jamming until the AFH map successfully excluded the Wi-Fi channel from Bluetooth's hopping sequence.

The result in Fig. 13(a) shows that combo Bluetooth experienced an AFH map update latency more than six times longer, whereas the combo Bluetooth with *D-SCAN* only incurred a small additional delay of about 3 s. The reasons *D-SCAN* combo Bluetooth shows a slightly longer delay than standalone Bluetooth are (i) the overheads of executing *D-SCAN* and (ii) the delay between issuing the Bluetooth HCI command to change the AFH map and completing the master–slave handshaking. Unlike standalone Bluetooth, where Bluetooth is always active, combo Bluetooth operates with Wi-Fi in a time-division manner, which makes achieving precise channel quality assessment difficult. However, with the aid of *D-SCAN*, the AFH map of combo Bluetooth can also be rapidly updated to mitigate CTI using detailed information about Wi-Fi channels.

If the AFH map is not updated properly in the presence of interference, Bluetooth encounters collisions and performs retransmission attempts. This increases the portion of the medium occupied by Bluetooth and eventually deteriorates Wi-Fi performance as well, shortening the medium occupation by Wi-Fi. We conducted an experiment to evaluate the impact of Bluetooth/Wi-Fi coexistence (Fig. 13(b)). We measured the saturated Wi-Fi throughput under three different scenarios: (i) only Wi-Fi in a static environment, (ii) Wi-Fi and Bluetooth in a static environment, and (iii) Wi-Fi and Bluetooth in a dynamic environment. When Bluetooth was off in a static environment, combo Wi-Fi with and without *D-SCAN* achieved its maximum data rate. However, when we turned on Bluetooth to stream music on a headset, the throughput was almost halved. Finally, when we alternately generated jamming signals on neighboring Wi-Fi channels every 30 s, the performance gap between combo Wi-Fi with and without *D-SCAN* became prominent. The music streamed on a headset with the legacy combo Bluetooth experienced high levels of noise and delay throughout the experiment, and Wi-Fi throughput decreased significantly to 8 Mbps, 26.7% compared with the interference-free scenario. The combo Bluetooth with *D-SCAN* also experienced noise and delay early in the interference but subsequently stabilized as the updated AFH map promptly responded to the CTI. Thus, we confirmed that *D-SCAN* improves the immunity of the combo-module to interference and synergistically advances Bluetooth and Wi-Fi performance. Other IoT protocols that require cost-effective spectrum analysis can also employ the variants of *D-SCAN*.



(a) AFH map update latency (b) Wi-Fi throughput on different combo-module scenarios

Fig. 13. Prompt AFH map update prevents performance degradation in Bluetooth/Wi-Fi combo-modules.

6. Discussion

6.1. Sub-1 GHz and 5 GHz Wi-Fi

Wi-Fi, depending on the amendment and usage, coexists with other communication protocols in various frequency bands. The applicability of *D-SCAN* extends beyond the 2.4 GHz band.

In other frequency bands, Wi-Fi signal characteristics are also distinct. In Sub-1 GHz unlicensed band, 802.11ah Wi-Fi channel widths (1, 2, 4, 8, and 16 MHz) are wider than those of coexisting low-power wide-range protocols such as LoRa (<500 kHz) and Sigfox (100 Hz). In 5 GHz band, 802.11ac Wi-Fi has different channel widths, center frequencies, and/or medium access schemes compared to those of other coexisting heterogeneous protocols such as LTE, radar, and dedicated short-range communication (DSRC), and is possibly distinguished. Similarly, 802.11ax, the new generation of Wi-Fi standards, inherits these channel characteristics from both 2.4 and 5 GHz bands and also operates in the much wider 6 GHz band. In the future, we will generalize *D-SCAN* by arranging the sampling points with an appropriate space for Wi-Fi existing in other bands, collecting RSSIs with a low-power radio [38], and learning temporal and spatial characteristics of Wi-Fi signals.

6.2. Bonded Wi-Fi channels

Since 802.11n, the concept of channel bonding in which two adjacent channels in 2.4 and 5 GHz spectrums are combined to increase the data rate has been implemented. In the 2.4 GHz band, the channel width can be up to 40 MHz, and in the 5 GHz band, it can be up to 160 MHz.

Technically, 40 MHz signals can be detected by *D-SCAN*, because they also leave their traces on successive four edge projections. The strength of deep learning methods is that they learn features on their own given data with appropriate labels. In the worst case, bonded channels may produce false-positive detections, and the inspection range of *D-SCAN* should be extended to 40 MHz to prevent this. However, because the presence of Bluetooth, Zigbee, and Wi-Fi on other overlapping channels often limits the availability of channel bonding, the impact on *D-SCAN*'s detection accuracy is negligible.

6.3. Differentiation of APs and clients

D-SCAN detects Wi-Fi signals effectively, but cannot distinguish whether they are from APs or clients. This is a type of exposed node problem and can cause *D-SCAN* devices to make false-positive decisions on AP presence; however, the rate was kept low in our experiment. The detection of close-enough Wi-Fi clients suggests that the AP it communicates with can also empirically provide satisfactory service to the *D-SCAN* device. In particular, the estimated channel utilization is valid regardless of the source of the signal, providing a good inference on the achievable throughput.

7. Conclusion

In this paper, we present *D-SCAN* that efficiently acquires comprehensive Wi-Fi channel information using a Bluetooth radio and synergistically improves the performance of both Wi-Fi and Bluetooth. Our results from three use cases: Wi-Fi scanning, handover, and the coexistence of Bluetooth with Wi-Fi, show this data-driven approach is promising. The well-defined problem of *D-SCAN* and novel RSSI representation method, called edge projection, *D-SCAN* enables a CNN-based deep learning model to deliver robust predictions on Wi-Fi channel information even in various network dynamics. In the future, we plan on applying federated learning techniques, where a central server and edge devices collaborate for data collection and model training, to make *D-SCAN* more effective in various environments. Furthermore, we anticipate that the positive impact of *D-SCAN* to extend beyond these two wireless technologies and beyond the 2.4 GHz band. The general strategy of using low-power wireless radios to accurately collect broad information about more energy-intensive wireless technology can be applied to other ISM bands including the widely used Sub-1, 2.4, and 5 GHz.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jaehyuk Choi reports financial support was provided by National Research Foundation of Korea. The co-author has published an earlier version of the work as a dissertation - Junhyun Park * J. Park, Environment-Aware Resource Management Strategies in IoT Protocols, Dissertation, Seoul National University, 2022.

Data availability

Data will be made available on request.

Acknowledgements

This research was supported in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2020-0-01602, IITP-2021-0-02048) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation), and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2020R1A2C1013308).

References

[1] M. Hou, F. Ren, C. Lin, M. Miao, HEIR: Heterogeneous interference recognition for wireless sensor networks, in: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, IEEE, 2014, pp. 1–9.
[2] C.-F. Chiasserini, R.R. Rao, Coexistence mechanisms for interference mitigation in the 2.4-GHz ISM band, IEEE Trans. Wireless Commun. 2 (5) (2003) 964–975.
[3] C.d.M. Cordeiro, S. Abhyankar, R. Toshiwal, D.P. Agrawal, BlueStar: Enabling efficient integration between bluetooth WPANs and IEEE 802.11 WLANs, Mob. Netw. Appl. 9 (4) (2004) 409–422.
[4] X. Zhang, K.G. Shin, Gap sense: Lightweight coordination of heterogeneous wireless devices, in: 2013 Proceedings IEEE INFOCOM, IEEE, 2013, pp. 3094–3101.
[5] Y. Wang, Q. Wang, G. Zheng, Z. Zeng, R. Zheng, Q. Zhang, WiCop: Engineering WiFi temporal white-spaces for safe operations of wireless personal area networks in medical applications, IEEE Trans. Mob. Comput. 13 (5) (2013) 1145–1158.
[6] Specification of the bluetooth system, covered core package version: 4.2, The bluetooth Special Interest Group (SIG), 2013.
[7] L. Ophir, Y. Bitran, I. Sherman, Wi-Fi (IEEE 802.11) and bluetooth coexistence: Issues and solutions, in: 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No. 04TH8754), Vol. 2, IEEE, 2004, pp. 847–852.
[8] A.E. Xhafa, Y. Sun, Mechanisms for coexistence of collocated WLAN and bluetooth in the same device, in: 2013 International Conference on Computing, Networking and Communications, ICNC, IEEE, 2013, pp. 905–910.
[9] A.E. Xhafa, X. Lu, D.P. Shaver, Coexistence of collocated ieee 802.11 and bluetooth technologies in 2.4 GHZ ISM band, in: International Conference on Access Networks, Springer, 2008, pp. 138–145.
[10] T. Kessler, Turn off bluetooth to fix airplay mirroring bug in OS X, 2014, http://www.cnet.com/how-to/.
[11] N. Mishra, K. Chebrolu, B. Raman, A. Pathak, Wake-on-WLAN, in: Proceedings of the 15th International Conference on World Wide Web, 2006, pp. 761–769.
[12] N. Mishra, D. Golcha, A. Bhadauria, B. Raman, K. Chebrolu, S-WOW: Signature based wake-on-WLAN, in: 2007 2nd International Conference on Communication Systems Software and Middleware, IEEE, 2007, pp. 1–8.
[13] K. Chebrolu, A. Dhekne, Esense: Energy sensing-based cross-technology communication, IEEE Trans. Mob. Comput. 12 (11) (2012) 2303–2316.
[14] J. Choi, G. Lee, Y. Shin, J. Koo, M. Jang, S. Choi, Blend: BLE beacon-aided fast WiFi handoff for smartphones, in: 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON, IEEE, 2018, pp. 1–9.
[15] W. Park, D. Ryoo, C. Joo, S. Bahk, BLESS: BLE-aided Swift Wi-Fi Scanning in Multi-protocol IoT Networks, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.
[16] R. Zhou, Y. Xiong, G. Xing, L. Sun, J. Ma, ZiFi: Wireless LAN discovery via ZigBee interference signatures, in: Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking, 2010, pp. 49–60.
[17] J. Yi, W. Sun, J. Koo, S. Byeon, J. Choi, S. Choi, BlueScan: Boosting Wi-Fi scanning efficiency using bluetooth radio, in: 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON, IEEE, 2018, pp. 1–9.
[18] J. Chung, J. Park, C.-K. Kim, J. Choi, C-SCAN: Wi-Fi scan offloading via collocated low-power radios, IEEE Internet Things J. 5 (2) (2018) 1142–1155.
[19] J. Choi, WidthSense: Wi-Fi discovery via distance-based correlation analysis, IEEE Commun. Lett. 21 (2) (2016) 422–425.
[20] Ieee 802.11, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification, 2012, IEEE Std..
[21] X. Chen, D. Qiao, HaND: Fast handoff with null dwell time for IEEE 802.11 networks, in: 2010 Proceedings IEEE INFOCOM, IEEE, 2010, pp. 1–9.
[22] A. Graves, N. Jaitly, A.-r. Mohamed, Hybrid speech recognition with deep bidirectional LSTM, in: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, IEEE, 2013, pp. 273–278.
[23] R. Johnson, T. Zhang, Supervised and semi-supervised text categorization using LSTM for region embeddings, in: International Conference on Machine Learning, PMLR, 2016, pp. 526–534.
[24] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, D. Yang, Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.
[25] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[27] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, K. Chowdhury, Oracle: Optimized radio classification through convolutional neural networks, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 370–378.
[28] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv preprint arXiv:1704.04861.

[29] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[31] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[32] W. Wang, M. Motani, V. Srinivasan, Opportunistic energy-efficient contact probing in delay-tolerant applications, IEEE/ACM Trans. Netw. 17 (5) (2009) 1592–1605.
[33] X. Hu, L. Song, D. Van Bruggen, A. Striegel, Is there WiFi yet? How aggressive probe requests deteriorate energy and throughput, in: Proceedings of the 2015 Internet Measurement Conference, 2015, pp. 317–323.
[34] D. Jaisinghani, V. Naik, S.K. Kaul, S. Roy, Realtime detection of degradation in WiFi network's goodput due to probe traffic, in: 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt, IEEE, 2015, pp. 42–47.
[35] H.D. Balbi, D. Passos, R.C. Carrano, L.C. Magalhães, C.V. Albuquerque, Association stability and handoff latency tradeoff in dense IEEE 802.11 networks: A case study, Comput. Commun. 159 (2020) 175–185.
[36] Linux backports website. https://backports.wiki.kernel.org/.
[37] Ubertooth-one, http://ubertooth.sourceforge.net/.
[38] M. Hessar, A. Najafi, V. Iyer, S. Gollakota, TinySDR: Low-power SDR platform for over-the-air programmable IoT testbeds, in: 17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 20, 2020, pp. 1031–1046.

Junhyun Park received his B.S. degree in computer science from University of Illinois at Urbana-Champaign, IL, USA, in 2013 and master's and Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2022. His current research interests include wireless networks, specifically IEEE 802.11 WLAN and Internet of Things, deep learning-based networking, and social networks security.

Junyoung O. Park is an Assistant Professor of Chemical and Biomolecular Engineering and Co-Director of the Metabolomics Center at UCLA. His research group focuses on systems-level analysis of metabolic networks using multi-omics data and machine learning to elucidate regulatory mechanisms and engineer metabolism. He aims to apply this knowledge to solving energy and environmental problems and curing human diseases such as cancer and diabetes. Before moving to Los Angeles, he conducted postdoctoral research at MIT. He received his bachelor's degrees in mathematics and bioengineering from UC San Diego and a master's and PhD in chemical engineering from Princeton University.

Jaehyuk Choi (GS'07–M'11) received the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2008. He is currently an Associate Professor with the Department of Software, Gachon University, Seongnam, South Korea. He was with the Real-Time Computing Laboratory, University of Michigan, Ann Arbor, MI, USA, as a Post-Doctoral Researcher from 2008 to 2011. His current research interests include wireless/mobile systems, Internet of Things connectivity and sensing systems, with an emphasis on wireless LAN/PAN, edge computing, multipath TCP, network management, and ubiquitous passive sensing.

Ted Taekyong Kwon received the BS, MS, and PhD degrees from Seoul National University (SNU) in 1993, 1995, and 2000, respectively. He is a professor with the Department of Computer Science and Engineering, Seoul National University. Before joining SNU, he was a postdoctoral research associate at the University of California Los Angeles and City University New York. During his graduate program, he was a visiting student at the IBM T.J. Watson Research Center and at the University of North Texas. He was a visiting professor at Rutgers University in 2010. His research interest lies in future Internet, network security, and wireless networks.