Physical Layer-Based IoT Security: An Investigation Into Improving Preamble-Based SEI Performance When Using Multiple Waveform Collections

JOSHUA H. TYLER\(^1\), (Student Member, IEEE), DONALD R. REISING\(^1\), (Senior Member, IEEE), FARAH I. KANDAH\(^2\), (Senior Member, IEEE), ERKAN KAPLANOGLU\(^3\), (Senior Member, IEEE), AND MOHAMED M. K. FADUL\(^1\)

\(^1\)Department of Electrical Engineering, The University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA
\(^2\)Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA
\(^3\)Department of Engineering Management and Technology, The University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA

Corresponding author: Donald R. Reising (donald-reising@utc.edu)

ABSTRACT The Internet of Things (IoT) is a collection of inexpensive, semi-autonomous, Internet-connected devices that sense and interact within the physical world. IoT security is of paramount concern because most IoT devices use weak or no encryption at all. This concern is exacerbated by the fact that the number of IoT deployments continues to grow, IoT devices are being integrated into key infrastructures, and their weak or lack of encryption is being exploited. Specific Emitter Identification (SEI) is being investigated as an effective, cost-saving IoT security approach because it is a passive technique that uses inherent, distinct features that are unintentionally imparted to the waveform during its formation and transmission by the IoT device’s Radio Frequency (RF) front-end. Despite the amount of research conducted, SEI still faces roadblocks that hinder its integration within operational networks. Our work focuses on the lack of feature permanence across time and environments, which is designated herein as the “multi-day” problem. We present results and analysis for six distinct experiments focused on improving multi-day SEI performance through multiple waveform representations, deeper Convolutional Neural Networks (CNNs), increasing numbers of waveforms, channel model impacts, and two-channel mitigation techniques. Our work shows improved multi-day SEI performance using the waveform’s frequency-domain representation and a CNN comprised of four convolutional layers. However, the traditional channel model and both channel mitigation techniques fail to sufficiently mitigate or remove real-world channel impacts, which suggests that the channel may not be the dominant effect hindering multi-day SEI performance.

INDEX TERMS Deep learning, specific emitter identification (SEI), convolutional neural network (CNN), communications, Internet of Things (IoT), multiple collection SEI.

I. INTRODUCTION

The Internet of Things (IoT) is a collection of inexpensive, semi-autonomous, Internet-connected devices that sense and interact within the physical world [1]. IoT security is of paramount concern because most IoT devices implement weak or no encryption at all [2]. The use of weak or no encryption is attributed to (i) limited on-board resources such as memory and power; (ii) high manufacturing costs that prohibit the integration of effective encryption techniques on low-cost devices; and (iii) difficulties associated with the implementation and management of encryption at the scales often associated with large IoT deployments [3], [4]. The concern surrounding IoT security continues to grow as more...
and more IoT devices are integrated into key infrastructures such as the electric power grid, and abused by nefarious actors [5], [6], [7], [8], [9], [10], [11], [12], [13]. Recently, a physical layer technique known as Specific Emitter Identification (SEI) has received a lot of attention as a possible means through which IoT devices and their associated infrastructure can be secured.

SEI was introduced almost thirty years ago to grant electronic warfare systems the capability of detecting, characterizing, and identifying radar systems using unintentional features present within their emitted signals. These features have been attributed to slight manufacturing and assembly process defects that exist within the components, subsystems, and systems that comprise a radar system’s Radio Frequency (RF) front end. SEI features have been shown to be sufficient enough to achieve serial number discrimination (i.e., the emitters are all of the same manufacturer and model) without their presence negatively impacting normal system operations such as signal detection and demodulation.

The success of radar-focused SEI has resulted in it being recommended as a physical layer mechanism for enhancing existing, higher-layer network security approaches such as username & password and MAC address filtering. SEI proves advantageous because: (i) it is a passive technique so the emitters are unaware they are being identified, (ii) it quantitatively measures the exploited features, (iii) it can ensure information is coming from known, authorized devices instead of unknown, unauthorized devices that can negatively impact the dependability and timeliness of communications within an IoT infrastructure [14], [15], and (iv) the exploited features are generated during normal emitter function so there is no need for modification, stimulation, or augmentation of the emitter to enable identification. The last advantage is ideally suited for IoT applications because it means that SEI-based security can be applied to even the simplest, lowest-cost devices and deployments.

Despite the amount of research conducted, SEI still faces roadblocks that hinder its adoption and integration within operational networks. Our work focuses on addressing one of these roadblocks as it relates to the lack of SEI exploited feature permanence across time and environments [16], [17]. We designate this lack of permanence herein as the “multi-day” problem. Currently, SEI processes are trained/developed using a single set of signals that are collected from each device during a single collection event. However, these SEI processes have been shown to perform poorly when tasked with identifying the same set of emitters using signals collected during a separate event conducted a day or days after the initial set of signals was collected and used to develop the SEI process. In other words, SEI does not currently support emitter discrimination using signals collected over multiple days when the process is trained using a single day’s worth of signals. Although the time between collections can vary from hours to weeks, we designate all such cases as part of the multi-day problem. The contributions of this work are as follows.

- An analysis of time and frequency representations of preambles for improving multi-day SEI performance.
- An analysis of CNN network depth to maximize multi-day SEI performance.
- An analysis of using additive, white Gaussian noise to model wireless interference to improve multi-day SEI performance.
- An analysis of using traditional channel mitigation techniques to improve multi-day SEI performance.

The remainder of this paper is outlined as follows. Sect. II provides a brief description of related SEI works and how our work differs from them. Sect. III describes the signal of interest, signal collection, and post-processing procedures while Sect. IV describes the methodology followed to generate the results presented in Sect. V. Finally, the paper is concluded in Sect. VI.

II. RELATED WORKS

Performing SEI using waveforms collected over multiple days is relatively recent; thus, there are not many published investigations into the multi-day SEI problem. However, this section summarizes those investigations that considered multi-day SEI.

The authors of [17] show that SEI performance degrades when using waveforms collected over multiple days and environments. The authors use two USRP B210 Software-Defined Radios (SDRs) to capture LoRa packets from sixty Pycom emitters transmitting at a carrier frequency of 915 MHz. When classifying the Fast Fourier Transform (FFT) coefficients, of the emitters’ waveforms, an accuracy of 82% is achieved when the training and testing waveforms are collected on the same day. However, accuracy drops to just 7% and 5% when classifying waveforms from the second and third collections. When using the In-phase and Quadrature (IQ) samples an accuracy of 72% is achieved when classifying waveforms collected on the same day as those used for training and drops to 48% and 46% when classifying waveforms from the two subsequent collections. The authors propose the “Tweak” [18] and “WideScan” [19], [20] methods to improve classification performance across collections. However, the authors do not provide results supporting either approach’s effectiveness.

The authors of [21] improve multi-day SEI by reducing the effect of the wireless channel on each emitter’s set of SEI features. The channel is mitigated by implementing IQ balancing and equalization on the payload portion of a BPSK waveform. This increases SEI accuracy from 23% to 34% when training a network on fifty Wireless-Fidelity (Wi-Fi) emitters collected on a single day and tested using waveforms collected on a later date. These emitters are recorded across multiple WiFi-compliant carrier bands and are sampled at 200 MHz. IoT devices may not be capable of achieving such a high sampling frequency. In this work,
we address this by sampling the waveforms at a frequency of 20 MHz.

III. BACKGROUND
A. SIGNAL OF INTEREST
The signal of interest is the IEEE 802.11a Wi-Fi frame, which is an Orthogonal Frequency Division Multiplexing (OFDM) waveform that is capable of data rates as high as 54 Mbps and operates within the 5 GHz Industrial, Scientific, and Medical (ISM) band. The first 16 µs of every IEEE 802.11a Wi-Fi frame is dedicated to the transmission of a preamble that is used for time synchronization, channel compensation, as well as frequency and phase offset correction. Fig. 1 shows the structure of the 802.11a preamble, which is comprised of ten Short Training Symbols (STS) that are designated from $t_1$ through $t_{10}$, a Guard interval (GI), and two Long Training Symbols (LTS) that are designated $T_1$ and $T_2$.

The use of IEEE 802.11a Wi-Fi is motivated by the facts that: (i) Wi-Fi, which includes 802.11a, is an IoT-designated communications standard [22], [23], (ii) it is a preamble-based communications standard that facilitates SEI using a steady-state portion of the signal, (iii) allows the extension to other preamble-based IoT standards such as 802.11ax (a.k.a., Wi-Fi 6), ZigBee, Z-Wave, and Bluetooth, (iv) the IEEE 802.11a preamble and first 16 µs of the Ultra-Reliable Low Latency Communications (URLLC) compliant IEEE 802.11ax preamble are constructed using the same symbols [24], (v) it has been used extensively in our and other published SEI works [25], [26], [27], [28], [29], and [30], and (vi) we have an on-hand set of commercial-off-the-shelf, IEEE 802.11a Wi-Fi compliant emitters.

| Signal Detect, AGC, Diversity Selection | Coarse Freq Offset Estimation | Channel and Fine Freq Offset Estimation | Timing Synchronize |
|----------------------------------------|------------------------------|----------------------------------------|-------------------|
| $10 \times 0.8 = 8\mu s$               | $2 \times 0.8 + 2 \times 3.2 = 8\mu s$ | $T_1$                                 | $T_2$ |

![FIGURE 1. Block diagram of the IEEE 802.11a OFDM Wi-Fi preamble that includes ten short training symbols $t_1$ to $t_{10}$, a guard interval GI, and two long training symbols $T_1, T_2$ that occurs within $16\mu s$.](image)

B. SIGNAL COLLECTION
The signal collection is performed in an open-air laboratory over a period of eight, consecutive weeks. Signals are collected from thirty-two TP-Link Archer T3U USB Wi-Fi emitters that each transmit the same 2 Gb binary file over an IEEE 802.11a Wi-Fi connection. Each week an Ettus Research USRP B210 collects four seconds worth of transmissions from each TP-link emitter at a sampling rate of 40 MHz. Once collected, each recording is gated to remove the channel-only segments that exist between transmissions. Initially, the preamble of every transmission is coarsely detected by finding every point in the recording where the magnitude changes from zero to greater than zero (i.e., the start of each transmission). Once coarsely detected, the preamble—along with thirty-two samples before and after the preamble—is extracted and stored for subsequent fine preamble detection.

C. FINE PREAMBLE DETECTION
After course detection, each preamble is finely detected using the Mean Squared Error (MSE). In this case, the magnitude representation of an ideal 802.11a preamble is slid across the coarsely detected preambles’ magnitude representations and MSE is calculated. Fine detection using the magnitude representation is advantageous because it is not negatively impacted by the presence of Carrier Frequency Offset (CFO), which can adversely impact the average percent correct classification performance of amplitude- and phase-based detection approaches. A preamble is detected when the MSE is at its minimum. For each emitter, this process is repeated for all transmissions within its corresponding collection records and ten thousand preambles are retained. The retained preambles correspond to the smallest MSE values for that emitter.

D. CARRIER FREQUENCY OFFSET CORRECTION
CFO is estimated using the process stated in [31] in which the offset is calculated by,

$$f_{\Delta} = \frac{f_e}{2\pi N} \angle \left( \sum_{j=0}^{N} R_1(j)R_2(j) \right),$$  \hspace{1cm} (1)

where $R_1$ and $R_2$ represent the Fourier coefficients of the first and second training symbols consecutively selected from the received waveform $r[n]$, respectively. $N$ is the length of the training symbol, and * represents the complex conjugate. Once calculated, CFO present in the $j^{th}$ received waveform of emitter $e$ is corrected by,

$$r'_{e,j}(n) = r_{e,j}(n) \exp(-j2\pi f_{\Delta} t) .$$  \hspace{1cm} (2)

All waveforms undergo a two-stage CFO correction process. The first stage performs course CFO correction using equation (1) and the Fourier coefficients of the eighth, $t_8$, and ninth, $t_9$, STS. The second stage performs fine CFO compensation using both LTS sequences, $T_1$ and $T_2$, in lieu of $t_8$ and $t_9$. Following the course and fine CFO compensation, all waveforms are filtered using a fourth-order elliptical filter that has a passband ripple of 0.5 dB, stopband attenuation of 20 dB, and corner frequency of 8.865 MHz. Filtering after CFO correction is advantageous because CFO manifests as a shift in the center frequency of the waveform’s Power Spectral Density (PSD). This shift causes the PSD to be misaligned with the filter’s passband, which can result in unnecessary coloration being applied that can negatively impact SEI performance [32]. All waveforms are then normalized to unit energy [28] by,

$$\tilde{r}_{e,j}(n) = \frac{r'_{e,j}(n)}{\sqrt{\sum_{n=0}^{N_p} |r'_{e,j}(n)|^2}} .$$  \hspace{1cm} (3)
where $N_p$ is the number of samples in the CFO-corrected preamble $r'_{e,j}(n)$, and the result stored for experimentation.

**IV. METHODOLOGY**

**A. NOISE SCALING AND WAVEFORM FILTERING**

Wireless channel interference is modeled using complex-valued preamble $r_{e,j}(n)$ where $\delta_i$ is the desired SNR in decibels. The waveform at the desired SNR is expressed as,

$$r_{e,j}(n) = \bar{r}_{e,j}(n) + \chi w(n),$$  \hspace{1cm} (5)

where $n = 1, 2, \ldots, N_p$ and $w(n)$ is AWGN. The resulting waveform, $r_{e,j}(n)$, is filtered using the same elliptical filter used in Sect. III-D and normalized to unit energy. This process is repeated ten times for all waveforms to ensure all presented numerical results are generated using Monte Carlo methods.

**B. PREAMBLE REPRESENTATIONS**

SEI is performed using either a time or frequency domain-based 802.11a preamble representation that are adopted from our published work in [26] and [28]. The time domain representation is,

$$p^t_{e,j}(m, n, z) = \left[ i_{e,j}(1, 1) \ i_{e,j}(1, 2) \ \cdots \ i_{e,j}(1, N_p) \ q_{e,j}(2, 1) \ q_{e,j}(2, 2) \ \cdots \ q_{e,j}(2, N_p) \ \lambda_{e,j}(3, 1) \ \lambda_{e,j}(3, 2) \ \cdots \ \lambda_{e,j}(3, N_p) \ \theta_{e,j}(4, 1) \ \theta_{e,j}(4, 2) \ \cdots \ \theta_{e,j}(4, N_p) \right]$$  \hspace{1cm} (6)

where $i$ is the in-phase component of the complex-valued preamble, $q$ is the complex-valued preamble’s quadrature-phase component, $\lambda$ is the real part of the complex-valued preamble’s natural logarithm that captures magnitude, and the phase of the complex-valued preamble is captured by the imaginary part of its natural logarithm and is defined here as $\theta$. The frequency domain representation is,

$$p^f_{e,j}(m, k, z) = \left[ I_{e,j}(1, 1) \ I_{e,j}(1, 2) \ \cdots \ I_{e,j}(1, N_p) \ Q_{e,j}(2, 1) \ Q_{e,j}(2, 2) \ \cdots \ Q_{e,j}(2, N_p) \ \Lambda_{e,j}(3, 1) \ \Lambda_{e,j}(3, 2) \ \cdots \ \Lambda_{e,j}(3, N_p) \ \Theta_{e,j}(4, 1) \ \Theta_{e,j}(4, 2) \ \cdots \ \Theta_{e,j}(4, N_p) \right],$$  \hspace{1cm} (7)

where $I$ is the Fourier coefficients’ real component, $Q$ is the Fourier coefficients’ imaginary component, $\Lambda$ and $\Theta$ are the Fourier coefficients’ magnitude and phase, respectively.

**C. EXPERIMENTS CONDUCTED**

A total of six multi-day SEI experiments are conducted using Convolutional Neural Network (CNN) architectures. Our use of a CNN is motivated by it being the most common DL architecture used in SEI experimentation [18], [19], [20], [21], [25], [30] as well as its use in our prior works [26], [27], [28], and [29]. The remainder of this section describes each experiment and provides the details needed to replicate them.

1) EXPERIMENT #1: PREAMBLE REPRESENTATION SELECTION

This experiment focuses on determining the preamble representation combination that results in the highest multi-day emitter identification performance. A total of two-hundred and fifty-five combinations are tested. Every preamble representation—described in Sect. IV-B—is investigated to determine the representation that maximizes multi-day SEI performance. Based on equations (6) and (7), there are

---

**TABLE 1. CNN architecture with eight stacked, two-dimensional convolutional layers used in Experiment #2.**

| Layer | Layer Type       | Dimensions | Filters | Stride | Activation Function             |
|-------|------------------|------------|---------|--------|--------------------------------|
| 1     | Image Input      | [4x320]    |         |        |                                 |
| 2     | 2D Convolution   | [4x4]      | 64      | 2      | maxPooling([2 x 2],Stride=2) $+$ relu            |
| 3     | 2D Convolution   | [4x4]      | 32      | 2      | maxPooling([2 x 2],Stride=2) $+$ relu            |
| 4     | 2D Convolution   | [4x4]      | 16      | 2      | maxPooling([2 x 2],Stride=2) $+$ relu            |
| 5     | 2D Convolution   | [4x4]      | 8       | 2      | maxPooling([2 x 2],Stride=2) $+$ relu            |
| 6     | 2D Convolution   | [4x4]      | 8       | 2      | maxPooling([2 x 2],Stride=2) $+$ relu            |
| 7     | 2D Convolution   | [4x4]      | 8       | 2      | maxPooling([2 x 2],Stride=2) $+$ relu            |
| 8     | 2D Convolution   | [4x4]      | 8       | 2      | maxPooling([2 x 2],Stride=2) $+$ relu            |
| 9     | 2D Convolution   | [4x4]      | 8       | 2      | maxPooling([2 x 2],Stride=2) $+$ relu            |
| 10    | Fully Connected (Dense) | 32       |         |        | Softmax Normalization            |
| 11    | Classification   | 32         |         |        |                                 |
FIGURE 2. Experiment #4: Flowchart of the STS averaging process. STS three through nine, designated $t_3$ and $t_9$, is selected from the original preamble (Method #4) and averaged into one 16-sample symbol. The average symbol is then inserted into the original preamble (Method #1), repeated and inserted into the original preamble (Method #2), or repeated and not inserted (Method #3).

This investigation is performed by adopting an $n$-choose-$k$ preamble representation selection approach in which $n=8$ and $k$ ranges from one to eight. This generates a total of two hundred and fifty-five preamble representation combinations. These combinations, ranked from highest to lowest multi-day average percent correct classification performance, are shown in Table 2. The CNNs used to conduct this experiment are configured such that their convolutional filter (a.k.a., kernel) size matches the selected combination value $k$. Other than this experiment-specific configuration, the CNNs are as described in Sect. IV-C2.

3) EXPERIMENT #3: INCREASING PREAMBLES AND NOISE REALIZATIONS
This experiment assesses multi-day SEI performance when CNNs are trained using multiple noise realizations and an increasing number of real-world collected preambles. One-, two-, five-, and ten-thousand preambles are randomly chosen from each emitter. For each set of preambles, a total of one, two, four, eight, and sixteen noise realizations are generated. All results are generated at an SNR of 30 dB using a four-layer CNN, Sect. IV-C2, and the frequency representation $P_f$, Sect. IV-B.

4) EXPERIMENT #4: AVERAGING STS SEQUENCES
This experiment attempts to improve multi-day SEI performance by reducing cross-week channel effects through the use of one of four average STS methods. The average STS is calculated over the third, $t_3$, through ninth, $t_9$, STS of the $j^{th}$ received preamble, $r_{ej}^{t_3}$, of emitter $e$. Average STS-based multi-day SEI is performed using one of the following four cases: (i) the average STS replaces the third through ninth STS in the corresponding preamble, (ii) the average STS is replicated $N_S$ (a.k.a., the number of STS used to compute the average) times and the result replaces the third through ninth STS sequences in the corresponding preamble, (iii) only the $N_S$ replicated, average STS is used for SEI, and realization of the preambles collected during Week #1 at an SNR of 30 dB. For this experiment, the preambles are represented using equation (7), Sect. IV-B.

2) EXPERIMENT #2: INCREASING CNN DEPTH
This experiment investigates whether or not increasing CNN depth improves multi-day SEI performance. This experiment uses CNNs based on the architecture shown in Table 1 with depths of one, two, four, eight, sixteen, or thirty-two stacked convolutional layers. The first, second, and third convolutional layers of all CNNs are constructed using a total of sixty-four, thirty-two, and sixteen $[4 \times 4]$ filters, respectively. All remaining layers are constructed using eight filters. CNNs constructed using one, two, four, and eight stacked convolutional layer(s) use a maximum pooling (a.k.a., MaxPooling) window size of $[2 \times 2]$, a stride of two, and a rectified Linear Unit (ReLU) layer. Only the first eight of the sixteen and thirty-two convolutional layer CNNs are activated. Every CNN—regardless of depth—is trained using a single noise realization of the preambles collected during Week #1 at an SNR of 30 dB. For this experiment, the preambles are represented using equation (7), Sect. IV-B.
| #  | \( i \) | \( q \) | \( \lambda \) | \( \theta \) | \( I \) | \( Q \) | \( \hat{\lambda} \) | \( \Theta \) | %  | #  | \( i \) | \( q \) | \( \lambda \) | \( \theta \) | \( I \) | \( Q \) | \( \hat{\lambda} \) | \( \Theta \) | %  |
|----|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|
| 1  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 19.33 | 36 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.31 |
| 2  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 19.01 | 37 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.31 |
| 3  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 19.00 | 38 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.30 |
| 4  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.99 | 39 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.29 |
| 5  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.94 | 40 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.27 |
| 6  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.91 | 41 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.27 |
| 7  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.91 | 42 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.27 |
| 8  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.86 | 43 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.27 |
| 9  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.80 | 44 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.25 |
| 10 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.79 | 45 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.22 |
| 11 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.77 | 46 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.21 |
| 12 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.75 | 47 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.20 |
| 13 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.71 | 48 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.20 |
| 14 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.68 | 49 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.16 |
| 15 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.60 | 50 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.14 |
| 16 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.58 | 51 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.14 |
| 17 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.57 | 52 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.14 |
| 18 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.56 | 53 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.13 |
| 19 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.56 | 54 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.09 |
| 20 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.51 | 55 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.06 |
| 21 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.50 | 56 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.06 |
| 22 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.48 | 57 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.04 |
| 23 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.48 | 58 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.04 |
| 24 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.47 | 59 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.03 |
| 25 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.46 | 60 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.03 |
| 26 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.45 | 61 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.00 |
| 27 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.44 | 62 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.99 |
| 28 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.44 | 63 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.95 |
| 29 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.43 | 64 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.94 |
| 30 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.41 | 65 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.92 |
| 31 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.39 | 66 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.91 |
| 32 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.36 | 67 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.90 |
| 33 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.33 | 68 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.90 |
| 34 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.32 | 69 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.88 |
| 35 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 18.32 | 70 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | 17.87 |
TABLE 2. (Continued.) Experiment #1: Every $n$-choose-$k$ preamble representation combination comprised of $i$, $q$, $\lambda$, $\theta$, $I$, $Q$, $\Lambda$, or $\Theta$. The presence of a specific representation—within a combination—is indicated using a “✓”. Combinations are ranked from highest (a.k.a., Combination #1) to lowest (a.k.a., Combination #255) AWC performance (%).

| #  | $i$ | $q$ | $\lambda$ | $\theta$ | $I$ | $Q$ | $\Lambda$ | $\Theta$ | %    |
|----|-----|-----|------------|----------|-----|-----|----------|----------|------|
| 71 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.87 |
| 72 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.84 |
| 73 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.81 |
| 74 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.81 |
| 75 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.80 |
| 76 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.77 |
| 77 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.77 |
| 78 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.76 |
| 79 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.75 |
| 80 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.75 |
| 81 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.74 |
| 82 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.74 |
| 83 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.73 |
| 84 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.72 |
| 85 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.72 |
| 86 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.72 |
| 87 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.71 |
| 88 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.71 |
| 89 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.70 |
| 90 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.69 |
| 91 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.68 |
| 92 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.65 |
| 93 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.64 |
| 94 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.63 |
| 95 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.62 |
| 96 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.61 |
| 97 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.60 |
| 98 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.58 |
| 99 | ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.57 |
| 100| ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.57 |
| 101| ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.53 |
| 102| ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.52 |
| 103| ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.52 |
| 104| ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.51 |
| 105| ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.50 |
| 106| ✓   | ✓   | ✓          | ✓        | ✓   | ✓   | ✓        | ✓        | 17.49 |

AWC: Average White Space.
TABLE 2. (Continued.) Experiment #1: Every $n$-choose-$k$ preamble representation combination comprised of $i$, $q$, $\hat{\lambda}$, $\theta$, $I$, $Q$, $\hat{\lambda}$, or $\Theta$. The presence of a specific representation—within a combination—is indicated using a "✓". Combinations are ranked from highest (a.k.a., Combination #1) to lowest (a.k.a., Combination #255) AWC performance (%).

| #   | $i$ | $q$ | $\hat{\lambda}$ | $\theta$ | $I$ | $Q$ | $\hat{\lambda}$ | $\Theta$ | %    |
|-----|-----|-----|------------------|---------|-----|-----|------------------|---------|------|
| 143 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.12 |
| 144 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.11 |
| 145 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.09 |
| 146 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.07 |
| 147 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.06 |
| 148 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.05 |
| 149 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.05 |
| 150 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.04 |
| 151 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.01 |
| 152 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.01 |
| 153 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 17.00 |
| 154 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.99 |
| 155 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.99 |
| 156 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.97 |
| 157 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.97 |
| 158 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.94 |
| 159 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.92 |
| 160 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.91 |
| 161 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.90 |
| 162 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.88 |
| 163 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.88 |
| 164 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.85 |
| 165 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.82 |
| 166 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.82 |
| 167 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.79 |
| 168 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.78 |
| 169 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.76 |
| 170 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.72 |
| 171 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.71 |
| 172 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.70 |
| 173 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.69 |
| 174 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.68 |
| 175 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.65 |
| 176 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.63 |
| 177 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.62 |
| 178 | ✓   | ✓   | ✓                | ✓       | ✓   | ✓   | ✓                | ✓       | 16.61 |
TABLE 2. (Continued.) 

| #  | i   | q   | \(\lambda\) | \(\theta\) | \(I\) | \(Q\) | \(\hat{\lambda}\) | \(\Theta\) | %    | #  | i   | q   | \(\lambda\) | \(\theta\) | \(I\) | \(Q\) | \(\hat{\lambda}\) | \(\Theta\) | %    |
|----|-----|-----|-------------|-----------|------|------|-------------|-----------|------|----|-----|-----|-------------|-----------|------|------|-------------|-----------|------|      |
| 215| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.17      |           |      | 236| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.31      |           |      |
| 216| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.17      |           |      | 237| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.23      |           |      |
| 217| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.17      |           |      | 238| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.21      |           |      |
| 218| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.10      |           |      | 239| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.07      |           |      |
| 219| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.10      |           |      | 240| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.06      |           |      |
| 220| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.09      |           |      | 241| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.05      |           |      |
| 221| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.03      |           |      | 242| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.01      |           |      |
| 222| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.01      |           |      | 243| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 14.87      |           |      |
| 223| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 16.00      |           |      | 244| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 14.75      |           |      |
| 224| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.99      |           |      | 245| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 14.54      |           |      |
| 225| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.98      |           |      | 246| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 14.37      |           |      |
| 226| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.84      |           |      | 247| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 13.93      |           |      |
| 227| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.76      |           |      | 248| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 13.72      |           |      |
| 228| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.75      |           |      | 249| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 13.72      |           |      |
| 229| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.75      |           |      | 250| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 13.59      |           |      |
| 230| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.56      |           |      | 251| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 13.35      |           |      |
| 231| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.49      |           |      | 252| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 13.13      |           |      |
| 232| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.45      |           |      | 253| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 12.50      |           |      |
| 233| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.41      |           |      | 254| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 12.45      |           |      |
| 234| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 15.38      |           |      | 255| ✓   | ✓   | ✓           | ✓         | ✓    | ✓    | 9.60       |           |      |

(iv) no averaging is performed (i.e., the original preamble). For Method #3 the first and tenth STS along with the GI and both LTS sequences are discarded and not used for multi-day SEI. In other words, Method #3-based multi-day SEI makes use of only the highlighted portion—enclosed in a dashed line box—of Method #2. Fig. 2 provides a representative illustration of the STS sequences used to calculate the average STS and implement Method #1, Method #2, and Method #3. Due to the 20 MHz sampling frequency, the preamble \(r^X\) is comprised of three-hundred twenty, complex-valued samples. Method #2 and Method #4 use preambles comprised of three-hundred twenty, complex-valued samples. However, Method #1 and Method #3 use preambles or preamble regions comprised of two-hundred forty and ninety-six complex-valued samples, respectively. For these two methods, the result has zeros appended to its end to reach a length of three-hundred twenty complex values. For this experiment, multi-day SEI is performed using frequency representation \(P^f\), Sect. IV-B, and the four-layer CNN in Sect. IV-C2.

5) EXPERIMENT #5: STS AND LTS SYMBOL RESIDUAL

This experiment attempts to improve multi-day SEI performance by reducing cross-week channel effects through the use of “residual” preambles. A residual preamble is generated by dividing the complex coefficients of STS \(t_1\) through \(t_{10}\) by STS \(t_6\) through \(t_{10}\) and LTS \(T_1\) by LTS \(T_2\). The results of both divisions are then concatenated to form a residual preamble. The residual has a sample length of one hundred forty-four complex coefficients. The result is padded to have an overall sample length of three-hundred twenty before the frequency representation \(P^f\) is calculated. This is done to ensure fairness when comparing multi-day SEI performance to the original three-hundred twenty sample length preamble. The multi-day SEI results associated with this experiment are generated using the four-layer CNN in Sect. IV-C2.

6) EXPERIMENT #6: TRAIN USING MULTIPLE DAYS OF COLLECTED SIGNALS

For the sixth and final experiment, multi-day SEI performance is assessed using CNN training sets comprised of
preambles selected from more than one of the weekly collections. The intent is for the CNN to learn a set of SEI features that facilitate emitter discrimination regardless of when the set of to-be-classified preambles are collected. In other words, this experiment investigates whether or not a CNN can be trained using preambles collected Week #1 through Week $N_w$—where $N_w$ is greater than 1 and not equal to 8—and classify Week #8 collected preambles with the same AEC as it classifies the test set of preambles associated with training weeks’ collections. This experiment uses the preambles’ frequency representation $P_f$, Sect. IV-B, and a four-layer CNN, Sect. IV-C2.

D. DEEP LEARNING CONFIGURATION

1) CONVOLUTIONAL NEURAL NETWORK PARAMETERS
All CNNs are trained using a learning rate of 0.01, a gradient threshold of 1, ADAM optimization, five hundred training epochs, and a mini-batch size of fifty-thousand predictors. Additionally, all CNNs are randomly initialized using the same random seed value(s) to ensure fairness across all experiments. Batch normalization is implemented directly after each convolutional layer in every CNN to enable multi-Graphical Processing Unit (GPU) distributed learning. Please note that batch normalization is not required when training on a single node.

2) COMPUTER ARCHITECTURE
All results are generated using MATLAB® R2022b. Training and testing is conducted across four Dell PowerRidge server nodes, each with dual Epyc 7662 64C/64T 64-bit CPUs, 512 Gb of DDR4 EEC RAM, and a Nvidia® A100 80Gb PCI-E GPU accelerator running CUDA 11.7. The operating system is RedHat Linux 3.10.0-1160.21.1.e17.x86_64.

V. RESULTS
This section provides the results generated for each of the six experiments detailed in Sect. IV-C. For Experiment #1 through Experiment #5, training is performed using the preambles collected during Week #1 (a.k.a., Day Zero). The only SNR level used for training and testing is 30 dB. When testing each case, ten noise realizations are generated for each week’s preambles and multi-day performance presented using average percent correct classification performance. Results are presented using either Average Emitter Classification (AEC) or Average Week Classification (AWC) performance. Results presented herein, all tables show AWC and all figures include the AEC and AWC.

Table 3 presents AWC performance—calculated across all emitters and each week’s set of preambles—for each of the two hundred fifty-five preamble representation combinations generated using the $n$-choose-$k$ selection approach in Sect. IV-C1. The presence of a specific preamble

VOLUME 10, 2022

TABLE 3. Experiment #2: AWC performance and total training times associated with multi-day SEI performed using a CNN architecture comprised of one, two, four, eight, sixteen, and thirty-two convolutional layers.

| Convolutional Layers | AWC (%) | Time (Seconds) |
|----------------------|---------|----------------|
| 1                    | 15.55%  | 217.42         |
| 2                    | 18.55%  | 542.58         |
| 4                    | 18.57%  | 972.12         |
| 8                    | 18.42%  | 1,261.19       |
| 16                   | 16.81%  | 1,822.80       |
| 32                   | 15.39%  | 3,268.45       |
representation (a.k.a., $i, q, \lambda, \theta, I, Q, \Lambda$, or $\Theta$) within each combination is denoted using a “$\bigvee$” under that representation’s symbol, Sect. IV-B. Combinations are ranked from highest (a.k.a., Combination #1) to lowest (a.k.a., Combination #255) AWC performance. The highest ranked preamble representation combination consists of $i, q, \lambda, I, Q,$ and $\Lambda$ (a.k.a., all but the time- and frequency-representation phase components are present) and achieves an AWC performance of 19.33%. Use of only the four frequency components (a.k.a., $I, Q, \Lambda$, and $\Theta$) result in an AWC of 18.56%, which corresponds to the nineteenth highest, overall AWC, an AWC that is only 0.75% lower than Combination #1, and a 33% data reduction. In fact, all of the highest AWC combinations (a.k.a., Combination #88 and higher) are associated with frequency representation components and without a high reliance (i.e., use of three or more components) upon time representation components. This is shown by the $I$ and $Q$ representation being present in nine of the top ten ranked combinations, and $\Lambda$ being in eight of the top ten ranked combinations. However, $\lambda$ is present in only seven of the top ten ranked combinations, and $q$ is in five of the top ten ranked combinations. Additionally, $\theta$ is not included in any of the top ten ranked combinations. This suggests that the preambles’ time representations do not provide discriminating features that are beneficial for multi-day SEI. Use of only the time representation (a.k.a., $i, q, \lambda$, and $\theta$) results in an AWC performance of 16.41%, which corresponds to a rank of one-hundred and ninety-eight out of two hundred fifty-five.

Based upon the Experiment #1 results and analysis, all remaining experiments are conducted using only the preambles’ frequency representations comprised of $I, Q, \Lambda$, and $\Theta$.

B. RESULTS FOR EXPERIMENT #2

When considering CNN of increasing depth, the AWC performance results—presented in Table 3—show that an eight convolutional layer CNN achieves an AWC performance of 22.81% that is the highest amongst all considered CNN depths. When sixteen and thirty-two convolutional layers are used, the AWC performance decreased to 16.81% and 22.59%, respectively. Four convolutional layers achieve an AWC performance of 22.25%, which is 0.56% lower than the eight convolutional layers CNN. However, only 972.12 seconds is needed to train the four convolutional layer CNN versus the 1, 261.19 seconds needed to train the eight convolutional layer CNN; thus, reducing the training time by 23%. Fig. 3 shows a decrease in AEC performance as the number of convolutional layers increases from eight to sixteen and onto thirty-two. For the one and two convolutional layers CNNs, the results suggest that the trainable parameters are insufficient to learn the discriminating features necessary for multi-day SEI. Conversely, there is not enough training data to sufficiently tune all of the sixteen and thirty-two convolutional layer CNNs’ trainable parameters to achieve effective multi-day SEI. Based upon the Experiment #2 results and observations, all subsequent experiments are performed using CNNs comprised of four convolutional layers.

C. RESULTS FOR EXPERIMENT #3

Fig. 4 shows AEC performance when a four convolutional layer CNN is trained using the frequency representation, equation (7), of Week #1’s preambles as their numbers increase from 1,000 to 10,000 and one, two, four, eight, or sixteen noise realizations are generated per preamble. Regardless of the number of preambles per emitter or number of noise realizations per preamble used, all Experiment #3 results are generated by classifying 10,000 preambles per emitter with ten noise realizations per preamble. Fig. 4 shows that a CNN trained using 1,000 preambles with one noise realization per preamble achieves an AEC performance of 55%. However, the AEC performance reaches a maximum of 61.43% when sixteen noise realizations are generated for each of the 1,000 preambles per emitter collected Week #1. Fig. 4 shows that AEC performance increases when the number of preambles per emitter is increased. For instance, the AEC performance is 64.77% when using 2,000 preambles per emitter with one noise realization per preamble. This is a 3.43% increase in AEC performance and only

| Noise Realizations | AWC (%) | Time (Seconds) |
|--------------------|---------|----------------|
| 1                  | 19.78%  | 964.61         |
| 2                  | 20.31%  | 2,120.1        |
| 4                  | 22.25%  | 4,487.8        |
| 8                  | 22.81%  | 9,038.4        |
| 16                 | 22.59%  | 18,097.8       |

FIGURE 4. Experiment #3: AEC performance when a four convolutional layer CNN—trained using the Week #1 preambles’ frequency representations, 1,000, 2,000, 5,000, or 10,000 (using “e3” to denote thousand) preambles per emitter with one, two, four, eight, or sixteen noise realizations per preamble—classifies 10,000 preambles per emitter with ten noise realizations per preamble collected on Week #1.
one-eighth of the data when compared to the 1,000 preambles per emitter with sixteen noise realizations per preamble case. When using 2,000 preambles per emitter and two, four, eight, and sixteen noise realizations per preamble, AEC performance increases to 68.46%, 69.73%, 71.13%, and 72.59%, respectively. Increasing the number of noise realizations per preamble to sixteen results in a maximum AEC performance improvement of 7.82% when using 2,000 preambles per emitter; however, it is at the expense of a sixteen-fold increase in the amount of data being stored and processed. It is clear from Fig. 4 that increasing the number of preambles per emitter provides the greatest improvement in AEC performance. When using one noise realization per preamble, increasing the number of preambles per emitter—from 1,000 to 10,000—increases AEC performance to a maximum of 93.34%, which is a 38.34% increase (55% versus 93.34%). This maximum is 10.71% higher than the 82.43% AEC performance achieved using 5,000 preambles per emitter with sixteen noise realizations per preamble. For the 10,000 preambles per emitter case, AEC performance increases to 95%, 96.02%, 96.99%, and 97.1% when combined with two, four, eight, and sixteen noise realizations per preamble, respectively. The difference between the eight and sixteen noise realizations per preamble results is marginal and within the statistical error of the experiment; thus, the use of eight noise realizations per preamble is beneficial because it cuts the data set size in half without severely degrading performance.

Since 10,000 preambles per emitter results in the highest AEC performance, all AWC performance results are generated using the same number of preambles per emitter.

Table 4 provides AWC performance when 10,000 preambles per emitter are combined with one, two, four, eight, or sixteen noise realizations per preamble. AWC performance is 19.75% with a training time of 964.61s when using one noise realization per preamble. When the number of noise realizations per preamble is increased to two, four, eight, and sixteen the AWC performance increases to 20.31% (a 0.53% increase), 22.25% (a 2.5% increase), 22.81% (a 3.06% increase), and 22.59% (a 2.84% increase), respectively. However, this increase in AWC performance comes at the expense of increased training times. Doubling the number of realizations in the training set roughly doubles the training time of the network. The longest training time—the median of 18,097.8 seconds (a.k.a., 301.63 minutes or just over 5 hours)—corresponds to the sixteen noise realizations per preamble case that achieves only a negligible improvement in AWC. Overall, the results in Table 4 show that increasing the number of noise realizations does not appreciably increase multi-day SEI performance; thus, the remaining three experiments are conducted using
10,000 preambles per emitter with one noise realization per preamble.

**D. RESULTS FOR EXPERIMENT #4**

AEC performance results are presented in Fig. 5 for each of the eight weeks of collected preambles classified by a four convolutional layer CNN trained using Week #1 preambles. Recall that each emitter is represented using the frequency representation of their 10,000 preambles with one noise realization per preamble. These results show that Week #1 AEC performance is 96% for Method #1 (red circle), Method #2 (green triangle), and Method #4 (black diamond) while Method #3 (blue square) results in an AEC performance of 91%. When classifying Week #2 preambles, Method #4 results in the highest AEC performance of 6.74%, which is only 2% higher than the three average STS methods’ AEC performances and 1% above the “guess threshold” of 3.13%. Week #3 is the only one in which Method #4 results in an AEC performance of 16.63% that is distinctly higher than those of the three average STS methods. Classification of Week #4, Week #5, and Week #7 collected preambles results in Method #4 achieving the highest AEC performance of 8.52%, 17.85%, and 10.31%, respectively, which is only 1% higher than the AEC performances of the three average STS methods. When classifying Week #6 and Week #8 collected preambles, Method #3 results in an AEC performance of 14.38% and 10.13%, respectively, which is only marginally higher than the performances of the other three.

AWC performance is presented in Table 5, which shows that Method #4 (a.k.a., the original preambles without any averaging) achieves the highest performance with an AWC of 21.53%. Method #1, Method #2, and Method #3 achieve AWC performances of 20.8%, 20.48%, and 20.32%, respectively. These results all fall within each other’s margin of error.

Overall, Experiment #4 results show that the averaging of STS sequences may be doing more harm than good in terms of improving multi-day SEI performance. This is attributed to the possibility that averaging across STS sequences may remove STS-specific coloration that is essential to discriminating one emitter from another rather than mitigating channel effects that hinder multi-day SEI performance. Based upon these observations as well as the removal of a pre-processing step, no average STS methods are performed in Experiment #5 and Experiment #6.

**E. RESULTS FOR EXPERIMENT #5**

Experiment #5’s AEC performance results are shown in Fig. 6. These results show—with the exception of Week #2—that the use of the residual (red circle) preamble does not improve multi-day SEI performance. This is most evident when looking at Week #1 results, which show an AEC
performance of 35% and 96.5% when using the residual and “no residual” (a.k.a., collected, black diamond) preambles. Recall that the four convolutional layers CNN is trained using only Week #1 collected preambles; thus, these results correspond to CNN classification of preambles drawn from the same collection used to train it and should result in the highest AEC performance when compared to the remaining seven weeks’ AEC performances (i.e., these results represent the “best” case scenario within multi-day SEI). AWC performance results are presented in Table 6 and show that performance decreases from 20.71% to 11.62% when using the residual preamble in lieu of the no residual (a.k.a., collected) preamble. The results presented in Fig. 6 and Table 6 confirm that the use of the residual preamble hinders same-day SEI while providing no benefit to addressing the multi-day SEI problem. It is for these reasons that Experiment #6 is performed without the use of the residual preambles.

F. RESULTS FOR EXPERIMENT #6

Fig. 7 shows both AEC and AWC performance for a four convolutional layer CNN trained using one, two, three, four, five, six, or seven weeks of collected preambles and tested using the remaining weeks’ collected preambles. For example, when the CNN is trained using Week #1, Week #2, and Week #3 collected preambles, it classifies preambles collected during Week #4, Week #5, Week #6, Week #7, and Week #8. It is important to note that this experiment focuses on whether or not a CNN can be trained using preambles collected Week #1 through Week #Nw—where Nw is greater than 1 and not equal to 8—and classify Week #8 collected preambles with the same AEC as it classifies the test set of preambles associated with the training weeks’ collections. Thus, the remainder of this section focuses on the analysis of Week #8 classification performance.

When trained using only Week #1 collected preambles, classification of Week #8 collected preambles results in an AEC performance of 8%. Adding Week #2 collected preambles to the CNN’s training set results in an AEC performance of 15%. Similarly, when Week #3, Week #4, and Week #5 collected preambles are added to the training set, then the AEC performance increases to 19%, 24%, and 31%, respectively. However, adding preambles collected during Week #6 and Week #7 does not increase AEC performance. This might be the maximum generalization that can be achieved by the selected CNN architecture. In fact, Week #8’s AEC performance decreases to 28% when the training set is comprised of seven weeks worth of collected preambles. This suggests that the CNN could be overfitting and losing its general representation of each emitter’s discriminating SEI features.

AWC performance linearly increases for each week represented within the CNN’s training set, Table 7. However, this increase in AWC performance comes at the cost of increased
training times, which is roughly an additional 1,000 seconds for every week added to the training set.

VI. CONCLUSION

The work presented in this paper shows that multi-day SEI performance improves when using the preambles’ frequency-domain representation—in lieu of the time-domain representation—and four convolutional layers CNN. The latter provided a sufficient number of trainable parameters to maximize multi-day SEI performance without providing so many that there is insufficient data or time to support the learning of emitter-specific features. This work also shows that an AWGN channel model is an insufficient representation of real-world channels when it comes to SEI performed over multiple collections or days. If this were not the case, then one would expect multi-day SEI performance to significantly improve as more and more noise realizations are added to the training set. However, this is not the case and in fact, SEI performance sees greater improvement when the number of preambles per emitter is increased versus when noise realizations increase. The work also shows that the average STS and residual preamble channel mitigation approaches do not mitigate or remove the real-world channel impacts present in the collected preambles. Our results suggest that the channel may not be the dominant effect hindering multi-day SEI performance. Therefore, future research will continue to investigate possible methods for isolating channel agnostic emitter-specific features as well as determining whether or not the channel is indeed the cause of degraded SEI performance across multiple collections or days. Additionally, since SEI is being put forward as an IoT security solution, future research will investigate more computationally efficient and less memory-intensive machine learning approaches.

REFERENCES

[1] Chief Information Officer, U. S. Department of Defense. (Dec. 2016). DoD Policy Recommendations for The Internet of Things (IoT). [Online]. Available: https://www.bsi.gov.uk/?viewid=799676

[2] K. Rawlinson. (Jul. 2014). HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack. [Online]. Available: https://www8.hp.com/us/en/hp-news/press-release.html?id=1744676

[3] I. Ray, D. M. Kar, J. Peterson, and S. Goeringer, “Device identity and trust in IoT-sphere forsaking cryptography,” in Proc. IEEE Int. Symp. Collaborate Internet Comput. (CIC), Dec. 2019, pp. 204–215.

[4] N. Neshenko, E. Bou-Harb, J. Crichtigno, G. Kaddoum, and N. Ghani, “Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploits,” IEEE Commun. Surveys Tuts., vol. 21, no. 3, pp. 2702–2733, 3rd Quart, 2019.

[5] Statista. (2019). Internet of Things (IoT) Connected Devices Installed Base Worldwide From 2015 To 2025 (in Billions). [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[6] S. Larsen. (Jul. 2017). A Smart Fish Tank Left a Casino Vulnerable to Hackers. [Online]. Available: https://money.cnn.com/2017/07/19/technology/fish-tank-hack-darktrace/index.html

[7] J. Wright and J. Cache, Hacking Wireless Expo: Wireless Security Secrets & Solutions, 3rd ed. New York, NY, USA: McGraw-Hill, 2015.

[8] M. Stanislav and T. Beardsley. (2015). Hacking IoT: A Case Study on Baby Monitoring Exposures and Vulnerabilities. Rapid7. [Online]. Available: https://www.rapid7.com/globalassets/external/docs/Hacking-IoT-A-Case-Study-on-Baby-Monitor-Exposures-and-Vulnerabilities.pdf

[9] J. Wright. KillerBee: Practical ZigBee Exploitation Framework or "Wireless Hacking and the Kinetic World." Accessed: Dec. 5, 2019. [Online]. Available: https://www.killerbelforwsu.com/presentations/toorcon11-wright.pdf

[10] P. Shipley and R. Gooler, “Insteon: False security and deceptive documentation,” presented at the DEFCON, 2015, doi: 10.5446/36420.

[11] P. Shipley. (2015). Tools for Insteon RF. [Online]. Available: https://github.com/evilpete/insteonrf

[12] C. Brook. (Dec. 2017). Mirai IoT Botnet Co-Authors Plead Guilty. [Online]. Available: https://digitalguardian.com/blog/mirai-iot-botnet-co-authors-plead-guilty/

[13] S. Simon. (Oct. 2016). Internet of Things’ Hacking Attack Led To Widespread Outage Of Popular Websites. [Online]. Available: https://www.wbur.org/npr/498954197/internet-outage-update-internet-of-things-hacking-attack-led-to-outage-of-popular

[14] F. Kandah, J. Cancellieri, D. Reising, A. Altarawneh, and A. Skjellum, “A hardware-software codesign approach to identity, trust, and resilience for IoT/CPS at scale,” in Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData), Jul. 2019, pp. 1125–1134.

[15] D. Reising, J. Cancellieri, T. D. Loveless, F. Kandah, and A. Skjellum, “Radio identity verification-based IoT security using RF-DNA fingerprints and SVM,” IEEE Internet Things J., vol. 8, no. 10, pp. 8356–8371, May 2021.

[16] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, “A review of radio frequency fingerprinting techniques,” IEEE J. Radio Freq. Identificat., vol. 4, no. 3, pp. 222–233, Sep. 2020.

[17] B. Hamadaoui and A. Elmaghabib, “Uncovering the portability limitation of deep learning-based wireless device fingerprints,” 2022, arXiv:2211.07687.

[18] J. Gaskin, B. Hamadaoui, and W.-K. Wong, “Tweak: Towards portable deep learning models for domain-agnostic LoRa device authentication,” 2022, arXiv:2209.00786.

[19] A. Elmaghabib and B. Hamadaoui, “LoRa device fingerprinting in the wild: Disclosing RF data-driven fingerprint sensitivity to deployment variability,” IEEE Access, vol. 9, pp. 142893–142909, 2021.

[20] A. Elmaghabib, B. Hamadaoui, and A. Natarajan, “WideScan: Exploiting out-of-band distortion for device classification using deep learning,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2020, pp. 1–6.

[21] A. Al-Shawabka, F. Restuccia, S. D’Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia, “Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Jul. 2020, pp. 646–655.

[22] WiSilica. (Aug. 2020). Top 6 IoT Communication Protocols. [Online]. Available: https://wislica.com/company/top-6-iot-communication-protocols/

[23] IEEE. (2022). IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Redline, IEEE Standard 802.11-2020 (Revision of IEEE Standard 802.11-2016), pp. 1–7524, 2021.

[24] IEEE 802.11ax: The Sixth Generation of Wi-Fi, Cisco Systems, Inc., San Jose, CA, Standard C11-740788-03, Apr. 2020. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/wireless/wireless-paper-c11-740788.pdf

[25] F. Restuccia, S. D’Oro, A. Al-Shawabka, M. Belgioio, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, “DeepRadioID: Real-time channel-resilient optimization of deep learning-based radio fingerprinting algorithms,” in Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput., Jul. 2019, pp. 51–60, doi: 10.1145/3323679.3326503.

[26] J. H. Tyler, M. K. M. Fadul, D. R. Reising, and E. Kaplanoglu, “Simplified denoising for robust specific emitter identification of preamble-based waveforms,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2021, pp. 1–7.

[27] M. K. M. Fadul, D. R. Reising, and M. Sartipi, “Identification of OFDM-based radios under Rayleigh fading using RF-DNA and deep learning,” IEEE Access, vol. 9, pp. 17100–17113, 2021.

[28] J. H. Tyler, M. K. M. Fadul, D. R. Reising, and F. I. Kandah, “An analysis of signal energy impacts and threats to deep learning based SEI,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2022, pp. 2077–2083.
[29] M. Taha, M. Fadul, J. Tyler, D. Reising, and T. Loveless, “An assessment of entropy-based data reduction for SEI within IoT applications,” in Proc. IEEE Mil. Commun. Conf. (MILCOM) Nov. 2022.

[30] F. Restuccia, S. D’Oro, A. Al-Shawabka, B. C. Rendon, S. Ioannidis, and T. Melodia, “DeepFIR: Channel-robust physical-layer deep learning through adaptive waveform filtering,” IEEE Trans. Wireless Commun., vol. 20, no. 12, pp. 8054–8066, Dec. 2021.

[31] T. M. Schmidl and D. C. Cox, “Robust frequency and timing synchronization for OFDM,” IEEE Trans. Commun., vol. 45, no. 12, pp. 1613–1621, Dec. 1997.

[32] D. R. Reising, M. A. Temple, and M. J. Mendenhall, “Improving intra-cellular security using air monitoring with RF fingerprints,” in Proc. IEEE Wireless Commun. Netw. Conf., Apr. 2010, pp. 1–6.

JOSHUA H. TYLER (Student Member, IEEE) received the Bachelor of Science and Master of Science degrees in electrical engineering from The University of Tennessee at Chattanooga (UTC), in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree. He is also a Researcher with UTC. He is also employed as a Research Associate with the Department of Electrical Engineering, UTC. His research interests include specific emitter identification, power quality analysis, digital communications, deep learning, and edge computing.

DONALD R. REISING (Senior Member, IEEE) received the B.S. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 2006, and the M.S. and Ph.D. degrees in electrical engineering from the Air Force Institute of Technology, Dayton, OH, USA, in 2009 and 2012, respectively. From 2008 to 2014, he was an Electronics Engineer with the U.S. Air Force Research Laboratory at Wright-Patterson Air Force Base, Dayton. He is currently a University of Chattanooga (UC) Foundation Associate Professor of electrical engineering with The University of Tennessee at Chattanooga, Chattanooga, TN, USA. His research interests include wireless device discrimination using RF distinct native attribute fingerprints, compressive sensing, cognitive radio, and deep learning. He is a member of Eta Kappa Nu and Tau Beta Pi.

FARAH I. KANDAH (Senior Member, IEEE) received the B.S. degree in computer science from Hashemite University, Jordan, in 2002, the M.S. degree in computer science from the University of Jordan, in 2005, and the Ph.D. degree in computer science from North Dakota State University, in 2012. He is currently an Associate Professor with the Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, USA. His research interests include smart communications to support real-time interactions in wired and wireless networks, threat hunting, blockchain, and trust management in areas of the Internet of Things, smart networking design, smart connected vehicles, cybersecurity, cyber-physical systems, and software-defined networks. He served as a Technical Committee Member, the Co-Chair, and the Session Chair for a number of conferences in the field of wireless communications and networking, such as CHINACOM, IEEE ICNC, and IEEE CCNC. He also served as a Reviewer for several international journals, including the Security and Communication Networks, IEEE Sensors Journal, IEEE Network, International Journal of Information Processing and Management, and the Journal of Computer Systems, Networks, and Communications.

ERKAN KAPLANOGLU (Senior Member, IEEE) is currently an Associate Professor with the Department of Engineering Management and Technology, Mechatronics, The University of Tennessee at Chattanooga (UTC). He founded the Biomechatronics and Assistive Technology Laboratory (BioAstLab), in 2019. The establishment of this laboratory has led to collaborations with the Departments of Biomedical and Medical at other partner higher education institutions. His research interests include robotics, biomechatronics, automation, and industry 4.0. He is a Senior Member of the International Association of Engineers (IAENG) and the European Embedded Control Institute (EIECI).

MOHAMED M. K. FADUL received the B.S. degree in electrical and electronics engineering from the University of Khartoum, Khartoum, Sudan, in 2012, and the M.S. and Ph.D. degrees in electrical engineering and computational engineering from The University of Tennessee at Chattanooga (UTC), Chattanooga, TN, USA, in 2018 and 2022, respectively. He is currently a Postdoctoral Researcher with UTC. His research interests include software-defined radios, wireless device discrimination using RF distinct native attribute fingerprints, and deep learning.

* * *

133616