WRIST: Wideband, Real-time, Spectro-Temporal RF Identification System using Deep Learning

Hai N. Nguyen, Marinos Vomvas, Triet D. Vo-Huu, and Guevara Noubir, *Senior Member, IEEE*

Abstract—RF emissions' detection, classification, and spectro-temporal localization are essential not only for understanding, managing, and protecting the radio frequency resources, but also for countering today's security threats such as jammers. Achieving this goal for wideband, real-time operation remains challenging. In this paper, we present WRIST, a Wideband, Real-time, Spectro-Temporal RF Identification system. WRIST can detect, classify, and precisely locate RF emissions in time and frequency using RF samples of 100 MHz spectrum in real-time. The system leverages an *one-stage object detection* Deep Learning framework, and transfer learning to a multi-channel visual-based spectral representation. Towards developing WRIST, we devised an iterative training approach which leverages synthesized and augmented RF data to efficiently build a large dataset with high-quality labels. WRIST achieves over 99% class detection accuracy, 94% emission precision and recall, with less than 0.08 bandwidth and time offset ratios in a large anechoic chamber over-the-air environment. In the extremely congested in-the-wild environment, WRIST still achieves over 80% precision and recall. WRIST currently supports five 2.4 GHz technologies (Bluetooth, Lightbridge, Wi-Fi, XPD, and ZigBee) and is easily extendable to others. We are making our curated dataset available to the whole community. It comprises over 10 million labelled RF emissions from off-the-shelf wireless radios spanning the five classes of technologies.

Index Terms—Spectro-temporal RF Identification, Deep Learning, Wideband and Real-time System, Dataset

1 Introduction

Mobile technologies, fueled by advances in wireless communications, revolutionized our society beyond the pioneers dreams. It enables a ubiquitous access to information, and connects people to each other, and to a rapidly increasing number of services. However, a plethora of emerging applications, such as Massive IoT (MIoT), autonomous cars, robotics, and augmented reality are driving the demand for spectrum to new heights. Spectrum scarcity is becoming a critical issue. At the same time, wireless systems are increasingly softwarized, and SDR platforms are highly capable, with small form factor and low cost. For instance, the XTRX SDR platform is capable of 2x2 120MSps in a mini PCIe form factor and costs few hundreds of dollars [1]. This is both a *blessing* for developing new sophisticated communications techniques (that are agile and flexible, exploiting every pocket of the spectrum), and a *curse* as it calls for new mechanisms for spectrum management and it lowered the barrier for attacks from smart jammers to compromised wireless chips [2], or weaponizing drones [3], [4]. While the DHS, FAA, and FCC have regulations against such threats [5], [6], [7], [8], [9], [10], they unfortunately, still lack the necessary technology to enforce them. This confluence of trends raises challenging research questions as to the development of scalable techniques for understanding, managing, and protecting the RF spectrum, in particular in dynamic mobile environments. Some of the traditional areas that will benefit from such techniques include spectrum management, as dynamic and fine-grain spectrum sharing is becoming a necessity even for 5G mobile systems [11], [12]. Crucial to all these applications is the ability to understand the spectrum, both in *real-time* and *a-posteriori*, detect, classify, and predict the time and frequency information of the communications. Traditional spectrum sensing techniques are insufficient as they cannot classify emissions, detect collisions, and adequately summarize the view of wideband spectrum.

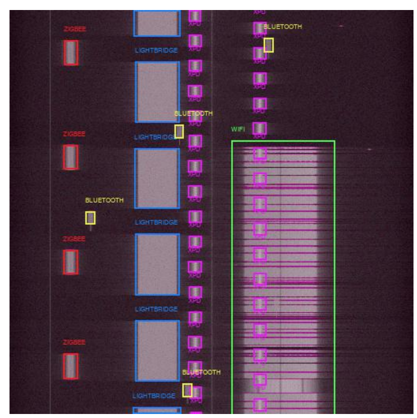


Fig. 1: Wideband, real-time identification of RF emissions.

Towards this objective, we propose *systematic and generalizable approaches* to detect and classify RF emissions with two key unmet requirements: *real-time* and *wideband* spectrum processing. These techniques are based on our team (Sprite) solution in the Spectrum Collaboration Challenge (SC2) organized by DARPA [13]. Sprite was a winning team in 2017, 2018 and a finalist in 2019 (a total of $\$2M$ prizes). To

- *H. N. Nguyen, M. Vomvas and G. Noubir are with the Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA (E-mail: {nguyen.hai, vomvas.m, g.noubir}@northeastern.edu).*
- *T. D. Vo-Huu is with Novowi LLC (Email: vohuudtr@novowi.com). This work was done while a faculty at Northeastern University.*

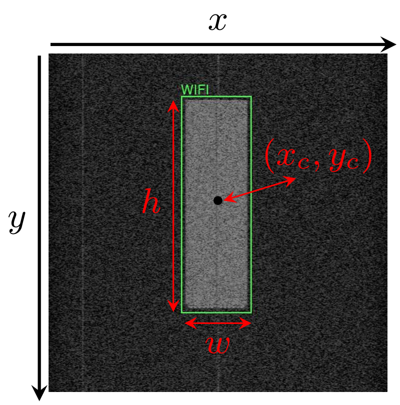


Fig. 2: A detection bounding box located at center frequency x_c and time y_c identifies a present Wi-Fi emission in the wide spectrum.

the best of our knowledge, previous work only focused on a subset of these objectives. We systematically develop RF-Centric ML models and techniques to detect and classify a wide variety of existing wireless standards, to be easily extensible to new and unknown RF emissions (We refer to "RF emissions" as electromagnetic waves emitted from RF devices. Henceforth, the terms "RF emissions" and "RF signals" are used interchangeably). Our approach is inspired by the success achieved by computer vision in several ways. For the *real-time* spectro-temporal detection and classification of RF emissions, our approach is inspired by YOLO [14], [15]. In this paper, we generalize and extend the principles underlying YOLO's success to the RF domain. These include (1) analyzing a multi-channel image-like representation of RF emissions with a single forward propagation neural network (unlike prior work that iterates through sliding and resizing, or complex multi-stage pipelines) by creating a grid and detecting/classifying objects per cell, (2) direct spectro-temporal location prediction combined with a small number of bounding boxes bootstrapped with anchor training and specialized to learn typical RF emission patterns, and (3) fine-grain features detection through passthrough network design and multiscaling. Moreover, we improve the speed and accuracy of the core DL network simultaneously by optimizing the layers and anchor boxes based on special characteristics of RF emissions.

Towards building the deep learning models, and in the absence of an initial labelled dataset, we developed a set of techniques to maximize labelling automation. We first reused some of YOLO existing layers and weights (transfer learning). On the other hand, we developed an approach to bootstrap an iterative process of using synthetic intermediate data, building increasingly large datasets and accurate DL models. Our final DL model achieves over 99% of class detection accuracy in various over-the-air environments, including the extremely congested in-the-wild spectrum. Moreover, to highlight our system's capabilities of locating the emissions in time and frequency in the wide spectrum and to better evaluate these capabilities, we devised new evaluation metrics for the evaluated locations. We emphasize that prior work only focuses on class accuracy metric, and does not consider metrics to evaluate the time and frequency locations of emissions in the wide spectrum, because of the lack of spectro-temporal identification capabilities. Therefore, we address this with new emission detection metrics including emission precision, recall, F1-score, and time/bandwidth offset ratios (described in Section 3.2), and show that our DL model achieves very good performance.

We believe that sharing a large and curated dataset with the wireless community will spur the creation of novel RFML models and techniques. Towards this goal, we developed a dataset of over $800,000$ RF images containing over 10 millions fully-labelled emissions (corresponding to over 13 Terabytes of raw RF samples collected from the 100 MHz-wide spectrum) from a variety of radios that operate in the 2.4GHz ISM band including Wi-Fi, Bluetooth, ZigBee, Lightbridge, XPD. The dataset is the result of the efficient data collection process with the automatic labeling and RF augmentation techniques. The emission data is annotated with time and frequency information, and with corresponding RF technologies. The dataset is complemented with the API that supports expansion of training data with RF augmentation, and serves as the building block to develop new models supporting other RF technologies (Section 5). Our contributions include:

- An extensible DL framework for real-time RF identification inspired by and leveraging state-of-the-art deep learning detection network, and a visual-based RF signal representation to enable the learning. We optimized the DL network and algorithm with RF-centric techniques for anchor bounding boxes and CNN layers, achieving better and faster RF identification (Section 2).
- An efficient iterative learning approach to develop DL models consisting of two stages: (1) Transfer learning from a dataset of synthesized and augmented RF data, and (2) Refining the DL model using a large dataset of over-the-air RF emissions acquired by the initial model (Sections 3 and 4). This two-step process can be iterated to support increasingly larger datasets and types of wireless technologies by minimizing manual efforts for annotating data.
- DL architecture and models for real-time, wideband RF emissions detection (anywhere in a 100 MHz band), classification, and spectro-temporal analysis achieving over 99% class detection accuracy even in extremely congested environment in the wild for five classes of RF technologies: Wi-Fi, Bluetooth, ZigBee, Lightbridge [27] and XPD [28].
- New evaluation metrics for spectro-temporal RF identification that assess the DL model's capabilities on the single-emission level. Our DL model achieves very high emission precision (up to 0.99), recall (up to 0.97), F1-score (up to 0.98) and low bandwidth and time offset ratios (less than 0.1).
- A curated dataset of over 10 million labelled emissions spanning five 2.4 GHz RF technologies, along with the API to generate synthetic data and expand the dataset by using RF data augmentation. The API provides an efficient and systematic way to extend the RF identification approach to new RF technologies, serving as a building block for RFML research

TABLE 1: Comparison with prior work in RF emission identification. Our work is the first in the literature that successfully and adequately addresses the requirements for spectro-temporal RF identification, real-time and wideband processing, and sufficient real RF training data. We emphasize that the wideband processing refers to the ability to cover the whole 80 MHz-wide 2.4 GHz ISM band in real-time, which is only present in our work (WRIST implemented with Ettus USRP X310 can cover a 100 MHz bandwidth), while the other systems only support up to 25 MHz (as in [16]). [17] provides evaluation for over-100MHz signal classification of eight technologies (DECT, Bluetooth, ATSC, LTE, UAS, Vulos, ADSB, and Wi-Fi), however, only synthetic RF signals are considered, and the approach is not evaluated in real-time.

	RF Identification		Real-time Processing		Dataset		
	Detection & Classification	Spectro-temporal Location Prediction	Narrowband	Wideband	Real RF Emissions	Open Data	Number of Technologies
This work	✓	✓	✓	✓	✓	✓	5
Schmidt et al. [18]	✓					✓	3
Bitar et al. [19]	✓				✓		3
Baset et al. [16]	✓		✓		✓		3
O'shea et al. [20]		✓					N/A
Lo et al. [21]	✓	✓			✓		1
Fonseca et al. [22]	✓	✓			✓		2
Prasad et al. [23]	✓	✓					3
Vagollari et al. [24]	✓	✓					N/A
Franco et al. [17]	✓	✓					8
ModRec[1] [25], [26]	✓				✓	✓	N/A

and applications (Section 5).

2 SPECTRO-TEMPORAL RF IDENTIFICATION

Aiming to build a practical and efficient RF identification framework, we design WRIST with the following objectives: accurate classification and spectro-temporal characterization, real-time processing and wideband spectrum support. To the best of our knowledge, prior work only addressed a subset of these objectives, as shown in Table 1. In this section, we identify the challenges towards developing such system and present our approach.

2.1 Challenges and Elements of Approach

Wideband RF Identification. As today's wireless systems are required to flexibly operate in a shared, wide RF spectrum (e.g., 2.4GHz ISM band is 80MHz wide and is home to various communication standards), wideband operation is critical for practical RF emission identification applications. Conventional approaches rely on sensing specific, small frequency bands, and classifying the RF signal within that band, without considering other emissions present in other parts of a much larger spectrum [18], [25]. Authors in [19] attempted to cover the whole 2.4 GHz ISM band with multiple co-operating neural networks classifying based on sensed data collected in much narrower bands. While such approach is theoretically feasible, it is very challenging to deploy in practice due to computation overhead as well as synchronization requirements for the classifiers. Moreover, we emphasize that RF classifiers used in those systems are unable to provide the spectro-temporal locations (time period and frequency information) of emissions. Our solution to these problems leverages techniques from *object detection* approaches of computer vision. We transform wideband RF samples into a 2D time-frequency image (Section 2.2) and develop RF-centric Deep Learning (DL) models to identify all individual and overlapping emissions with their RF categories and 2D positions (cf. an example of Wi-Fi packet detection in Figure 2).

1. Existing works on classification of different modulation schemes.

Real-time Capability. As seen in Table 1, most of existing work lacks the real-time processing capability and analysis, despite its importance for practical RF recognition systems. We incorporated two features into the DL framework to solve this problem. In essence, we first designed and deployed a *RF-centric compression* first layer (cf. Section 2.4) of the neural network to reduce the computational load for the rest of the network, while selectively preserving important RF data features. Second, we optimized the state-of-the-art deep learning detection network (discussed in Section 2.3) with RF-centric techniques and eventually achieved faster processing speed and better performance.

Efficient Data Collection and Training. Effective supervised Machine Learning and Deep Learning frameworks require a lot of training data, which takes significant effort to carefully label. Building a good RF dataset has even more challenges due to the massive I/Q data of the wide spectrum requiring expert knowledge to collect and annotate. For instance, a 100 MHz wideband receiver can generate 800 Mbytes of data every second. In consequence, building a large labelled RF dataset is very time-consuming and costly. To reduce the manual efforts for building the dataset and extending it for future use, we employ an iterative training process. First, we collected and labelled a small set of individual RF emissions. Then, we converted them to 2D image-like *spectrum snapshots*, and applied various RF emission augmentations on the snapshots (e.g., shifting emission locations, or adjusting the SNR) to obtain a *Synthetic Labelled Dataset* to train the first model, called *supporting model*. Next, we collected a much larger dataset of RF samples from over-the-air transmissions and used the *supporting model* for the initial annotations which are manually corrected afterwards. Furthermore, this dataset was processed by our RF compression algorithm, and expanded by our RF sample augmentations (e.g., generating additional emission collisions) to obtain an *Extended Labelled Dataset*. Using this dataset, the *real-time models* were trained and evaluated leading to WRIST's final model. The overall workflow of our system is depicted in Figure 3. It is emphasized that the RF-centric compression layer is only present in the *real-time model*. In the next sections, we describe the core components of our system. We leave the description and evaluation of the *supporting model* in Section 3 and the *real-time model* in Section 4.

2.2 Spectral Representation

Our first step of designing the Deep Learning framework is defining the representation of RF signals as the input layer. While investigating the characteristics of different RF emissions, we observed and were motivated by the unique features characterizing various RF technologies, such as bandwidth, frequency range, frame format, modulation and coding schemes. For example, IEEE 802.15.1 standard (or Bluetooth) operates on 1 MHz channels using GPSK, DQPSK and DPSK modulation; while Wi-Fi signals (IEEE 801.11a/g/n/ac) use OFDM modulation for channels of 20–160 MHz. Such features are recognizable by visual analysis of frequency and/or time domain of the RF samples. Motivated by that, we use a image-like spectral representation towards designing WRIST, enabling the use of numerous cutting-edge DL solutions specially designed for images [15], [29], [30].

Our process of transforming raw RF samples to visual data is described as follows. We divide the I/Q data stream into equal chunks, and transform the data of each chunk into the frequency domain with an N-point Fast Fourier Transform (FFT) algorithm. The FFT outputs of M chunks are grouped to form a $M \times N$ matrix of complex samples. Then, a 2D grayscale image representing the 2D view of frequency spectrum (called *spectrum snapshot*) is created from the matrix by mapping each element $m_{x,y}$ (in column x and row y) to the corresponding integer value $p_{x,y}$ of pixel at coordinate (x, y):

$$\begin{aligned} p_{x,y} &= f(A_{x,y}) \\ &:= \gamma * (\min(\max(A_{x,y}, A_{min}), A_{max}) - A_{min}) \end{aligned} \tag{1}$$

where $A_{x,y} = 20 * \log_{10}|m_{x,y}| - N_0$ representing the SNR (in dB) of received emission at frequency bin x of the y-th chunk with respect to the noise floor N_0, while $A_{min} = -10, A_{max} = 50$ are the pre-calculated minimum/maximum SNR values that we consider for each spectrum snapshot. Lastly, $\gamma = 255/(A_{max} - A_{min})$ is the scaling factor of the SNR-pixel mapping.

Figure 4 shows the examples of visualized Wi-Fi, Bluetooth, and ZigBee emissions using our described transformation technique. We emphasize that, while only the magnitudes of complex samples $m_{x,y}$ are used in the transformation formula and the phases are omitted, the distinguishing RF features of different RF technologies, such as bandwidth or emission texture, are clearly visible. It should be noted that we used the representation in Equation (1) only for the *supporting model* (Section 3), where all RGB color channels receive the same value (grayscale mapping). For the *real-time model* (Section 4), we used the *RF-centric compression* scheme (described in section 2.4) to map each color channel to a different value, which results in full RGB spectrum snapshot as the input data for the final system.

2.3 Optimized Deep Learning Identification Network

The Deep Learning module of WRIST is inspired by YOLO [15], which is a popular *one-stage object detection* method to detect different objects in an image. YOLO is currently the fastest detection solution [31] with the capability of end-to-end processing using a single neural network. Because of that, this approach is much faster than the *two-stage object detection methods* relying on slow and complex pipelines [32], [33]. Towards achieving real-time RF identification, we optimized the DL architecture and algorithm to achieve better, faster RF identification results. Our optimizations are based on the observation that visualized RF emissions have distinctive characteristics such as size or texture, compared to real-life objects. We note that while YOLO was considered in a small number of prior work [20], [21], [22], the mechanisms for real-time, wideband identification of multiple RF technologies, as well as efficient data collection for large dataset compliant with DL operation requirements, are still lacking. Our techniques for the development of WRIST make significant enhancement to YOLO and adequately address the aforementioned problems.

First Optimization: RF-centric Anchor Boxes. Our neural network identifies emissions by outputting a set of *bounding boxes*, each for a potential emission in the input spectrum snapshot. All features, including emissions and noise, in every time and frequency slot need to be considered for detection. To achieve this, the network splits the input into a $S \times S$ grid, where each grid cell generates B bounding boxes predicting the emissions whose centers are located within that cell. The YOLO algorithm predicts using a set of pre-defined bounding boxes of specific sizes for each grid cell (called *anchor boxes*) as the references for the predicted objects. The anchor boxes are fundamental in many advanced detection methods [15], [31], [33], enabling the capability to capture objects of different aspect ratios. Therefore, it is essential for the anchor boxes to be suitable to the target objects that the DL model learns and predicts. A big observable difference between typical real-life objects (which are the learning targets of the original YOLO framework) and RF emissions is that the latter typically have highly-varying sizes (due to the variations of packet duration and bandwidth), instead of fixed sizes for the former. Hence, using *RF-centric* anchor boxes can enhance the learning, and provide more precise detection information for RF emissions. It is noted that the original anchor boxes in YOLO are the result of K-means clustering on the ImageNet dataset [34] used for computer vision tasks. Because the ImageNet dataset only contains data of real-life objects, the original anchor boxes of YOLO are not optimized for the task of RF identification. For that reason, we replaced the original anchor boxes in YOLO with our *RF-centric* anchor boxes generated by using K-means clustering algorithm on the RF training dataset. The later evaluation shows that using *RF-centric* anchor boxes achieves performance improvement compared to the unmodified YOLO algorithm.

An output bounding box has a *confidence score* to determine its relation with an RF emission, which is calculated as the product of the predicted emission presence ($P(E) \in \{0, 1\}$) and the Intersection-over-Union (IoU)

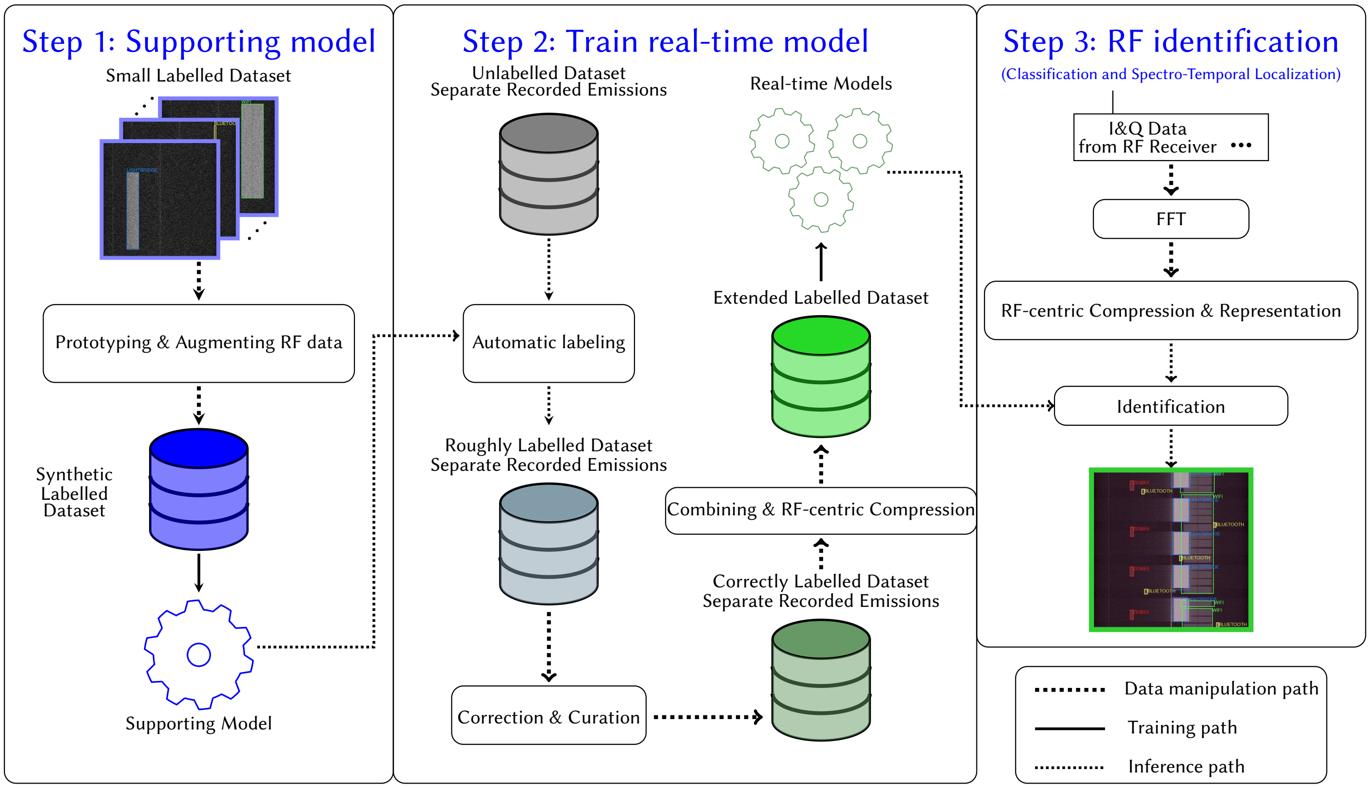
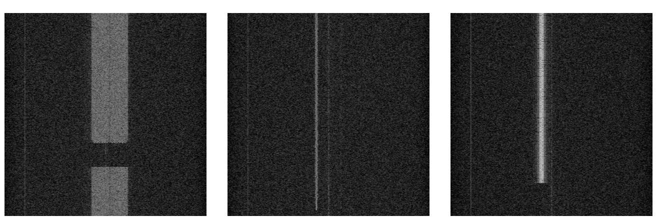


Fig. 3: The spectro-temporal RF identification workflow of WRIST.



(a) Wi-Fi (b) Bluetooth (c) ZigBee

Fig. 4: Different RF emissions are clearly distinguishable with the spectral representation.

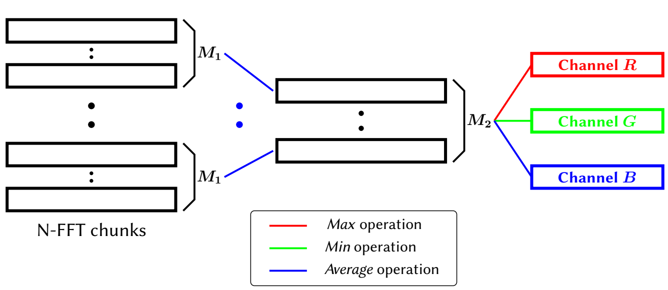


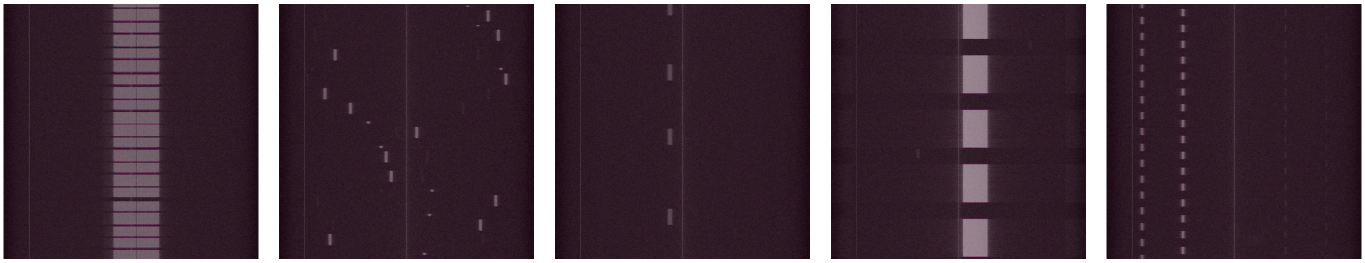
Fig. 5: RF-centric compression mechanism.

score:

$$IoU = \frac{S_I}{S_U} \tag{2}$$

where S_I is the overlapping area and S_U is the combining area of the detection box and the ground truth. The confidence score is equal to the IoU if there is an emission in the cell, otherwise is zero. A bounding box is predicted with the conditional probabilities $(P(\mathrm{RFclass}_l|E), l \in [1 \dots C])$ for all C different RF technologies (classes). The position of the box (spectro-temporal information) is described by four predicted variables: the center coordinates of the box (x_c, y_c) relative to the cell position, the width w and the height h relative to the size of input image.

Second Optimizations: Neural Network Layers. YOLO uses a deep convolutional neural network that incorporates three detection layers that split the input image into grids of three different scales. Prediction at larger scales utilizes the combination of upsampled feature maps from predictions at smaller scales, and the feature maps from the first layers of the network. This flexibility makes the network predict better the RF emissions whose visualized object sizes are significantly different (e.g., small Bluetooth vs. long and wide Wi-Fi emissions). The YOLO network can achieve real-time processing in computer vision [31]. However, utilizing the off-the-shelf YOLO network for wideband, real-time RF identification remains a challenge. We addressed this and achieved high accuracy of real-time identification by optimizing the original convolutional layers, along with utilizing an RF compression algorithm (to be presented in Section 2.4). We selectively reduced the volume of convolutional filters, based on an observation that visualized RF emissions are sharp and simpler than real-life objects (which are the initial targets for YOLO design). Therefore, there are less useful features needed to extract, and equivalently, smaller volume of convolutional filters is sufficient to identify such emissions. The filter reduction was performed step-by-step until we observed a significant increase in the validation error, using the following formula:



(a) Wi-Fi (b) Bluetooth (c) ZigBee (d) Lightbridge (e) XPD

Fig. 6: Spectrum snapshots of different RF emissions resulting from RF-centric compression.

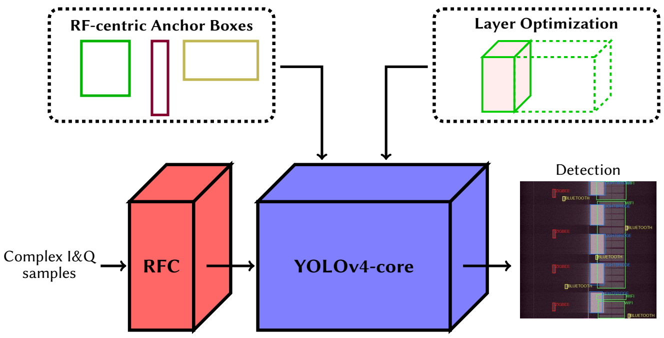


Fig. 7: SYL-4 real-time model comprises the RF-centric compression layer (RFC) and the optimized version of YOLOv4.

$U_i = U_{i-1} \times (1 - \sigma^i)$ where $\sigma = 0.5$ and U_i is the filter volume at the i^{th} step. In our experiments, we stopped the filter reduction after $i = 2$, resulting in the total reduction of 62.5%. Our optimized model is now more than 2.2 times faster while preserving detection performance.

Our neural network is trained by optimizing a loss function consisting of three components, each penalizes the error in one of the three categories: Box coordinates, confidence score, and class probabilities. The mean squared error loss is used for box coordinates error, while the cross-entropy loss is used for the other errors. The total loss is the sum of the losses calculated at the three detection layers of the neural network. During the network forward propagation, the i^{th} detection layer outputs a 3D *tensor* of size $S_i \times S_i \times [B \times (1 + 4 + C)]$ where $S_i \times S_i$ is the grid size of the i-th scale; B is the number of anchor boxes, fixed at 3; and C is the number of prediction classes. Also, there are cases when large RF emissions (e.g., Wi-Fi) spanning multiple cells result in redundant prediction boxes for the same object. We used non-maximal suppression algorithm introduced in [14] to remove these unnecessary boxes, which have IoU with the main prediction bounding box (i.e. one with the highest confidence score) exceeding 0.5, if they predict the same RF category.

2.4 RF-centric Compression

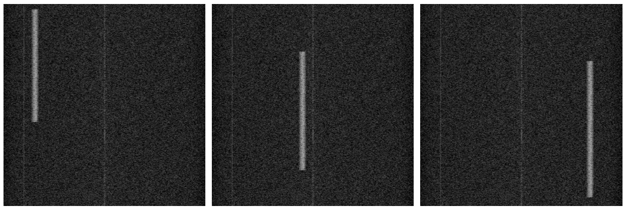
Utilizing the *one-stage object detection* can improve the detection speed, however, is insufficient for the real-time RF identification of wideband spectrum. Our initial model for RF emission identification required tens of milliseconds to process 100 MHz I/Q samples that span only a few milliseconds. If we increase the duration of input data, the spatial size of data passing through the network will subsequently increase, which makes the identification even slower. We explored several approaches and converged on a *RF-centric compression* layer as the first layer of the *real-time model*. This layer compresses multiple input spectrum snapshots into one that retains important features of the original data. The compression consists of two steps, illustrated in Figure 5. In the first step, the layer combines M_1 FFT outputs into one average chunk, i.e., for every group of M_1 chunks of FFT output $\{m_{x,y_1}\}$, where $0 \leq x < N$ and $0 \leq y_1 < M_1$, the layer computes the signal energy average $\frac{1}{M_1}\sum_{y_1}|m_{x,y_1}|^2$ on each individual frequency bin x across the time dimension.

In the second step, the layer processes the outputs from the first step to provide the mappings to RGB color channels of the final output. Let E_{x,y_2} denote the first step's results, where y_2 is the first step's output chunk index. Here, the layer again compresses M_2 chunks of $\{E_{x,y_2}\}$, where $0 \leq y_2 < M_2$, into one average chunk and obtain $E^{avg}_{x,y} = \frac{1}{M_2}\sum_{y_2} E_{x,y_2}$ for this second step's y-th output chunk. In addition to the average, the layers also computes the maximum and minimum value per frequency bin: $E^{max}_{x,y} = \max_{y_2}(E_{x,y_2})$ and $E^{min}_{x,y} = \min_{y_2}(E_{x,y_2})$. Each output chunk provided by the second compression is a row of the final 2D *spectrum snapshot*, where the SNR of a frequency bin is mapped to corresponding pixel values of RGB image channels with the mapping equations:

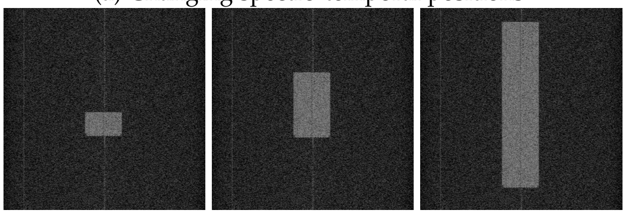
$$\begin{aligned} R_{x,y} &= f(10 \times \log_{10} E^{max}_{x,y} - N_0) \\ G_{x,y} &= f(10 \times \log_{10} E^{min}_{x,y} - N_0) \\ B_{x,y} &= f(10 \times \log_{10} E^{avg}_{x,y} - N_0) \end{aligned} \tag{3}$$

where $f(z)$ mapping function is defined in Equation (1). We emphasize that the mapping to create different RGB channels in this step is important to distinguish between different wireless technologies and improve the correctness of the predictions. We compare the model using our RGB transformation with one using only the averaging operation to create a grayscale transformation and discuss the advantages of our method in Section 4.1.

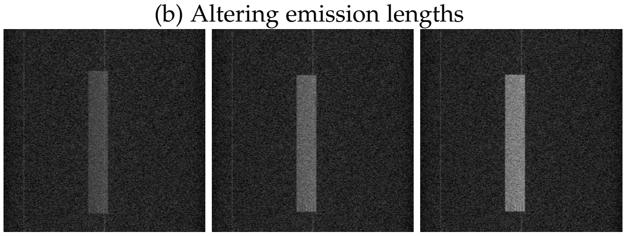
Although the compression discards some information in the original data, it still preserves important properties in the final spectrum snapshot such as the high and low peaks of RF emissions or signal strength variations over time in the wideband spectrum. The preserved properties are useful to distinguish different RF technologies. The resulting images in Figure 6 show how "compressed" RF emissions are clearly distinguishable. Towards developing this layer,



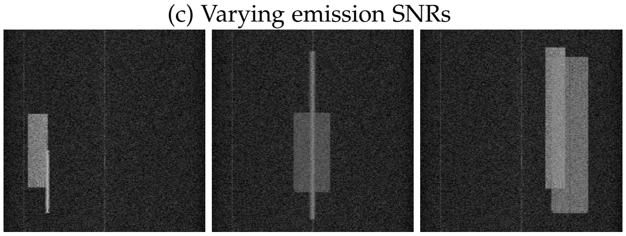
(a) Changing spectro-temporal positions



(b) Altering emission lengths



(c) Varying emission SNRs



(d) Simulating RF collisions

Fig. 8: Synthetic spectrum snapshots generated by image augmentation techniques.

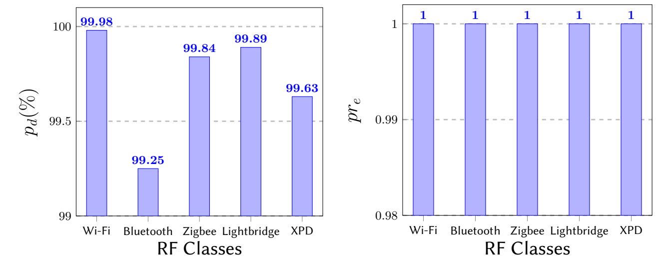


Fig. 9: Evaluation of class detection accuracy p_d (Left) and emission precision pr_e (Right) of the *supporting model*.

we are inspired by the pooling method in image processing neural networks which filters the most important features of the data and reduces the computation for the succeeding layers of the network.

We also considered dropping I/Q samples as an alternative approach for the real-time processing problem. However, simply discarding a large number of samples results in a high possibility of missing essential RF features to recognize RF emissions. As the extreme case, very short transmissions (e.g., Wi-Fi ACK packets) can be frequently missed. We concluded that the selective, RF-centric compression is a better choice. Figure 7 illustrates our finalized *real-time model* SYL-4 where the RF-centric compression layer is integrated into the optimized YOLOv4 [31] (YOLO-version 4) model.

3 The Supporting Model

In this section, we present the development of the *supporting model*, an important step towards building a high-quality training dataset for practical Deep Learning models. The *supporting model* was trained on a large synthetic dataset, which was built using efficient data augmentation techniques on a small labelled dataset. Because of the efficiency of training, this model was utilized to support automatic labelling of a much larger dataset of real RF emissions. We also refer to this model as *offline* or *non-realtime model*. We note that since the supporting model is only used to assist the training of real-time models, the real-time requirement is relaxed and the RF-centric compression layer is disabled.

3.1 Dataset of Synthetic RF Emissions

To train the *supporting model*, we created a *Synthetic Labelled Dataset* using efficient data generation process described as follows. We collected a small amount of RF data, generated and labelled the RF images. Then, we cropped the RF emissions out of the images and saved them to separated files, called the *prototypes* of the emissions. In the next step, we used those prototypes to generate new RF images by using various image augmentations: (1) adjusting the emission SNR resulting in the change of object brightness in the image, (2) changing the length of object by cropping or concatenating to vary the transmitted duration, (3) moving the object to different locations of an image to vary the spectro-temporal positions of emission (Note that the positions are selected following the frequency bands specified in the standard protocols), and (4) making different emissions overlap with each other to synthesize real-life wireless collisions (Augmentation results are depicted in Figure 8). These techniques allow us to efficiently generate sufficient training data mimicking real RF data. We note that while doing these augmentations, we systematically generated the annotations (including the category and four coordinates) for all emissions without manual efforts.

Consequently, we created a dataset of 99,067 RF images (47,672 images of single emission and 51,395 of colliding emissions) of size 512×512, where each image captures the view of a 100 MHz spectrum over 2.62ms time span with resolution of $N = 512$ frequency bins and $M = 512$ time slots. In total, the synthetic dataset contains 150,830 fully labelled synthetic emissions of five RF technologies: Wi-Fi, Bluetooth, ZigBee, Lightbridge (DJI protocol for robust aerial communications [27]), and XPD (Samson's cordless microphones [28]). The dataset is split into the training, validation, and test sets with the ratio $0.64 : 0.16 : 0.2$ correspondingly.

We reused the open-source YOLOv4 implementation [35], where Batch Normalization [36] is utilized to significantly improve the convergence and remove the requirements for regularization. We used the training batch size of 64 and learning rate $\alpha = 0.001$. We utilized Stochastic Gradient Descent optimizer with momentum $\beta = 0.9$ and weight decay $\lambda = 0.0005$. The neural network was trained

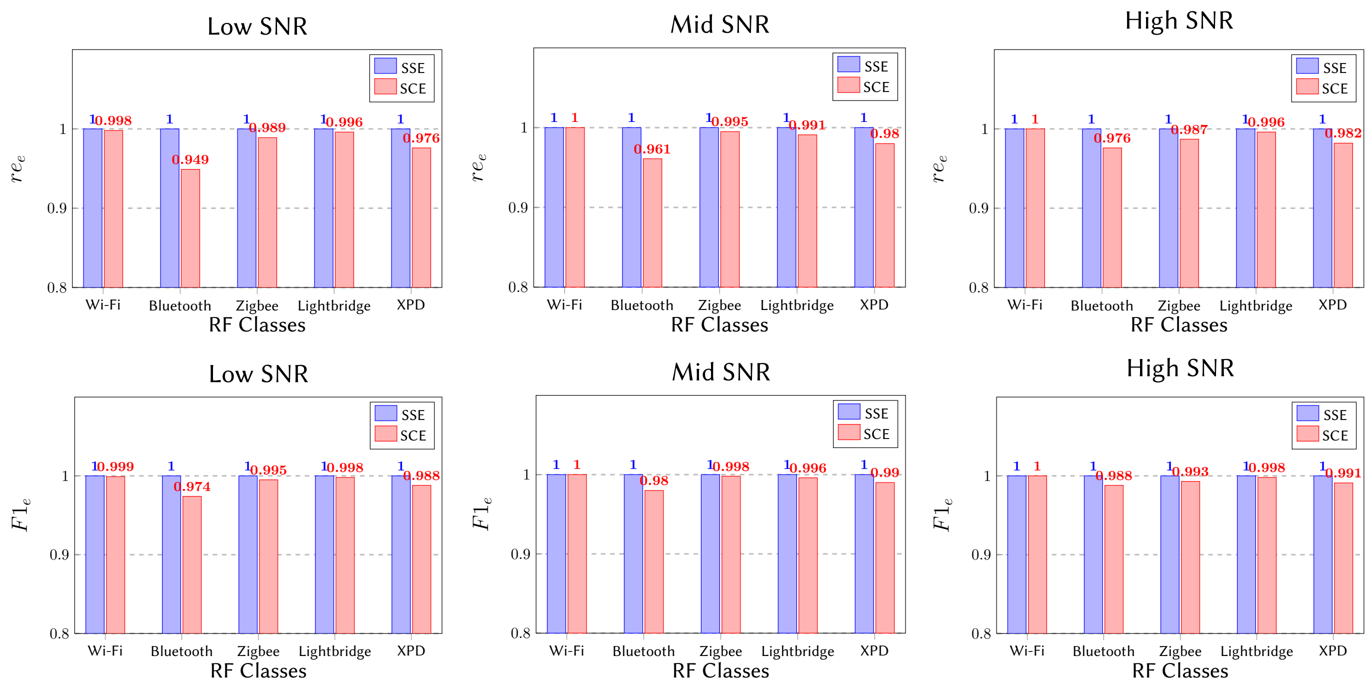


Fig. 10: Evaluation results of emission recall re_e (Upper row) and F1-score $F1_e$ (Lower row) for the *supporting model* with regards to RF classes, SNRs, and emission properties (Single (SSE)/Collision (SCE))

TABLE 2: Evaluation results of the *supporting model* on the synthetic dataset.

	Single emissions	Colliding emissions	Total
p_d	99.99%	99.46%	99.72%
pr_e	1.0	1.0	1.0
re_e	1.0	0.987	0.991
$F1_e$	1.0	0.993	0.995
$r_{\Delta BW}$	0.033	0.04	0.038
$r_{\Delta t}$	0.015	0.027	0.023

on a NVIDIA GeForce GTX 1080 GPU with the first 53 convolutional layers pre-trained on the ImageNet [34] dataset to reuse the visual feature maps learned from various real-life objects.

3.2 Evaluation Metrics

There is a lack of standard evaluation metrics for spectro-temporal RF identification. In our previous work [37], we only used mean average precision to evaluate the deep learning models. This is a popular metric in computer vision and is able to assess both the classification and the spectro-temporal information of the prediction. However, this metric does not explicitly provide important evaluation information for the classification and localization in the RF domain, such as the precision and recall of the emissions, or the error of the predicted bandwidths and time periods. Therefore, we use a set of fine-grain evaluation metrics to fully assess the proposed techniques:

Class Detection Accuracy p_d: This metric evaluates how well the model can recognize the presence of specific RF classes in the wideband spectrum. If an RF class has at least one emission in a spectrum snapshot, then that class is labelled 1 for this snapshot, otherwise 0. Meanwhile, the identification model classifies 1 for a class if it detects at least one emission of that class in the spectrum snapshot, otherwise 0. p_d is then calculated for every class as follows:

$$p_d = \frac{N_{cor}}{N_{tot}} \times 100\% \tag{4}$$

with N_{cor} and N_{tot} are the number of correct classification, and the total number of spectrum snapshots, respectively. The total accuracy is the average accuracy of all classes.

Emission Detection Metrics: We define a set of metrics to evaluate the detection capability on the emission level. To calculate these metrics, each output detection box is required to be mapped to the corresponding ground-truth emission that has *the same RF class* and *the highest non-zero Intersection-over-Union score* (calculated by Equation (2)). A detection is a true positive (TP) if it is mapped successfully, otherwise a false positive (FP). Also, any ground-truth emission not mapped to a detection is counted as a false negative (FN).

- Precision pr_e: This metric evaluates the correctness of an output detection and is calculated by:

$$pr_e = \frac{TP}{TP + FP} \tag{5}$$

- Recall re_e: This metric evaluates the ability to recognize all ground-truth emissions:

$$re_e = \frac{TP}{TP + FN} \tag{6}$$

- F1-score $F1_e$: This is a popular metric that is especially effective against unbalanced data [38]. The calculation combines precision and recall as follows:

$$F1_e = 2 \times \frac{pr_e \times re_e}{pr_e + re_e} \tag{7}$$

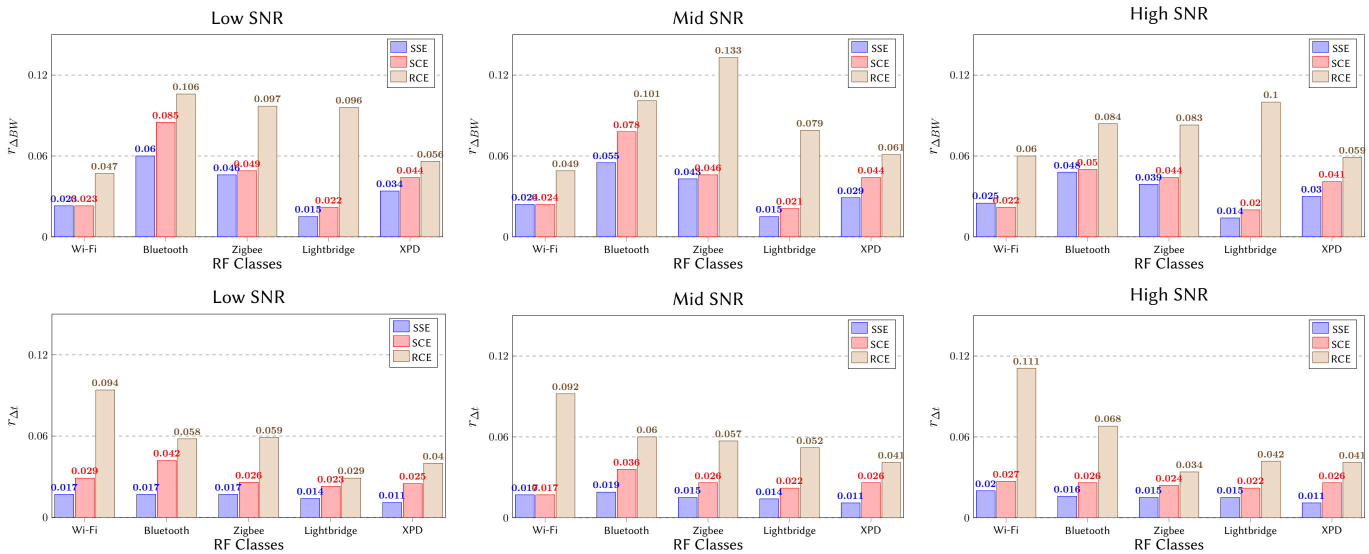


Fig. 11: Evaluation of BW offset ratio $r_{\Delta BW}$ (upper row) and time offset ratio $r_{\Delta t}$ (lower row) of the *supporting model* on the synthetic dataset regarding two cases - Single emissions (SSE) and Collision emissions (SCE), and the *real-time model* on the real dataset of compressed emissions (RCE).

- Bandwidth (BW) offset ratio $r_{\Delta BW}$: Inspired by the F1-score, we introduce this metric to simultaneously capture the correctness of the predicted center frequency and bandwidth in a single number. It is calculated for each detection-emission mapping by:

$$r_{\Delta BW} = \frac{|x_{lo} - \hat{x}_{lo}| + |x_{hi} - \hat{x}_{hi}|}{\hat{x}_{hi} - \hat{x}_{lo}} \tag{8}$$

  where x_{lo}, $\hat{x}_{lo}$ are the smaller x-values and x_{hi}, $\hat{x}_{hi}$ are the bigger x-values (in the x-axis of the spectrum snapshot, as shown in fig. 2) of the detection and the ground truth, respectively. We calculated the final $r_{\Delta BW}$ by taking the average of $r_{\Delta BW}$ from all the mappings.
- Time offset ratio $r_{\Delta t}$: Similarly, this metric simultaneously captures the correctness of the predicted start time and duration of emissions in a single number:

$$r_{\Delta t} = \frac{|y_{lo} - \hat{y}_{lo}| + |y_{hi} - \hat{y}_{hi}|}{\hat{y}_{hi} - \hat{y}_{lo}} \tag{9}$$

  where y_{lo}, $\hat{y}_{lo}$ are the respective smaller y-values and y_{hi}, $\hat{y}_{hi}$ are the respective bigger y-values (in the y-axis of the spectrum snapshot) of the detection and the ground truth. We calculated the final $r_{\Delta t}$ by taking the average of $r_{\Delta t}$ from all the mappings.

3.3 Experimental Results

The overall test result of the *supporting model* on the synthetic dataset is shown in Table 2. It is clear that our model can achieve over 99% of class detection accuracy in both cases of single emissions and colliding emissions. Furthermore, it also achieves the maximum score of 1.0 for emission precision pr_e. Meanwhile, the recall and F1-score slightly decrease in the collision case (re_e and $F1_e$ drops by 0.013 and 0.007 respectively, compared to the maximum score in single emission data). As a result, the offsets of predicted bandwidth and time period increase in the collision case, yet still remain very small (up to 0.04 for $r_{\Delta Bw}$ and 0.027 for $r_{\Delta t}$.

Figure 9 shows the variations of class detection accuracy and emission detection precision across the RF classes. It is evident that the *supporting model* achieves over 99% accuracy and maximum precision for all classes. Interestingly, classes with larger bandwidths like Wi-Fi and Lightbridge gain higher accuracy compared to others. Bluetooth emissions which have smallest bandwidths, has the lowest accuracy of 99.25%.

Figure 10 presents more details on how the model performance on re_e and $F1_e$ changes regarding different SNRs in cases of Synthetic Single Emissions (SSE) and Synthetic Colliding Emission (SCE). Emission SNRs are categorized into three groups: Low (5-14 dB), Mid (15-24 dB), and High (25 dB and above). We note that we do not focus on emissions with very low SNRs (significantly below decodability) in this work. As the minimum recommended SNR for data network is around 20 dB [39], such emissions do not pose an issue to existing RF communications. From the results, we see that the *supporting model* achieves maximum re_e and $F1_e$ in the SSE case across all classes and SNR levels. Meanwhile, both re_e and $F1_e$ slightly decreases in the SCE case, with the most significant drops observed by Bluetooth (re_e drops by 0.051 and $F1_e$ drops by 0.026) and XPD (re_e drops by 0.024 and $F1_e$ drops by 0.012) classes in low SNR. For all other cases, the model can provide the emission recall and F1-score higher than 0.98. The evaluation results of bandwidth and time offset ratios are shown in Figure 11, where we can see again that the model performs worse in the SCE case compared to the SSE case. Additionally, Bluetooth has the biggest prediction offsets, with $r_{\Delta BW} = 0.85$ for SCE and 0.06 for SSE, along with $r_{\Delta t} = 0.058$, all with low SNR level. For the other cases, the offset ratios of emission detections are negligible, below 0.08. Moreover, it is clear that the bandwidth offset ratios are generally higher than

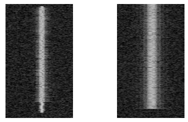


Fig. 12: Example of an RF property of real emissions that causes confusion for the *supporting model*. A saturated Bluetooth emission (Left) with leakage can be misclassified as a Zigbee emission (Right), and vice versa.

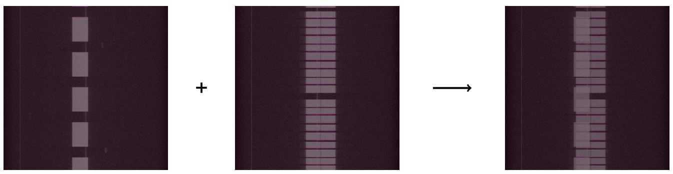


Fig. 13: The I/Q samples are combined in the time domain to generate new data patterns of the wideband spectrum, assuming the Additive White Gaussian Noise environment.

the time offset ratios across all classes and SNR levels.

4 The Real-time Model and System

In this section, we present the process of developing WRIST, which is centered around the *real-time model*. The real-time deep learning model is trained on the extended dataset of recorded RF emissions from various types of wireless radios, which was built by utilizing the labelling automation with the *supporting model*. Moreover, using our techniques of combining recorded RF samples, the training data can be scaled efficiently. We show in our evaluation that WRIST achieves real-time, wideband operation. Table 1 indicates that WRIST is the first Deep Learning-based approach capable of real-time, wideband spectro-temporal identification of RF emissions from commercial wireless radios.

4.1 Training and Evaluating Real-time Models

In this section, we evaluate and compare the performance of a minimally modified YOLO retrained on RF data (*rf*YOLO) and our optimized models to determine the best option for the *real-time model*. The optimized models are faster than the minimally modified models *rf*YOLO, and also provide better predictions. The ability to quickly generate predictions allows the system to either handle more data (i.e., by increasing the sample rate which can cover a larger bandwidth) or preserve more features to predict more precisely (i.e., by reducing the compression factor of the RF-centric compression layer). For this purpose, we reduced the volume of the convolutional filters, due to the fact that RF emissions are coarse and have significantly less features than real-life objects. Furthermore, the optimized models are more RF-centric with the anchor boxes derived (using k-means clustering) from the RF dataset instead of computer vision-based dataset of real-life objects (e.g., ImageNet [34]), as in YOLO models. Using the anchor boxes that better reflect the shape variations of RF emissions provides better precision in the detection. We applied our RF-centric optimizations (Section 2.3) to both YOLOv3 and YOLOv4, and retrained on RF data to generate two optimized models: SYL-3 and SYL-4, respectively. The impact of optimizations is analyzed and discussed below, through comparison between the models and their performance.

TABLE 3: 2.4 GHz RF transmitters used for this work.

Technology	Device model	Frequency range
Wi-Fi	TP-LINK TL-WN722N	2.412 - 2.462 GHz
Bluetooth	Avantree DG60	2.402 - 2.480 GHz
ZigBee	TI CC2430EM	2.400 - 2.483 GHz
Lightbridge	DJI Phantom 4	2.404 - 2.470 GHz
XPD	Samson XPD2 Lavalier	2.404 - 2.476 GHz

It is widely acknowledged that having a high quality dataset is essential to train a good Deep Learning model. Although a model trained on synthetic RF data can learn certain RF features, it is still incapable of capturing specific RF variations from over-the-air wireless emissions, as well as recovering from incorrect assumptions in the synthetic data. Upon testing the *supporting model* on several recorded RF emissions, the most common error was misclassification due to unexpected RF-related factors in the synthetic data such as out-of-band leakage (Figure 12). Fortunately, such errors presented patterns which could easily be corrected using automated tools. Because of that, the *supporting model* was utilized for automatic labelling to produce a large dataset of recorded over-the-air RF emissions. The process of building a training dataset for the real-time models is illustrated in ***Step 2*** of Figure 3. We collected RF emissions from five radio technologies: Wi-Fi, Bluetooth, ZigBee, Lightbridge and XPD. We adjusted the transmission SNR in three ranges (measured in dB units): Low (5-14), Mid (15-24), and High (25 and above). Again, we emphasize that recognition of RF emissions with very low SNR levels that have minimal impact to coexisting communications is not the focus of this work. Firstly, for our goal of obtaining clean recordings with minimal interference in the 2.4 GHz ISM band, we recorded different RF emission types separately. Secondly, we generated uncompressed RF images and used the *supporting model* to automatically labelled the images. Then, we manually corrected and adjusted the annotation boxes. In the next step, we expanded the dataset to avoid overfitting and enhance the generalization for the trained models. To do that, we combined the separate recordings' I/Q samples in the time domain to produce a much larger dataset with coexisting and overlapping patterns of different types of RF emissions without incurring additional effort in labeling, by re-using the corrected and curated annotations from the first step (Figure 13). We note that, in contrast to the synthetic dataset used for the supporting model, the dataset for the real-time model was built from the recorded emissions from real RF radios, and was extended using RF combining manipulations.

The final step before training the *real-time model* is the RF-centric compression. We chose the two compression parameters based on two criteria: First, the product of two parameters (i.e., the total compression factor) is sufficient to extend the duration of each spectrum snapshot to meet the detection time of the models. Second, the two integer-valued parameters are roughly equal to balance the effects of the two levels of compression. Consequently, we used parameters $M_1 = M_2 = 5$ to allow real-time capabilities

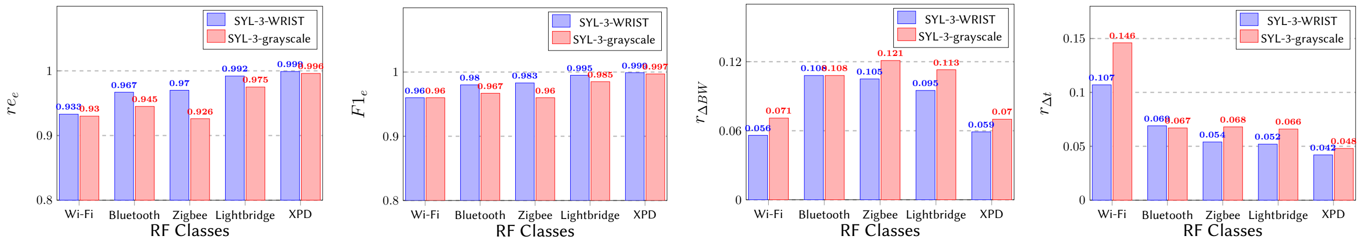


Fig. 14: Performance comparison between a model using RF-centric compression with our RGB transformation and a model using fully grayscale compression with averaging operation.

for the original models *rf*YOLO and generated a dataset of $253,397$ compressed RF images of size 512×512 containing over 4 million labelled RF emissions. Meanwhile, our optimized models SYL-3 and SYL-4 have faster detection speeds (shown in Table 4) that allow us to reduce the compression with $M_1 = 3, M_2 = 4$ in the new dataset. We split the datasets with ratio $0.64 : 0.16 : 0.2$ for the training, validation, and test sets, respectively. The same training hyperparameters were used for the *supporting model*. We emphasize that the *real-time model* was trained from scratch, in contrast to the *supporting model* which used transfer learning with pre-trained preceding layers.

Evaluation for RF Transformation. We validate our design of RF-centric compression by comparing two models: One uses our RGB transformation (with three different mapping functions in the second step of the compression) described in Section 2.4 and the other uses the grayscale transformation with the same averaging operation for both three image channels. We used our optimized YOLOv3 architecture (SYL-3) and trained the models using the same settings, compression factor, and hyper-parameters. The results in Figure 14 show that our method outperforms the grayscale compression. Our model has generally higher precision and F1-score, achieves up to 4.4% better for precision and 2.3% better for F1 (in Zigbee). Moreover, our predictions are more aligned with the ground truths. WRIST's predictions have lower time and bandwidth offset ratio, with the most significant difference seen in Wi-Fi (nearly 4% lower for time offset). The results confirm the effectiveness of our RF-centric compression design.

TABLE 4: Detection time of different *real-time models*.

Model	YOLOv3	SYL-3	YOLOv4	SYL-4
Detection time	44.19 ms	17.23 ms	51.35 ms	22.96 ms

Evaluation for Model Optimization. Figure 15 compares the performance of the optimized and original real-time models using the test datasets. It is clear that SYL-3 achieves significant improvement for Wi-Fi compared to the corresponding original model *rf*YOLOv3 with re_e of 0.93 compared to 0.85 and $F1_e$ of 0.96 compared to 0.92. Similarly, SYL-4 also gain an improvement compared to *rf*YOLOv4 with re_e of 0.06 higher and $F1_e$ of 0.02 higher. While there are no significant changes for other classes, the big improvement for Wi-Fi emissions observed in both SYL-3 and SYL-4 justifies the effectiveness of RF-centric optimizations. More importantly, besides having competitive performance, SYL-3 and SYL-4 models are more than $2.2\times$ faster than the corresponding original models, as shown in Table 4. Consequently, we chose SYL-4 trained with lower-compressed data as our final *real-time model* for WRIST.

Evaluation for Different SNRs and Classes. Using the final model, we achieve over 99% of class detection accuracy p_d and over 0.99 of emission detection precision pr_e for all classes, as shown in Figure 16. Figure 17 provides more details on how the final *real-time model* performs with regards to different classes as well as SNR levels. It is clear that our model gets over 0.99 of pr_e regardless of SNR levels and RF classes. Furthermore, we achieve over 0.94 of emission recall re_e for most of the cases except high SNR Wi-Fi, where the emissions start to create confusing visual patterns such as the RF leakage. Meanwhile, F1-score maintains very high values, over 0.96 in all cases. Furthermore, in this real dataset, XPD is the most recognizable category with over 0.997 for both precision, recall, and F1-score regardless of SNR levels. We also observe significant improvements between Low and Mid SNR levels with Zigbee class (re_e increases from 0.961 to 0.998) and between Mid and High SNRs with Bluetooth (re_e increases from 0.947 to 0.993).

The evaluation of bandwidth (BW) and time offset ratios for the *real-time model* (with Real Compressed Emissions - RCE) with regards to RF classes and SNR levels is illustrated in Figure 11. Is it easy to see that compared to the model trained on uncompressed data (in SSE and SCE cases), the *real-time model* trained on compressed data exhibits higher offset ratios for both bandwidth and time period. This is expected because we need to reduce the amount of processed information (by RF-centric compression) to enable the real-time capabilities for our model and system. Nonetheless, we are still able to maintain $r_{\Delta BW}$ below 0.14 and $r_{\Delta t}$ below 0.12 for all cases. Furtheremore, while Wi-Fi and XPD have the lowest $r_{\Delta BW}$ in all three SNR levels, XPD and Lightbridge provide better time offset ratios, with $r_{\Delta t} = 0.029$ and 0.041 for Low SNR Lightbridge, and XPD in Mid and High SNR, respectively.

4.2 WRIST System

Implementation. Our hardware setup consists of an Ettus USRP X310 connected with a host computer via the 10G Ethernet interface. The host computer is equipped with a 6-core Intel Core i7-8700@3.2GHz processor, a NVIDIA GeForce GTX 1080 Graphics Card, and 32 GB RAM. The integrated implementation of WRIST has two main parts: The first part, written in C++, comprises (1) the module for RF sample

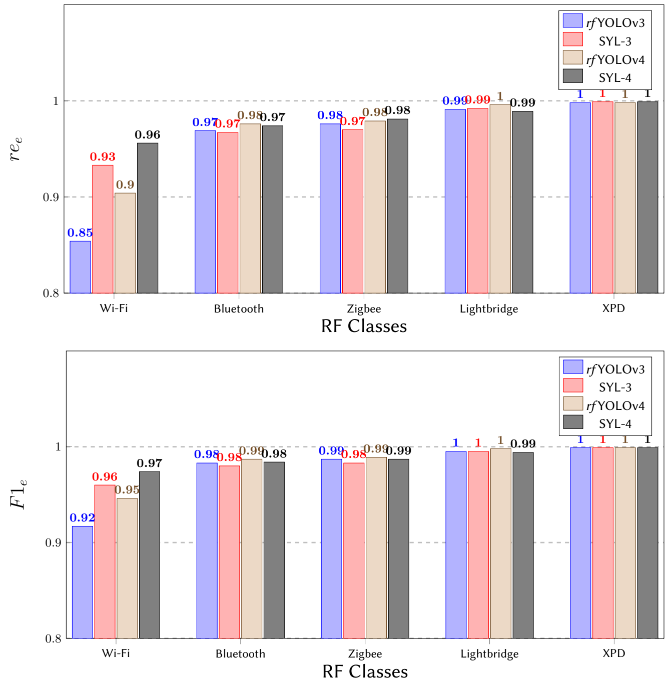


Fig. 15: Results of SYL-3, SYL-4 compared with the original models. *rf*YOLOv3 and *rf*YOLOv4 use higher compression parameters $M_1 = M_2 = 5$ to enable real-time WRIST, while SYL-3 and SYL-4 use lower compression with $M_1 = 3, M_2 = 4$ to exploit the speed improvement after model optimizations.

collection from the USRP and (2) the RF-centric compression module. The RF compression algorithm is optimized and implemented with the help of GNU Radio VOLK Library [40]. Also, it is noted that the FFT computation is handled on the host CPU instead of the GPU. The second part includes the detection module, which is written in Python based on the SYL-4 framework and utilizes the GPU. We handle the data communications between the two parts by a custom message passing protocol using Google Protobuf and the ZeroMQ messaging library.

Real-time Microbenchmarks. We ran our implemented WRIST numerous times over extended period of time (over 100 times, each time for at least 5 minutes each) validating the real-time capability and without experiencing any overflows (which would indicate if the system is slower than the incoming rate of RF samples, resulting in sample drops). Therefore, WRIST achieves real-time performance. To provide more fine-grained details of the system pipeline, we benchmarked each of WRIST's modules of the pipeline. The requirement of real-time processing for a 100MHz monitored spectrum is that the processing rate of each module needs to exceed 100 Msamps/s (Million samples per second). The throughput of the modules was measured on the host computer to evaluate the real-time capability of WRIST, shown in Table 5. We emphasize that the detection module, along with the FFT and RF-centric compression modules, are parallelized in our system. It is clear that the RF detection module is the bottleneck in the system. Nonetheless, this module still sustains the incoming sample

TABLE 5: WRIST's real-time microbenchmarks.

Module	Throughput
Detection	130.8 Msamps/s
FFT	182.14 Msamps/s
RF-centric compression	3679.04 Msamps/s
WRIST	130.8 Msamps/s

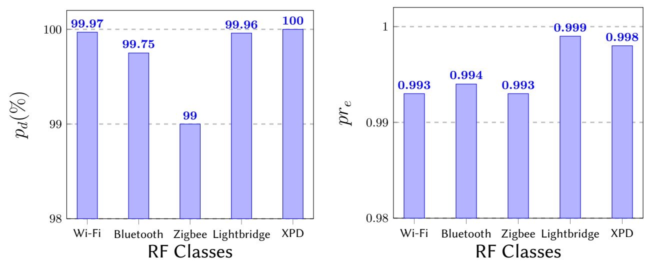


Fig. 16: Evaluation of class detection accuracy p_d (Left) and emission precision pr_e (Right) of the final *real-time model* SYL-4 on the test dataset.

rate of 100 Msamps/s, ensuring the real-time operation of WRIST.

Performance in anechoic chamber. To validate the practicality of WRIST's RF identification in different over-the-air environments, we evaluate with data collected in a $60 \times 60 \times 30$ ft anechoic chamber (Figure 18). To create a crowded spectrum environment which is common in real life scenarios, we positioned the RF devices Figure 19 in different locations inside the chamber and set up their transmissions in different bands. We recorded and evaluated on $89,281$ labeled RF emissions. The results are shown in Table 6, where we can see that WRIST's model still achieves over 99% class detection accuracy. The performance slightly declines with respect to some metrics evaluating fine-grained emission detections compared to the previous evaluation on test dataset, due to the complex patterns introduced by the crowded spectrum. Nonetheless, our system still maintains very high score of pr_e (0.969) , re_e (0.94) and $F1_e$ (0.954), while only exhibit very small BW and time offset ratios (0.055 and 0.062, respectively). Figure 20a illustrates the spectro-temporal identifications of Wi-Fi, Bluetooth, ZigBee, Lightbridge and XPD emissions with the respective green, yellow, red, blue and purple rectangular boxes correctly positioned at the corresponding emissions. These illustrations show that WRIST accurately detects, classifies and precisely identifies the spectro-temporal position for every single RF emission under various transmission and collision patterns.

Performance in the wild. We evaluated WRIST's capabilities in the extreme case when operating in the congested spectrum in the wild. We collected and labelled $1,101$ emissions in this environment. The results shown in Table 6 indicate that in this scenario, the performance of WRIST further degrades compared to previous evaluation settings. This is understandable because the task difficulty also increases due to the greater volume of emissions and more complex collision patterns, which make the task challenging even for human vision (examples shown in Figure 20b). However, it is interesting that we achieve $p_d = 100\%$. The reason might

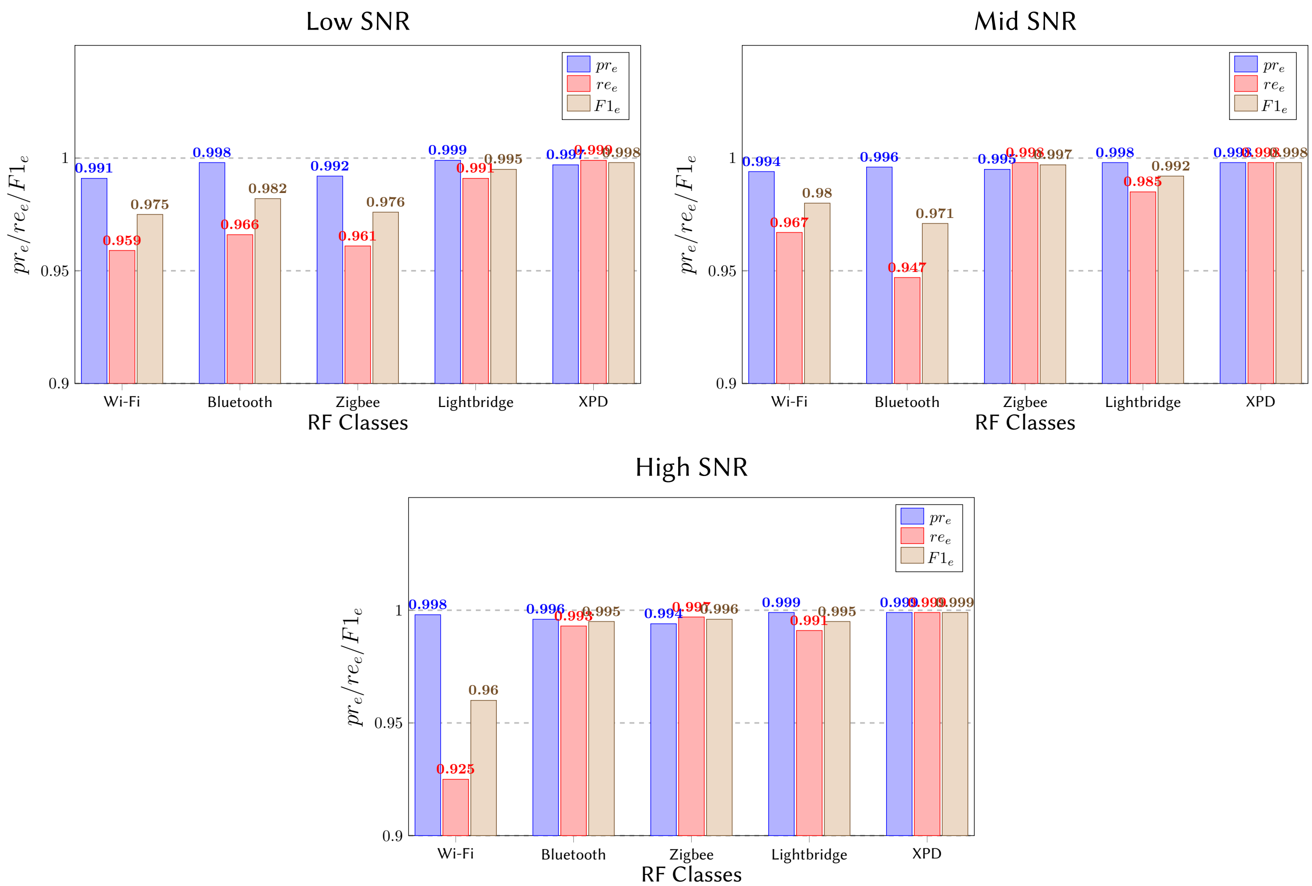


Fig. 17: Evaluation results for the *real-time model* (with SYL-4 as our final choice) with regards to RF classes and SNRs.

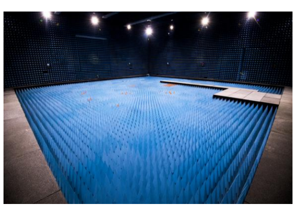


Fig. 18: Anechoic chamber.

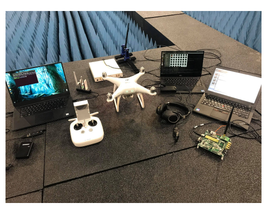


Fig. 19: Wireless devices used in this work.

seem to be because there are many emissions of different RF classes within a spectrum snapshot, and therefore one can achieve high prediction accuracy by just always returning 1. However, it is not the fundamental reason of WRIST's good performance. WRIST can infer the category, time and frequency information of every single emission in the busy spectrum, and therefore can recognize the presence of RF technologies. Thanks to that, both the offset ratios $r_{\Delta BW}$ and $r_{\Delta t}$ are below 0.06. These results are comparable to the previous settings, indicate that our system can predict correctly the spectro-temporal information of emissions even in the extremely congested frequency spectrum. The results of pr_e, re_e and $F1_e$ degrade compared to previous over-the-air environments, however, still maintain the high scores of 0.87, 0.83 and 0.849, respectively.

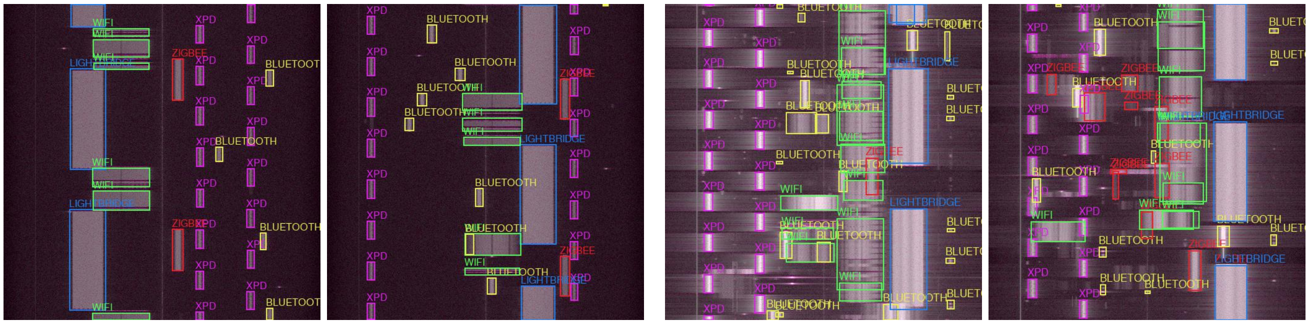
5 SPREAD DATASET

Sharing a large, labelled datasets of RF emissions is critical to scientific research. It allows researchers to collaborate, enables access to valuable data to people who are unable to acquire the necessary equipment, and stimulates the development of new techniques, models, and DL architectures for RF research. In the absence of a large, curated dataset for spectro-temporal identification of different RF technologies, we are making our dataset SPREAD [2] available to the research community. SPREAD is complemented with the supporting API for the reproduction of the iterative learning approach, as well as the expansion of the dataset with new RF technologies. In the following, we present the elements of the dataset and how it is structured, then provide important details of the API.

5.1 Data and Format

SPREAD currently supports five RF categories: Wi-Fi, Bluetooth, Zigbee. Lightbridge (Wireless communication protocol of DJI drones [27]), and XPD (Samson's wireless microphone [28]). The device models of the transmitters that we collected data from are shown in Table 3. The top level of SPREAD contains the following components:

2. Abbreviated from **Sp**ectro-temporal **R**F **E**mission **A**nalysis **D**ataset. The dataset and API are shared at https://sprite.ccs.neu.edu/datasets/SPREAD/.



(a) Inside anechoic chamber

(b) In the wild

Fig. 20: WRIST's detections in congested environments

Off-the-shelf Datasets. This includes the datasets that we used for the training, validation, and evaluation of the DL models, described in the previous sections. Each dataset consists of the spectrum snapshots/images and annotations of emissions present in each image. The datasets contain both synthetic data (that we generated to train the *supporting model*) as well as the real, compressed data of emissions collected in different over-the-air environments. It is noted that in-the-wild data and labels are not made publicly available for privacy reasons, while the rest of the recordings were carefully collected using randomly generated data, streamed music or generic video footage. In total, SPREAD provides $99,067$ images for synthetic data and $780,664$ images for real emission data (The latter corresponds to over 10 million labelled emissions in over 13 Terabytes of RF samples). We emphasize that, while the datasets can be used as-is for training and testing of new models, users can utilize our supporting API (described later in Section 5.2) to expand to an even larger dataset based on their needs.

The RF images are stored in JPG format, while the annotations are stored in an accompanying TXT file of the image. An emission annotation (a bounding box exampled in Figure 2) is represented as one line in the TXT file with the format `<class_No> <x_center> <y_center> <width> <height>`. The class index number `class_No` for each RF category is defined beforehand in the global metadata. It should be noted that the positions and sizes of the annotated boxes are relative to image sizes. The images and annotation files are named similarly, where `<class_name>_<image_No_in_class>` format is used for synthetic data. Meanwhile, the real compressed data uses the format `<rec_type>_<rec_No>_pic_<pic_No_in_rec>`, where `rec_type` is `syn` if the recording results from RF combining of recorded samples, otherwise `rec` (implies recorded emissions from devices).

Recorded Samples. The raw I/Q samples collected from RF devices using software-defined radios are stored in binary files with the 64-bit complex format (Real and imaginary parts are 32 bits each). Each sample file is named with format `rec_<rec_No>`. We note that the raw RF samples resulting from RF combining transformation do not need to be published, as they can be regenerated by combining the recorded raw samples using our API.

TABLE 6: Performance evaluation of WRIST in different realistic environments.

	Dataset recording	Anechoic chamber	In-the-wild
p_d	99.74%	99.87%	100%
pr_e	0.996	0.969	0.87
re_e	0.977	0.94	0.83
$F1_e$	0.986	0.954	0.849
$r_{\triangle BW}$	0.078	0.055	0.056
$r_{\triangle t}$	0.058	0.062	0.054

Metadata. SPREAD provides the metadata for both the image datasets and the sample recordings. The metadata is important to describe the datasets, as well as to allow different tools to operate on the datasets. An image dataset is associated with a global metadata file that specifies the following:

- *RF properties*: The number of RF categories as well as the mapping between the categories and indexes.
- *Image properties*: The type of the dataset (synthetic or compressed), the number of images and image sizes (width and height), and the compression factors M_1, M_2 (the values are `null` if the dataset is synthetic).
- *Others*: Auxiliary information such as the date when the dataset is generated, or the environment and author of the dataset.

Additionally, each RF samples file is represented by a corresponding recording metadata file with the following fields:

- *Information of RF emissions*: This includes the RF categories present in the recording, as well as the corresponding SNR levels, channel numbers and transmitting devices.
- *Recording condition*: The parameters of the receiving radios, such as sample rate, center frequency, and device model, as well as the physical recording environment and measured noise floor power (in dB).
- *Auxiliary information*: The name, duration, date and the author of the recording.

All the metadata is structured and stored in JSON files. Figure 21 illustrates the example contents of the metadata files.

5.2 Supporting API

SPREAD also provides a Python API to support the use and manipulation of data. Here, we emphasize the functionality needed to reproduce the iterative learning process and expand the dataset with more categories, devices supported, and data.

Creating Synthetic Data. In our API, the function `gen_synthetic_data()` is responsible for generating synthetic RF images towards the training of *supporting model*. This function sequentially calls `gen_synthetic_single_emission()` to generate single emission data and `gen_synthetic_colliding_emission()` to generate colliding emission data. Those two functions, which users can use independently, take the list of RF categories and their corresponding emission prototypes, and apply image augmentations to create synthetic spectrum snapshots with different emission characterictics, such as pattern, SNR and spectro-temporal position (described in Section 3.1). The ranges of SNR and positions can be specified by users and passed to those functions as the input parameters.

RF Combining Transformation. Combining recorded I/Q samples is an important step of our iterative learning process, as it enriches the training data and improves data diversity with new patterns without requiring extra manual effort for labelling. We implemented the function `combine_recordings()` to combine raw I/Q sample files (recordings) recorded from emissions of different RF categories and output new sample files in the directory specified by the users. This functions adds the samples together in the time domain using GNURadio flowgraphs [40]. Furthermore, the metadata information and annotations of the original recordings are selectively copied over the new metadata and annotation files of the combined recording. We note that the original annotations for the recorded samples (provided by automatic labeling process using the *supporting model* and corrected manually afterwards) are made for the uncompressed RF emissions.

RF Compression. Training the *real-time model* requires a dataset of real, compressed RF emissions. We provide the function `gen_compressed_data()` that processes RF recordings to generate RGB spectrum snapshots using the compression algorithm described in Section 2.4. The user specifies the compression parameters and the input/output directories, and this function will go through each recording, map SNRs to corresponding image pixel values, and generate compressed RF images from pixel matrices. Moreover, the original annotations of each recording are adjusted to account for the compression.

Finally, we provide several helper functions such as to create the metadata files necessary for the training of RF identification models and the expansion of dataset. By making the API available along with the dataset, we hope SPREAD will serve as a building block of new DL models and techniques for wideband spectro-temporal RF identification in the future.

6 Discussion and Related Work

In this paper, we propose an RF data transformation technique to represent RF data in images. Our technique uses Fast Fourier Transform (FFT) to convert RF signal to frequency domain, and makes it clearly visible in the image representing the wideband 100 MHz spectrum. This design is specific to our system that support wideband, real-time RF identification. For other systems that support very narrow bandwidth, we would recommend Short-Time Fourier Transform (STFT) as an alternative approach for RF representation. Similarly, signals with very short durations (in the order of a microsecond) would also benefit from using an STFT.


```
{
    "n_classes": 5,
    "class_indexes": {
        "wifi": 0,
        "bluetooth": 1,
        "zigbee": 2,
        "lightbridge": 3,
        "xpd": 4
    },
    "n_images": 10000,
    "data_type": "compressed",
    "img_width": 512,
    "img_height": 512,
    "compress_M1": 4,
    "compress_M2": 3,
    "environment": "over-the-air",
    "date": "XX-XX-20XX",
    "author": "sprite-team"
}
```

```
{
    "rec_name": "rec_11",
    "classes": [
        "wifi"
    ],
    "snr": {
        "wifi": "high"
    },
    "channels": {
        "wifi": 7
    },
    "tx": {
        "wifi": "TP-LINK TL-WN722N"
    },
    "rx": "Ettus USRP X310",
    "samp_rate": 100000000,
    "fc": 2440000000,
    "noise_db": -44.16855294618486,
    "duration": 10,
    "environment": "over-the-air",
    "date_recorded": "20XX-XX-XX_XX:XX:XX",
    "author": "sprite-team"
}
```


Fig. 21: Example metadata of image dataset (Left) and recording (Right).

As we mentioned in previous sections, we do not focus on signals with very low SNRs in this work. Nonetheless, identifying such signals would also be helpful for some wireless systems, for example, systems that monitor long-range, low-SNR IoT communications. Therefore, it is an interesting direction for our future work, where we will incorporate more features specifically observed in RF signals (e.g., phase component or cyclostationary features) into the identification mechanism to improve the performance and support more scenarios.

RF Detection and Classification. RF detection and classification problems have attracted significant attention in the research community over the past decades. There has been significant effort investigating the performance of various expert features for the task of recognizing unique RF signal characteristics, such as higher order signal statistics (e.g., cumulants [41], moments [42]), or cyclostationary signatures [43]. However, these methods require domain knowledge, and moreover, new RF technologies are proliferating nowadays requiring the redesign of fingerprinting algorithms developed for old standards. Therefore, automatic feature extraction and learning have been increasingly important. Deep neural networks have the capabilities to learn distinguishing features from captured RF samples, and were investigated recently in RF classification tasks, such as modulation recognition [25], [44], radar detection [45], collision detection [46], or RF fingerprinting of ZigBee [47], LoRa [48], and Wi-Fi devices [49]. Recently, Deep Learning is also utilized for universal beamforming system [50]. The rapid development of new DL techniques and architectures such as Convolutional Neural Networks, Recurrent Neural Networks, and Generative Adversarial Networks and their success across application domains have spurred the progress and achievement of DL-based RF detection and classification. Nonetheless, the main drawback of aforementioned works is the absence of real-time, wideband capabilities to detect, classify, and spectro-temporally localize multiple multi-channel emissions.

Effective wideband RF detection and classification necessitate various information about coexisting RF radio technologies instead of a single signal property as modulation scheme. As a matter of fact, different RF technologies can share a common communication technique, with an example where Bluetooth Low Energy and WirelessHART standards both use spread spectrum modulation (DSSS/FHSS). This leads to the increasing interest for detection and classification of RF technologies in the wide, congesting, unlicensed bands. Previous work [18], [19] classifies three popular 2.4 GHz technologies: Wi-Fi b/g, Bluetooth, and ZigBee. The authors used ResNet [29] and CLDNN [51] models to classify either raw [19] or FFT-transformed [18] I/Q samples. Authors in [16] classifies similar RF categories in real-time, leveraging a sample-dropping algorithm. However, those works do not classify separate emissions, lack the real-time wideband processing capabilities, and do not infer emission's spectro-temporal information. Recently, there are several works that utilizes deep object detection networks to predict spectro-temporal locations of RF signals in the spectrum. In [21], [22], authors leverage YOLO network to infer the information for recorded RF signals. Authors in [24] also uses YOLO to detect and classify RF signals of different modulations in MATLAB simulations. Meanwhile, [23] instead uses R-CNN to predict time-frequency locations of MATLAB-simulated signals. In [17], authors also apply R-CNN to predict RF-signals of eight technologies (DECT, Bluetooth, ATSC, LTE, UAS, Vulos, ADSB, and Wi-Fi) produced by a synthetic data generation process. However, none of those works address the problem of real-time processing for either narrowband or wideband data. Compared to those works, our model and system achieve wideband, real-time spectro-temporal RF identification, and moreover, make considerable performance improvements on the original YOLO models in our task. The iterative learning approach, architectures, and final models, will provide a solid basis for the critical tasks of automatic non-cooperative spectrum management mechanisms in the future, as well as other applications such as the detection of malicious drones, and jammers.

Furthermore, large and curated datasets are crucial for training deep neural networks. It is even more important to make the data available to the research community to spur the development of new RFML techniques, architectures, and models. Authors in [52] published a dataset including both simulated and recorded over-the-air signals of different modulations for the modulation recognition task, where the RF technology information is not available. Authors in [18] built a dataset consisting of Wi-Fi, Bluetooth and Zigbee signals transmitted from a signal generator instead of commercial RF devices. These datasets have other common critical limitations, that are, the data samples in cases of concurrent and/or colliding transmissions, collected with a sample rate much higher than the standard bandwidth of considered RF technologies. Our developed dataset addresses these limitations allowing to develop DL solutions for practical real-time, wideband RF detection, classification, and spectro-temporal location prediction.

Deep Detection Networks. In one of the first efforts for DL-based object detection, Girshick et al. introduced the R-CNN detection framework that operates in two phases: Identifying possible regions where an object could be present (region selection process) and classifying objects in those regions using deep neural network. This *two-stages* method can provide very precise detections, but incurs an expensive computation cost. Following works improve the speed of R-CNN with the new region selection algorithms (the bottleneck of R-CNN) using a separately-trained network [53] or a sub-network sharing the feature maps with the detection network [33]. However, these approaches still lack the necessary real-time capability. Meanwhile, *one-stage* methods like YOLO [14], [15] prioritize the prediction speed. YOLO detect objects in a small set of regions obtained by gridding the image, instead of making excessive region selections. This mechanism of YOLO results in fairly accurate detections of distinguishing objects from the image background with very little amount of time. Later *one-stage object detection* methods such as RetinaNet [54] and EfficientDet [55] tried to improve the detection accuracy with new DL techniques including Focal Loss and Feature Pyramid Networks, at the expense of substantial loss of prediction speed. The latest YOLOv4 [31] incorporated various advanced DL techniques to enhance the previous versions, and consequently achieved considerable performance improvement with a minimal decrease of the speed. Based on that, we designed the SYL-4 framework for spectro-temporal RF identification with further enhancement in prediction speed, that allows faster, and better predictions of RF emissions.

7 Conclusion

Understanding RF emissions in real-time is a crucial capability. We presented WRIST, a wideband, real-time spectro-temporal RF identification system. The system provides high accuracy, low latency detection, classification, and localization of RF emissions. It relies on optimized *one-stage object detection* mechanisms integrated with a *RF-centric compression*. Our iterative learning approach consisting of training and leveraging a *supporting model* and a *real-time model* allowed us to create a curated and labeled dataset of over-the-air RF emissions. The deep learning models evaluation on commercial SDR peripherals proved that real-time, wideband identification is not only feasible, but also achieves over 99% of class detection accuracy even in the congested in-the-wild environment. Furthermore, the spectro-temporal emission detections provided by our system can achieve very high precision (up to 0.99), recall (up to 0.97), F1-score (up to 0.98) and low bandwidth and time offset ratios (less than 0.1). We also introduce SPREAD, a large, curated, and labelled dataset that we will open to the community for RFML research. Our iterative process developed within WRIST can be applied to new waveforms and RF emission patterns to expand the dataset.

Acknowledgments

This work was partially supported by grants NAVY/N00014-20-1-2124, NCAE-Cyber Research Program, and NSF/DGE-1661532.

References

[1] Fairwaves, "XTRX: The first ever truly embedded SDR," 2019. [Online]. Available: https://www.crowdsupply.com/fairwaves/xtrx

[2] Project Zero, "Over the air: Exploiting broadcom's wi-fi stack," 2017. [Online]. Available: https://www.crowdsupply.com/fairwaves/xtrx

[3] E. Schmitt, "Pentagon tests lasers and nets to combat a vexing foe: ISIS drones," *New York Times*, 2017. [Online]. Available: https://www.nytimes.com/2017/09/23/world/middleeast/isis-drones-pentagon-experiments.html

[4] DroneShield, "ISIS dropping grenades from drones," 2019. [Online]. Available: https://www.droneshield.com/isis-dropping-grenades-from-drones

[5] FCC, "Jammer enforcement," 2012. [Online]. Available: http://www.fcc.gov/encyclopedia/jammer-enforcement,http://transition.fcc.gov/eb/News_Releases/DOC-304575A1.html

[6] FCC, "FCC enforcement bureau rolls out new jammer tip line: 1-855-55nojam," 2012. [Online]. Available: http://transition.fcc.gov/Daily_Releases/Daily_Business/2012/db1015/DOC-316795A1.pdf

[7] FCC, "FCC fines jammers," 2013. [Online]. Available: ftp://ftp.fcc.gov/pub/Daily_Releases/Daily_Business/2013/db0409/FCC-13-47A1.txt,http://transition.fcc.gov/eb/Orders/2013/FCC-13-106A1.html

[8] FCC, "FCC enforcement bureau steps up education and enforcement efforts against cellphone and gps jamming," 2013. [Online]. Available: https://www.fcc.gov/document/fcc-enforcement-bureau-steps-education-and-enforcement-efforts-against

[9] FAA, "Faa drone zone," https://faadronezone.faa.gov/.

[10] DHS, "Countering Unmanned Aircraft Systems Fact Sheet," 2019. [Online]. Available: https://www.dhs.gov/publication/st-countering-unmanned-aircraft-systems-fact-sheet

[11] GSMA, "Spectrum Sharing: GSMA Public Policy Position," July 2019. [Online]. Available: https://www.gsma.com/spectrum/wp-content/uploads/2019/09/Spectrum-Sharing-PPP.pdf

[12] L. Bodusseau, "Spectrum sharing has potential – but needs careful planning," 2018. [Online]. Available: https://www.gsma.com/spectrum/spectrum-sharing-has-potential-but-needs-careful-planning/

[13] DARPA, "Spectrum Collaboration Challenge (SC2)," https://www.darpa.mil/news-events/2018-12-19, https://www.darpa.mil/news-events/2019-09-10.

[14] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[15] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[16] A. Baset, C. Becker, K. Derr, S. Ramirez, S. Kasera, and A. Bhaskara, "Towards wireless environment cognizance through incremental learning," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2019, pp. 256–264.

[17] H. Franco, C. Cobo-Kroenke, S. Welch, and M. Graciarena, "Wideband spectral monitoring using deep learning," in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, 2020, pp. 19–24.

[18] M. Schmidt, D. Block, and U. Meier, "Wireless interference identification with convolutional neural networks," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017.

[19] N. Bitar, S. Muhammad, and H. H. Refai, "Wireless technology identification using deep convolutional neural networks," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017.

[20] T. O'Shea, T. Roy, and T. C. Clancy, "Learning robust general radio signal detection using computer vision methods," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017.

[21] E. Lo and J. Kohl, "Internet of things (iot) discovery using deep neural networks," in *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[22] E. Fonseca, J. F. Santos, F. Paisana, and L. A. DaSilva, "Radio access technology characterisation through object detection," *Computer Communications*, vol. 168, pp. 12–19, 2021.

[23] K. S. V. Prasad, K. B. D'souza, and V. K. Bhargava, "A downscaled faster-rcnn framework for signal detection and time-frequency localization in wideband rf systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4847–4862, 2020.

[24] A. Vagollari, V. Schram, W. Wicke, M. Hirschbeck, and W. Gerstacker, "Joint detection and classification of rf signals using deep learning," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 2021, pp. 1–7.

[25] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks 2016*, 2016.

[26] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15 713–15 722, 2018.

[27] DJI Developer Technologies, "Airlink - DJI Mobile SDK Documentation: Lightbridge," 2017. [Online]. Available: https://developer.dji.com/mobile-sdk/documentation/introduction/component-guide-airlink.html#lightbridge

[28] Samson Technologies, "XPD Series," 2019. [Online]. Available: http://www.samsontech.com/samson/products/wireless-systems/xpd-series/

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *arXiv:1409.1556*, 2014.

[31] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," in *arXiv:2004.10934*, 2020.

[32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, 2015.

[34] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[35] AlexeyAB, "darknet," https://github.com/AlexeyAB/darknet.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[37] H. N. Nguyen, M. Vomvas, T. Vo-Huu, and G. Noubir, "Wideband, real-time spectro-temporal rf identification," in *Proceedings of the 19th ACM International Symposium on Mobility Management and Wireless Access*, 2021, pp. 77–86.

[38] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen *et al.*, "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2018, pp. 112–117.

[39] CISCO Meraki, "SNR and Wireless Signal Strength ," 2018. [Online]. Available: https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/Signal-to-Noise_Ratio_(SNR)_and_Wireless_Signal_Strength

[40] "Gnuradio," 2021. [Online]. Available: https://www.gnuradio.org

[41] O. A. Dobre, Y. Bar-Ness, and Wei Su, "Higher-order cyclic cumulants for high order modulation classification," in *IEEE Military Communications Conference, 2003. MILCOM 2003.*, 2003.

[42] S. S. Soliman and S.-Z. Hsue, "Signal classification using statistical moments," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 908–916, 1992.

[43] P. D. Sutton, K. E. Nolan, and L. E. Doyle, "Cyclostationary signatures in practical cognitive radio applications," *IEEE Journal on selected areas in Communications*, vol. 26, no. 1, pp. 13–24, 2008.

[44] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, 2019.

[45] S. Sarkar, M. Buddhikot, A. Baset, and S. K. Kasera, "Deepradar: a deep-learning-based environmental sensing capability sensor design for cbrs," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 56–68.

[46] H. N. Nguyen, T. D. Vo-Huu, T. Vo-Huu, and G. Noubir, "Towards adversarial and unintentional collisions detection using deep learning," in *Proceedings of the ACM Workshop on Wireless Security and Machine Learning, WiseML@WiSec 2019, Miami, Florida, USA, May 14, 2019*, 2019, pp. 22–24.

[47] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for rf device fingerprinting in cognitive communication

networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 160–167, 2018.

[48] A. Elmaghbub and B. Hamdaoui, "Lora device fingerprinting in the wild: Disclosing rf data-driven fingerprint sensitivity to deployment variability," *IEEE Access*, vol. 9, pp. 142 893–142 909, 2021.

[49] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "Deep learning for rf fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

[50] H. N. Nguyen and G. Noubir, "Universal beamforming: A deep rfml approach," in *Proceedings of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2022, pp. 165–172.

[51] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[52] DeepSig.ai, "Deepsig datasets for modulation recoginition," 2018, https://www.deepsig.ai/datasets.

[53] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

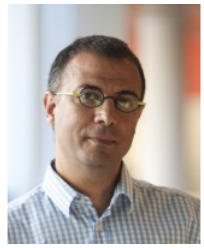
[54] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss dense object detection," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[55] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *arXiv:1911.09070*, 2019.

Hai N. Nguyen received the BS degree in the Talented Engineer Program in Electronics and Telecommunications Engineering at Hanoi University of Science and Technology, Hanoi, Viet Nam, in 2016. He is currently a PhD candidate in the Khoury College of Computer Sciences at Northeastern University, where he is investigating machine learning techniques for wireless communication systems. Prior to the PhD study, he had conducted research on Quality-of-Experience in multimedia networking. His research interests include machine learning, deep learning, and their applications in security and networked systems.

Marinos Vomvas received the Diploma in Electrical and Computer Engineering at National Technical University of Athens, Athens, Greece, in 2017. He is currently a PhD Candidate in Cybersecurity at Northeastern University in Boston. His research interests include wireless security, practical security and privacy, and multiparty computation (MPC).

Triet D. Vo-Huu holds a PhD in Computer Science from Northeastern University. His research interests focus on designing and building secure and robust wireless communication systems. Dr. Vo-Huu was an Assistant Research Professor at Northeastern University and co-led Northeastern University award winning team in the DARPA Spectrum Collaboration Challenge (SC2).



Guevara Noubir is a Professor at Northeastern University. His research interests span a range of problems on the theory and practice of privacy, security, and robustness in networked systems. Dr. Noubir holds a PhD in CS from EPFL and MS in CS (diplôme d'ingénieur) from ENSIMAG, France. He held research and visiting positions at CSEM SA, EPFL, Eurecom, MIT, and UNL. Professor Noubir received several research awards including the US National Science Foundation CAREER Award, Google Faculty Research Award on Privacy, multiple best paper awards, and led award winning teams in several competitions. He co-led (with Dr. Vo-Huu) Northeastern University's awards winning team in the DARPA Spectrum Collaboration Challenge (SC2).