IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 February 2025, accepted 16 March 2025, date of publication 24 March 2025, date of current version 31 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3554080

RESEARCH ARTICLE

Classification and Time-Frequency Localization of Arbitrary LPWAN Signals With Radial Deformable DETR

CHUN HO KONG AND HAIBO HU, (Senior Member, IEEE)
Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong
Corresponding author: Haibo Hu (haibo.hu@polyu.edu.hk)

ABSTRACT With the increasing adoption of Internet-of-Things (IoT) technologies, numerous devices utilizing protocols such as Sigfox and LoRa are now widely available inexpensively and operate in unlicensed ISM bands. However, challenges such as inventory management, unauthorized usage, and network performance must be addressed. Future adoption of emerging IoT protocols with various modulation schemes, bandwidth, and data rates can further complicate this. Therefore, it is important not only to classify but also to localize the frequency, bandwidth, and time of these LPWAN signals on the air for management, security, or band planning purposes. SOTA algorithms usually look through the whole received signal on the time domain or frequency domain only to perform classification tasks, without finding out the corresponding time-frequency location of the signal. This paper proposes to classify and localize time-frequency locations of LPWAN signals by an enhanced version of Deformable DEtection TRansformer (Deformable DETR). We devise an attention radius suitable for processing Low Power Wide Area Network (LPWAN) Spectrogram traces extracted from Software Defined Radios (SDRs) IQ data with Short-Time Fourier Transform (STFT). Inspired by Large Language Models (LLMs), sequences of STFT vectors from SDR IQs can leverage attention mechanisms, and finding out LPWAN signals in spectrograms is similar to object detection tasks in computer vision. Our method eliminates the need for hand-crafting CNN layers or signal processing pipelines for different LPWAN protocols provided that sufficient training samples are available. Therefore, we build a fully annotated dataset for Lora and Sigfox in multiple frequencies, bandwidths, packet data, and time, as well as data augmentation techniques that serve both training and validation datasets for our modified Deformable DETR model. The experimental results demonstrate an average precision of over 89.5% for LoRa signals and over 79.8% when mixed with ultra-narrow-band signals.

INDEX TERMS Attention mechanisms, multiple signal classification, radio spectrum management, reconnaissance.

I. INTRODUCTION

With the recent development of the Internet of Things (IoT) technologies, an abundance of inexpensive IoT devices equipped with various Low-Power Wide-Area Network (LPWAN) wireless protocols were deployed in areas such as security, smart homes, smart cities, healthcare, infrastructures, and more [1], [2]. By 2030, projections suggest over 31 IoT devices per person, contributing to a global market valued at over $8 trillion U.S. dollars [2]. To stay connected, these IoT devices usually form networks with LPWAN protocols, for example, LoRa [3] and Sigfox [4]. These protocols often coexist in the unlicensed Industrial, Scientific, and Medical (ISM) bands, where regional restrictions apply. For example, the European Union (EU) imposes that only 1% duty cycle of on-air-time is allowed for all LPWAN protocols on some frequency bands and transmission power [5]. As such, spectrum planning and management [2] are crucial to cater to the increasing number of IoT devices that are squeezing into the ISM band

The associate editor coordinating the review of this manuscript and approving it for publication was Chengpeng Hao.

and to increase the underlying Signal-to-Interference-plus-Noise Ratio (SINR) for better signal reception and device availability. In addition, mitigating privacy issues [6], [7] [8] and security threats [9] such as counterfeit and rogue [10] IoT Devices, also relies on the intelligence of spectrum status. In this paper, we propose transformer-based classification and time-frequency localization of the underlying LPWAN technologies in any unknown premises.

This work is empowered by Software Defined Radios (SDRs) [11], a relatively cheap radio frontend in small form-factor hardware, while most of the signal processing blocks are conducted on PC software, providing extreme flexibility. The application of SDRs to promote the effective usage of the radio spectrum is also known as Cognitive Radio (CR) [12]. The most crucial advantage of CR is reconfigurability, where the radio itself can be reconfigured to adapt to variations in the development of new wireless standards, incorporating new services and applications as they arise. With the aid of SDRs, we can reconfigure and receive various LPWAN Radio Frequency (RF) signals at different frequencies, channels, bandwidth, protocol, and modulation schemes, with minimal efforts and costs by scanning across these ISM bands.

Once SDRs form the basis for accessing the underlying LPWAN technologies in the spectrum, we adopt a modified version of Deformable DEtection TRansformer (Deformable DETR) [13] to process LPWAN spectrogram traces obtained from SDRs In-phase and quadrature (IQ) data. This allows us to perform LPWAN technology classification while obtaining the respective frequency, bandwidth, and time locations in the spectrogram traces. In other words, we effectively transfer object detection in machine vision to LPWAN protocol classification. More specifically, to detect the presence of RF signals, a received signal strength indicator (RSSI) threshold (RF squelch) is defined to determine the start and the end of the RF transmission [14]. For multiple LPWAN technologies coexisting in the spectrum, it is often difficult to determine the optimal universal threshold and the problem could be further complicated when there are multiple transmitters with different protocols, power levels, modulation schemes, and bandwidths. In the case of SDRs, as the number of protocols and their parameters are unknown, we often received multiple LPWAN protocols at once with overlapping packets across the spectrum. In recent research, Machine Learning (ML) approaches with Convolutional Neural Networks (CNNs) which were originally designed for image classification domain [15] can be employed as a means to perform modulation classification [16], technology classification [17], or source identification [18] for these LPWAN protocols, without tuning on various threshold parameters. However, most of these works **can only determine the presence of signals, but not the exact time and frequency of the packets**. Some even cannot work on Sub-GHz LPWAN signals due to their long transmission time and narrow bandwidth.

To summarize, our work has two main contributions. First, to analyze spectrum status in any premises, we propose transformer-based (Deformable DETR with Multiscale Deformable Radial Attention) classification on various LPWAN protocol signals from the spectrogram traces of SDR IQ data. Second, to validate our method as well as any LPWAN method in the future, we build a fully annotated LPWAN dataset of LoRA and Sigfox for both training and validation in LPWAN research, together with its generation pipeline and data augmentation.

The remainder of this paper is organized as below. Section II presents related work in the field of wireless technology detection, classification, and their applications with ML. In Section III, the method of generating, augmenting, and processing the annotated LPWAN dataset is outlined. Section IV presents the Deformable DETR model with our customization for spectrogram traces from SDR IQ data. Section V shows the empirical evaluation results against various parameters.

II. RELATED WORK

To perform technology classification or fingerprinting of LPWAN or generic RF signals, various methodologies can be employed. These methodologies can be broadly categorized based on whether they involve using machine learning techniques. Both categories will be outlined in this section.

For non-ML approaches, some work [19], [20] focused on transient-based identification on the frequency domain, which is based on the signal amplitude or power. These approaches utilize statistical analysis such as standard deviation, variance, skewness, and kurtosis on the transient portion of the signal before the preamble. Some work directly works [21], [22] [23] on the preamble, which is highly different across various RF protocols and modulations, or relying on modulation specifics like the inter-carrier spacing of orthogonal frequency-division multiplexing (OFDM) in different wireless standards. Reference [24] This implies these algorithms require pre-existence knowledge of the underlying signals. Moreover, some of these works are dependent on specialized hardware for data acquisition like oscilloscopes with high sample rates or dedicated spectrum analyzers, further increasing the cost of data collection and reducing the flexibility of the system. Thus, we would like to present an approach that can work with less prior knowledge of the LPWAN protocols.

For ML-based approaches, most of the work is based on CNNs, which is prevalent in image classification tasks. Some focused on modulation classification [25], [26] instead. For technology classification, in [27], the authors proposed a framework that allows compressing STFT traces into smaller sizes while being able to keep global signal features and augmenting small object highlights onto the compressed spectrogram traces and fed into a CNN network. However, they evaluated relatively high-bandwidth signals like Zigbee,

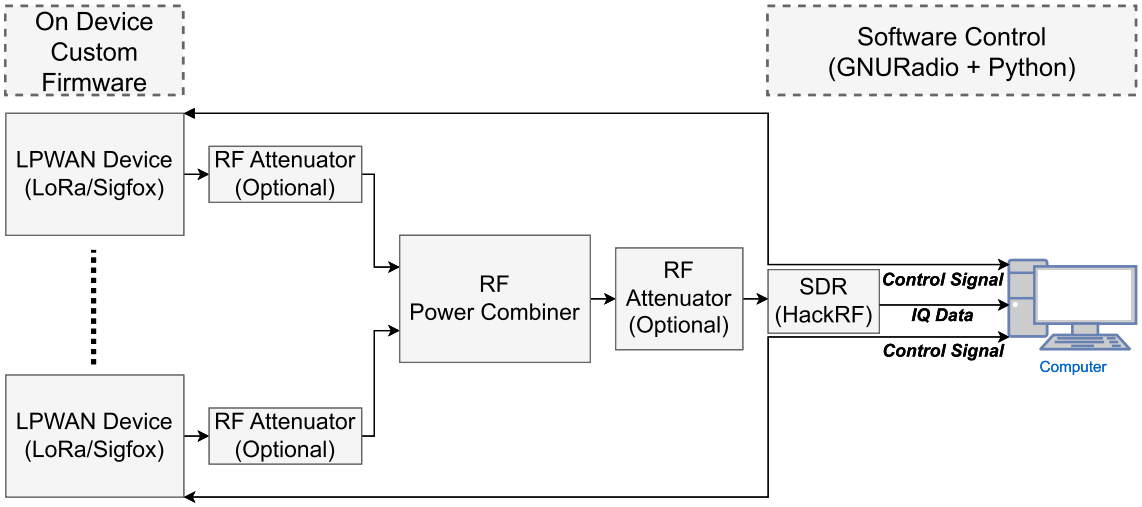

FIGURE 1. IQ dataset generation architecture.

TABLE 1. Comparison between this work and other state-of-the-art works.

Work	Time-Frequency Localization	Identifies	Contains real-world-liked overlapping signals	Full dataset generation pipeline with COTS SDRs
This work	Yes	LPWAN Protocols	Yes	Yes
[25]	No	Modulations	No	No
[17]	No	LPWAN Protocols	No	Partial
[19]	No	Individual Devices	No	No
[28]	No	LoRa Spreading Factors	No	No
[27]	No	2.4GHz Protocols	Yes	No
[26]	Yes	Modulations	No	No
[29]	Yes	4G/5G	Yes	No
[30]	Yes	Modulations	No	No

Wi-Fi, and Bluetooth on 2.4GHz bands, which marked the difference with our work. In [28], this work aims to utilize a CNN-based classifier to identify LoRa signals, with their protocol parameters like spreading factors (SFs) to determine if there are inter-SF interferences, therefore it is LoRa specific. The most promising work lies in [17], where the authors proposed a spectrum manager framework with two custom-designed CNN networks to handle IQ and FFT data aiming to classify narrow-band LPWAN technologies such as Sigfox, LoRA, and IEEE 802.15.4g. However, some of these works did not use SDRs for real-world evaluation and they did not allow their algorithm to output the exact time-frequency location of the LPWAN signals especially when multiple LPWAN signals occur, with possible overlapping positions in the incoming IQ data.

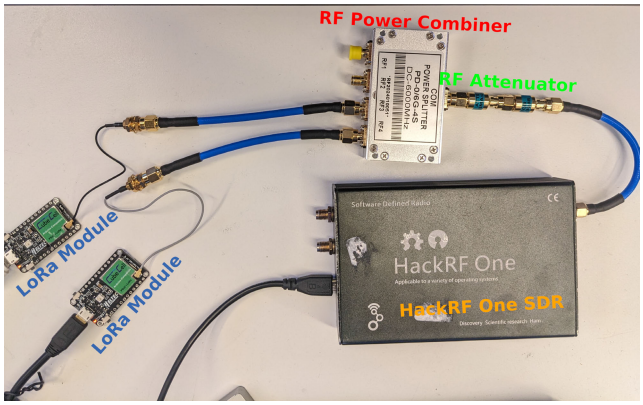

FIGURE 2. Actual setup of LoRa IQ dataset generation.

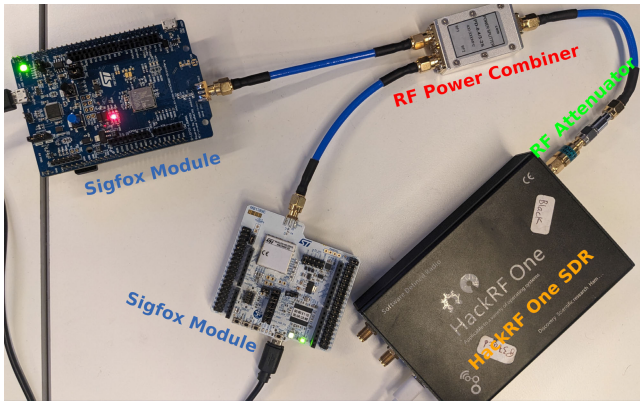

FIGURE 3. Actual setup of Sigfox IQ dataset generation.

To the best of our understanding, LPWAN technology classifications, especially in Sub-GHz ISM bands using SDRs and Transformer models have not been investigated

so far. This work will attempt to investigate this with our modified Deformable DETR model that is suitable for processing LPWAN Spectrogram traces extracted from SDRs IQ data with STFT to perform classification and time-frequency localization of LPWAN signals.

Additionally, Table 1 outlines the differences between our work and other state-of-the-art approaches that utilize Machine Learning methods.

III. LPWAN IQ DATASET GENERATION AND PROCESSING

To generate a custom LPWAN dataset for this work, we have employed two HackRF SDRs, LoRa Modules, and Sigfox Modules for data collection with proper dataset annotations. Fig. 1, Fig. 2 and Fig. 3 outline our IQ dataset generation architecture and our actual setup for LoRa and Sigfox, respectively.

A. DATASET GENERATION WITH LPWAN DEVICES AND SDRs

All of our LPWAN devices are initially preloaded with custom firmware, which provides flexibility for controlling the LPWAN radio modem and various parameters, including transmission power and protocol parameters such as LoRa Spreading Factors (SF_{lora}), preamble length ($PremLen_{lora}$), data length ($PktLen_{lora}$) and the actual data bytes per packets with our PC. Before starting the data collection process, the output power of the LPWAN devices is measured initially with an external power analyzer, and then with the SDR itself, to ensure consistency among the annotated data. RF attenuators are placed on both ends of the RF Power Combiner ports to achieve calibration purposes and to ascertain the output power per LPWAN devices will not overload the RF frontend of the SDRs, or damage the SDRs due to the direct connection of high-power RF path.

To generate LPWAN annotated datasets, software written in GNURadio [31] Blocks and Python on the computer starts recording IQ data with the SDR tuned to a center frequency $f_{c_{\text{sdr}}}$ that is wholly covering the transmission frequency and bandwidth of the LPWAN device and saving them to a file on the computer while instructing individual LPWAN devices to transmit on a random frequency $f_{c_{\text{dev}}}$, with random data bytes, power level and protocol parameters (if applicable). IQ samples from an SDR device can be represented as:

$$x[n] = \text{Re}(x[n]) + j\,\text{Im}(x[n]) = I[n] + jQ[n] \tag{1}$$

where, n is the IQ sample index and $I[n]$ and $Q[n]$ represent the real and imaginary parts of the signal, respectively.

Before the start and after the end of the transmission, relevant sample locations (n) in the IQ file will be marked in an annotation file, along with the protocol parameters, power levels, and the SDR acquisition parameters, including $f_{c_{\text{sdr}}}$ and the sample rate (SR) of the resulting IQ data. In the current generated dataset, $SR = 2Msps$, and the value of $f_{c_{\text{sdr}}}$ are $433.000MHz$ and $920.800MHz$ for LoRa and Sigfox datasets respectively. It is worth noting that the differences of $f_{c_{\text{sdr}}}$ have minimal impact on the resulting IQ dataset, as all of these IQ files are baseband signals.

After an LPWAN packet transmission is completed, we sleep the modem of an individual LPWAN device for a random amount of time before starting the process over again. We deliberately do not avoid overlapping the signals in the frequency domain from multiple LPWAN devices connecting on the same RF Power Combiner, since we are providing annotations of the signals per IQ files, outlining the time-frequency location of each LPWAN packet. As datasets for training and validation of ML models, this could help the model to adapt to overlapping of signals which is prevalent in real-world environments with overlapping LPWAN signals in ISM bands.

As a result, we have generated 50 packets per IQ file and signal class. With a total of 40000 packets across both LoRa and Sigfox classes, we have 400 IQ files per signal class. Each signal class bears LPWAN packets of randomized SNRs (SNR) of $SNR \in \{0, 3, 6\}dB$. With Sigfox, packet length of 12 bytes of random data and bandwidth (BW_{Sigfox}) of $BW_{Sigfox} = 100Hz$ are applied due to protocol specifications. For Lora, we have $8 \leq PktLen_{lora} \leq 48$ bytes and random data bytes. Additionally, we also varied LoRa protocol-specific parameters: $0 \leq PremLen_{lora} \leq 24$ symbols; $6 \leq SF_{lora} \leq 12$; and $BW_{lora} \in \{125, 250, 500\}KHz$.

To further increase the available training and validation data for our proposed model, we will incorporate data augmentation techniques against our generated datasets. We specifically choose to add Additive White Gaussian Noise (AWGN) along with our generated dataset to increase the robustness of the model and provide more variety of SINR combinations that are closer to real-world scenarios of coexisting LPWAN technologies with different signal strengths and channel conditions. After adding AWGN to our IQ datasets, these AWGN-added datasets, combined with the original datasets will be calculated for STFT, then converted to spectrogram, and undergo power-law normalization before feeding into the model for training and validation. The data augmentation and dataset processing pipeline are outlined in Fig. 4.

B. DATA AUGMENTATION WITH AWGN

To generate an AWGN with unity power, we can construct complex AWGN noise as:

$$w[n] = \frac{\mathcal{N}(0,1) + j\mathcal{N}(0,1)}{\sqrt{2}} \tag{2}$$

where, $\mathcal{N}(0, 1)$ denotes a standard normal distribution with $\mu = 0$ and $\sigma^2 = 1$

To obtain an AWGN noise-added signal, we can add the original IQ data with $w[n]$, scaled by the factor of α.

$$x_{noisy}[n] = x[n] + \sqrt{\alpha} \times w[n] \tag{3}$$

In our work, we have chosen $\alpha \in \{0.0001, 0.0002\}$ to mix with the original IQ datasets. For an original noise floor that is well below these chosen noise power levels, these

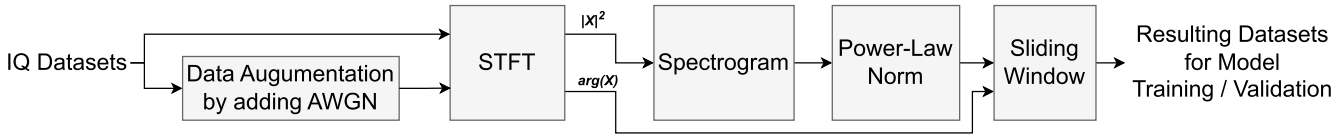

FIGURE 4. IQ dataset processing pipeline.

AWGNs will be the dominant noise in the IQ dataset, thus decreasing the SINR of each LPWAN transmission. After this data augmentation step, the total number of IQ files will be increased by the factor of 2, resulting in 1200 IQ files combined across all signal classes.

Selected visualization examples of the augmentation dataset versus the original datasets of one of the signal classes are presented in Fig. 5.

C. PROCESSING OF TRAINING AND VALIDATION DATASET

As we require the time-frequency location of the LPWAN signals, STFT will be taken on all of the IQ data. STFT operation on $x[n]$ to produce $X[k]$ can be expressed as:

$$X[k,m] = \sum_{n=kR}^{kR+N-1} x[n]W[n-kR]e^{-j\frac{2\pi mn}{N}} \tag{4}$$

where,

- k is the index of the time bin.
- m is the index of the frequency bin.
- $W[n-kR]$ is the window function
- N is the FFT size, and in our case $N = 2048$ to balance computation efficiency and frequency resolution.
- R is the hop size, and in our case $R = N = 2048$.

Here, we utilized blackman window in $W[n-KR]$, which is given by the following equation:

$$W[n] = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right) \tag{5}$$

where, n is the sample index, and N is the total number of samples in the window.

After obtaining the STFT vectors, to determine the power of the signal and obtaining a spectrogram, we can just square the magnitude of the STFT vector obtained in (4):

$$P[k,m] = |X[k,m]|^2 \tag{6}$$

Note that $\angle X[k,m]$ is unaltered. It will be used directly as a learning and inference feature for the model.

To increase the dynamic range of the spectrogram vector $P[k.m]$, we will apply power-law normalization, which can give the model a better "visual representation" for learning, because Deformable DETR was originally used in Image Detection Tasks, this would result in a more natural image look-alike spectrogram vector. This normalization step can be expressed as:

$$P_{normalized}[k,m] = \left(\frac{P[k,m] - \min(P)}{\max(P) - \min(P)}\right)^{\gamma} \tag{7}$$

where, γ is the power law exponent, chosen as $\gamma = 0.13$ to highlight both strong and weak LPWAN signals.

As the model presented in the later section requires running on a GPU for training and validation, it is impossible to feed the whole STFT vector without splitting the data. Hence, we have implemented a sliding window along the time-axis, allowing splitting up the Spectrogram traces and allowing more positional combinations of the LPWAN packets in the spectrum to train and validate the model. The sliding window SW can be represented as follows:

$$SW[i,k,m] = P_{normalized}[k,m] \times rect\left(\frac{k - iS_{SW}}{\min(L_{SW}, T - iS_{SW})}\right) \tag{8}$$

where,

- i is the index of the sliding window.
- S_{SW} is the step size of the window in terms of time bin indices. In our case, we have set this to 1 second equivalent.
- L_{SW} is the window length in terms of time bin indices. In our case, we have set this to 3 seconds equivalent.
- T is the total number of time bins in $P_{normalized}$ (i.e. $\max(k)+1$).
- $rect(n)$ is the rectangular sliding window function, as shown below in (9).

$$rect(n) = \begin{cases} 1, & \text{if } 0 \le n < 1 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

The number of sliding windows is given by:

$$Len_{SW} = \left\lceil \frac{T - L_{SW}}{S_{SW}} \right\rceil + 1 \tag{10}$$

The window length of 3 is chosen because it is desirable to have all of the LPWAN packets in the dataset to be at least in view in one of the sliding windows. Since the maximum on-air time of the LoRa packets in the datasets is 1.954 seconds, we chose 3 seconds to allow every LPWAN packet at least contained in a sliding window once, with a step size of 1 second to avoid discontinuity among LPWAN signals.

With 1200 IQ data files from the previous steps, we defined 80% of the data for training and 20% of the remaining data for validation of the model. As a result, we have generated a total of 32565 and 8196 Spectrogram sliding windows for training and validation respectively. Examples of spectrogram sliding windows are shown in Fig. 6.

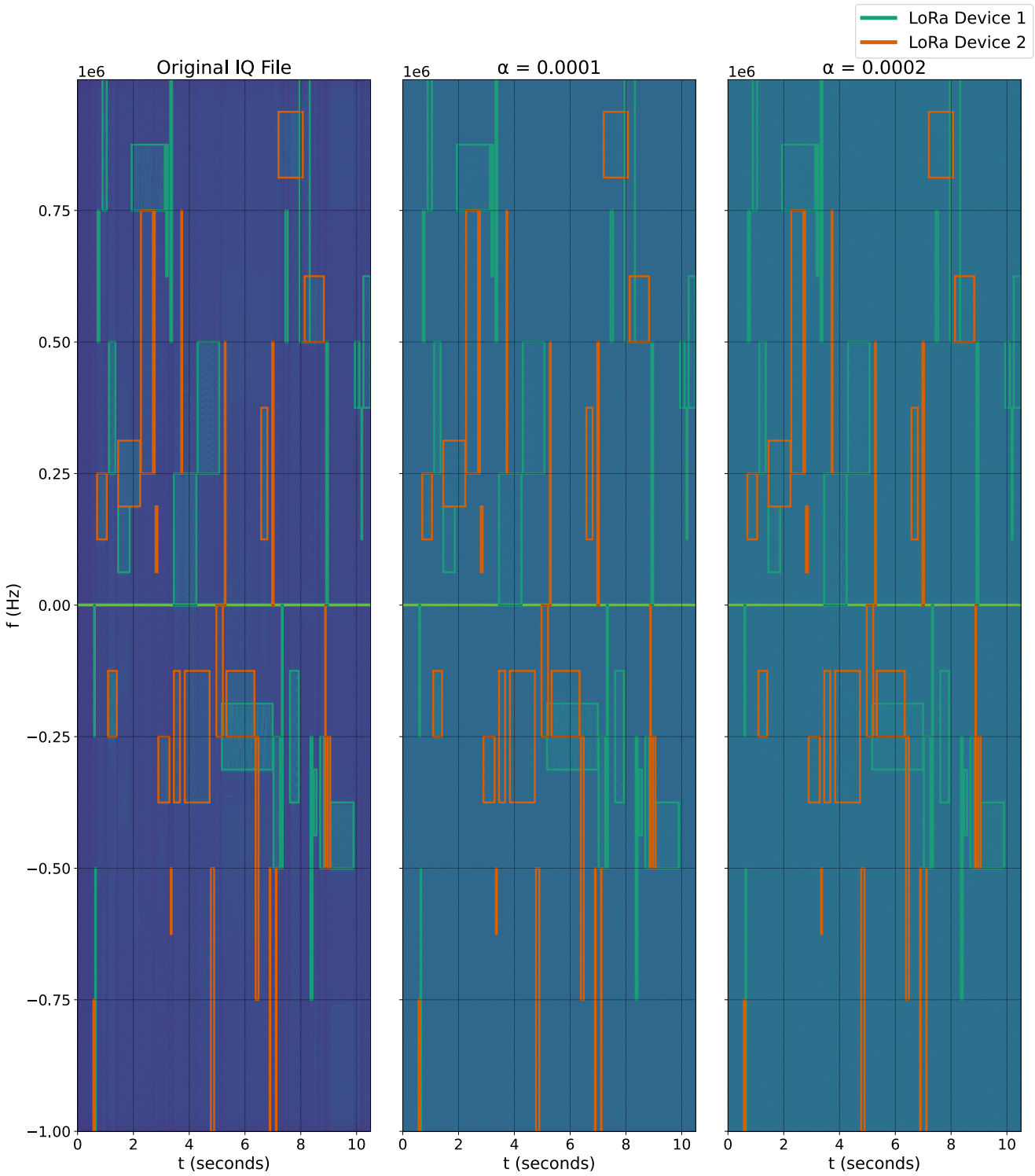

FIGURE 5. Original IQ and AWGN augmented datasets comparison.

IV. MODIFIED DEFORMABLE DETR MODEL DESIGN

DETR [32] applies the transformer model and attention mechanism [33] into the area of object detection. Traditionally, many hand-engineered components such as non-maximum suppression (NMS) to remove overlapping bounding boxes are designed in object detection systems.

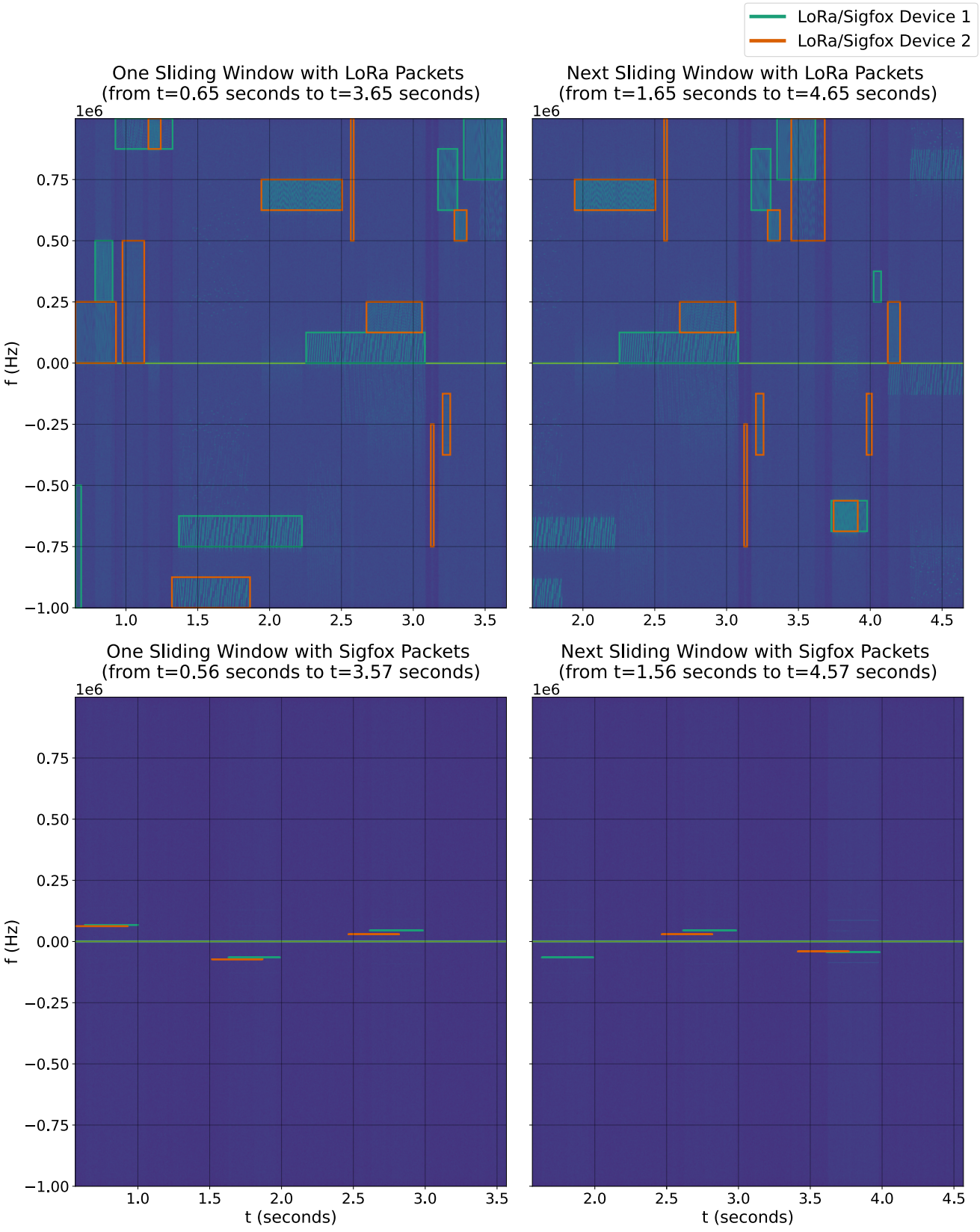

FIGURE 6. Example of spectrogram sliding windows.

DETR simplifies the detection pipeline by framing the problem as a direct set prediction task. It primarily consists of the following components:

- **Backbone**: This component extracts features from high-resolution images into lower-resolution activation maps.
- **Transformer Encoder-Decoder**: This component processes the flattened activation maps and learns global dependencies through a self-attention mechanism, given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{11}$$

  where, Q denotes the queries, K denotes the keys, V denotes the values, and d_k is the dimension of the key vectors, which helps provide stability during training. Positional encodings are also fed into the encoder because the transformer architecture is position invariant.
- **Object Queries**: These are learned embeddings that the decoder transforms. These queries attend to the output of the encoder through a cross-attention mechanism to generate class labels and bounding box coordinates through a Feed Forward Network (FFN).
- **Bipartite Matching Loss**: This component involves matching predicted objects to ground truth objects using a bipartite matching algorithm. The pairwise matching cost is given by:

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) \tag{12}$$

  where, $\hat{\sigma}$ is the permutation of N predictions that minimizes the sum of the matching costs, $\mathfrak{S}_N$ denotes the set of all permutations of N elements, y_i are the ground truth labels, and $\hat{y}_{\sigma(i)}$ are the predicted labels permuted by σ. To calculate the actual loss, we use $L_{Hungarian}$ which is given by:

$$L_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \Big[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbf{1}_{\{c_i \neq \emptyset\}} L_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) \Big] \tag{13}$$

  where, $\hat{p}_{\hat{\sigma}(i)}(c_i)$ is the probability of the predicted class c_i for the i-th ground truth object, $\mathbf{1}_{\{c_i \neq \emptyset\}}$ is the indicator function that is 1 if the ground truth class c_i is not an 'empty' class (meaning no object), and L_{box} is the loss for the bounding box coordinates.

In Deformable DETR [13], it further enhances the original DETR implementation by introducing deformable attention mechanisms, which was inspired by deformable convolution [34], allowing sparse spatial sampling of input features. The deformable attention mechanism attends to a selected group of sampling coordinates, acting as an initial filter to highlight key elements from the entire set of feature map pixels. The deformable attention mechanism can be modeled as:

$$DA(z_q, p_q, x) = \sum_{m=1}^{M} W_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \Big] \tag{14}$$

where,

- z_q is the query feature at position q. This is the feature vector.
- p_q is the reference point corresponding to the query position q.
- x is the input feature map.
- M is the number of attention heads.
- K is the number of sampling locations for each attention head.
- W_m is the output projection matrix for m'th attention head.
- W'_m is the input projection matrix for m'th attention head.
- A_{mqk} is the attention weight at m'th attention head and k'th sampling location, linear projected from z_q.
- Δp_{mqk} the sample offset at m'th attention head and k'th sampling location, linear projected from z_q.

This can also be extended to multi-scale feature maps, which are usually obtained from selected layers from the Backbone. The Multi-Scale Deformable Attention Module can be modeled as:

$$MSDA(z_q, p_q, \{x^l\}_{l=1}^{L}) = \sum_{m=1}^{M} W_m \Big[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot W'_m x^l(\phi_l(p_q + \Delta p_{mlqk})) \Big] \tag{15}$$

where,

- $\{x^l\}_{l=1}^{L}$ is a set of L input feature maps at different scales.
- L is the number of scales (feature levels).
- K is the number of sampling locations for each attention head at each level.
- A_{mlqk} is the attention weight at m'th attention head, l'th feature level, and k'th sampling location, linear projected from z_q.
- Δp_{mlqk} the sample offset at m'th attention head, l'th feature level and k'th sampling location, linear projected from z_q.
- $\phi_l(\cdot)$ is the transformation function that maps coordinates from the query feature map to the l'th level feature map.

Based on Deformable DETR, below we will present our modifications and enhancements that allow our new model to accept spectrogram sliding windows outlined in previous sections to perform classification and time-frequency localization of LPWAN signals.

A. RESNET50 BACKBONE MODIFICATIONS WITH TRANSFER LEARNING

In the original DETR [32] and Deformable DETR [13] implementations, the CNN backbone utilized the Resnet50 [35] model for feature extraction. Resnet50 is also used in some work as modulation pattern recognition [36], thus, it is possible to utilize the same model to perform feature extraction on our spectrogram traces, without implementing a new backbone from scratch.

However, unlike normal images that contain either 3 channels for RGB, or a single channel for grayscale images, our spectrogram data contain 2 channels instead, which are the power-law normalized power level and the STFT phase ($P_{normalized}$ and $\angle X[k,m]$). To expedite the training process and reuse pre-existing weights, we employed transfer learning techniques and modified the first Convolution layer ($Conv2d$) of Resnet50 from 3-channels input to 2-channels. After obtaining pre-trained weights of Resnet50 from PyTorch [37], we simply calculate the mean of the original 3-channels weights and apply them to the new 2-channels instead:

$$W_{new}[:,j,:,:] = \frac{1}{3}\sum_{i=0}^{2} W_{orig}[:,i,:,:], \quad \text{for } j \in \{0,1\} \tag{16}$$

where, i and j, are the indices of the original and new channels, respectively, and W_{orig} and W_{new} denote the original and the new learnable weights.

For the original first $Conv2d$ layer, *stride* is used to skip certain pixels (or time-frequency bins in our case) to speed up training and validation time. It is expressed by $Conv2d_{stride} = (h, w)$ where, h and w, denote the height and width for the convolution kernel to skip when convolving the input spectrogram sliding window. By default, Resnet50 uses $Conv2d_{stride} = (2,2)$. In our experiments, we varied this parameter to optimize feature extraction for narrowband signals for more frequency-domain details, especially for ultra-narrow-band signal classes such as Sigfox, as it occupies only 1 frequency bin before frequency expansion (see Section IV-C).

B. MULTI-SCALE DEFORMABLE ATTENTION WITH ATTENTION RADIUS

The idea behind having deformable attention is to allow the model to attend to certain sampling points that contain important features. However, by reverse thinking, we can also explicitly instruct the model *not* to attend to certain locations if we have prior knowledge of the nature of the application. In the case of LPWAN signals that are located in Sub-GHz bands, especially in ISM bands, there is an allocation bandwidth. To fully comply with relevant laws and standards, usually, the upper limit of the signal bandwidth should be well within the band allocation bandwidth. For instance, the maximum frequency of LoRa in the Sub-GHz band is 500KHz, while the ISM band in ITU Region 1 is 1.74MHz. [38] With this as prior knowledge, when the bandwidth of the SDR's received baseband signal is higher than the possible LPWAN signals, we can restrict the attention radius of the model for faster convergence of the model.

Recall in (14), we have A_{mqk} which is the attention weight, describing the importance of the sampling location k. We can define that for a certain attention radius (r), we set A_{mqk} to 0 when $|(\Delta p_{mqk})_y| < r$, which means the resulting sampling location $p_q + \Delta p_{mqk}$ is out of range (in the y-axis). This implies that the feature of the specific LPWAN signal should not exist at those points, and therefore, the model should not attend to them as they are irrelevant. Formally, to achieve this, we can modify DA in (14) as follows, we call this **Deformable Radial Attention**:

$$DRA(z_q, p_q, x, r) = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mqk} \cdot M_{mqk}(\Delta p_{mqk}, r) \cdot W'_m x(p_q + \Delta p_{mqk}) \right] \tag{17}$$

where,

$$M_{mqk}(\Delta p_{mqk}, r) = \begin{cases} 1 & \text{if } |(\Delta p_{mqk})_y| \le r \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where, $(\Delta p_{mqk})_y$, explicitly denotes the y-component (frequency axis) of Δp_{mqk}.

Note that renormalization is required if our mask has modified weights on the attention head. This is because the sum of the attention weights has to be 1. (i.e. $\sum_{k=1}^{K} A_{mqk} = 1$). Formally, we can do this by:

$$A'_{mqk} = \frac{A_{mqk} \cdot M_{mqk}(\Delta p_{mqk}, r)}{\sum_{k'=1}^{K} A_{mqk'} \cdot M_{mqk'}(\Delta p_{mqk'}, r)} \tag{19}$$

where, A'_{mqk}, is the renormalized attention weight

Additionally, this modified algorithm can also work on $MSDA$ in (15), allowing the application of attention radius across all feature levels, we call this **Multi-scale Deformable Radial Attention**:

$$MSDRA(z_q, p_q, \{x^l\}_{l=1}^{L}, r) = \sum_{m=1}^{M} W_m \left[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot M_{mlqk}(\Delta p_{mlqk}, r) \cdot W'_m x^l(\phi_l(p_q + \Delta p_{mlqk})) \right] \tag{20}$$

where,

$$M_{mlqk}(\Delta p_{mlqk}, r) = \begin{cases} 1 & \text{if } |(\Delta p_{mlqk})_y| \le r \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where, $(\Delta p_{mlqk})_y$ explicitly denotes the y-component (frequency axis) of Δp_{mlqk}.

C. EXPANDING BANDWIDTH ANNOTATIONS

Ultra-narrowband LPWAN signals such as Sigfox (100Hz in bandwidth) present unique challenges due to their limited spectral footprint by occupying only about 1 frequency bin in the Spectrogram. This lack of dynamic range on the frequency domain will negatively affect our model's ability to learn. This problem is more prevalent when Sigfox is using BPSK modulation, which implies each packet provides little to no frequency variation which in turn, we can only rely on phase information to determine Sigfox packets. We can indeed increase the frequency resolution by increasing the FFT size (N), or reduce the baseband bandwidth by reducing SDR's sample rate (SR), but in this way, we are sacrificing the ability to receive multiple LPWAN signals in-band for simultaneous time-frequency localization. Increasing N will also affect the memory usage of the GPU, in which, we can only decrease L_{SW} to compensate. A simpler solution is to increase the bandwidth of these ultra-narrow-band signals in our dataset annotations, which allows the model to not only learn from the signal itself but also the surrounding blank space in the spectrogram.

In the case of Sigfox, the 100Hz packets are being deployed in a 192KHz wide band and a 100Hz channel within this 192KHz is randomly selected for uplink. To the best of our knowledge, Sigfox did not release any information on whether there are guard bands or gaps between these 100Hz sub-channels. Even if there are none, the probability of one or more packets transmitting in the adjacent sub-channel at the same time is low: $P = 1/1920$. Moreover, for cases where overlap happens, the number of non-overlapping samples should be more than overlapping samples, which should produce minimal impact on the model, and the advantages should be outweighed.

Hence, two versions of our model are being trained with two levels of increase in Sigfox bandwidth annotation. $BW_{sigfox} \in \{5000, 7000\}Hz$, corresponds to additionally include 2 and 3 frequency bins for both top and bottom in the bounding boxes, respectively.

V. RESULT AND DISCUSSION

To evaluate the performance of our model, we have consumed the remaining 20% of our dataset as an evaluation dataset. It consists of 8196 spectrogram sliding windows containing 28503 and 28662 LoRa and Sigfox packets, respectively. It has been trained with NVIDIA V100 GPUs with 50 epochs.

Fig. 7 and Fig. 8 are examples of bounding box predictions from the model for LoRa and Sigfox respectively to demonstrate the ability of the model to perform time-frequency localization and classifications of LPWAN signals. To analyze the model's performance, similar to object detection tasks, we focus on its Average Precision (AP) [39] and Average Recall (AR) metrics, given by a certain IoU (Intersection over Union) threshold (0.5). We additionally introduce Y-AP and Y-AR metrics which are the AP and AR respectively when only considering the Y-axis (the frequency axis) IoU thresholds (Y-IoU). The replaced Y-IoU can be calculated by:

$$Y\text{-}IoU = \frac{|Y_p \cap Y_g|}{|Y_p \cup Y_g|} \tag{22}$$

where,

- Y_p is the predicted y-axis range (frequency axis)
- Y_g is the ground truth y-axis range (frequency axis)
- $|Y_p \cap Y_g|$ is the length of the intersection of the predicted and ground truth y-ranges
- $|Y_p \cup Y_g|$ is the length of the union of the predicted and ground truth y-ranges

In practical applications for technology classification and spectrum management, it is more important to find out which band (i.e. channel) the LPWAN signal occupies, rather than the exact location of the start of the packet in the time domain, since often the specific LPWAN device will transmit in a fixed subset of frequencies in the ISM bands. Introducing Y-AP and Y-AR provides a better picture of the capability of the model to extract the frequency locations of LPWAN signals. Unless specified otherwise, all AP, AR, Y-AP, and Y-AR are averaged over all classes.

A. TIME-FREQUENCY LOCALIZATION PERFORMANCE

To evaluate the time-frequency localization performance of the model, we utilized AP and AR metrics and compared them against different parameters for the model. More specifically, we adjusted the $Conv2d_{stride}$ parameter where $Conv2d_{stride} \in \{(1, 5), (2, 4), (2, 3)\}$, while keeping $BW_{sigfox} = \{7000\}Hz$ and MSDRA on. Overall performance in AP and AR are presented in Fig. 9, where a maximum AP of 77.62% can be obtained with $Conv2d_{stride} = (1, 5)$. It appears that decreasing the stride amount in the y-axis (frequency axis) can increase the AP of the model, the reason behind this can be seen in Fig. 10.

In Fig. 10, an interesting phenomenon can be observed, where, $Conv2d_{stride} = (2, 4)$, produces the best AP for LoRa signal instead. This discrepancy came from the Sigfox class, while we have a better AP by using $Conv2d_{stride} = (2, 4)$ for LoRa, using $Conv2d_{stride} = (1, 5)$ allows an even better increase in AP in the Sigfox class, which outweigh the AP decrease in LoRa class and increasing the overall class-averaged AP. This implies that LoRa (or relatively higher bandwidth LPWAN signal) may not benefit from the finer details of frequency axis details in the Backbone. On the contrary, ultra-narrow-band signals like Sigfox do require high-resolution details in the frequency domain to extract fine details for the model to converge. Interestingly, decreasing the x-axis (time axis) stride did not increase the AP and AR and worsened the metrics. This suggests that careful balance is required between the x and y axis of $Conv2d_{stride}$ and time-axis details are less important than frequency-axis details, especially a decrease in the stride of the y-axis will increase the amount of data in subsequent layers, for example, from $h/2$ to $h/1$ will increase the y-axis size of the output of $Conv2d$ layer by 2, which in turn increase

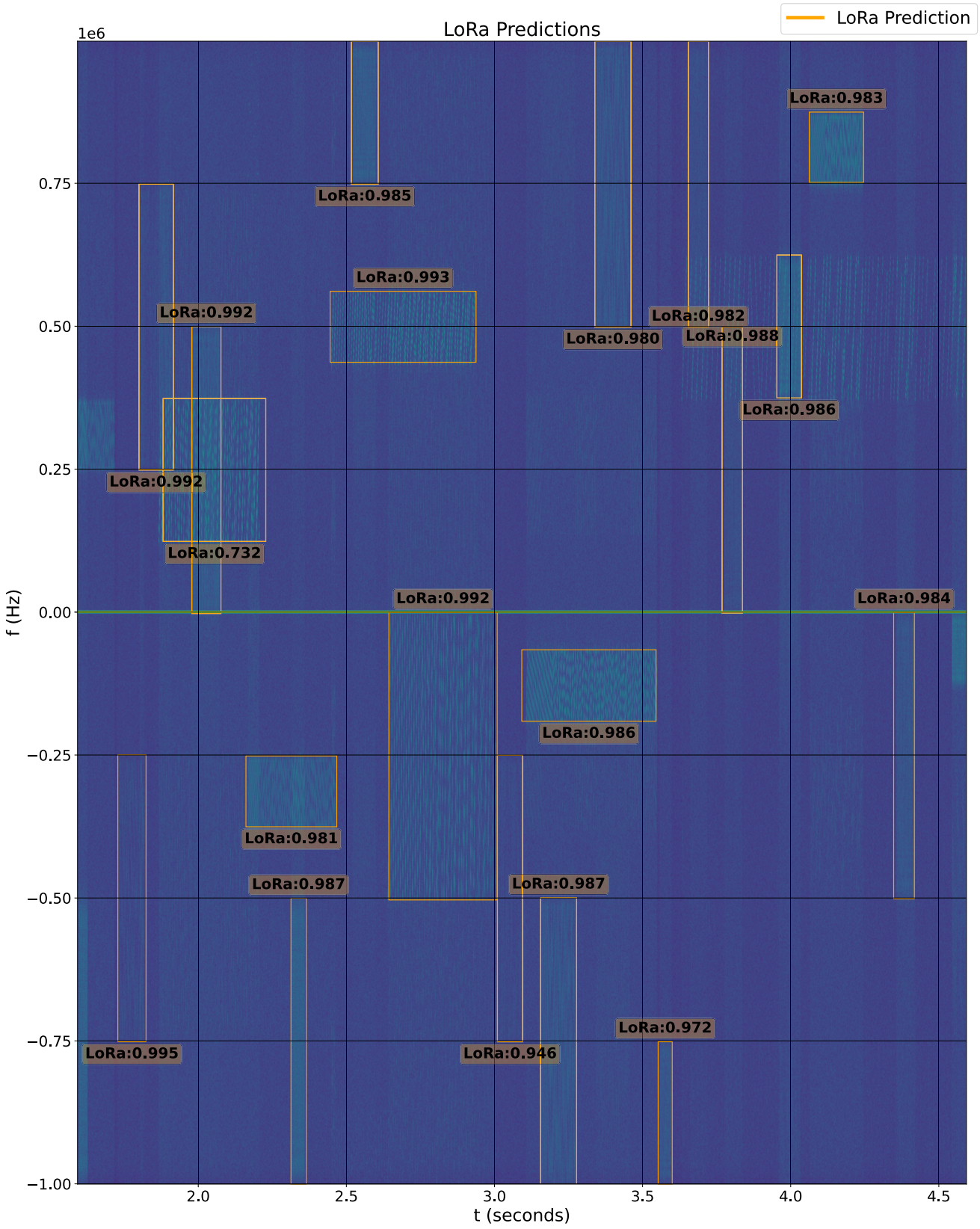

FIGURE 7. Example of LoRa prediction.

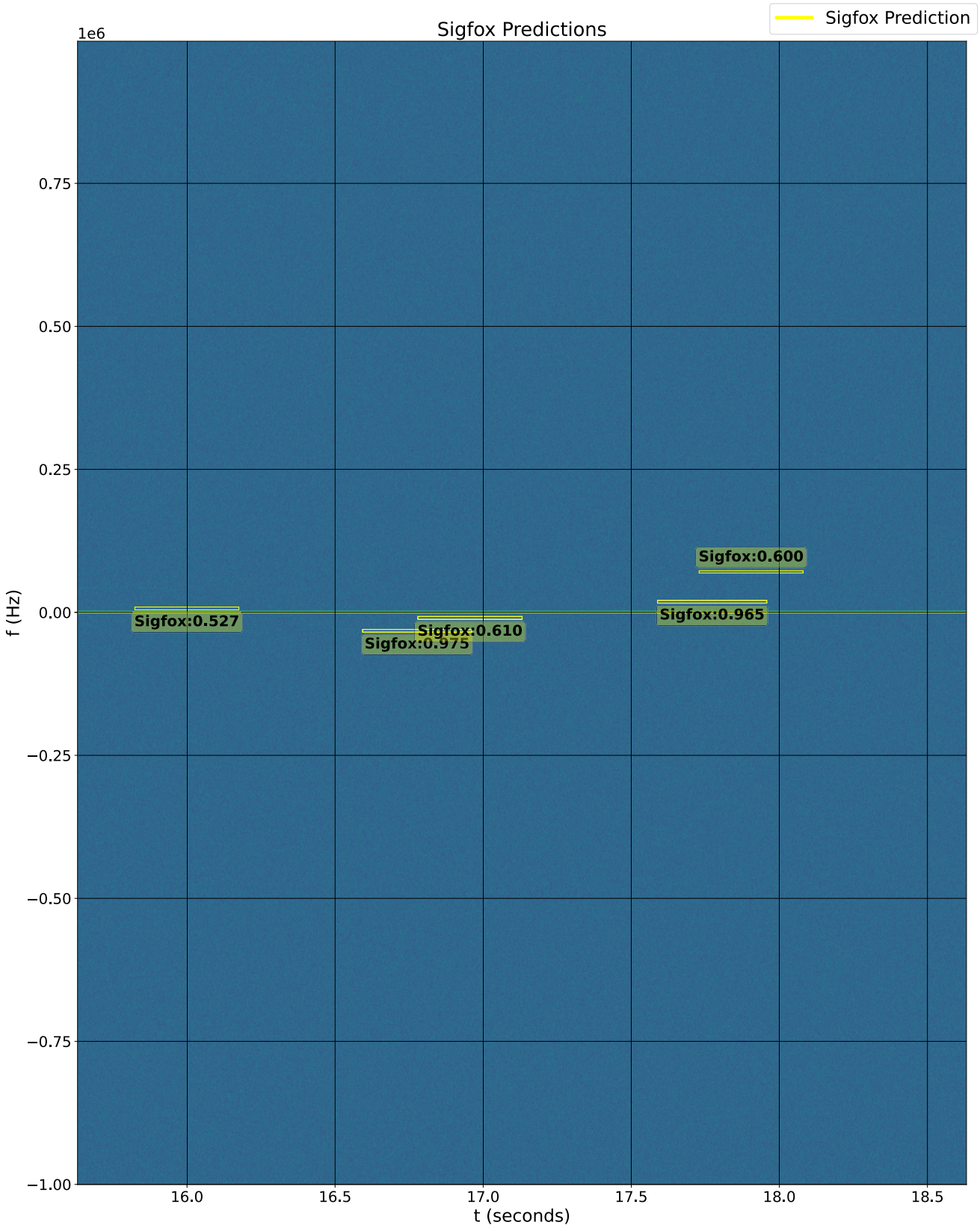

FIGURE 8. Example of Sigfox prediction.

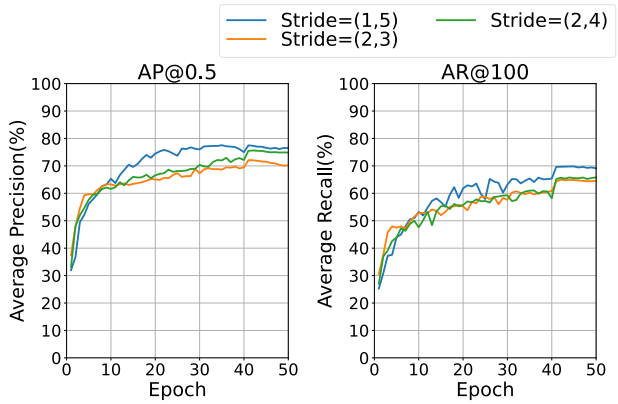

FIGURE 9. Time-frequency localization performance.

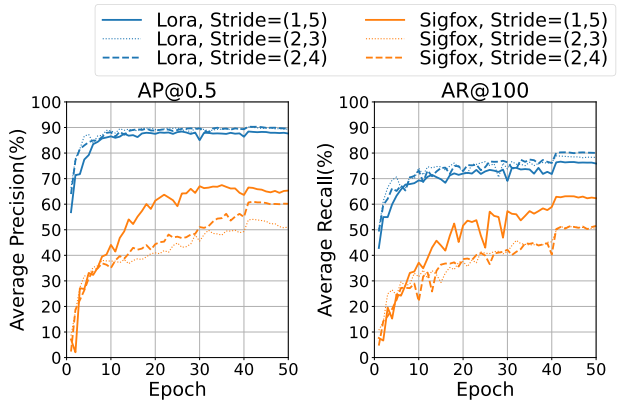

FIGURE 10. Time-frequency localization per-class performance.

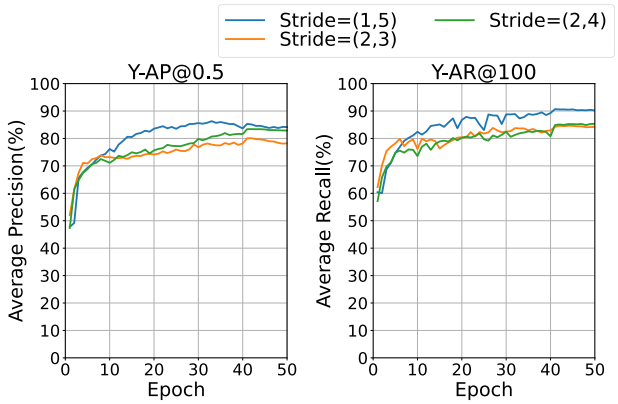

FIGURE 11. Frequency localization performance.

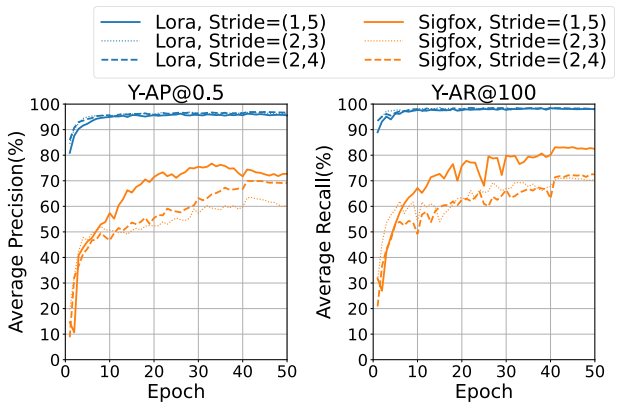

FIGURE 12. Frequency localization per-class performance.

the training time. At $Conv2d_{stride} = (1, 5)$, AP@0.5 for the LoRa signal class can still achieve nearly 90% AP, which we believe the merits for ultra-narrow-band outweigh the disadvantages for relatively higher bandwidth LPWAN signals.

B. FREQUENCY LOCALIZATION PERFORMANCE

The introduction of Y-IoU in (22) enables us to calculate Y-AP and Y-AR that characterize the performance of the model in the Y-axis (frequency-axis). This would give us a better picture of the frequency localization performance of the model. The relevant metrics are presented in Fig. 11 and Fig. 12.

Similar to the previous section of V-A, the best result is achieved by $Conv2d_{stride} = (1, 5)$. The metrics of Y-AP and Y-AR are higher than those of AP and AR (Y-AP > AP and Y-AR > AR) in the previous section. Y-AP@0.5 for LoRa and Sigfox reaches 95.94% and 74.23% respectively. The increase of Y-AP and Y-AR denotes that the model has greater ability at frequency localization than time-frequency localization. This provides the feasibility of application in spectrum management for this model.

C. PERFORMANCE UNDER NOISE

To further find out how noisy environments impact the model performances, we isolated out the AP values per class and per noise levels, as shown in Table 2.

TABLE 2. Comparison of AP values across different classes and noise levels.

Class	Metric	AP of both mixed dataset	AP of AWGN-free dataset	AP of AWGN-added dataset	AP Differences after AWGN-added
ALL	AP@0.5	77.6%	94.6%	70.7%	-23.9%
	Y-AP@0.5	85.1%	98.6%	80.0%	-18.6%
LoRa	AP@0.5	88.3%	98.9%	84.1%	-14.8%
	Y-AP@0.5	95.9%	99.1%	94.2%	-4.9%
Sigfox	AP@0.5	67.0%	90.3%	57.4%	-32.9%
	Y-AP@0.5	74.2%	98.1%	65.8%	-32.3%

All AWGN-free datasets yield AP values exceeding 90% and Y-AP values above 98%. After adding AWGN noise into the dataset, it is obvious that large-bandwidth signals like LoRa are less susceptible to AWGN noise due to the fact that there are still more frequency-axis features available

for the model to perform LPWAN signal classification and time-frequency localization. One way to tackle this is by increasing the number of STFT bins, which allows the model to learn more frequency-axis features from ultra-narrowband signals. Details of this are outlined in the next section.

D. CLASSIFICATION ONLY PERFORMANCE

To further extend the evaluation of the classification performance of the model, we presented the confusion matrix for the two classes in Fig. 13.

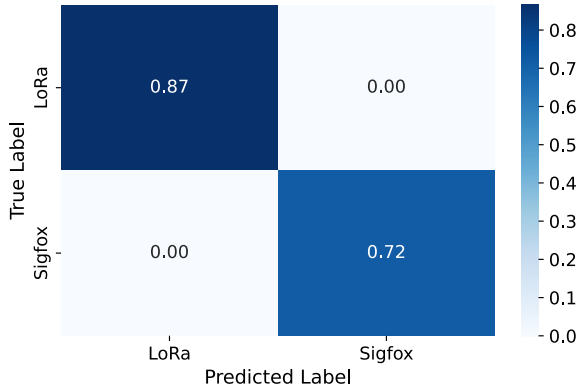

FIGURE 13. Confusion matrix of the model for classification.

Note that the reason for the sum of the column is not 1.0 owing to the underprediction of the model, where some of the ground truth labels were not detected. This issue is much more prominent in Sigfox as the lack of frequency-axis features in the backbone and the STFT bins is due to the ultra-narrow-band nature of the LPWAN packets. It is desirable if more features can be extracted by the backbone from the frequency axis, which could increase the performance of the model.

To improve the performance of the model to detect ultra-narrow-band signals like Sigfox, we could increase the amount of information available in the frequency-axis, by sacrificing some time-axis features. Originally, an STFT size of $N = 2048$ had been chosen to strike a balance between time and frequency resolutions among large bandwidth signals and narrow bandwidth signals, as well as computation time of performing the STFT operation itself, and favor to LoRa signals which is a much more common LPWAN protocol. When we set $N = 4096$ to favor more narrow band features, per-class AP of LoRa and Sigfox decreased from 89.5% to 86.9% (-2.6%) and increased from 66.9% to 72.6% (+5.7%), respectively. Per-class Y-AP for LoRa and Sigfox also slightly decreased from 95.9% to 95.2% (-0.7%) and significantly increased from 74.2% to 80.2% (+6.0%), respectively. **Overall AP reaches 79.8% in this case.** This results in an improved confusion matrix which favours Sigfox predictions outlined in Fig. 14.

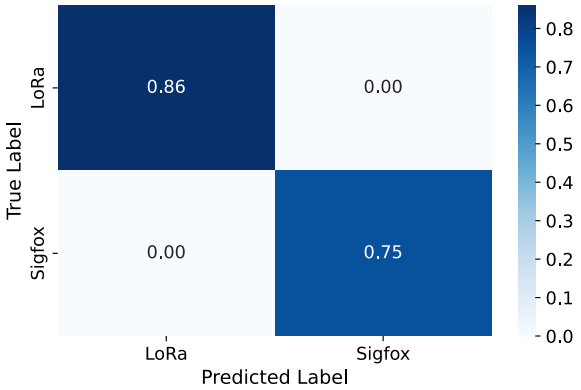

FIGURE 14. Confusion matrix of the model for classification with $N = 4096$.

E. ABLATION STUDY

1) MULTI-SCALE DEFORMABLE RADIAL ATTENTION (MSDRA)

The modification of MSDA in (15) to MSDRA in (20) by adding attention radius was supposed to expedite the model convergence when we have acquired prior knowledge of the target spectrum environment where the LPWAN signals are located. All of the conducted experiments were carried out with MSDRA for a 250KHz attention radius on the y-axis, because of the constraint of 500KHz maximum bandwidth of LPWAN signals in ISM bands. Here, we will try to use the original MSDA without any radial attention mechanism to study the impact of MSDRA on the convergence of the model.

As seen from Fig. 15, when MSDRA is not used, AP and Y-AP decreased by nearly 3% and 2.6% respectively. The model also converges faster with higher (Y-)AP as early as at epoch 20 when MSDRA is used. Thus, we conclude that MSDRA is essential and useful for processing LPWAN spectrogram data with the possession of prior spectrum knowledge.

2) EXPANDING BANDWIDTH ANNOTATIONS

In section IV-C, we have to expand the bandwidth of the Sigfox annotations to allow the model to learn about frequency features around the Sigfox signal (e.g. noise surrounded by the Sigfox signal). Note that when this modification is not added, the AP of the Sigfox class drops significantly to near 0. This is due to the way that IoU calculations work, where the model has to predict with 1-pixel accuracy, which results in a lack of dynamic range. If the prediction were off by 1 pixel in the y-axis, the IoU would result in 0, because there would be no overlapping at all. The reason is that the original Sigfox only occupies 1 bin, or more precisely about 0.1 bin because the bandwidth of Sigfox is 100Hz, and the width of each Spectrogram bin for $SR = 2MHz$ with STFT $N = 2048$ is about $976.5625Hz$. Thus, the lack of dynamic range coupled with the fact that

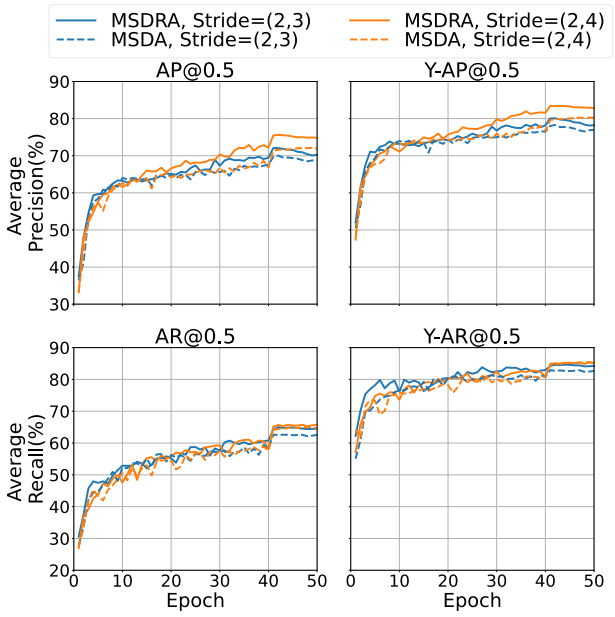

FIGURE 15. Impact of AP when removing MSDRA.

Sigfox uses BPSK with little frequency-axis variation results in poor prediction results for Sigfox.

Here, we try to use different Sigfox Bandwidth, as shown in Fig. 16.

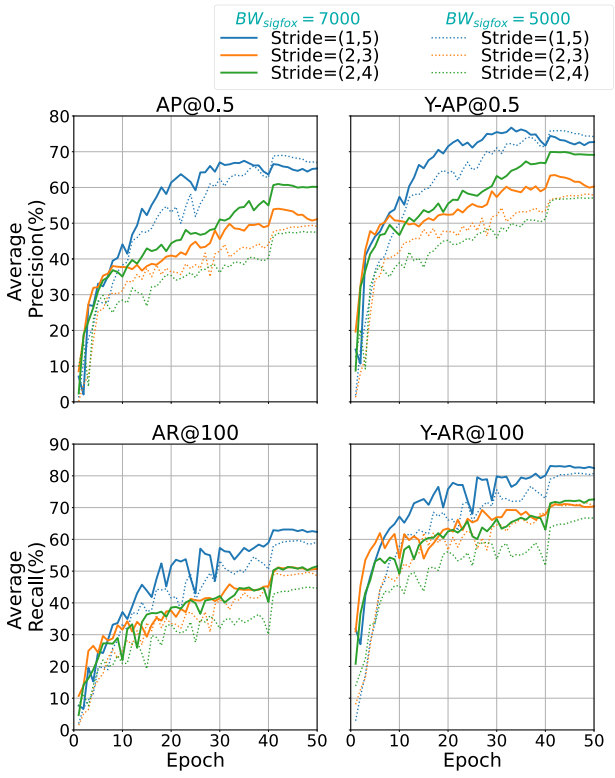

FIGURE 16. Impact of AP when varying BW_{sigfox}.

Improvements can be observed when we set $BW_{sigfox} = 7000$, but this only applies to $Conv2d_{stride} \in \{(2,4),(2,3)\}$. For $Conv2d_{stride} = (1,5)$, which is the best parameter so far, we should keep $BW_{sigfox} = 5000$. This means that when more details exist in the frequency axis (i.e. lower $Conv2d_{stride}$), it is possible to less artificially increase the annotated bandwidth of the ultra-narrow-band signals. To ensure the annotations are closer to reality, we can just use $BW_{sigfox} = 5000$, with finer details for $Conv2d_{stride}$, allowing the model to learn the extra features around the real Sigfox signal, increasing the dynamic range for (Y-)IoU calculations, while keeping the annotations closer to reality.

F. VISUALIZATION OF RADIAL ATTENTION

To better demonstrate the impact of the addition of radial constraints in (MS)DRA, Fig. 17 visualizes the non-zero attention-weighted sampling locations for each reference point in MSDA that contains at least one out-of-radius sampling location in a random spectrogram sliding window. We define "In-radius sampling points" as those lying within a specific radius ($\leq 250KHz$), and "Out-of-radius sampling points" as those lying outside this radius.

This visualization clearly shows that, without radial constraint, some reference points within a LoRa packet annotation attended to blank signal regions that are far away from actual in-packet signal features. Additionally, many reference points outside any annotated regions attended to sampling locations of pure noise, resulting in noisy reference points attending to noisy sampling locations. This behavior contributes minimally to the model's convergence and learning process.

Thus, by applying (MS)DRA, the model is constrained to focus only on meaningful regions within specified radius. This targeted attention mechanism facilitates faster convergence and improves the model's overall accuracy by preventing attention from being wasted on irrelevant or noisy areas.

G. COMPUTATIONAL COMPLEXITY

It is also important to evaluate the computational complexity of the model, especially if running at the edge with resource-constrained IoT devices is required. Note that the addition of (MS)DRA essentially applies a lightweight mask towards the original implementation of (MS)DA, which incurs limited additional asymptotic complexity to the model. Based on the original complexity of (MS)DA [13], the complexity of (MS)DRA can be given by:

$$\mathcal{O}(N_q C^2 + \min(HWC^2, N_q K C^2) + 5N_q KC + 3N_q CMK + N_q K) \tag{23}$$

where,

- N_q is the number of queries;
- HxW is the height and width of the input feature map, respectively;
- C is the embedding dimension;

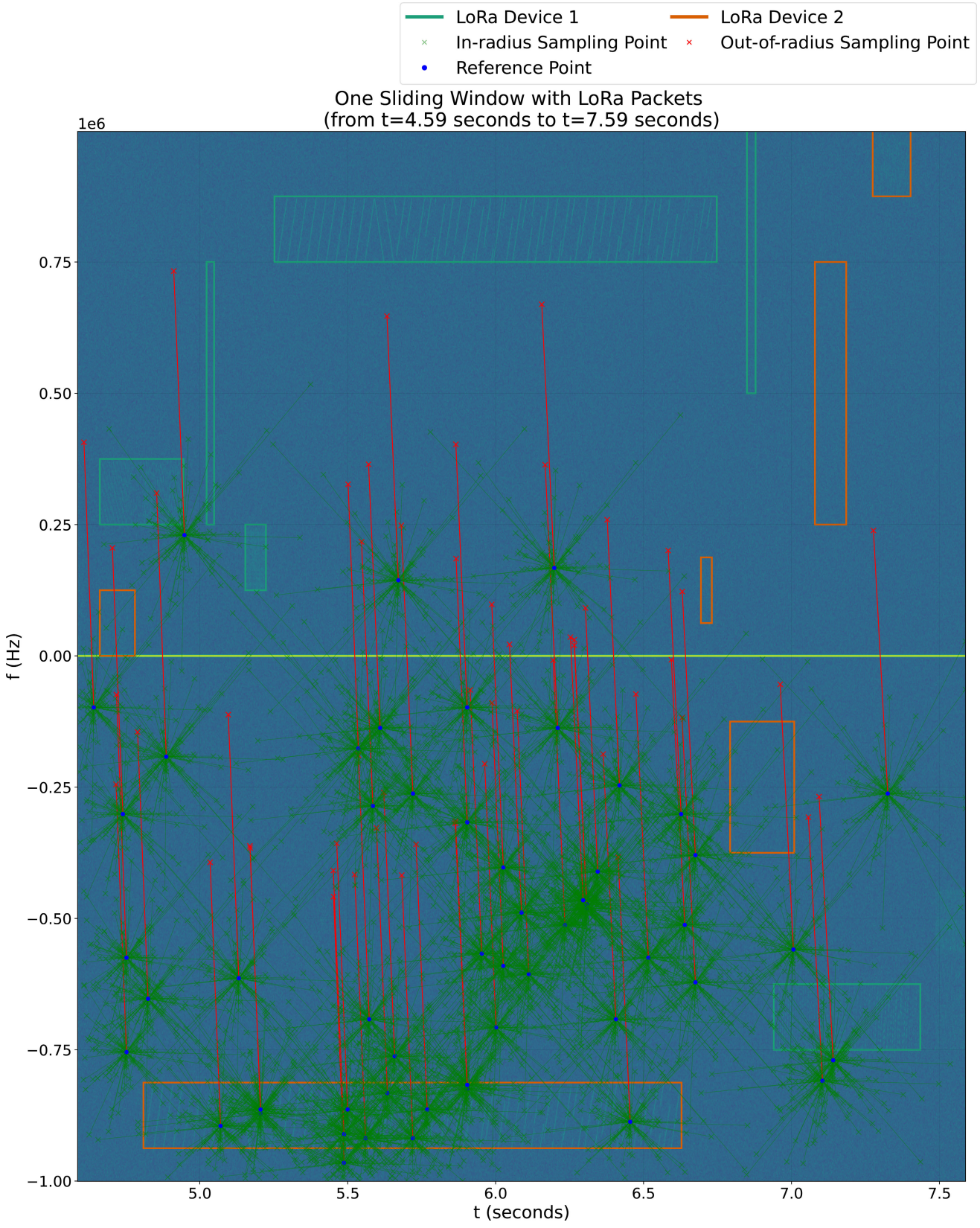

FIGURE 17. Visualization of attention sampling points.

- K is the number of sampling points per query per attention head;
- M is the number of attention heads.

The addition of N_qK term accounts for the extra normalization step introduced in (MS)DRA when an attention weight of a specific sampling point is modified by setting it to zero due to being out of radius. This step ensures that the remaining attention weights still sum to one after masking out out-of-radius points. However, this normalization step has a negligible impact on the overall complexity as terms involving H, W, N_q, K, and C remain dominant. On a separate note, decreasing $Conv2d_{stride}$ values in the Resnet50 backbone increases the frequency resolution of the feature map, which improves AP and Y-AP values, as demonstrated above. However, this comes at the cost of a larger input feature map (H and W), leading to a higher computational complexity.

In practice, MSDRA has tested on multi-generation old GPUs. With a single NVIDIA GTX1070, inferences of each sliding window of 3 seconds ($L_{SW} = 3$) with $Conv2d_{stride} = (2, 4)$ and $N = 2048$ require approximately 0.875 seconds of GPU runtime, demonstrating the feasibility of performing real-time inferences and highlighting the practicality of the proposed model. The computational bottleneck is more likely to arise from the encoder layers with global attention mechanisms in the original transformer model, which is inherently resource-intensive. If further decrease in computation complexity is required, it is possible to decrease the number of attention heads, scales, and sampling points to scale down the overall model, employ advanced feature extraction techniques that reduce the dimension of input features, or simply decrease the STFT resolution according to specific application needs. These approaches provide flexibility for adapting MSDRA and Transformer models to more resource-constrained environments, such as edge devices in IoT applications.

H. REAL-WORLD DATASETS

To demonstrate the robustness of the proposed model with real-world scenarios, we evaluated it using a real-world commercially available IoT device, a LoRa temperature sensor, transmitting in public ISM bands via actual antennas on both the transmitter and receiver sides. It is important to note that the signals transmitted by the commercial LoRa temperature sensor as shown in Fig. 18 were not included in the original training dataset, ensuring the model is inferring towards unseen real-world LPWAN signals. Furthermore, we confirmed that the model and manufacturer of the wireless modem Integrated Circuits (ICs) are different between the datasets: the LPWAN dataset was generated using the ASR6501 modem, while the commercial device uses the SX1268 modem. This ensures minimal correlation between signal features from the same family of ICs. The dataset generation process also followed the pipeline outlined in Section III, without any additional tuning or adjustments.

FIGURE 18. Commercial LoRa temperature sensor for real-world inferencing.

The LoRa packets transmitted by the temperature sensor were manually annotated after being received by a HackRF SDR over-the-air, with a physical separation of more than 20 meters within the same room without line-of-sight. The data collection spanned over 17 hours, resulting in 1,048 individual LoRa packets being received and annotated. Despite the model encountering entirely new data from a different device in real-world conditions, it achieved an **AP exceeding 90%**. This performance is comparable to the results obtained with the original LPWAN dataset, highlighting the model's robustness and generalization capabilities when applied to LPWAN signals not seen during training.

VI. CONCLUSION

In this paper, without hand-crafting specific models, we have proposed a customized Deformable DETR-based model with DRA and MSDRA to classify LPWAN technologies for time-frequency localization. The custom modifications transfer the model from image detection tasks to LPWAN signal detection, allowing the model to accept spectrogram traces obtained from STFT of SDR IQ data to perform classification and time-frequency localization. The IQ datasets captured by Software Defined Radios for this work contain two sub-GHz signal classes with narrow-band technologies. These annotated datasets have gone through data augmentation steps and are further processed to produce spectrogram sliding window traces that can be fed into our model for training and validation. In the experiments, our model localized

and classified LPWAN signals in the spectrogram traces, achieving an AP of 77.6% and 79.8% for $N = 2048$ and $N = 4096$, respectively at IoU@0.5 in the validation set. For LoRa and Sigfox, per-class AP can reach 89.5% and 66.9% respectively, and per-class Y-AP at 95.9% and 74.2% respectively. Additionally, we tested the model's robustness by training it with the custom LPWAN dataset generation pipeline and inferring on real-world commercially available IoT devices, ensuring the model is applicable in real-world scenarios. Although the modification of (MS)DRA spans from (MS)DA and incurs minimal additional computational complexity, it still heavily relies on the original Transformer model which could still require a global attention mechanism on the encoder layers. In future work, we plan to extend the annotated datasets to more real-world LPWAN technologies across ISM bands. We will also explore a new ML-based approach toward LPWAN technology classification with time-frequency localization. For example, to detect ultra-narrow-band LPWAN signals, we will modify the STFT processing pipeline with an adaptive number of STFT bins, to incorporate more fine signal details for feature extraction to the model. It is also worth investigating in non-transformer-based models which could further decrease the potential computational complexity from encoder layers. Last but not least, ultimately we plan to propose a low-cost federated LPWAN signal detection network on the edge.

REFERENCES

[1] S. Mahmood, "Review of Internet of Things in different sectors: Recent advances, technologies, and challenges," *J. Internet Things*, vol. 3, no. 1, pp. 19–26, 2021.

[2] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of Things market analysis forecasts, 2020–2030," in *Proc. 4th World Conf. Smart Trends Syst., Secur. Sustainability (WorldS4)*, Jul. 2020, pp. 449–453.

[3] (Jul. 2024). *LoRa Alliance*. Accessed: Jul. 24, 2024. [Online]. Available: https://lora-alliance.org/

[4] (Jul. 2024). *Sigfox 0G Technology*. Accessed: Jul. 24, 2024. [Online]. Available: https://www.sigfox.com/

[5] M. Saelens, J. Hoebeke, A. Shahid, and E. D. Poorter, "Impact of EU duty cycle and transmission power limitations for sub-GHz LPWAN SRDs: An overview and future challenges," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–32, Dec. 2019, doi: 10.1186/s13638-019-1502-5.

[6] R. Li, H. Hu, and Q. Ye, "RFTrack: Stealthy location inference and tracking attack on Wi-Fi devices," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 5925–5939, 2024.

[7] Z. Chen, H. Hu, and J. Yu, "Privacy-preserving large-scale location monitoring using Bluetooth low energy," in *Proc. 11th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, Dec. 2015, pp. 69–78.

[8] H. Zheng and H. Hu, "MISSILE: A system of mobile inertial sensor-based sensitive indoor location eavesdropping," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3137–3151, 2020.

[9] L. Tang and H. Hu, "OHEA: Secure data aggregation in wireless sensor networks against untrusted sensors," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1425–1434.

[10] S. Afzal, A. Faisal, I. Siddique, and M. Afzal, "Internet of Things (IoT) security: Issues, challenges and solutions," *Int. J. Sci. Eng. Res*, vol. 12, no. 6, p. 52, 2021.

[11] F. K. Jondral, "Software-defined radio—Basics and evolution to cognitive radio," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 3, pp. 1–9, Dec. 2005.

[12] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[13] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.

[14] N. Damak, C. Krall, and R. Storn, "Optimization of squelch parameters for efficient resource allocation in software defined radios," in *Proc. SDR-WInnComm-Europe 2013*, 2013, pp. 57–63.

[15] M. Hussain, J. J. Bird, and D. R. Faria, "A study on CNN transfer learning for image classification," in *Proc. 18th UK Workshop Comput. Intell.*, Nottingham, U.K. Cham, Switzerland: Springer, Aug. 2018, pp. 191–202.

[16] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. 17th Int. Conf.*, Aberdeen, U.K. Cham, Switzerland: Cham, Switzerland: Springer, Sep. 2016, pp. 213–226.

[17] A. Shahid, J. Fontaine, M. Camelo, J. Haxhibeqiri, M. Saelens, Z. Khan, I. Moerman, and E. D. Poorter, "A convolutional neural network approach for classification of LPWAN technologies: Sigfox, LoRa and IEEE 802.15.4G," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–8.

[18] Y. Jiang, L. Peng, A. Hu, S. Wang, Y. Huang, and L. Zhang, "Physical layer identification of LoRa devices using constellation trace figure," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–11, Dec. 2019.

[19] M. Köse, S. Tascioglu, and Z. Telatar, "RF fingerprinting of IoT devices based on transient energy spectrum," *IEEE Access*, vol. 7, pp. 18715–18726, 2019.

[20] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2009, pp. 25–36.

[21] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based RF fingerprinting to enhance wireless network security," *J. Commun. Netw.*, vol. 11, no. 6, pp. 544–555, Dec. 2009.

[22] W. C. Suski II, M. A. Temple, M. J. Mendenhall, and R. F. Mills, "Using spectral fingerprints to improve wireless network security," in *Proc. IEEE GLOBECOM Global Telecommun. Conf.*, Nov. 2008, pp. 1–5.

[23] M. D. Williams, S. A. Munns, M. A. Temple, and M. J. Mendenhall, "RF-DNA fingerprinting for airport Wimax communications security," in *Proc. 4th Int. Conf. Netw. Syst. Secur.*, Sep. 2010, pp. 32–39.

[24] A. Bouzegzi, P. Ciblat, and P. Jallon, "Maximum likelihood based methods for OFDM intercarrier spacing characterization," in *Proc. IEEE 19th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2008, pp. 1–5.

[25] Z. Chen, H. Cui, J. Xiang, K. Qiu, L. Huang, S. Zheng, S. Chen, Q. Xuan, and X. Yang, "SigNet: A novel deep learning framework for radio signal classification," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 529–541, Jun. 2022.

[26] A. Vagollari, V. Schram, W. Wicke, M. Hirschbeck, and W. Gerstacker, "Joint detection and classification of RF signals using deep learning," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–7.

[27] B. Li, W. Huang, W. Wang, and Q. Wang, "Spectrum painting for on-device signal classification," in *Proc. IEEE 25th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2024, pp. 229–238.

[28] A. Almohamad, M. Hasna, S. Althunibat, K. Tekbiyik, and K. Qaraqe, "A deep learning model for LoRa signals classification using cyclostationay features," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 76–81.

[29] R. Zhao, Y. Ruan, H. Xu, T. Li, R. Zhang, D. Yang, and Y. Li, "TRTFL: A transformer based robust time-frequency localization detector for spectrogram with overlapping signals," in *Proc. IEEE 99th Veh. Technol. Conf.*, Jun. 2024, pp. 1–6.

[30] H. Xing, X. Zhang, S. Chang, J. Ren, Z. Zhang, J. Xu, and S. Cui, "Joint signal detection and automatic modulation classification via deep learning," *IEEE Trans. Wireless Commun.*, vol. 23, no. 11, pp. 17129–17142, Nov. 2024.

[31] E. Blossom, "GNU radio: Tools for exploring the radio frequency spectrum," *Linux J.*, vol. 2004, no. 122, p. 4, Jun. 2004.

[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[34] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[36] X. Tian and C. Chen, "Modulation pattern recognition based on Resnet50 neural network," in *Proc. IEEE 2nd Int. Conf. Inf. Commun. Signal Process. (ICICSP)*, Sep. 2019, pp. 34–38.
[37] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan, "Pytorch," in *Programming With TensorFlow: Solution for Edge Computing Applications*. Cham, Switzerland: Springer, 2021, pp. 87–104.
[38] ITU Secretariat. (2020). *The Radio Regulations Volume 1, Edition of 2020*. International Telecommunication Union, Geneva, CH, USA. Accessed: Jul. 24, 2024. [Online]. Available: http://handle.itu.int/11.1002/pub/814b0c44-en
[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

CHUN HO KONG received the B.S. degree in information security from The Hong Kong Polytechnic University, in 2018. Since 2019, he has been a Research Assistant with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. His research interests include cybersecurity, radio frequency, the Internet of Things, and machine learning. He is the Capture the Flag (CTF) Coach of NuttyShell, the PolyU CTF Team, and led the team to win various championships in cybersecurity competitions.

HAIBO HU (Senior Member, IEEE) is currently a Professor with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. As a Principal Investigator, he has received over 25 million HK dollars of external research grants from Hong Kong and mainland China. He has published over 160 research papers in refereed journals, international conferences, and book chapters, and is granted five U.S. patents. His research interests include cybersecurity, data privacy, and adversarial machine learning. He was a recipient of a number of titles and awards, including the IWAIT 2021 Best Paper Award, the IEEE MDM 2019 Best Paper Award, the WAIM Distinguished Young Lecturer, the ICDE 2020 Outstanding Reviewer, the VLDB 2018 Distinguished Reviewer, the ACM-HK Best Ph.D. Paper, the Microsoft Imagine Cup, and the GS1 Internet of Things Award. He is a Senior Member of ACM and CCF, and a certified Cisco CCNA Security Trainer.