ATTACKING IOT WIRELESS PROTOCOLS WITH PREAMBLE EXTRACTION AND SDR

KONG CHUN HO

MPhil

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University
Department of Electrical and Electronic Engineering

Attacking IoT Wireless Protocols with Preamble Extraction and SDR

KONG Chun Ho

A thesis submitted in partial fulfillment of the requirements for
the degree of Master of Philosophy
August 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: KONG Chun Ho

Abstract

The wide adoption of Internet-of-Things (IoT) technologies enabled smart things to connect to the Internet easily by different means. The increasing number of devices are equipped with various IoT Wireless Protocols with Low-Power Wide-Area Network (LPWAN) technologies like Sigfox and LoRa, which can be obtained relatively inexpensively and operate in unlicensed Industrial, Scientific, Medical (ISM) bands.

As more IoT devices are being rolled out, some bear with novel proprietary wireless technologies with different security implications. The emergence of Software-Defined Radios (SDRs) provides the ability of Cognitive Radio (CR), which allows high flexibility and reconfigurability of radios with Commercial-Off-the-Shelf (COTS) hardware with signal processing blocks handled on a computer. In this regard, we wish to utilize SDRs to perform IoT Wireless Protocols LPWAN-based attacks.

One of the most vulnerable parts of an LPWAN packet is the preamble. It is usually found prefixed on a physical layer (PHY) packet of a wireless protocol, allowing these low-powered IoT receivers to wake up from deep sleep and to perform channel tasks including Automatic Gain Control (AGC), frequency, and phase offset estimation for the reception of packets. With the knowledge of the preamble, it is possible to perform high-accuracy jamming attacks and reverse-engineering of the underlying LPWAN protocols. In this thesis, we presented a work that aims to exploit the crucial nature of the preamble, focusing on attacking IoT Wireless Protocols by extracting the preamble part of arbitrary LPWAN signals with SDRs to support preamble attacks.

To extract the preamble of the LPWAN packets, our algorithm requires acquiring the time-frequency location of arbitrary LPWAN signals of different protocol parameters, data rates, bandwidth, and frequencies before doing any band-pass operations on the SDR in-phase / quadrature (IQ) data. To this end, unlike SOTA algorithms that only classify without time-frequency localization of whole IQ data for LPWAN technologies, we also proposed a time-frequency localization machine learning (ML) model for LPWAN signals, based on a Deformable DEtection TRansformer (DETR) architecture, which contains a new attention mechanism called "Multi-Scale Deformable Radial Attention" (MSDRA) based on original Deformable DETR architecture. Application of DETR in LPWAN signals effectively transforms the domain of image detection into IoT Wireless Protocols LPWAN time-frequency localization. This allows our ML model not only to support our preamble extraction attacks but also to enable better spectrum management and band planning with reconnaissance capability, further enhancing the security of IoT.

Publications Arising from the Thesis

1. C. KONG and H. Hu "Automatic Preamble Extraction System for LPWAN Signal", in *9th EAI International Conference on Smart Grid and Innovative Frontiers in Telecommunications (SmartGIFT '24)*, Hong Kong, Dec 2024. **(Outstanding Paper Award)**

2. C. KONG and H. Hu "Classification and Time-Frequency Localization of Arbitrary LPWAN Signals with Radial Deformable DETR", in *IEEE Access*, vol. 13, pp. 53065-53083, 2025, doi:10.1109/ACCESS.2025.3554080.

Personal Awards and Speeches

1. C. KONG "NuttyShell Sharing by PolyU CTF Coach", in *PolyU x NuttyShell Cybersecurity CTF 2025 Prize Presentation Ceremony*, 2025.

2. 2^{nd} runner-up in "HKCERT Capture the Flag Challenge 2024 (Open)", organized by *Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT)*, 2024.

3. C. KONG "NuttyShell Sharing by PolyU CTF Coach", in *PolyU x NuttyShell Cybersecurity CTF 2024 Prize Presentation Ceremony*, 2024.

4. 4^{th} runner-up in "Hong Kong Cyber Security New Generation Capture the Flag Challenge (Open)", organized by *Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT)*, 2023.

5. 2^{nd} runner-up in "Hong Kong Cyber Security New Generation Capture the Flag Challenge (Open)", organized by *Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT)*, 2022.

6. Champion in "HackTheBoo CTF", organized by *HackTheBox*, 2022.

Team Awards as CTF Coach

1. 2^{nd} runner-up in "HKCERT Capture the Flag Challenge 2024 (Tertiary)", organized by *Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT)*, 2024.

2. Champion in "HackaDay", organized by *PwC*, 2024.

3. Champion in "HKUST Firebird CTF Competition 2024", organized by *Hong Kong University of Science and Technology*, 2024.

4. Champion in "Hong Kong Cyber Security New Generation Capture the Flag Challenge (Tertiary)", organized by *Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT)*, 2023.

5. 2^{nd} runner-up in "Hong Kong Cyber Security New Generation Capture the Flag Challenge (Tertiary)", organized by *Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT)*, 2023.

6. Champion in "HackaDay", organized by *PwC*, 2023.

7. 2^{nd} runner-up in "Greater Bay Area Cybersecurity Competition", organized by *IVE*, 2023.

8. Champion in "HackaDay", organized by *PwC*, 2022.

9. Champion in "Hong Kong Cyber Security New Generation Capture the Flag Challenge (Tertiary)", organized by *Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT)*, 2022.

10. 1^{st} runner-up in "Cybersecurity Purple Team Competition", organized by *IVE*, 2022.

11. 2^{nd} runner-up in "HackaDay", organized by *PwC*, 2021.

12. 1^{st} runner-up in "CyberSecurity Blue Team Competition", organized by *IVE*, 2021.

13. 3^{rd} runner-up in "HackaDay", organized by *PwC*, 2020.

14. 1^{st} runner-up in "Hong Kong Cyber Security New Generation Capture the Flag Challenge", organized by *Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT)*, 2020.

Acknowledgments

It has been a long journey to complete my studies. First and foremost I would like to express my gratitude to my supervisor, Professor Haibo HU, who guided me toward a solid research direction and enriched my understanding of research writing and experimental frameworks. His research passion and support helped me go through various obstacles during my research. Next, as the Capture The Flag (CTF) Coach of our CTF Team, NuttyShell, I am incredibly thankful for the unwavering support from our teammates and students, which has allowed me to fully concentrate on my research without the need to focus heavily on managing the team, all while still achieving success in local and global CTF competitions. Finally, I would not be able to complete my studies without my family and friends, who will always stand behind me and make me feel alright at all times.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the emergence of the Internet of Things (IoT) technologies, more IoT devices are being deployed across the globe. For the global IoT market, it is expected to grow to the size of USD 8 Trillion by the year 2030. [1] IoT devices are widely used for Machine-to-Machine (M2M) communications or Wireless Sensor Networks (WSNs) in a variety of applications such as smart grids, smart cities, smart home, healthcare, etc. [2] [3] [4] However, due to limited computation power on these IoT devices, concerns regarding information security [5] [6] [7] [8] and privacy [9] [10] [11], including the appearance of rogue IoT devices [12] have been raised. In 2020, there was a 100% increase of infected IoT devices compared with 2019 and it was expected that IoT infections will grow intensely for later years. [13] IoT devices are commonly equipped with radio transceivers that enable them with wireless capabilities leveraging a huge range of IoT wireless protocols, often using Low-Power Wide-Area Network (LPWAN). Whether they are open or proprietary, well understood or poorly researched, these LPWAN wireless protocols come with a variety of security implications. [14] Often, IoT wireless protocols are transmitted with preambles that are prefixed on the packets before the real payload. Commonly, this could be useful to wake up a transceiver in low power mode, identify start of a new packet frame, or even perform Automatic

Gain/Frequency Control (AGC/AFC) on the receiver front-end. [15] The existence of preamble fields in physical (PHY) layers can be seen in protocols including 802.11(Wi-Fi) [16], BLE [17], LoRa[WAN] [18], Sigfox [19] (UNB) [20] and more. Due to the essential nature of preamble in wireless protocols for a wireless receiver to function properly, it became a high value target in terms of IoT attacks, as disallowing the preamble to be received by a legitimate receiver by means like reactive jamming [21] would cause degradation of wireless networks, or even denial of services. [22] For most of the LPWAN IoT Wireless Protocols (IoTWPs) that are having a fixed preamble, one does not require the knowledge of the actual payloads, even if they are encrypted, to perform attacks that are targeting preamble portion of a specific LPWAN IoTWP.

On the other hand, with cheap Software-Defined Radios (SDRs), which provides IQ (Inphase/Quadrature) samples at arbitrary frequencies and bands and sample rates [23], it is possible for an adversary to perform attacks to wireless systems more easily as they are highly flexible and configurable with software. [24] The reconfigurability of SDRs is closely related to Cognitive Radio (CR) [25], which provides reconfigurability for radios to adapt to variations of new wireless standards and incorporate new services and applications when they arise. By utilizing SDRs to acquire IQ data to a computer, it is possible to effortlessly reconfigure the radio receiver on the fly to receive various LPWAN signals across the ISM bands of different bandwidth, protocol parameters, or modulation schemes, with minimal costs.

In this work, we devise a scheme that will leverage the power of Commercial off-the-shelf (COTS) SDRs which are widely available in the market to perform Preamble Extraction from general frequency-changing LPWAN IoTWPs RF traces, regardless it contains proprietary or open protocol standards. Unlike some works that assume the modulation scheme of the target signal, our work is designed to work with minimal prior knowledge and wireless protocols parameters, including those that are not being currently known of.

Before extracting the preamble of a signal, it is required to detect the presence of sig-

nals. To this end, we also proposed a novel machine learning (ML) approach to classify and perform time-frequency localization of LPWAN signals in Spectrogram from IQ data received by SDRs. Based on Deformable DEtection TRansformer (Deformable DETR), which utilizes Multi-scale Deformable Attention (MSDA) mechanism that perform deformable attentions in multiple feature scales in place of ordinary attention mechanism, we introduced Multi-scale Deformable Radial Attention (MSDRA) mechanism to allow better handle of LPWAN data. Novel techniques like transfer learning are also introduced to shift the paradigm from traditional image detection to LPWAN RF signal detection and localization.

To summarize our work we proposed a scheme to:

1. Perform LPWAN IoTWPs Packet classification and time-frequency localization from spectrogram traces obtained from SDR IQ data, utilizing ML techniques including Deformable DETR, MSDRA and transfer learning

2. Extract these LPWAN IoTWP Packets as IQ data by performing Signal Processing tasks like band-pass filtering with finite impulse response (FIR) filter.

3. Compare and extract common signal features prefixing these RF packets, based on cross-correlation and thresholding

4. Generate similarity metric between LPWAN IoTWP packets across frequency-time domain that determines likely preambles locations

Compared with SOTA algorithms, our MSDRA-based LPWAN time-frequency localization machine learning algorithm allows time-frequency localization of LPWAN signals, while able to identify LPWAN protocols instead of just modulation formats. It was also one of the first algorithms that trained and evaluated real-world-like overlapping LPWAN signals in variable SNR conditions. In addition, we are one of the first that provide a full dataset generation pipeline that relies on COTS SDRs to generate these training datasets. For LPWAN preamble extraction, our method also

eliminated the requirement to label datasets for training, while enabling preamble localization on narrow-band LPWAN signals.

The remainder of this thesis will be organized as follows. In chapter 2, we will walk though some background and related work in the field of RF signal classification, time-frequency localization, detection and preamble extraction, followed by the importance of LPWAN preambles by demonstrating various LPWAN attack and defense methods that are powered by preamble. Then, chapter 3 will lay out the problem statements and assumptions for this work, which are essential for a correct understanding of the research objective. Next, we will walk through the system design in detail in chapter 4, which contains a novel algorithm of LPWAN packet extraction with our proposed Modified Deformable DETR with MSDRA, and the signal comparison and preamble extraction engine. In chapter 5, we present the experimental results and discussions were given, along with some interpretations. We will then move on to discuss improvements and future work in chapter 6 and conclude this work in chapter 7.

Chapter 2

Research Background and Related Work

In this section, we review literature on related work and provide some research background. It is organized into two parts, namely Preamble Detection from RF Packets, Preamble Supported Attacks and Defenses, and Packet Extraction Techniques.

2.1 Preamble Detection from RF Packets

Several works focus on detecting the start of signals (and hence the preamble) either as a standalone algorithm, or as a part of an attack procedure that requires usage of preamble. Note that these works focus on detecting the start of the preamble (i.e. start of the signal), but not the end of the preamble, hence, they did not accurately localize the both the time where the preamble starts and ends, plus utilizing COTS SDRs in LPWAN signals.

For instance, in [26], a standalone transient detection technique is presented to detect the turn-on transients of Wi-Fi radios. Wi-Fi radios are sometimes required to have

a slow turn-on time of their RF signals [26] to avoid interfering with adjacent RF channels. Hence, a Bayesian change detector was created which aims to estimate the time instant that when a transmitter starts to power up. Note that the equipment used in this paper was an expensive digital scope with a computer attached, rather than a COTS SDR, and requires prior knowledge of the underlying RF protocol.

Another paper [27] demonstrated an alternate transient detection method, which utilizes variance trajectory sequences. By using a moving window and calculating the variance value of all samples within the window, a trajectory of variance value could be calculated along with increasing window number. A customizable threshold value could be supplied to determine the start of a signal. This paper [27] also demonstrated a way to perform preamble extraction, which is by using Discrete Fourier Transform (DFT) on the overall samples and subsequently obtaining the Power Spectral Density (PSD) of the samples. Finally, cross-correlation is used on the PSD sequence. This paper assumes the signal is 802.11a Wi-Fi.

A more recent paper [28] also leveraged similar techniques with cross-correlation to extract the preamble features for high-bandwidth Unmanned Aerial Vehicle (UAV) signals for Machine Learning. This is different with other related work as it wants to achieve similar goals with this work, which is trying to exploit preamble from unknown signals independent from their modulation types, in this case, UAV drones. However, the work did not mention any transient detection techniques. However, this work does not aim to obtain the exact time location of the preamble, but rather obtain a feature map of the preamble as training dataset for subsequent machine learning applications to identify different drones. They also employed cross-correlation directly on the IQ data in the time domain with a moving window, on high-bandwidth data, which is different from our work as we focus on frequency-time domain, on narrow-band LPWAN packets, which inherently contains fewer usable information due to narrower band in nature. The feature extraction technique in this paper is a part of an attack or a workflow that aims to classify different kinds of UAV model in the area, with

neural network and federated learning, with 99.97% of accuracy. Note that this work utilized a USRP B210 SDR as hardware.

Another powerful software used for automatic wireless protocol reverse engineering, called Universal Radio Hacker (URH) [29] is widely used in reverse engineering of wireless protocols, including LPWAN IoTWPs. The way that it detects and guesses the preamble is by assuming a repeating bit pattern in the preamble, which is normal in preamble as they need to have a specific repeating characteristic for a receive device to be aware of. URH counts 1's and 0's for each position and try to estimate the range of the ending location of the preamble, and the subsequent sync word in the bit stream. However, it assumes the user has a basic understanding on the modulation scheme of the target signal, which it can in turn demodulate and convert the wireless symbols back to bit streams. This implies that URH works only on raw bit streams after demodulation. URH is easily incorporated with SDR hardware and I/Q traces.

From these works we can discover that some methods require prior knowledge of the target wireless protocol, and some do not. For detecting start of signal, they employ various mathematical and statistics models and to detect the presence of the preamble from a(n) (unknown) trace, cross-correlation is often used on time-domain, or in frequency domain for all samples. SDR hardware can also be used in these schemes. However, these works do not accurately determine the end location of the preambles. To close the research gap, we present a method that works directly on the RF data without the requirement of demodulation, while allowing preamble localization on narrow-band LPWAN signals, especially Sub-GHz ones. This is achieved by using the spectrogram traces which will be outlined in the later chapters.

2.2 Preamble Supported Attacks and Defenses

As the preamble field is an extremely crucial location in LPWAN IoTWPs (or other protocols), attacking it could bring severe effects to the whole wireless system compared with attacking other parts of the same packet. Also it could undermine privacy as it already provides enough information to uniquely identify an RF transmitter. Here, we outline two types of attack that are supported by the usage of preamble in wireless protocols.

2.2.1 Preamble Jamming

There are many works that target the preamble part of a signal to perform jamming attacks. It is particularly useful to create a low-power jammer in reactive jamming as the preamble of a target signal can be used to synchronize the jammer with the target. Moreover, the jammer are not required to transmit the jamming signal all the time as it can wait until the reception of the target preamble before start to jam, this not only reduce the power consumption for a jammer to jam longer time, but also reduce the chances of the jammer being discovered and located. These ideas can be easily implemented easily on a SDR. [30] Several works focused on jamming the preamble of a wireless protocol to degrade the performance of such networks. In [31], the authors used a counter-intuitive method, which was an extremely smart move to jam Vehicular ad hoc networks (VANETs), which undermines road safety. Not only does it work in a lab environment in an anechoic chamber, but also outdoors which resulted in great impact with a large black-out area despite very short communication distances between legitimate devices. The authors have used a weak preamble-like structure as a jamming signal, and tricked the receiver to receive the false signal and perform AGC to increase its receiver gain for the reception of such weak signal. However, when there is a legitimate signal on the air, the receiver will fail to receive it because there is dynamic range overload at the frontend analog-to-digital converter (ADC) which was

triggered by a high AGC from the weak preamble-like jamming signal. Besides from VANETs, general approaches to attack Orthogonal Frequency-Division Multiplexing (OFDM) receivers were also proposed. In [32], the author mentioned "Preamble Whitening" attack which involves generating white noise jamming signal to jam only the preamble part of the OFDM receiver. "False Preamble Timing Attack" involves corrupt the symbol timing estimation on the receiver side by moving or destroying the timing metric peak, by retransmitting the same/different preamble in an incorrect time slot. Finally, a complicated method called "Preamble Nulling Attack" requires the jammer to have a full knowledge of the preamble waveform on the receiver side. By inverting the preamble symbols, it is possible to destructive interfere with the preamble symbol on the receiver side. However, it requires the existence of a jammer and receiver and have the knowledge of the same or similar channel condition in order to be effective.

A more SDR-centric approach can be seen in [33] with the USRP N210 SDR. The authors directly implemented custom FPGA code on the SDR, designed to interface and function with the GNURadio [34] software on the host side. The FPGA code that is running on the SDR itself allows rapid reactive jamming capability as it can respond to on-air RF events in a timely manner, as low as 80ns. The system could operate in two modes, the first one is targeted signal detection, which requires a prior knowledge of the preamble from a specific wireless standard for the implemented internal cross-correlator inside the FPGA. Without a known preamble, the second mode is useful as it makes use of an internal differential energy detector to jam any kind of RF signals on a specific predefined frequencies and bands. With the first method, the authors have successfully jammed a Wi-Fi network which results in the UDP bandwidth to be 0Kbps, with a relatively low jamming power. For the second method, they have successfully jammed 100% of the downlink WiMAX packets which could result in the total degradation of the mobile wireless network. It is notable that these preamble jamming techniques are not only applicable in RF medium, but also in

underwater acoustic links. In [35], the authors used 3 commercial underwater acoustic modems to generate jamming signals similar to those in RF on the air, which also result in loss of packet when jamming the preamble. In [36], the authors discovered that jamming the preamble in Underwater Wireless Sensor Networks (UWSNs) is the most effective way to jam such networks. When launching the jamming attack in the first second, it was discovered that it will always jam the underlying communication channel. The authors then observed the packet delivery ratio had dropped to 0% in constant and reactive jamming for UWSNs. These works outlined the importance of preamble in wireless protocols. Whether the transmission medium is electromagnetic RF or underwater acoustic links, the basic principle for preamble jamming is the same. With reactive jamming, it is possible to precisely jam specific packets without the need to constantly turn on the jammer. With the usage of SDRs, it is possible to perform preamble jamming on arbitrary RF Signals (or LPWAN Protocols) with ease.

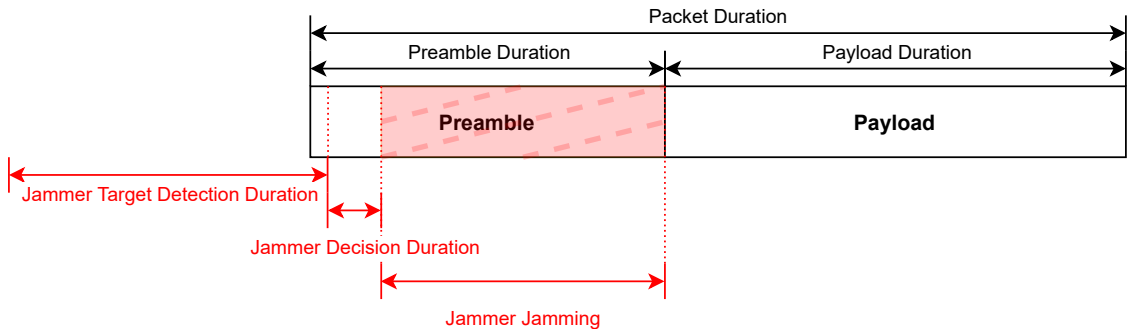

Figure 2.1: **General Reactive Jamming Attack on Preamble**

2.2.2 RF Fingerprinting and Target Identification

Besides from using the preamble as a tool for jamming wireless protocols, it also could be used to undermine privacy of the underlying RF users by performing RF fingerprinting and device identification.

In [28], after extracting preamble features from UAV downlink signals, which have unknown modulation schemes and protocol formats, the authors generated cross-correlation feature maps from these preamble traces. These feature maps are in the form of 3D matrices and with the application of 3D convolutional neural networks to classify different UAV make/model. Note that in this case, the authors do not want to identify each individual UAVs, but rather the same make/model. To detect a new UAV which could inevitably be classified into one of the classes that have been trained, the authors tried to perform novelty detection, by using the Euclidean distance to find anomalies in their predictions from known UAVs and unknown UAVs.

There are also other works that focus on classifying wireless devices individually. One such work is [37], where it uses techniques for feature reduction such as Partial Least Squares (PLS) regression and Principal Component Analysis (PCA). They have performed classifications between devices by calculating the lowest distances of fingerprint samples between a test set and a reference set. Finally, for an experiment of 20 devices, 2 per manufacturer, which includes a variety of Wi-Fi devices, from smartphones to Wi-Fi dongles. They have achieved 70% accuracy, with the help of the PLS algorithm. It is worth noting that for some devices with the same manufacturer, they have a very similar fingerprint that avoids proper identification between such devices.

For an actual LPWAN IoTWP, this work involving the LoRa protocol which is widely used in IoT devices, [38] have successfully extracted RF fingerprint features from LoRa preambles, regardless whether the payload is encrypted or not. The authors used two types of LoRa modules, which consist of 3 modules with SX1278 chip, and 1 module

with SX1276 chip. By leveraging the frequency offset between the transmitter and receiver, RF fingerprint could be extracted with a differential constellation trace figure (DCTF). With a simple clustering algorithm, the authors have succeeded in clustering 4 DCTF clusters for each module and in turn uniquely identifying all individual 4 modules.

Besides attacking LPWAN Protocols with preamble, it is also possible to further secure the network with the knowledge of preambles. For example, in [39], a PHY level wireless intrusion detection system is proposed, through preamble manipulation. This works because multiple make and model of hardware produce different different response patterns when the preamble is being manipulated. The authors essentially created a wireless multi-factor authentication (MFA) and intrusion detection system through device fingerprinting, by varying the actual preamble transmitted on the transmitter. In [40], an intrusion detection system is implemented for the inherently unauthenticated and unencrypted ADS-B protocol, operating at 1090MHz. It allows the detection of False Aircraft Injection Attack by creating a database of RF features from legitimate aircraft ADS-B transmitters, which is created by ADS-B signal signature extraction, containing the ADS-B preambles, further enhancing aviation safety.

From these related works, it is obvious that using preamble from wireless protocols can also identify individual devices at different levels, from manufacturer/vendor level, to the level of individual devices. RF fingerprinting and device identification with preamble could be used as a tool to attack or defend the security and privacy of LPWAN IoTWPs.

2.3 Packet Extraction Techniques

Before our system can reliably compare and extract the preambles of the LPWAN packets, we require individual packets being extracted to start with. These methodologies can be broadly categorized based on whether they involve using machine learning techniques. Both categories will be outlined in this section.

2.3.1 Non ML-based Packet Extraction Techniques

For non-ML approaches, some work [41] [42] 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 [43] [44] [45] 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. [46] 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.

2.3.2 ML-based Packet Extraction Techniques

For ML-based approaches, most of the work is based on CNNs, which is prevalent in image classification tasks. Some focused on modulation classification [47] [48] instead. For technology classification, in [49], 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, Wi-Fi, and Bluetooth on 2.4GHz bands, which marked the difference with our work. In [50], 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 [51], 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.

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. Deformable DETR [52], which is based on DEtection TRansformer (DETR) [53] applies the transformer model and attention mechanism [54] 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. DETR simplifies the detection pipeline by framing the problem as a direct set prediction task, to reduce the subsequent post processing steps. Deformable DETR introduced deformable attention mechanisms to DETR, which allows the model to attend to certain points in an image, increasing accuracy and achieving faster convergence. Henceforth, this work will attempt to investigate LPWAN technology classification and time-frequency localization with our modified Deformable DETR model with Radial Attention Mechanism 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.

Table 2.1: **Comparison between our 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
[47]	No	Modulations	No	No
[51]	No	LPWAN Protocols	No	Partial
[41]	No	Individual Devices	No	No
[50]	No	LoRa Spreading Factors	No	No
[49]	No	2.4GHz Protocols	Yes	No
[48]	Yes	Modulations	No	No
[55]	Yes	4G/5G	Yes	No
[56]	Yes	Modulations	No	No

Additionally, Table 2.1 outlines the differences between our work and other state-of-the-art approaches that utilize Machine Learning methods in the regard of technology classification and time-frequency localization for LPWAN packets.

Chapter 3

Problem Definition and Objectives

3.1 Objectives

We have set out 3 objectives for this research work and aim to achieve all of them.

3.1.1 Reliable for Reverse Engineering

For an arbitrary LPWAN IoTWPs, regardless if it is open or proprietary, a modulation method that is seen before or not, giving it is frequency changing in nature, we seek to automatically extract a common prefix from every packets of such protocols, and such common prefix is regarded as preambles. With these automatically extracted preambles, these datasets are more trivial for an experienced reverse engineer to perform automatic/manual reverse engineering on, or as a basis to defend against LPWAN IoTWPs attacks. Hence, we seek to attack or defend an IoTWPs, by lowering the amount of work needed to manually extract such traces, and rejecting false positives on possible preamble traces and thus lower the difficulty to do reverse engineering on LPWAN IoTWPs.

3.1.2 Support Preamble-based Attack/Defense

With a set of preamble extracted for a given LPWAN IoTWP, these traces can now be used as datasets for training, testing, or validating and supporting preamble-supported attacks or defenses. From chapter 2, it is obvious that preamble samples are needed in order to carry out a variety of attacks or defenses. Our work seeks to incorporate in these attack or intrusion detection pipelines for better attack and defense performances.

3.1.3 Universality

Unlike many of the related works, this work aims to work on a variety of LPWAN IoTWPs and modulation schemes, given that it is frequency changing in nature, regardless of existence discovery by the research community. This work is also aimed to work on PHY layers irrespective of upper layer design. (e.g. Works on LoRa meaning also works on LoRaWAN). Regardless if the payload is encrypted or not, or an encrypted application layer is used, the preamble part of a wireless signal should still be able to be attacked on. Nonetheless, we would like to have our scheme to work on the same LPWAN IoTWP but with another model, manufacturer, and devices. Most importantly, this work should be able to work on COTS SDR devices without the need of expensive equipment to be operated on.

3.2 Assumptions

We assumes the following for our work:

- **LPWAN IoTWPs are digital, packet-based:** As we would like to extract preambles from each of the packets, LPWAN IoTWPs have to be a packet-based protocol. In fact, existing protocols like BLE, Wi-Fi, LoRa, Zigbee, Sigfox, etc. are all packet-based, and of course, they are also digital signals.
- **Most, if not all OTA packets consists of preamble:** We assumes that packets on the air (OTA), we would observe preamble-included packets as it is normally used to synchronize a legitimate receiver and optionally perform AGC/AFC tasks.
- **Preambles occur immediately in presence of signal:** Preambles should be prefixed at the beginning of a packet as it was designed to perform the tasks we mentioned earlier.
- **Average signal power of the preamble portion is similar, or higher then rest of the signal:** As the preamble is used for receiver synchronization, there is no point if it signal strength is lower than the rest of the signal, especially AGC are being based on it. We assume that we could observe the preamble in a spectrogram plot like we could observe the payload in the later part of the plot.
- **LPWAN IoTWPs are frequency-changing:** For the scope of this work, we limit IoTWPs to frequency changing. For non-frequency changing signals, please refer to chapter 6.
- **The LPWAN IoTWP signal could be fully captured by SDR, with given F_s, F_c and BW:** This is a hardware and configuration limitation and not related with this work. It is expected the specification of the SDR is sufficiently enough to receive the LPWAN IoTWP at the target center frequency (F_c), with enough Bandwidth (BW) and I/Q sample rate (F_s).

- **Parameters are allowed to exist for adjustment:** Parameters, such as threshold values should be allowed to be adjusted in our work, in order to accommodate different RF environments, channel conditions and hardware in our signal path like antennas.

3.3 Threat Model

The work proposed has the following threat model.

- **Accessing the RF spectrum via SDR:** SDR is the only way for us to gain access to the RF spectrum. It will be connected to a PC via appropriate methods, e.g. Universal Serial Bus (USB), depending on the model of the SDR.
- **ISM Bands:** For the simplicity of the threat model, we assume the LPWAN IoTWPs occur in the Industrial Scientific Medical (ISM) Band. Although the algorithms presented in this paper work on multiple bands with the SDRs changing reception center frequencies (F_c), we can legally receive (and transmit) signals in ISM bands.
- **Passive collection:** There should be no need to actively transmit any RF signals on the air in order to capture the preamble for the known, or unknown LPWAN IoTWPs. We seek to passively perform data collection and extract preamble out of it in stealth to perform further attacks, exploitations, or intrusion detection.
- **Collison allowed:** Given that we work on congested bands, there are risks for protocol and packet overlapping or collisions. Thus, especially for packet detection and extraction, we allow overlapping LPWAN signals to increase the difficulty of LPWAN IoTWPs packet extractions.
- **Signal Processing on Computer:** All of the signal processing tasks outlined in this thesis are solely running on a computer, without dedicated hardware like spectrum analyzers (except SDRs).

Chapter 4

System Design

In this chapter, we will lay out in detail on how we designed our system to extract preamble traces using SDRs. IQ data are received to a computer from a SDR, then sent to the stage of LPWAN Packets Extraction, followed by down-sampling and the signal comparison and extraction engine. The overview of the system is outlined in Figure 4.1.

Overall, the system consists of the following components:

1. **IQ Data Acquisitions from SDRs and GNU Radio:** To transfer data from the SDR to the computer for subsequent data and signal processing tasks and LPWAN packets extraction tasks, we have utilized GNU Radio [34] to perform IQ data acquisitions.

2. fLPWAN Packet Extraction with Radial Deformable DETR: With the IQ data saved locally on the computer or directly streamed from the SDR devices, we can proceed to extract individual LPWAN packets from the IQ data, utilizing our proposed trained Modified Deformable DETR with MSDRA mechanism.

3. **Signal Filtering and Frequency Shifting:** As the baseband signal from SDR might contain multiple LPWAN packets at once across different frequen-

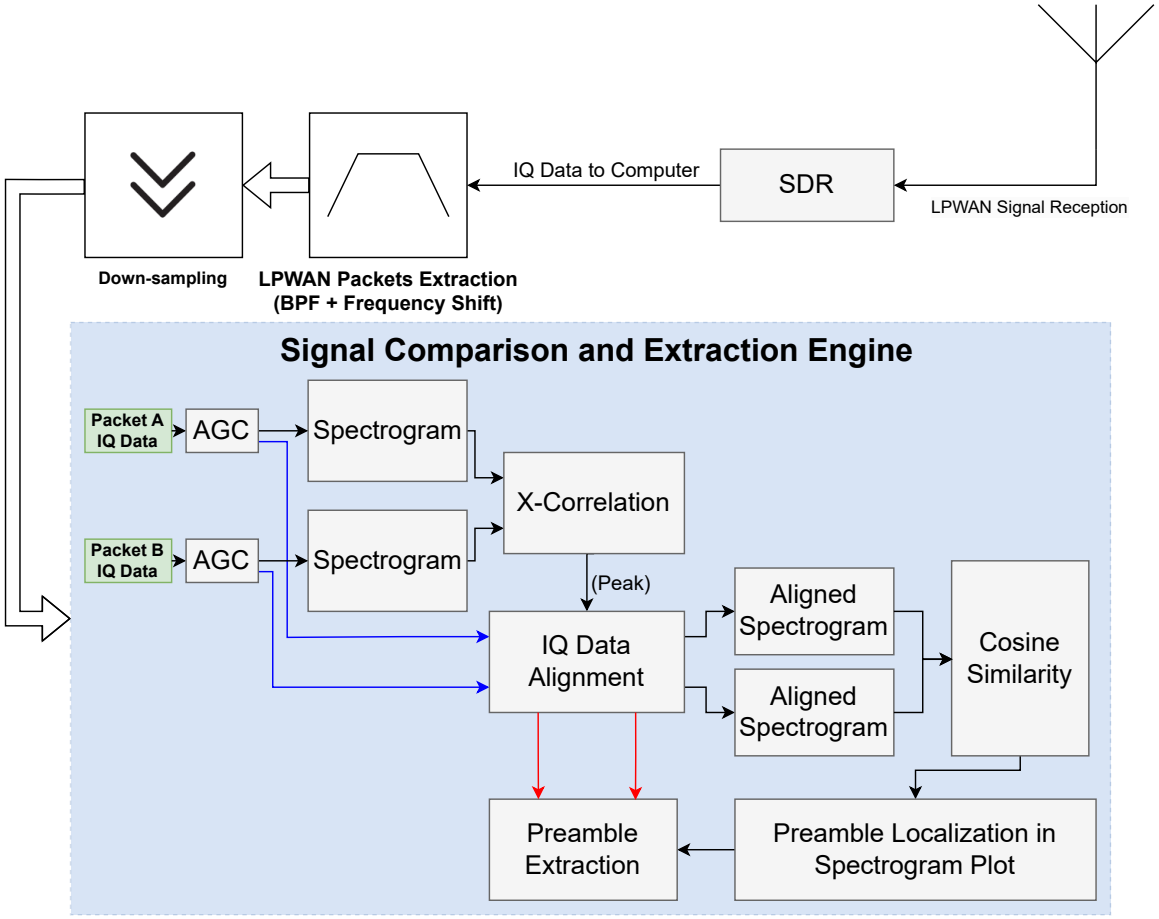

Figure 4.1: **System Overview**

cies, signal filtering is used to further reduce other uninterested frequencies in our IQ data. The LPWAN Packet is further shifted to around 0Hz.

4. **Down-sampling:** Downsampling is used to reduce the amount of data for subsequent processing. As the baseband signal no longer contains other LPWAN packets, we can reduce the sampling rate (and the resulting bandwidth) of the signal.

5. **Signal Comparison and Extraction Engine:** We only compare two LPWAN packets at a time to find out their common preamble. To compare each two packets, we first took Short-Time Fourier Transform (STFT) for each packet, squaring their magnitude, which obtained two spectrograms. We then

align the two spectrograms with cross-correlation by determining the peak of the cross-correlation output and trimming IQ samples to align both packet traces. We then plot two spectrograms again on both of the aligned packets, perform cosine similarity matching for each STFT chunk. For a configurable threshold value, if the similarity is higher than a defined threshold, that specific STFT chunk is considered similar. After we have obtained the preamble location of the two traces, we simply extract them from the packet from the IQ data.

In the later sections, we will describe these algorithms and systems.

4.1 IQ Data Acquisitions from SDRs and GNU Radio

The first step of this system is to acquire IQ samples from the SDR, and save the IQ file locally for further analysis, or directly stream the IQ data to the next components. The user have to broadly know which the frequency the target signal ($f_{c(target)}$) is located in (e,g, in ISM bands), and also setting an appropriate sampling rate (f_s) and center frequency ($f_{c(sdr)}$) for the SDR. f_s should be chosen to satisfy the Nyquist rate of the target signal. Note that as we are now capturing **both** I and Q data at the same time, we are effectively capturing twice of the samples per unit time. Thus, we can satisfy the Nyquist rate by simply setting the SDR sampling rate to be the signal bandwidth. For example:

$$f_s = 2Msps$$

$$f_{c(sdr)} = 432.800MHz$$

This will result in:

$$BW = 2MHz$$
$$f_{start} = f_{c(sdr)} - \frac{BW}{2} = 432.800 - 1.000 = 431.800MHz$$
$$f_{end} = f_{c(sdr)} + \frac{BW}{2} = 432.800 + 1.000 = 433.800MHz$$

Where BW is the SDR signal acquisition bandwidth, f_{start} to f_{stop} is the RF frequency range that the SDR is capturing I/Q data on.

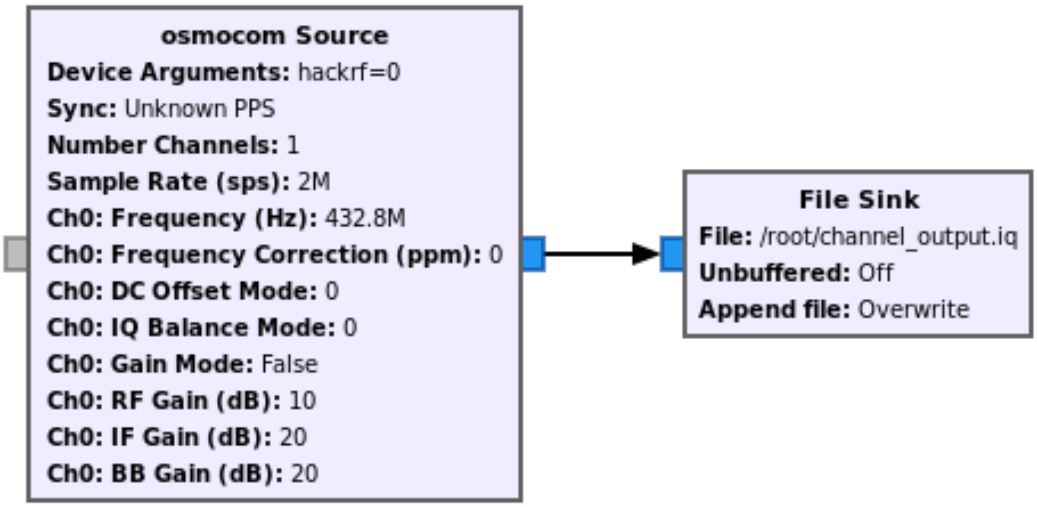

Figure 4.2: **Example GNURadio Flowgraph for Recording IQ Signals ($f_s = 2Msps$, $BW = 2MHz$, $f_{c(sdr)} = 432.800MHz$, SDR Used: HackRF One)**

4.2 LPWAN Packet Extraction with Radial Deformable DETR

4.2.1 Overview

A modified version of Deformable DEtection TRansformer (Deformable DETR) [52] to process LPWAN spectrogram traces obtained from SDRs In-phase and quadrature (IQ) data is proposed in this section. More specifically, we replaced the Multi-scale Deformable Attention (MSDA) to Multi-scale Deformable Radial Attention (MSDRA), which allows better process of RF spectrogram 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 novelly transfer object detection in machine vision to LPWAN protocol classification and time-frequency localization.** Traditionally, 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 [57]. For multiple LPWAN technologies coexisting in the spectrum, especially in the ISM bands, 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 [58] can be employed as a means to perform modulation classification [59], technology classification [51], or source identification [60] 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.

Hence, for our preamble extraction framework, we proposed the novel use of Deformable DETR with our MSDRA mechanism to extract LPWAN packets for our subsequent preamble extraction engine. In the following, we will outline the dataset generation and augmentation for training the model, description of the Modified Deformable DETR design, and transfer learning techniques.

4.2.2 Dataset Generation

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. Figure 4.3, Figure 4.4, and Figure 4.5 outline our IQ dataset generation architecture and our actual setup for LoRa and Sigfox, respectively.

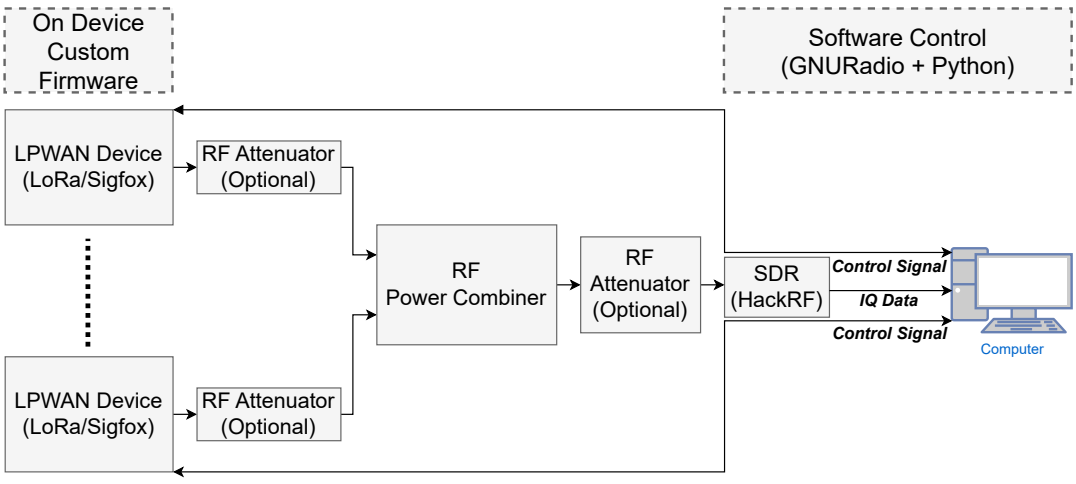

Figure 4.3: **IQ dataset generation architecture**

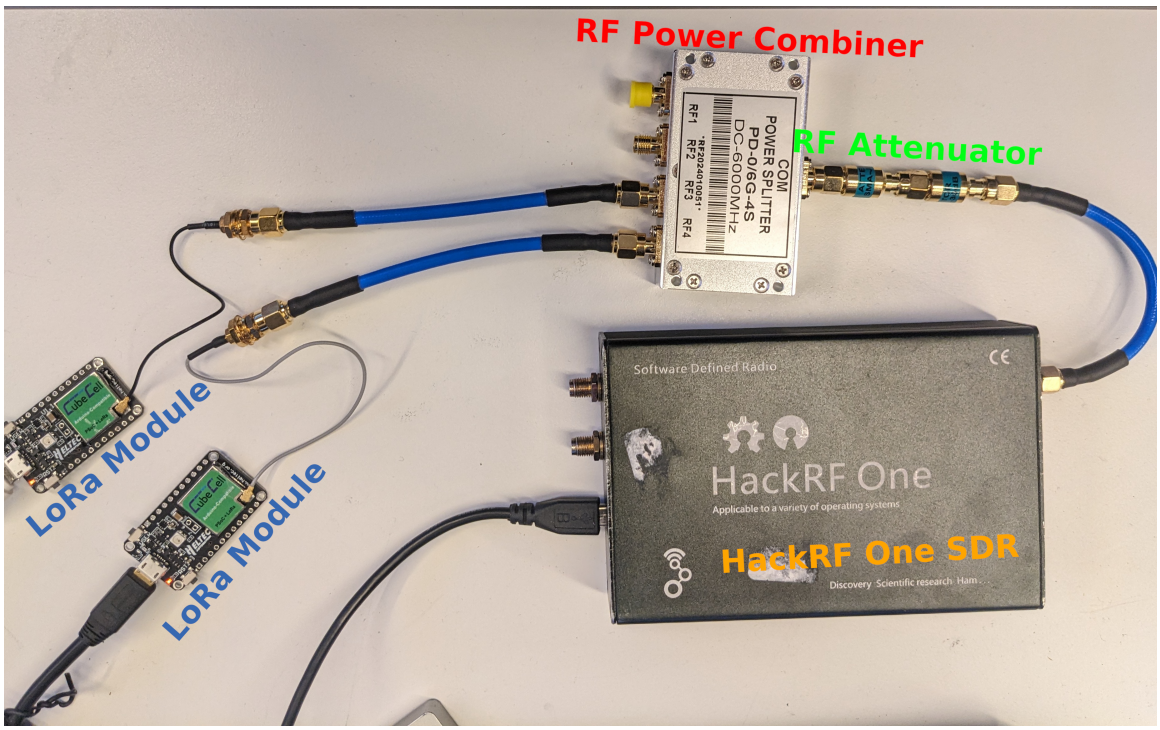

Figure 4.4: **Actual setup of LoRa IQ dataset generation**

4.2.2.1 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 [34] Blocks

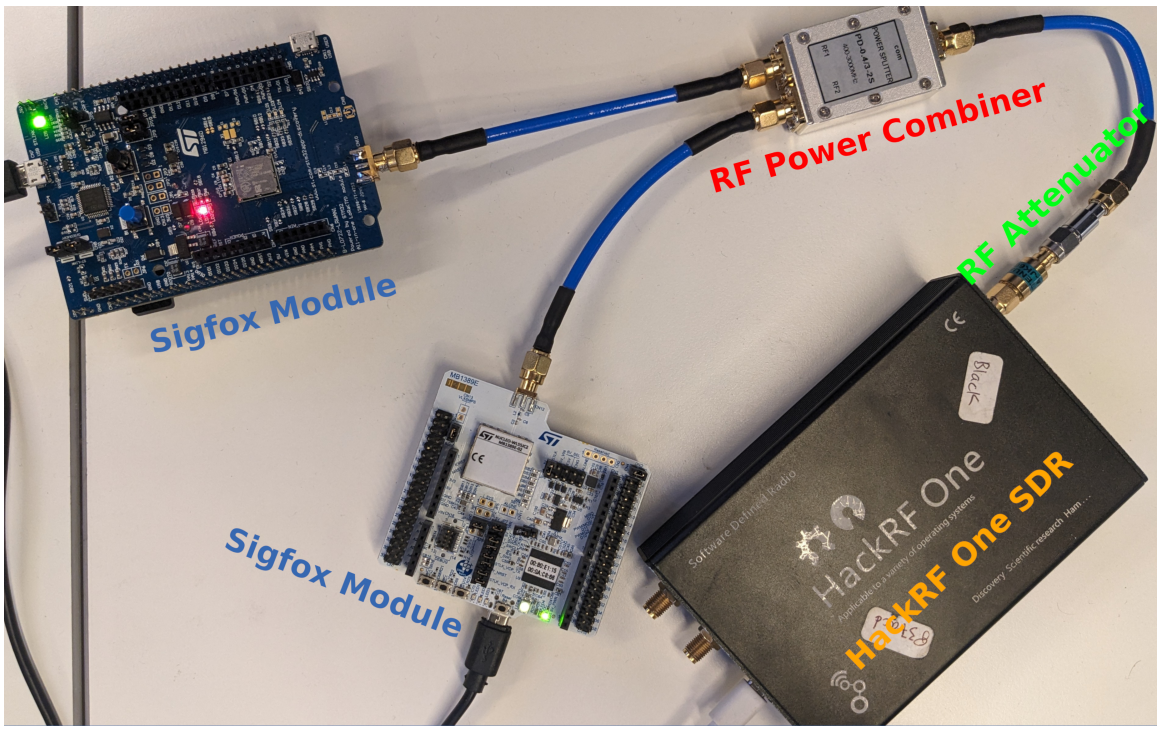

Figure 4.5: **Actual setup of Sigfox IQ dataset generation**

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:

$$x[n] = \text{Re}(x[n]) + j\,\text{Im}(x[n]) = I[n] + jQ[n] \tag{4.1}$$

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

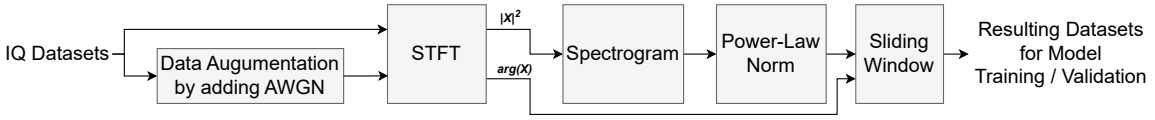

Figure 4.6: **IQ dataset processing pipeline**

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

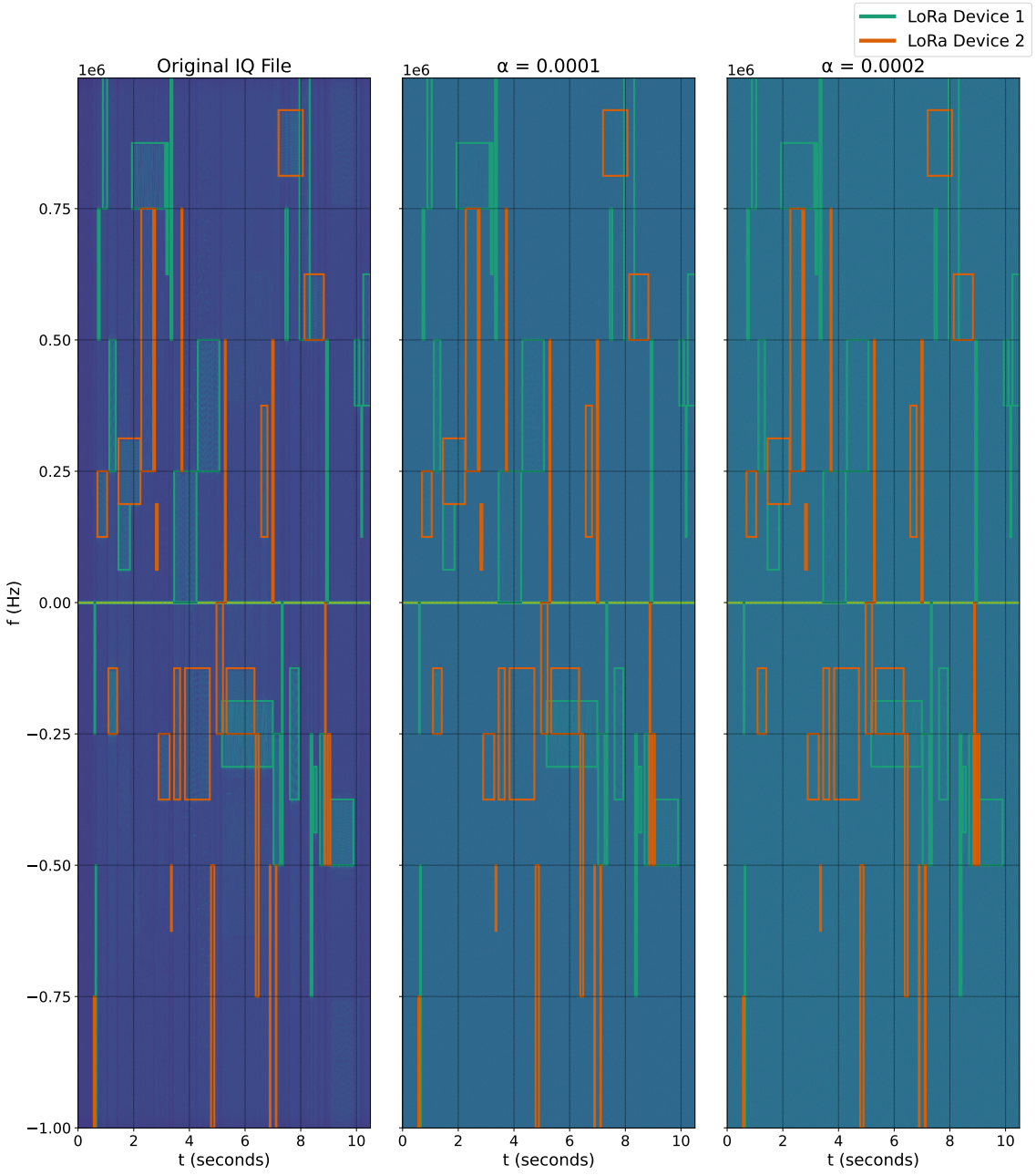

Figure 4.7: **Original IQ and AWGN augmented datasets comparison**

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 be-

fore feeding into the model for training and validation. The data augmentation and dataset processing pipeline are outlined in Figure 4.6.

4.2.2.2 Data Augmentation with AWGN

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

$$w[n] = \frac{\mathcal{N}(0,1) + j\mathcal{N}(0,1)}{\sqrt{2}} \tag{4.2}$$

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.

$$x_{noisy}[n] = x[n] + \sqrt{\alpha} \times w[n] \tag{4.3}$$

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 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 Figure 4.7.

4.2.3 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:

$$X[k,m] = \sum_{n=kR}^{kR+N-1} x[n]W[n-kR]e^{-j\frac{2\pi mn}{N}} \tag{4.4}$$

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:

$$W[n] = 0.42 - 0.5\cos(\frac{2\pi n}{N-1}) + 0.08\cos(\frac{4\pi n}{N-1}) \tag{4.5}$$

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 (4.4):

$$P[k, m] = |X[k, m]|^2 \tag{4.6}$$

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:

$$P_{normalized}[k, m] = (\frac{P[k, m] - \min(P)}{\max(P) - \min(P)})^{\gamma} \tag{4.7}$$

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

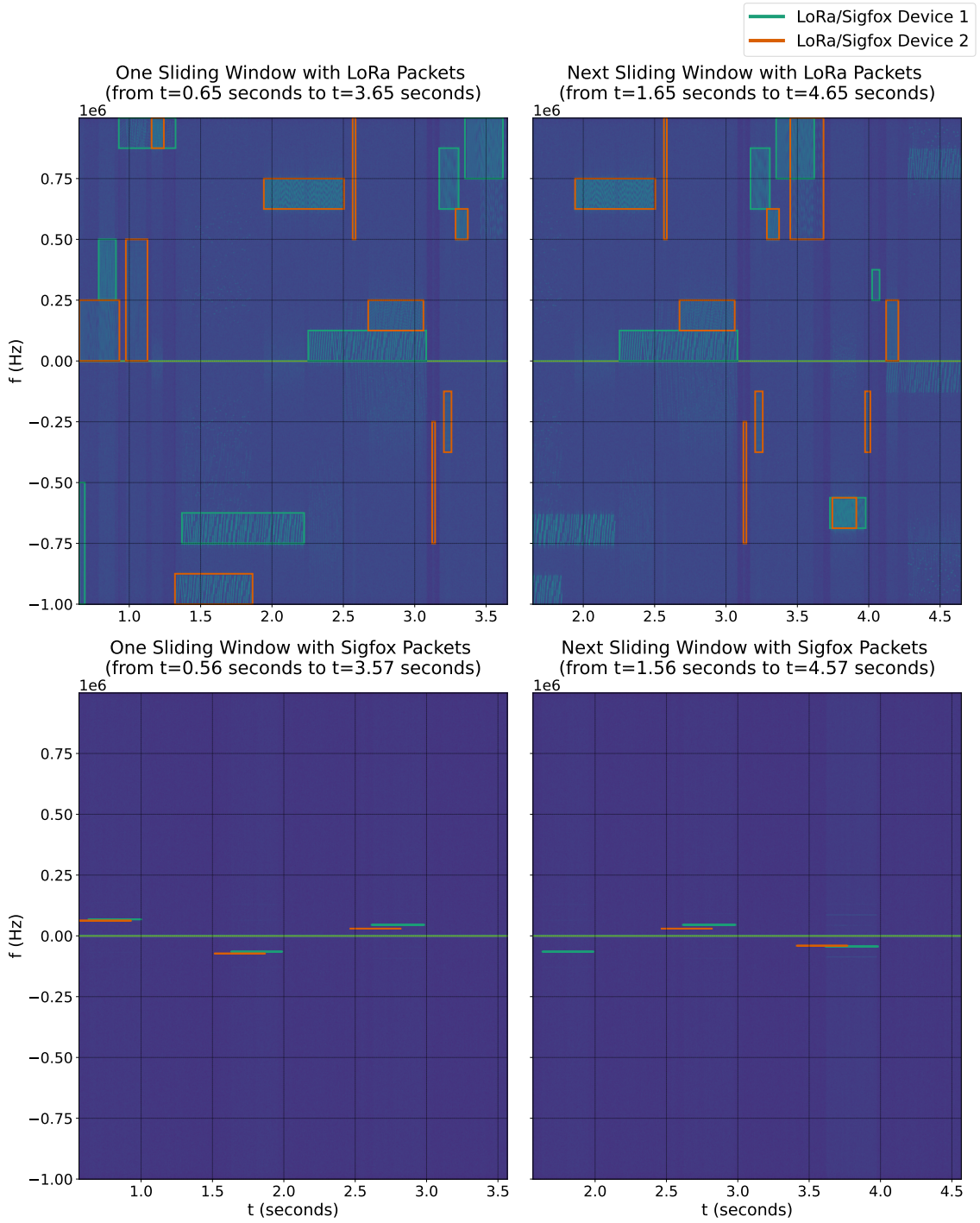

Figure 4.8: **Example of spectrogram sliding windows**

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:

$$SW[i,k,m] = P_{normalized}[k,m] \times rect(\frac{k-iS_{SW}}{\min(L_{SW},T-iS_{SW})}) \tag{4.8}$$

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 (4.9).

$$rect(n) = \begin{cases} 1, & \text{if } 0 \leq n < 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.9}$$

The number of sliding windows is given by:

$$Len_{SW} = \left\lceil \frac{T - L_{SW}}{S_{SW}} \right\rceil + 1 \tag{4.10}$$

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 Figure 4.8.

4.2.4 Modified Deformable DETR Model Design

DETR [53] allows performing object detection by utilizing the transformer model and attention mechanisms [54]. 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:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.11}$$

  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:

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) \tag{4.12}$$

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:

$$\begin{aligned} L_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \big[&- \log \hat{p}_{\hat{\sigma}(i)}(c_i) \\ &+ \mathbf{1}_{\{c_i \neq \emptyset\}} L_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) \big] \end{aligned} \tag{4.13}$$

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 [52], it further enhances the original DETR implementation by introducing deformable attention mechanisms, which was inspired by deformable convolution [61], 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:

$$\begin{aligned} DA(z_q, p_q, x) = \sum_{m=1}^{M} W_m \Bigg[&\sum_{k=1}^{K} A_{mqk} \\ &\cdot W'_m x(p_q + \Delta p_{mqk}) \Bigg] \end{aligned} \tag{4.14}$$

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:

$$MSDA(z_q, p_q, \{x^l\}_{l=1}^{L}) = \sum_{m=1}^{M} W_m \left[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot W'_m x^l(\phi_l(p_q + \Delta p_{mlqk})) \right] \tag{4.15}$$

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.

4.2.5 Resnet50 Backbone Modifications with Transfer Learning

In the original DETR [53] and Deformable DETR [52] implementations, the CNN backbone utilized the Resnet50 [62] model for feature extraction. Resnet50 is also used in some work as modulation pattern recognition [63], 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 greyscale 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 [64], we simply calculate the mean of the original 3-channels weights and apply them to the new 2-channels instead:

$$W_{new}[:,j,:,:] = \frac{1}{3}\sum_{i=0}^{2} W_{orig}[:,i,:,:], \quad \text{for } j \in \{0,1\} \tag{4.16}$$

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 subsection 4.2.7).

4.2.6 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. [65] 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 (4.14), 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 (4.14) as follows, we call this **Deformable Radial Attention**:

$$\begin{aligned} DRA(z_q, p_q, x, r) = \sum_{m=1}^{M} W_m \Bigg[\sum_{k=1}^{K} A_{mqk} \\ \cdot M_{mqk}(\Delta p_{mqk}, r) \cdot W'_m x(p_q + \Delta p_{mqk}) \Bigg] \end{aligned} \tag{4.17}$$

where,

$$M_{mqk}(\Delta p_{mqk}, r) = \begin{cases} 1 & \text{if } |(\Delta p_{mqk})_y| \leq r \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

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:

$$A'_{mqk} = \frac{A_{mqk} \cdot M_{mqk}(\Delta p_{mqk}, r)}{\sum_{k'=1}^{K} A_{mqk'} \cdot M_{mqk'}(\Delta p_{mqk'}, r)} \tag{4.19}$$

where, A'_{mqk}, is the renormalized attention weight Additionally, this modified algorithm can also work on $MSDA$ in (4.15), allowing the application of attention radius across all feature levels, we call this **Multi-scale Deformable Radial Attention (MSDRA)**:

$$\begin{aligned} MSDRA(z_q, p_q, \{x^l\}_{l=1}^{L}, r) = \sum_{m=1}^{M} W_m \Bigg[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \\ \cdot M_{mlqk}(\Delta p_{mlqk}, r) \cdot W'_m x^l(\phi_l(p_q + \Delta p_{mlqk})) \Bigg] \end{aligned} \tag{4.20}$$

where,

$$M_{mlqk}(\Delta p_{mlqk}, r) = \begin{cases} 1 & \text{if } |(\Delta p_{mlqk})_y| \leq r \\ 0 & \text{otherwise} \end{cases} \tag{4.21}$$

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

4.2.7 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-narrowband 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.

With this Modified Deformable DETR with MSDRA in mind, we can now perform LPWAN packets extraction by supplying continuous IQ data into the trained model

as multiple sliding windows and obtaining the bounding boxes (i.e. time-frequency locations) and the label of the LPWAN packets for subsequent preamble extraction engine.

4.3 Signal Filtering and Frequency Shifting

After we have obtained the time-frequency locations of the LPWAN packets, a band-pass FIR filter will be applied followed by a down-sampling process.

Formally, individual LPWAN packets IQ data can be represented as:

$$x[n,k] = \mathrm{Re}(x[n,k]) + j\,\mathrm{Im}(x[n,k]) = I[n,k] + jQ[n,k] \tag{4.22}$$

, where n is the LPWAN packet index, k is the IQ sample index in n and $I[n,k]$ and $Q[n,k]$ represent the in-phase and quadrature parts of n LPWAN packet, respectively. To decrease the data required for signal processing, while allowing higher frequency resolution in the subsequent steps, the overall sample rate of the IQ data will be reduced.

An FIR band-pass filter on the LPWAN packet n will first be designed:

$$h_n[m] = \sum_{k=-\infty}^{\infty} H(e^{j\omega})e^{j\omega m}d\omega \tag{4.23}$$

, where $h_n[m]$ is the impulse response of the FIR filter for specific LPWAN signal, and $H(e^{j\omega})$ is the desired frequency response that encompasses the bandwidth of the actual LPWAN signal. Then this filter is applied to the wide-band SDR baseband signal:

$$y[n,k] = \sum_{m=0}^{M-1} h[m]x[n,k-m] \tag{4.24}$$

, where $y[n,k]$ is the filtered output for packet n, $x[n,k]$ is the input signal, and M is the filter order.

An example of a LPWAN Packet before and after filtering can be observed in Figure 4.9 and Figure 4.10.

The filtered LPWAN signal is then shifted to 0Hz by multiplying the frequency offset:

$$z[n,k] = y[n,k] \cdot e^{-j2\pi f_c k/f_s} \tag{4.25}$$

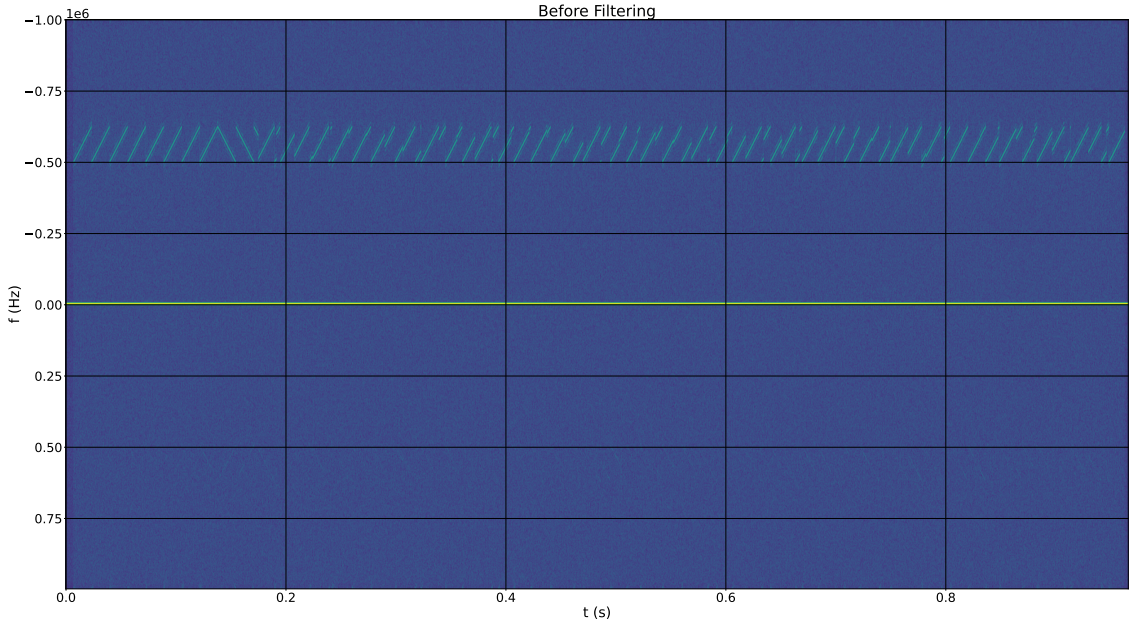

Figure 4.9: **LPWAN Packet Before Filtering**

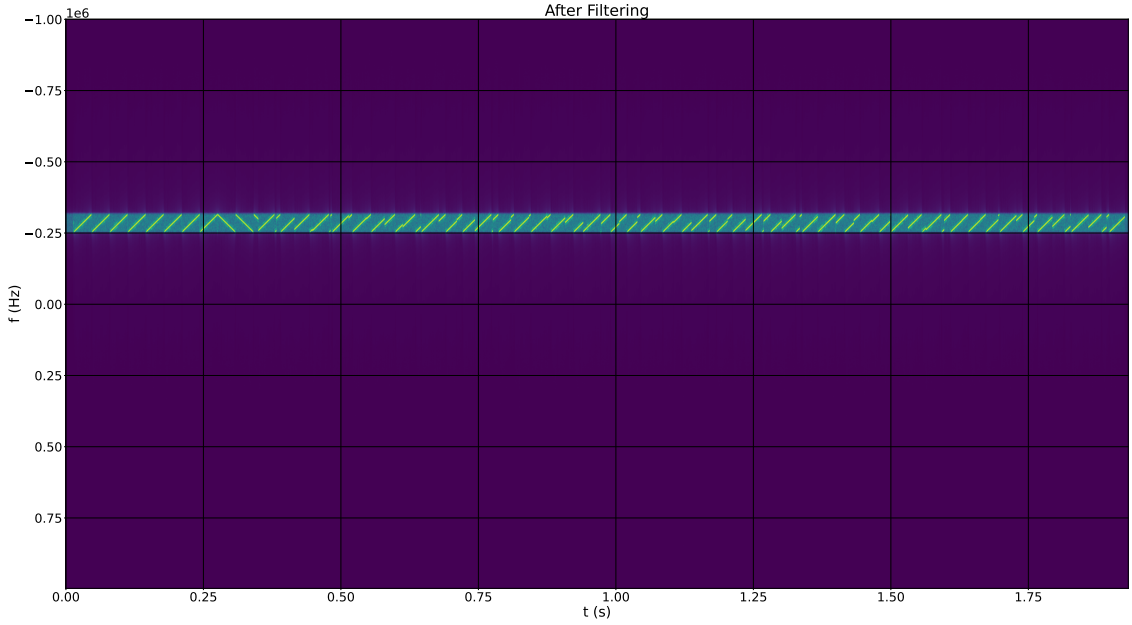

Figure 4.10: **LPWAN Packet After Filtering**

, where $z[n,k]$ is the new baseband signal, f_c is the center frequency of the LPWAN packet, and f_s is the original sampling rate. An example of shifting to 0Hz can be

seen in Figure 4.11.

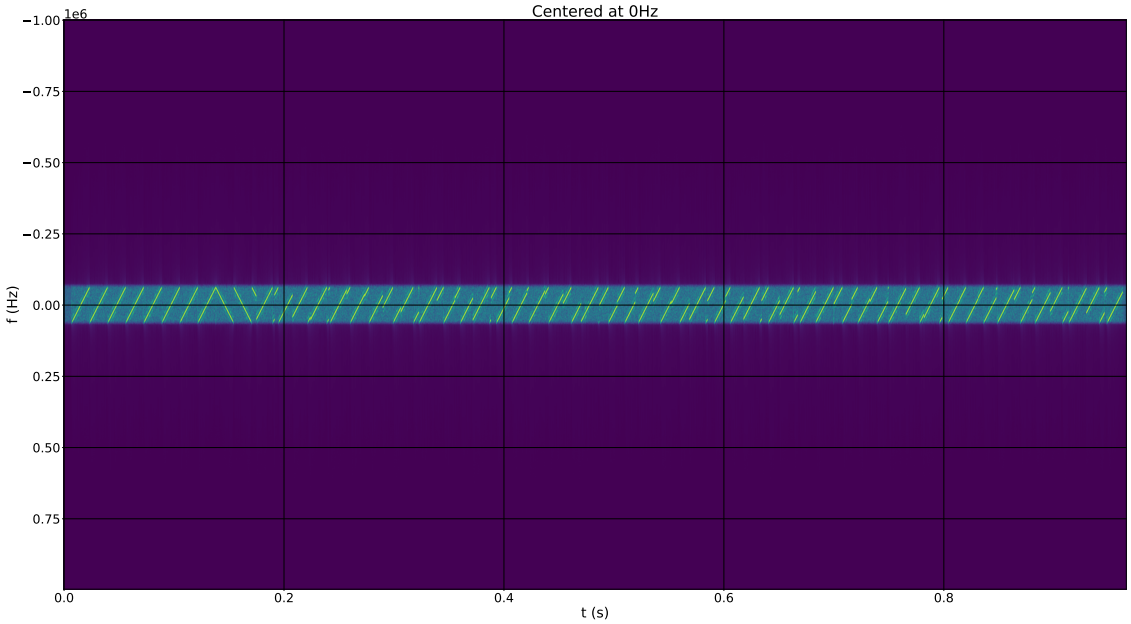

Figure 4.11: **LPWAN Packet Centered at 0Hz After Filtering**

4.4 Down-sampling

To reduce the sampling rate, which is subject to the Nyquist rate:

$$w[n, l] = z[n, kD] \tag{4.26}$$

, where $w[n, l]$ is the downsampled signal, D is the downsampling factor, and $l = \lfloor k/D \rfloor$. The new sampling rate is subjected to the Nyquist rate: $f_{s,new} = f_s/D \geq 2B$, where B is the actual bandwidth of the LPWAN signal, to satisfy the Nyquist criterion.

4.5 Signal Comparison and Extraction Engine

This engine performs actual LPWAN packets IQ data comparison and preamble extraction. To ascertain the consistency of the signal power of IQ data, Automatic Gain Control (AGC) is used by iteratively applying the following algorithm to all input IQ samples:

1. Apply current gain (initial gain is 1):

$$y[n] = G[n] \cdot x[n] \tag{4.27}$$

2. Calculate sample power:

$$|y[n]|^2 = \Re(y[n])^2 + \Im(y[n])^2 \tag{4.28}$$

3. Calculate error:

$$e[n] = |y[n]|^2 - R^2 \tag{4.29}$$

, where $R = 1$ is the reference magnitude.

4. Calculate scaled error:

$$e_{scaled}[n] = \frac{\alpha^2 \cdot e[n]}{\alpha^2 \cdot |y[n]|^2 + \epsilon} \tag{4.30}$$

, where α is the adaptation rate and $\epsilon = 1 \times 10^{-10}$ is a small constant to provide stability and to prevent division by zero.

5. Update gain:

$$G[n+1] = G[n] \cdot (1 - e_{scaled}[n]) \tag{4.31}$$

6. Limit gain:

$$G[n+1] = \text{clip}(G[n+1], G_{min}, G_{max}) \tag{4.32}$$

, where $G_{min} = 1 \times 10^{-6}$ and $G_{max} = 65536$ are the minimum and maximum allowable gains, respectively.

After AGC, Short-Time Fourier Transforms (STFTs) are taken to generate spectrograms of individual LPWAN packets. STFT operation on $x[n]$ (which is the down-sampled $w[n, l]$) to produce $X[k]$ can be expressed as:

$$X[k, m] = \sum_{n=kR}^{kR+N-1} x[n]W[n - kR]e^{-j\frac{2\pi mn}{N}} \tag{4.33}$$

, 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, Blackman window in $W[n - KR]$ is utilized, which is the same given by (4.5). After obtaining the STFT vectors in (4.33), to obtain a spectrogram, the magnitudes of the STFT bins are squared:

$$P[k, m] = |X[k, m]|^2 \tag{4.34}$$

Aiming further aligning the LPWAN signals, cross-correlation is performed for two LPWAN packet spectrograms:

$$R_{x_1x_2}[l] = \sum_{n=-\infty}^{\infty} x_1[n] \cdot x_2^*[n - l] \tag{4.35}$$

, where x_2^* denotes the complex conjugate of x_2. The peak magnitude of the output is used to align both IQ data:

$$l_{peak} = \arg\max_l |R_{x_1x_2}[l]| \tag{4.36}$$

The second LPWAN signals are aligned to the first by shifting it by l_{peak} samples:

$$x_2^{aligned}[n] = x_2[n + l_{peak}] \tag{4.37}$$

For each pair of LPWAN packets, cosine similarity is calculated along the aligned spectrogram to acquire the overall similarity metric along the frequency-time spectrogram plot: Let $P_1[k, m]$ and $P_2[k, m]$ be the aligned spectrograms of two LPWAN packets, where k represents the frequency bin and m represents the time frame. For a window of size $WinSz$, the cosine similarity at time index i is given by:

$$CS(i) = \frac{\sum_{k=0}^{K-1} \sum_{j=0}^{WinSz-1} P_1[k, i+j] \cdot P_2[k, i+j]}{\sqrt{\sum_{k=0}^{K-1} \sum_{j=0}^{WinSz-1} P_1[k, i+j]^2} \sqrt{\sum_{k=0}^{K-1} \sum_{j=0}^{WinSz-1} P_2[k, i+j]^2}} \tag{4.38}$$

, where K is the number of frequency bins, $WinSz$ is the window size, and i ranges from 0 to $M - WinSz$, with M being the total number of time frames. Finally, the localization of the preambles are executed according to a certain similarity threshold $T_{SimStart}$, T_{SimEnd} which denotes the threshold that considers the two LPWAN signals started to be similar and dissimilar, respectively, from the aligned spectrogram plot. To extract the preambles of two given LPWAN Packets, we simply extract by the sample index that satisfies $T_{SimStart}$, T_{SimEnd} from the **aligned** packets.

Chapter 5

Experimental Results and Discussions

5.1 Overview

In this chapter, we will demonstrate our experimental results and discussions. This will be divided into two parts. Firstly, we will introduce the performance metrics of the LPWAN packets extraction algorithm that uses our Modified Deformable DETR and MSDRA. Then, we will evaluate the performance of our signal comparison and preamble extraction engine.

5.2 Modified Deformable DETR with MSDRA

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.

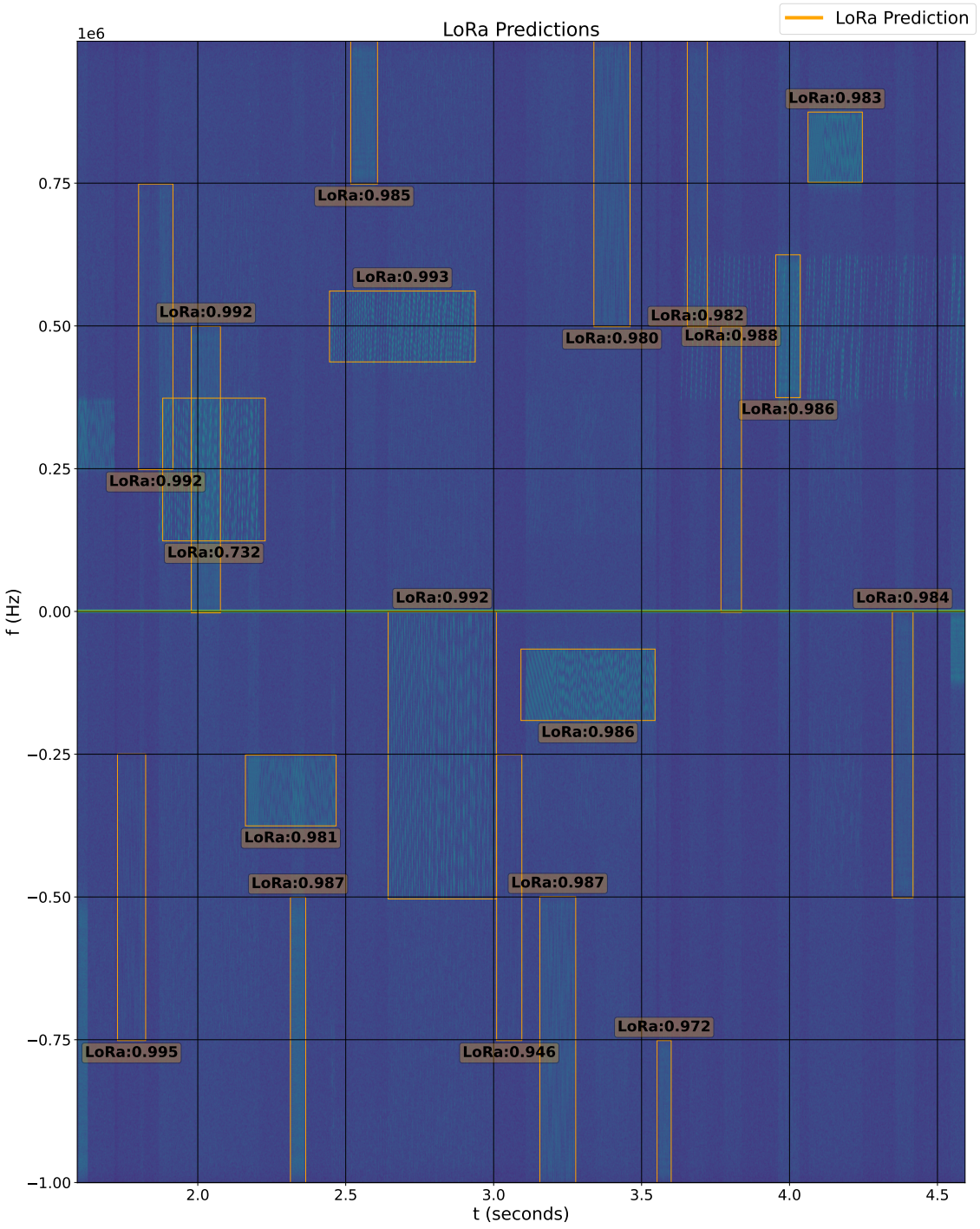

Figure 5.1: **Example of LoRa prediction**

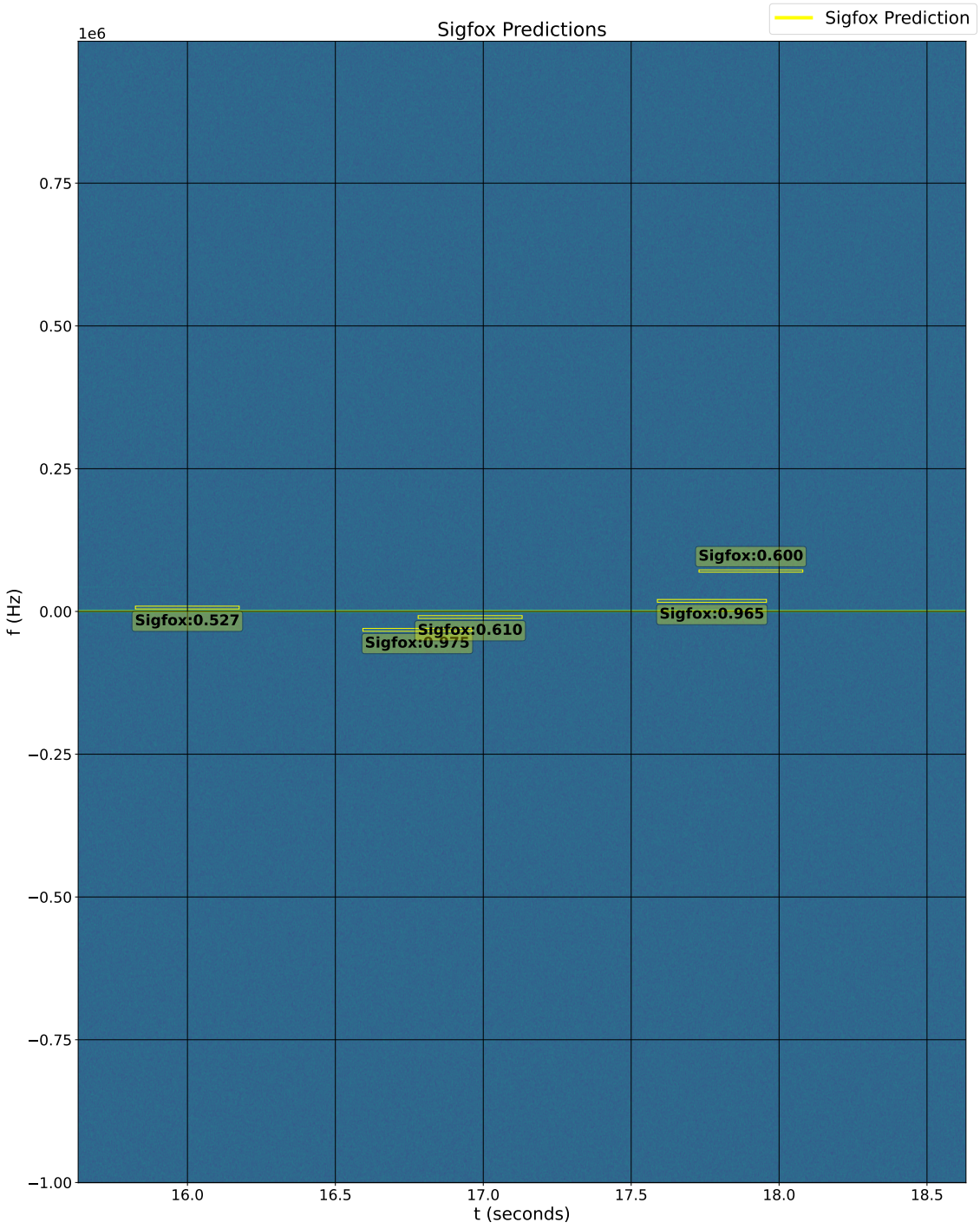

Figure 5.2: **Example of Sigfox prediction**

Figure 5.1 and Figure 5.2 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) [66] 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:

$$Y\text{-}IoU = \frac{|Y_p \cap Y_g|}{|Y_p \cup Y_g|} \tag{5.1}$$

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.

5.2.1 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 Figure 5.3, 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 Figure 5.4. In Figure 5.4, an interesting phenomenon can be ob-

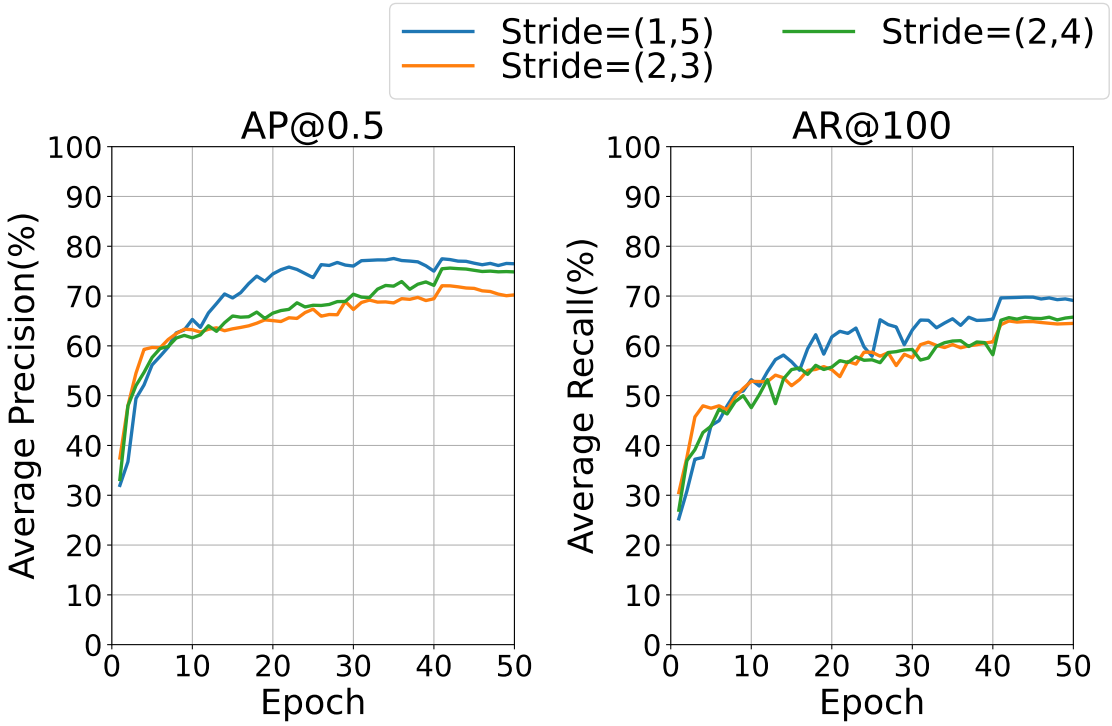

Figure 5.3: **Time-Frequency Localization Performance**

served, 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 in-

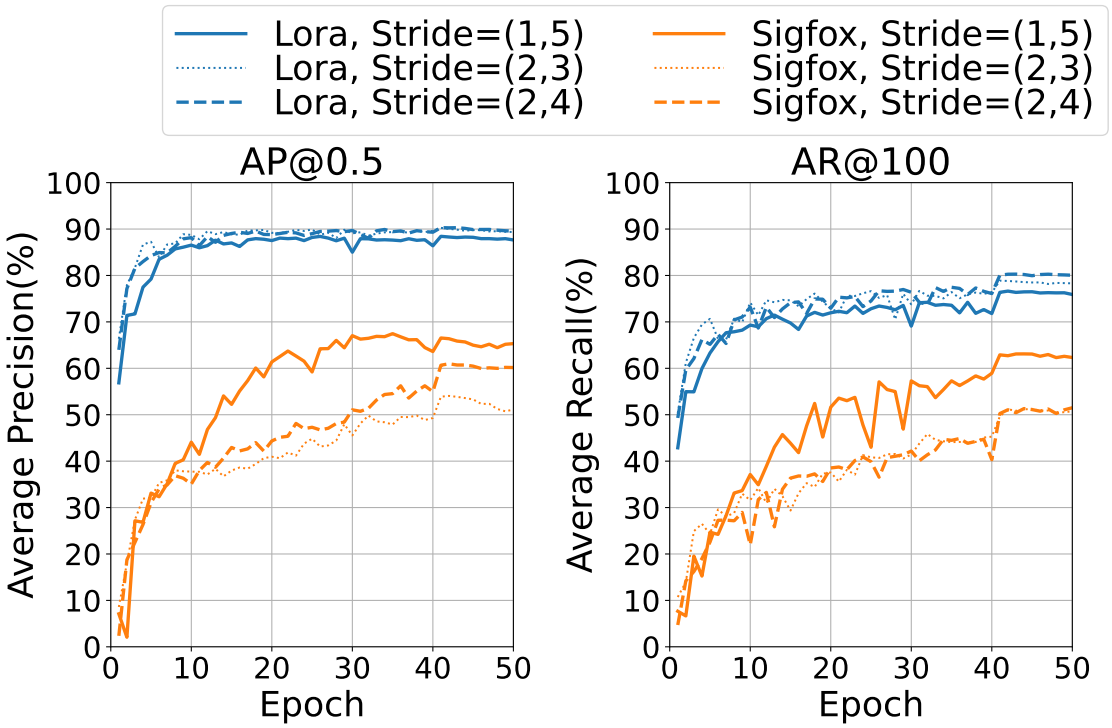

Figure 5.4: **Time-Frequency Localization Per-Class Performance**

crease 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 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.

5.2.2 Frequency Localization Performance

The introduction of Y-IoU in (5.1) 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 Figure 5.5 and Figure 5.6. Similar to the previous

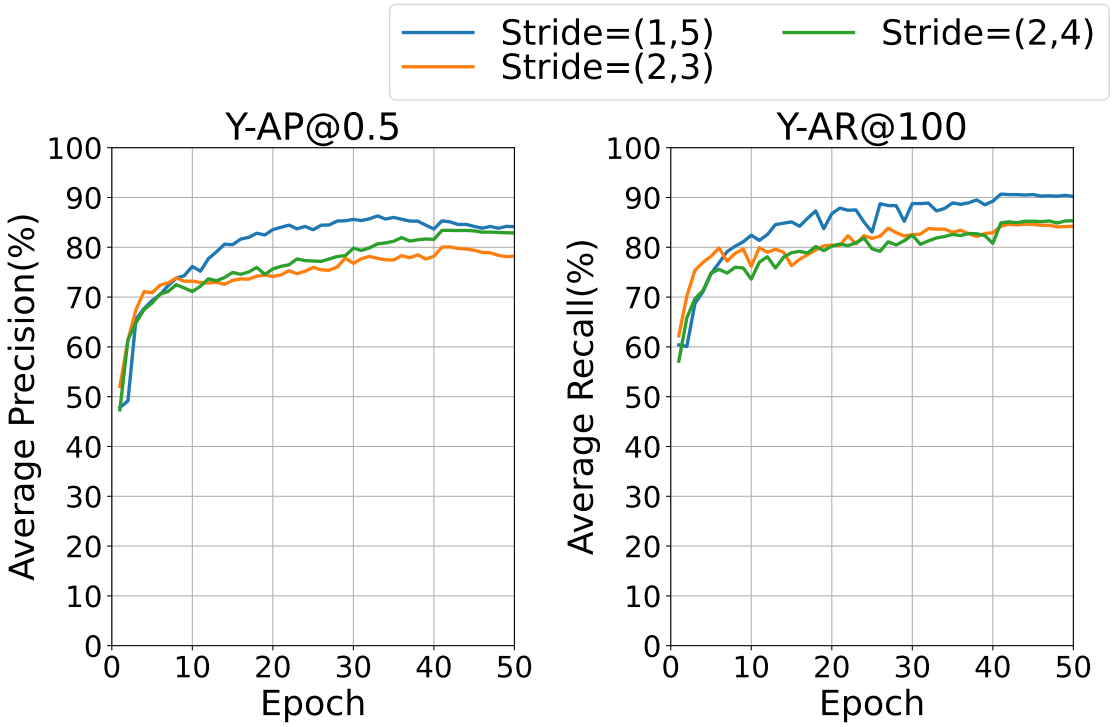

Figure 5.5: **Frequency Localization Performance**

subsection - subsection 5.2.1, 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

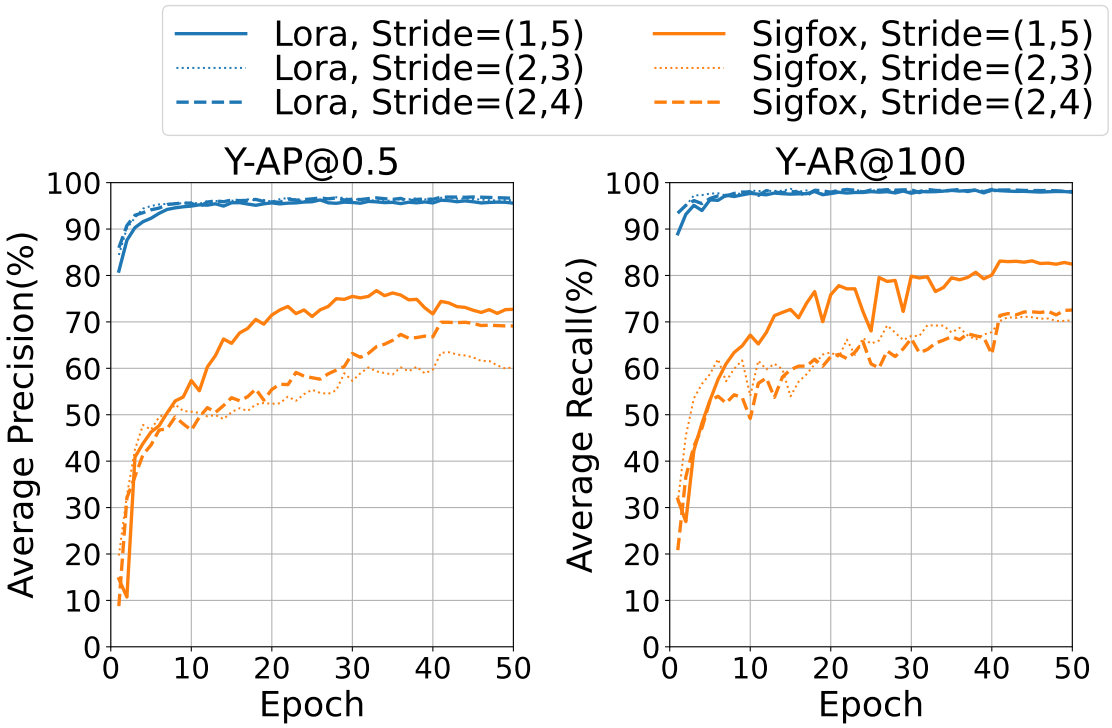

Figure 5.6: **Frequency Localization Per-Class Performance**

provides the feasibility of application in spectrum management for this model.

5.2.3 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 5.1. 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.

Table 5.1: **Comparison of AP values across different classes and noise levels**

Class	**Metric**	**AP of 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%

5.2.4 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 Figure 5.7. 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

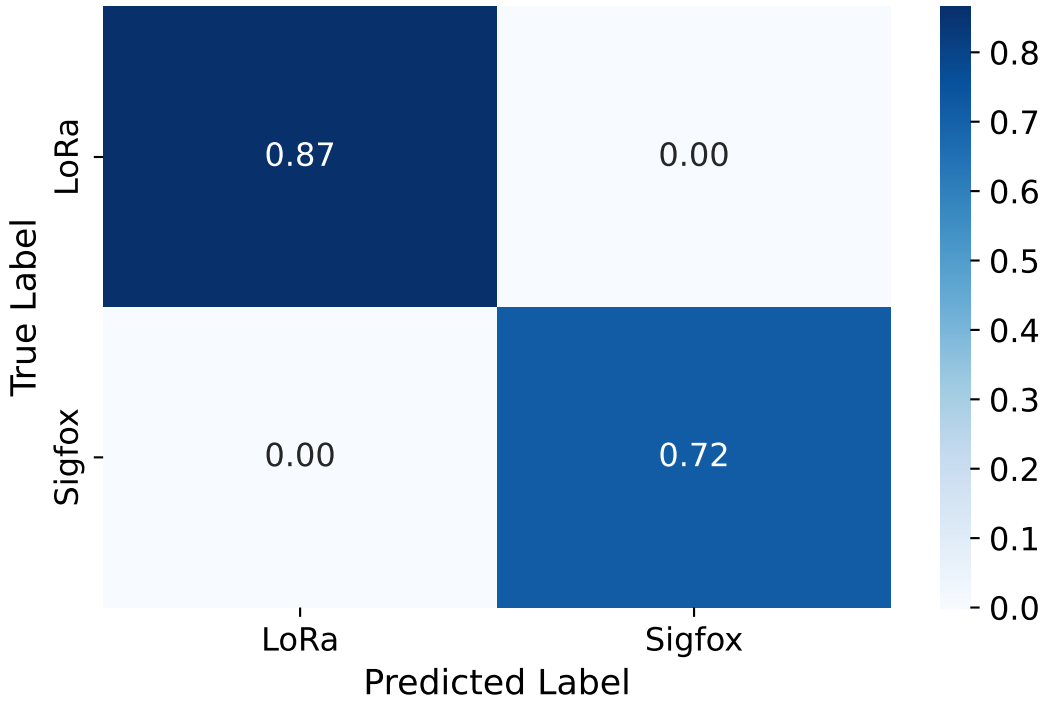

Figure 5.7: **Confusion matrix of the model for classification**

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 Figure 5.8.

5.2.5 Ablation Study

5.2.5.1 Multi-scale Deformable Radial Attention (MSDRA)

The modification of MSDA in (4.15) to MSDRA in (4.20) 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

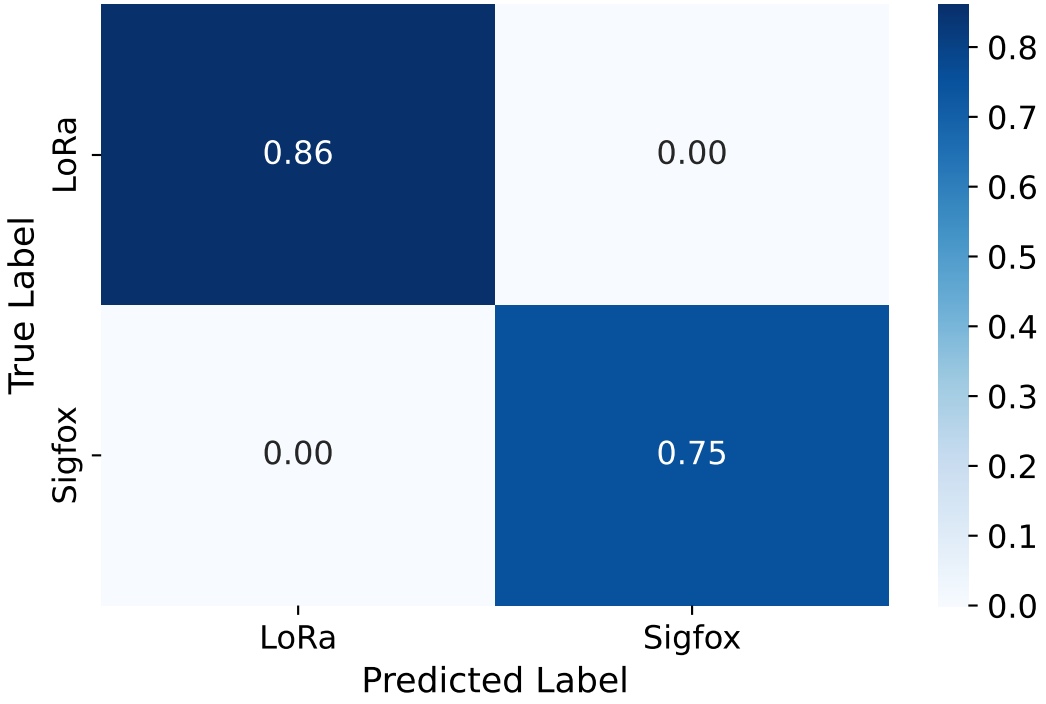

Figure 5.8: **Confusion matrix of the model for classification with** $N = 4096$

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 Figure 5.9, 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.

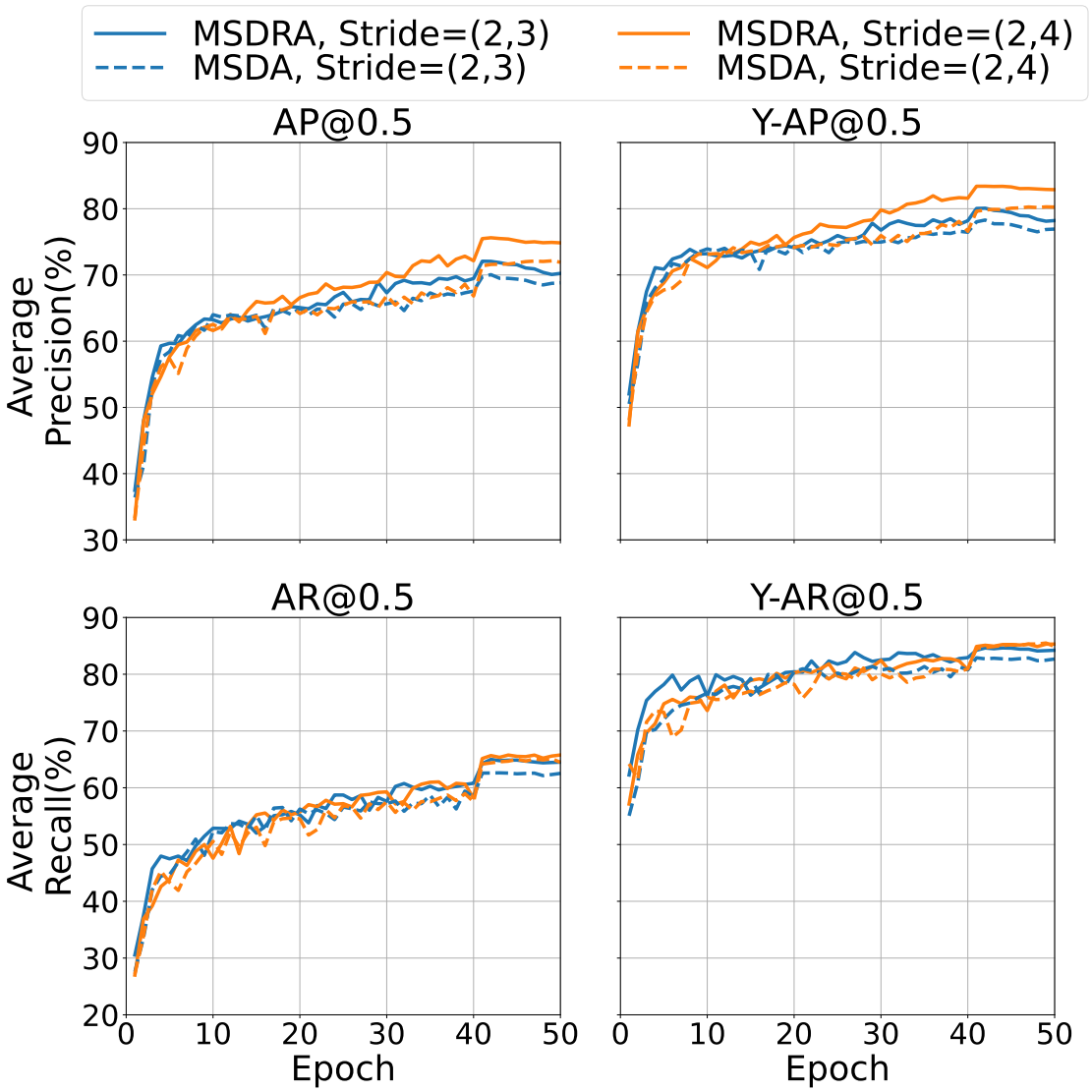

Figure 5.9: **Impact of AP when removing MSDRA**

5.2.5.2 Expanding Bandwidth Annotations

In subsection 4.2.7, 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 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 Figure 5.10. 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.

5.2.6 Visualization of Radial Attention

To better demonstrate the impact of the addition of radial constraints in (MS)DRA, Figure 5.11 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

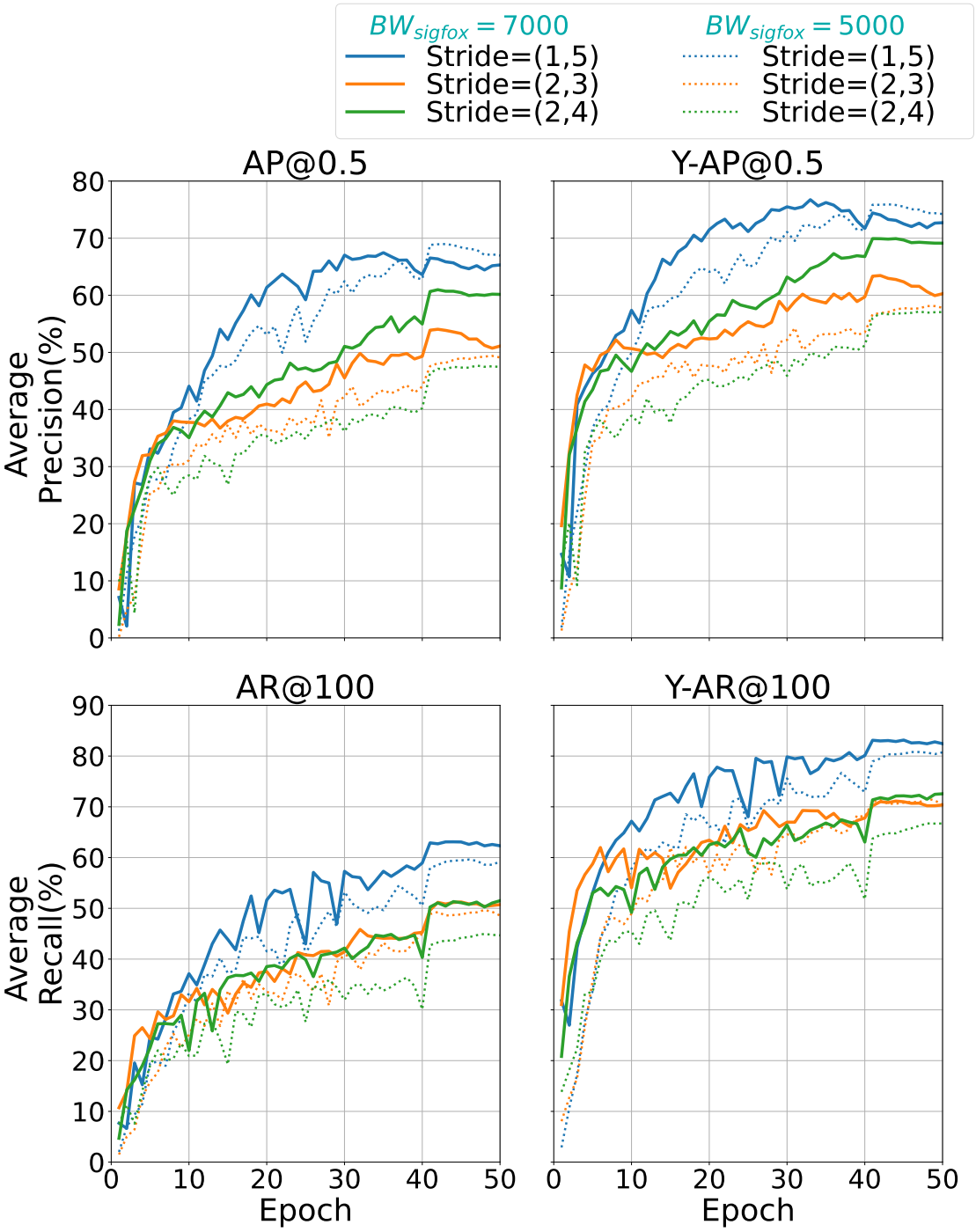

Figure 5.10: **Impact of AP when varying** BW_{sigfox}

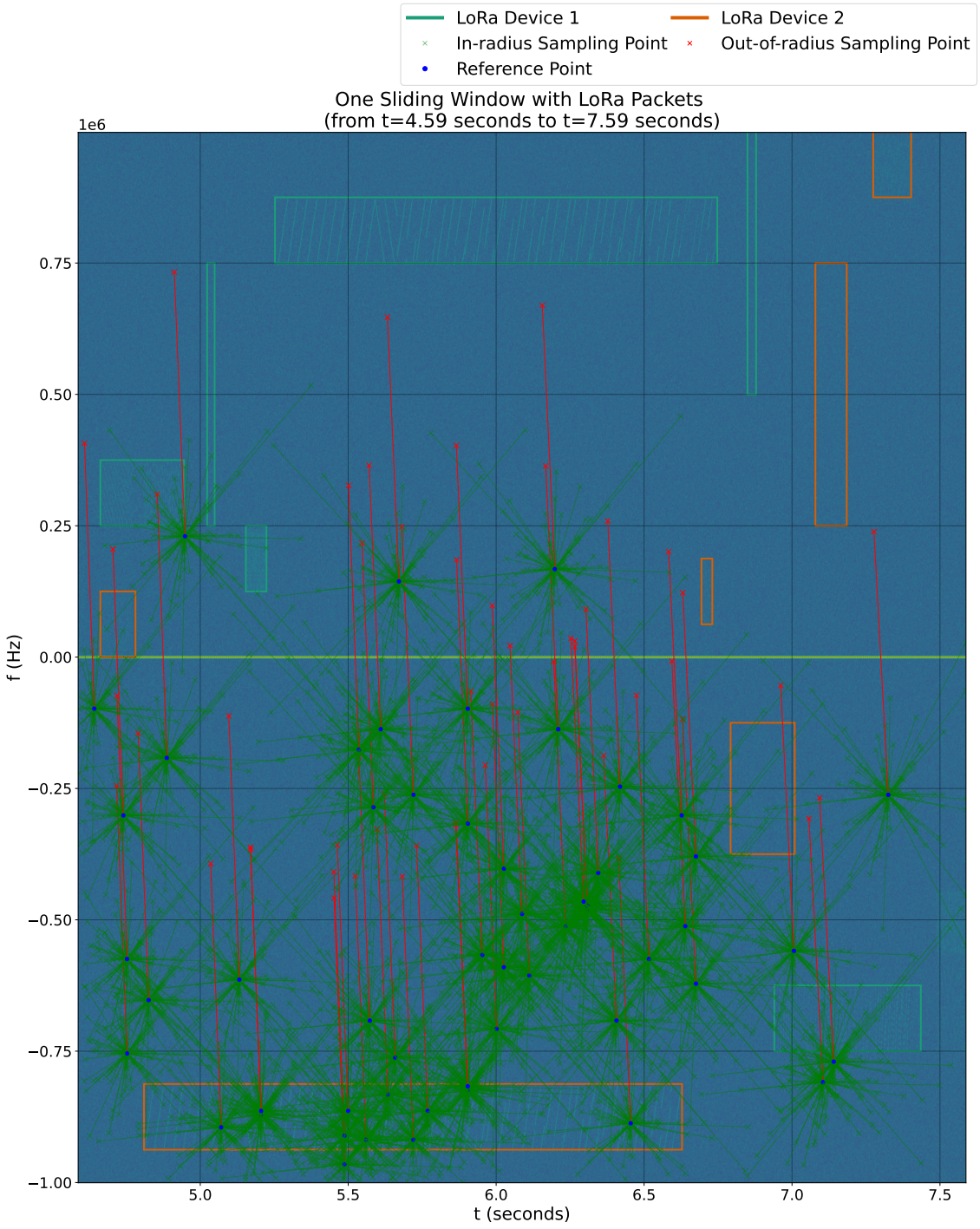

Figure 5.11: **Visualization of Attention Sampling Points**

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.

5.2.7 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 [52], the complexity of (MS)DRA can be given by:

$$\begin{aligned}\mathcal{O}(N_q C^2 + \min(HWC^2, N_q K C^2) + 5N_q KC + \\ 3N_q CMK + N_q K)\end{aligned} \tag{5.2}$$

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;
- 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.

5.2.8 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 Figure 5.12 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 subsection 4.2.2, without any additional tuning or adjustments.

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.

Figure 5.12: **Commercial LoRa Temperature Sensor for Real-World Inferencing**

Table 5.2: **Dataset description for each batch**

Bandwidth	Spreading Factor	Preamble Length (Symbols)	Number of Packets
125KHz	7	6	25
		12	25
	11	6	25
		12	25
250KHz	7	6	25
		12	25
	11	6	25
		12	25
500KHz	7	6	25
		12	25
	11	6	25
		12	25

5.3 Signal Comparison and Preamble Extraction Engine

To evaluate the performance of the proposed signal comparison and preamble extraction system, we have generated 10 batches of fully annotated LoRa signal datasets, with each batch containing 300 LoRa packets varied in transmission power, frequency, signal bandwidth, spreading factor, random data payload, and most importantly, preamble lengths. Table 5.2 below describes the datasets for evaluation: Here are the parameters supplied to the algorithm:

- AGC $\alpha = 1 \times 10^{-8}$
- Cosine similarity matching window size $WinSz = 32$
- Cosine similarity similar thresholds $T_{SimStart} = 0.35$, $T_{SimEnd} = 0.2$

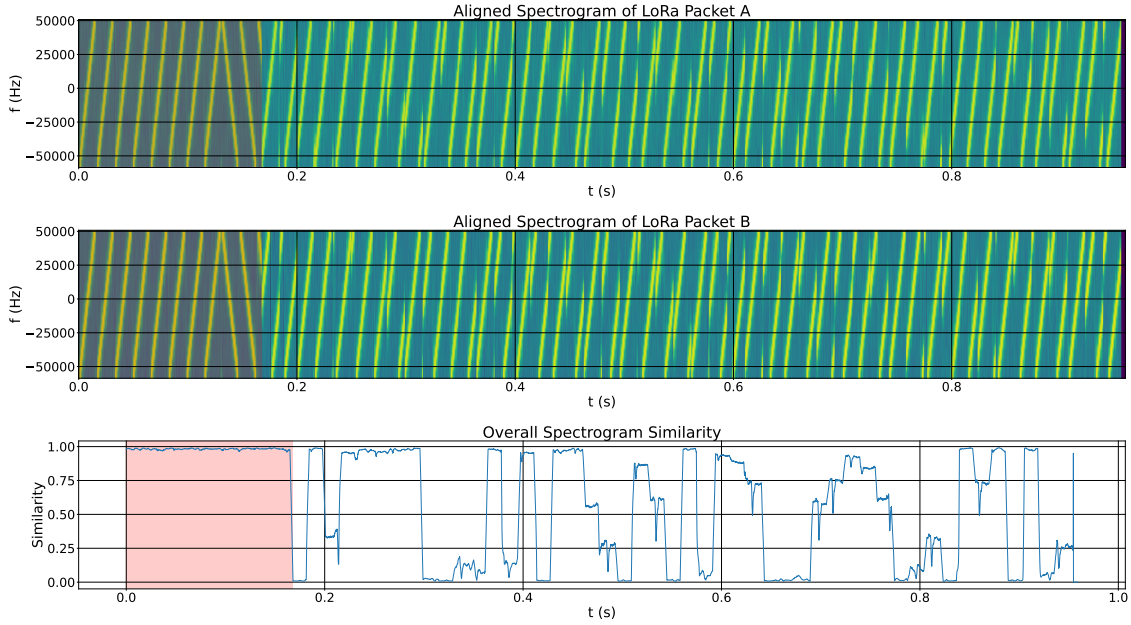

Figure 5.13: **Comparison and Extraction Example (BW=125KHz, SF=11, Preamble Length=6), areas in red indicates detected preamble locations**

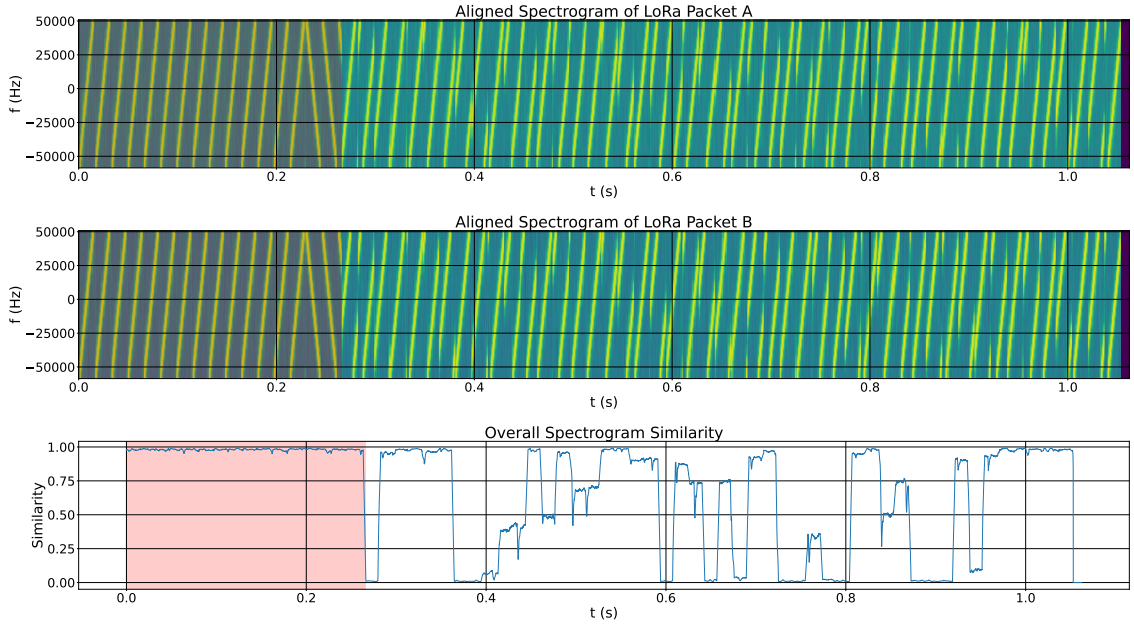

Figure 5.14: **Comparison and Extraction Example (BW=125KHz, SF=11, Preamble Length=12), areas in red indicates detected preamble locations**

All of the LPWAN packets of the same class are supplied to the system for comparison, resulting in 300 comparisons for each class. To evaluate the performance of the system, we calculated the Intersection over Union (IoU) against the preamble range of the ground truth of the actual common prefix of the LPWAN signals, which is given by:

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} \tag{5.3}$$

,where P denotes the prediction interval, G the ground truth interval, $|\cdot|$ denotes the length of the interval. Two IoU thresholds are selected $t \in \{0.5, 0.75\}$, and their Average Precision (AP) is calculated by:

$$\text{AP} = \sum_{k=1}^{N} P(k) \cdot \Delta R(k) \tag{5.4}$$

where:

- N is the total number of predictions.
- $P(k) = \frac{\sum_{i=1}^{k} \mathbb{1}[\text{IoU}_i \geq t]}{k}$ is the precision at position k.
- $R(k) = \frac{\sum_{i=1}^{k} \mathbb{1}[\text{IoU}_i \geq t]}{N}$ is the recall at position k.
- $\Delta R(k) = R(k) - R(k-1)$ is the change in recall between position k and $k-1$.
- $\mathbb{1}[\text{IoU}_i \geq t] = 1$ if $\text{IoU}_i \geq t$, 0 otherwise.

For each signal class, the AP is given in Table 5.3. Example comparison and extraction results can be found in Figure 5.13 and Figure 5.14. For narrow-band signals, AP of above 95.6% can be observed, especially for longer preamble lengths and spreading factors because of more time-axis features. If the signal is higher in bandwidth (i.e. 500KHz), there is a drop in AP which indicates the frequency-axis features do not benefit from the increasing amount of data. This is anticipated as larger bandwidth signals result in higher noise power (AWGN in this case), which reduces correlation. Moreover, these signals could also dilute the usable features as a fixed number of

Table 5.3: **AP of each classes**

Bandwidth	**Spreading Factor**	**Preamble Length (Symbols)**	**AP@0.5**	**AP@0.75**
125KHz	7	6	95.67%	75.33%
		12	98.33%	84.00%
	11	6	100.00%	96.33%
		12	100.00%	96.33%
250KHz	7	6	97.67%	79.67%
		12	99.33%	85.67%
	11	6	99.67%	94.67%
		12	100.00%	94.67%
500KHz	7	6	74.00%	56.00%
		12	78.33%	64.33%
	11	6	89.00%	71.67%
		12	83.67%	73.67%

STFT bins is used in correlation. In general, this algorithm shows promising results in localizing and extracting LPWAN preambles that bear different protocol parameters, bandwidth, and preamble lengths.

Chapter 6

Future Work

There are limitations of current work that can be improved in the future.

- **Phase Information:** Currently, we have only utilized phase information to perform LPWAN IoTWP packets extraction. For our signal comparison and preamble extraction engine, we assume frequency-varying signals. This is due to the complexity of the algorithm, in which, for phase modulated signals (e.g. ultra-narrow band Sigfox signals), additional multiple blocks have to incorporated such as coarse and fine frequency offset estimation, phase offset estimation with costas loop, and with a lot of prior knowledge, including the symbol rate of the target signals, which contradicts with our assumptions that requires as minimum prior knowledge as possible. It is possible to train another Machine Learning model for such tasks, but this would change our engine requiring prelabelled datasets, which undermine the flexibility of the system.
- **Performance Improvement:** The Modified Deformable DETR works great on LoRa signals with over 90% accuracy. However, for Sigfox signals that are ultra-narrowband, we can only obtain near 70% accuracy. As demonstrated in the experimental results section above, we can optimize the model or modify the existing backbone of the model to learn finer details for ultra-narrowband sig-

nals. However, doing so may lose the ability of the application transfer learning if the model architecture has been modified tremendously. In the future, instead of using the transformer model and attention mechanisms, we may explore various machine learning techniques to build a light-weight model to perform these LPWAN signal classification and time-frequency localization tasks. For the signal comparison and preamble extraction engine, we may further increase the AP of the algorithm for example, by utilizing multi-scale architecture similar in MSDRA and compare multiple features, at the expense of computational costs.

- **Diversify Our Training and Testing Set:** Although we have considered LoRa protocols with multiple preamble lengths, spreading factors, bandwidths, packet lengths and transmission frequencies, we hope to incorporate more LPWAN IoTWPs. This can further build a database containing multiple LPWAN IoTWPs for signal identification purposes and reverse engineering intentions.
- **Testing Preambles on Attacks and Defenses:** The goal of performing LPWAN signal detection and the subsequent preamble extractions is to support preamble based attacks and defenses. In the future, we could expand our work by utilizing the output of the preamble locations and the extracted preambles to perform these kinds of novel attacks and defenses.
- **Alternate Learning Algorithms:** We presented a (supervised) machine learning-based algorithm for LPWAN IoTWPs technology classification and time-frequency localization. In the future, we plan to look into light-weight unsupervised learning techniques and compare their performance with the current approach. Likewise, we also proposed an algorithm to perform preamble localization and extraction, in which in the future it might also be worth comparing its performance against machine-learning based algorithms, although this might require annotated preamble location datasets for training.

Chapter 7

Conclusion

In this thesis, we devised a system that contains a signal comparison and preamble extraction engine that allows preamble localization and extraction on arbitrary frequency-varying LPWAN IoTWP signals. The engine itself does not require any labeled training datasets, in which we utilized methods involving Short-Time Fourier Transform (STFT), Spectrograms, Cross-Correlation (XCorr) and cosine similarity techniques to extract common prefixes in LPWAN packets. With these preambles localized and extracted, it enables preamble-based IoT wireless protocols attacks and defenses. We experimented on multiple frequencies, bandwidth, and protocol parameters of LoRa packets and discovered that over 95.6% AP can be achieved with narrow-band signals. In the future, we plan to extend the algorithm to more real-world LPWAN technologies, especially those that are not frequency-varying in nature, like Phase-Shift Keying (PSK). We believe that the existing phase information inside the spectrogram can be obtained and used against preamble extraction, given that the original phase and frequency offset are correctly compensated. We will also explore a hybrid lightweight ML approach that combines the current algorithm to further enhance this system, especially for signals that are larger in bandwidth.

In addition, before feeding LPWAN Packets into the signal comparison and preamble

extraction engine, LPWAN Packets have to be classified and time-frequency localized. Without hand-crafting specific models, we have also proposed a customized Deformable DETR-based model with a custom deformable attention module with radial constraint - Multi Scale Deformable Radial Attention (MSDRA) to classify LPWAN technologies with time-frequency localization. The custom modifications essentially converted the original purpose of the model for image detection 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, we ultimately plan to propose a low-cost federated LPWAN signal detection network on the edge.

References

[1] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of things market analysis forecasts, 2020-2030," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 449–453.

[2] F.-C. T. M. and F.-L. P., "A review on the use of blockchain for the internet of things," *IEEE Access*, vol. 6, pp. 32 979–33 001, 2018.

[3] S. Mahmood, "Review of internet of things in different sectors: recent advances, technologies, and challenges," *Journal on Internet of Things*, vol. 3, no. 1, p. 19, 2021.

[4] F.-C. T. M. and F.-L. P., "A review on the use of blockchain for the internet of things," *IEEE Access*, vol. 6, pp. 32 979–33 001, 2018.

[5] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[6] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804517301455

[7] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[8] L. Tang and H. Hu, "Ohea: Secure data aggregation in wireless sensor networks against untrusted sensors," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1425–1434.

[9] R. Li, H. Hu, and Q. Ye, "Rftrack: Stealthy location inference and tracking attack on wi-fi devices," *IEEE Transactions on Information Forensics and Security*, 2024.

[10] Z. Chen, H. Hu, and J. Yu, "Privacy-preserving large-scale location monitoring using bluetooth low energy," in *2015 11th international conference on mobile ad-hoc and sensor networks (MSN)*. IEEE, 2015, pp. 69–78.

[11] H. Zheng and H. Hu, "Missile: A system of mobile inertial sensor-based sensitive indoor location eavesdropping," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3137–3151, 2019.

[12] 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.

[13] "Infected iot device numbers grow 100% in a year," Oct 2020, accessed: 2021-08-20. [Online]. Available: https://www.securitymagazine.com/articles/93731-infected-iot-device-numbers-grow-100-in-a-year

[14] R. Krejčí, O. Hujňák, and M. Švepeš, "Security survey of the iot wireless protocols," in *2017 25th Telecommunication Forum (TELFOR)*, 2017, pp. 1–4.

[15] Y. S. Lee and Y. O. Park, "Ber performance of agc in high-speed portable internet system," in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, vol. 7, 2004, pp. 4794–4797 Vol. 7.

[16] H. Labiod, H. Afifi, and C. D. Santis, *WI-FI ™, BLUETOOTH ™, ZIGBEE ™ AND WIMAX ™*. Dordrecht: Springer Netherlands, 2007, appendix A.

[17] BluetoothSIG, *Specification of the Bluetooth System, Core Version 4.0*. Bluetooth SIG, 2010, volume 2. [Online]. Available: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737

[18] "Lora alliance," Jun 2024, accessed on Jul 24, 2024. [Online]. Available: https://lora-alliance.org/

[19] "Sigfox 0g technology," Jul 2024, accessed on Jul 24, 2024. [Online]. Available: https://www.sigfox.com/

[20] G. Ferré and E. P. Simon, "An introduction to Sigfox and LoRa PHY and MAC layers," Apr. 2018, working paper or preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01774080

[21] H. Pirayesh and H. Zeng, "Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 24, no. 2, pp. 767–809, 2022.

[22] H. Rahbari, M. Krunz, and L. Lazos, "Swift jamming attack on frequency offset estimation: The achilles' heel of ofdm systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1264–1278, 2015.

[23] F. K. Jondral, "Software-defined radio: Basics and evolution to cognitive radio," *EURASIP J. Wirel. Commun. Netw.*, vol. 2005, no. 3, p. 275–283, Aug. 2005. [Online]. Available: https://doi.org/10.1155/WCN.2005.275

[24] G. Baldini, T. Sturman, A. R. Biswas, R. Leschhorn, G. Godor, and M. Street, "Security aspects in software defined radio and cognitive radio networks: A survey and a way ahead," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, pp. 355–379, 2012.

[25] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE journal on selected areas in communications*, vol. 23, no. 2, pp. 201–220, 2005.

[26] O. Ureten and N. Serinken, "Bayesian detection of wi-fi transmitter rf fingerprints," *Electronics Letters*, vol. 41, pp. 373–374(1), March 2005. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/el_20057769

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

[28] G. Reus-Muns and K. R. Chowdhury, "Classifying uavs with proprietary waveforms via preamble feature extraction and federated learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 6279–6290, 2021.

[29] J. Pohl and A. Noack, "Automatic wireless protocol reverse engineering," in *13th USENIX Workshop on Offensive Technologies (WOOT 19)*. Santa Clara, CA: USENIX Association, Aug. 2019. [Online]. Available: https://www.usenix.org/conference/woot19/presentation/pohl

[30] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, "Short paper: Reactive jamming in wireless networks: How realistic is the threat?" in *Proceedings of the Fourth ACM Conference on Wireless Network Security*, ser. WiSec '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 47–52. [Online]. Available: https://doi.org/10.1145/1998412.1998422

[31] O. Puñal, C. Pereira, A. Aguiar, and J. Gross, "Experimental characterization and modeling of rf jamming attacks on vanets," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 524–540, 2015.

[32] M. J. L. Pan, T. C. Clancy, and R. W. McGwier, "Jamming attacks against ofdm timing synchronization and signal acquisition," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, 2012, pp. 1–7.

[33] D. Nguyen, C. Sahin, B. Shishkin, N. Kandasamy, and K. R. Dandekar, "A real-time and protocol-aware reactive jamming framework built on software-defined radios," in *Proceedings of the 2014 ACM Workshop on Software Radio Implementation Forum*, ser. SRIF '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 15–22. [Online]. Available: https://doi-org.ezproxy.lb.polyu.edu.hk/10.1145/2627788.2627798

[34] E. Blossom, "Gnu radio: tools for exploring the radio frequency spectrum," *Linux journal*, vol. 2004, no. 122, p. 4, 2004.

[35] M. Samir, M. Kowalski, S. Zhou, and Z. Shi, "An experimental study of effective jamming in underwater acoustic links," in *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, 2014, pp. 737–742.

[36] M. Zuba, Z. Shi, Z. Peng, and J.-H. Cui, "Launching denial-of-service jamming attacks in underwater sensor networks," in *Proceedings of the Sixth ACM International Workshop on Underwater Networks*, ser. WUWNet '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi-org.ezproxy.lb.polyu.edu.hk/10.1145/2076569.2076581

[37] J. Padilla, P. Padilla, J. Valenzuela-Valdés, J. Ramírez, and J. Górriz, "Rf fingerprint measurements for the identification of devices in wireless communication networks based on feature reduction and subspace transformation," *Measurement*, vol. 58, pp. 468–475, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224114003893

[38] X. Wu, Y. Jiang, and A. Hu, "Lora devices identification based on differential constellation trace figure," in *Artificial Intelligence and Security*, X. Sun,

J. Wang, and E. Bertino, Eds. Cham: Springer International Publishing, 2020, pp. 658–669.

[39] B. W. Ramsey, B. E. Mullins, M. A. Temple, and M. R. Grimaila, "Wireless intrusion detection and device fingerprinting through preamble manipulation," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 585–596, 2014.

[40] M. Leonardi and D. Di Fausto, "Ads-b signal signature extraction for intrusion detection in the air traffic surveillance system," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2564–2568.

[41] M. Köse, S. Taşcioğlu, and Z. Telatar, "Rf fingerprinting of iot devices based on transient energy spectrum," *IEEE Access*, vol. 7, pp. 18 715–18 726, 2019.

[42] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *2009 International Conference on Information Processing in Sensor Networks*. IEEE, 2009, pp. 25–36.

[43] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based rf fingerprinting to enhance wireless network security," *Journal of Communications and Networks*, vol. 11, no. 6, pp. 544–555, 2009.

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

[45] M. D. Williams, S. A. Munns, M. A. Temple, and M. J. Mendenhall, "Rf-dna fingerprinting for airport wimax communications security," in *2010 Fourth International Conference on Network and System Security*. IEEE, 2010, pp. 32–39.

[46] A. Bouzegzi, P. Ciblat, and P. Jallon, "Maximum likelihood based methods for ofdm intercarrier spacing characterization," in *2008 IEEE 19th International*

Symposium on Personal, Indoor and Mobile Radio Communications. IEEE, 2008, pp. 1–5.

[47] 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 Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 529–541, 2021.

[48] 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.

[49] B. Li, W. Huang, W. Wang, and Q. Wang, "Spectrum painting for on-device signal classification," in *2024 IEEE 25th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2024, pp. 229–238.

[50] A. Almohamad, M. Hasna, S. Althunibat, K. Tekbıyık, and K. Qaraqe, "A deep learning model for lora signals classification using cyclostationay features," in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2021, pp. 76–81.

[51] A. Shahid, J. Fontaine, M. Camelo, J. Haxhibeqiri, M. Saelens, Z. Khan, I. Moerman, and E. De Poorter, "A convolutional neural network approach for classification of lpwan technologies: Sigfox, lora and ieee 802.15. 4g," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2019, pp. 1–8.

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

[53] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[55] 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 *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, 2024, pp. 1–6.

[56] 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 Transactions on Wireless Communications*, vol. 23, no. 11, pp. 17 129–17 142, 2024.

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

[58] M. Hussain, J. J. Bird, and D. R. Faria, "A study on cnn transfer learning for image classification," in *Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK*. Springer, 2019, pp. 191–202.

[59] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks: 17th International Conference, EANN 2016, Aberdeen, UK, September 2-5, 2016, Proceedings 17*. Springer, 2016, pp. 213–226.

[60] Y. Jiang, L. Peng, A. Hu, S. Wang, Y. Huang, and L. Zhang, "Physical layer identification of lora devices using constellation trace figure," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, pp. 1–11, 2019.

[61] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[63] X. Tian and C. Chen, "Modulation pattern recognition based on resnet50 neural network," in *2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP)*. IEEE, 2019, pp. 34–38.

[64] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan, "Pytorch," *Programming with TensorFlow: solution for edge computing applications*, pp. 87–104, 2021.

[65] ITU Secretariat, "The radio regulations volume 1, edition of 2020," International Telecommunication Union, Geneva, CH, Regulations, 2020, accessed on Jul 24, 2024. [Online]. Available: http://handle.itu.int/11.1002/pub/814b0c44-en

[66] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.