ICARUS: Learning on IQ and Cycle Frequencies for Detecting Anomalous RF Underlay Signals

Debashri Roy*, Vini Chaudhury*, Chinenye Tassie*, Chad Spooner†, and Kaushik Chowdhury*

*Institute for the Wireless Internet of Things, Northeastern University and †NorthWest Research Associates
Email: {droy, vchaudhury, ctassie}@ece.neu.edu, cmspooner@nwra.com, krc@ece.neu.edu

***Abstract*—The RF environment in a secure space can be compromised by intentional transmissions of hard-to-detect underlay signals that overlap with a high-power baseline transmission. Specifically, we consider the case where a direct sequence spread spectrum (DSSS) signal is the underlay signal hiding within a baseline 4G Long-Term Evolution (LTE) signal. As compared to overt actions like jamming, the DSSS signal allows the LTE signal to be decodable, which makes it hard to detect. ICARUS presents a machine learning based framework that offers choices at the physical layer for inference with inputs of (i) in-phase and quadrature (IQ) samples only, (ii) cycle-frequency features obtained via cyclostationary signal processing (CSP), and (iii) fusion of both, to detect the underlay DSSS signal and its modulation type within LTE frames. ICARUS chooses the best inference method considering both the expected accuracy and the computational overhead. ICARUS is rigorously validated on multiple real-world datasets that include signals captured in cellular bands in the wild and the NSF POWDER testbed for advanced wireless research (PAWR). Results reveal that ICARUS can detect DSSS anomalies and its modulation scheme with 98-100% and $67-99\%$ accuracy, respectively, while completing inference within $3-40$ milliseconds on an NVIDIA A100 GPU platform.**

***Index Terms*—Anomaly detection, LTE, DSSS, IQ, CSP.**

I. Introduction

The presence of unauthorized wireless signals in secure spaces constitutes a security risk, and several examples exist where private information was transmitted to external entities [1], [2]. It becomes hard to detect a signal when it hides (henceforth referred to as an *anomaly*) within a preexisting and stronger standard-compliant signal like cellular LTE (henceforth referred to as a *baseline*). The research community has just started to explore the impact of such anomalies under assumptions of different signal types [3] that are relevant to both consumer [4] and federal [5] stakeholders. We have considered the 4G LTE signal as baseline to generate a complex detection scenario for anomaly signal, in contrast to a more bursty 5G signal where the underlay can become more easily visible.

• **An example anomaly signal.** We specifically consider in this paper the noise-like direct sequence spread spectrum (DSSS) anomaly that hides within the LTE baseline. Moreover, the processing gain in the despreading operation enables the anomaly to exist with a low signal-to-interference-and-noise (SINR) ratio. Several wireless standards utilize a variant of DSSS, such as code-division multiple access (CDMA), IEEE 802.11b specifications used in Wi-Fi, and the Global Positioning System [6], any of which can be used by a malicious actor to construct an 'anomaly' hiding within the baseline. The baseline signal that contains an underlying anomaly signal is referred as *anomalous baseline signal* in the remainder of the paper. Next, we list several challenges related to the specific problem of DSSS detection.

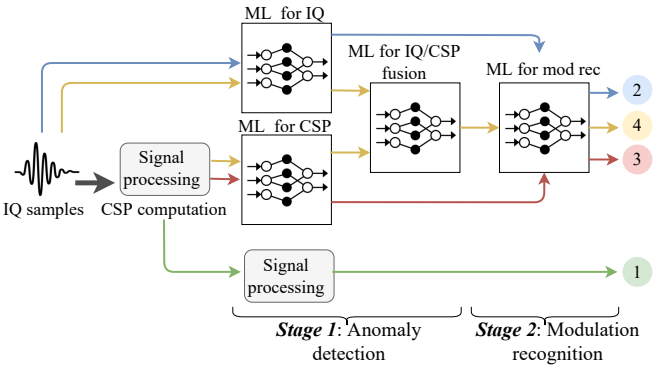
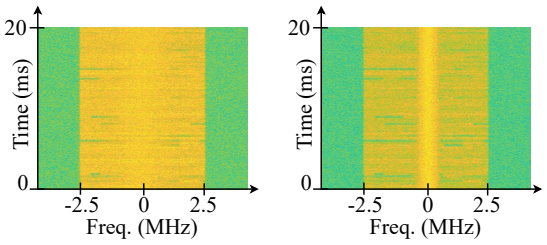


Fig. 1. Steps in detecting an anomaly and recognizing its modulation type within the baseline signal by selecting one of the four color-coded pipelines composed of neural networks (NNs) and/or signal processors: (1) signal processing on CSP features, (2) NN on IQ only, (3) NN on CSP features only, (4) NN fusing both IQ and CSP features. One of these four inference pipelines must be chosen considering computational resource constraints.

• **Challenge 1. Accuracy of methods for detecting DSSS anomaly:** In general, we list four different methods (Fig. 1) for DSSS detection, each with its own overhead and detection accuracy that combine signal-processing and machine learning (ML). However, detection of a subtle anomaly like DSSS is specifically challenging because it spectrally resembles the broadband noise present within the baseline signal, as shown in Fig. 2. Hence, we present the basics and associated problems of these so called '`Pipelines`' below (see `1-4` in Fig. 1).

`Pipeline 1`: *Pure signal processing.* Cyclostationary Signal Processing (CSP) algorithms can be used to extract highly discriminative features from most RF signals based on periodically time-variant probabilistic parameters, and their estimates, for the signal. Simple threshold-based algorithms can use these features to detect the presence of DSSS signals. `Pipeline 1` requires significant processing power to generate CSP features, longer signal collection times, and expert knowledge, but no *a priori* dataset gathering or training is needed [7], [8]. **Prob-**



(a) SIR of -0.091dB **(b)** SIR of -0.91dB

Fig. 2. Spectrograms of the DSSS signal overlapping with LTE show the impact of relative power levels on signal discriminative ability. The signal to interference ratio (SIR) is the power of the LTE signal divided by the power of the DSSS signal. In (b) the central higher power band of DSSS is clearly visible, but not in (a).

lem: Using expert knowledge to select the most representative CSP features for *Stages 1* and *2*, and quantify the impact of block lengths of IQ samples needed to compute them.

`Pipeline 2`: *ML using In-phase and Quadrature (IQ) samples.* Once trained on representative datasets, neural network-based ML models can be used for rapid inference tasks, while consuming very little real-time data. Prior ML approaches that use IQ samples have been demonstrated for problems like transmitter identification [9], channel estimation [10], outlier detection [11]. **Problem:** IQ-only ML models may suffer from lack of generalizability on encountering channel conditions different from those represented by the training dataset.

`Pipeline 3`: *ML using CSP features.* This approach partially includes `Pipeline 1` by giving all CSP features obtained from the former to the ML model as input, allowing the latter 'black box' to take over for final inference. By not using IQ inputs from `Pipeline 2`, this approach is robust to varying channel conditions. **Problem:** Selecting the best low-dimensional approximation to the inherently high-dimensional CSP features, and generating a comprehensive dataset of labeled CSP features for training.

`Pipeline 4`: *ML fusing CSP features and IQ.* This approach fuses the latent embeddings from both the pipelines for IQ (`Pipeline 2`) and for CSP features (`Pipeline 3`) for the final inference. This approach combines the representative power of both pipelines. **Problem:** Designing a robust fusion model that accepts latent embedding of different dimensions.

• **Challenge 2. Computational cost:** `Pipelines 1-4` incur different computational costs and elapsed times depending on the hardware specifications. For example, a 4-core Intel i7 CPU is better suited for `Pipeline 3` as it supports ∼38 giga floating point operations per second (FLOPS). Similarly, an NVIDIA A100 GPU is better for `Pipeline 2`, as it supports ∼ 312 tera FLOPS [12]. Thus, only considering the accuracy of the pipeline is not sufficient for a practical anomaly detection system. The data preprocessing latency that shapes the inputs for the respective pipelines described above, the computational requirements of the component inference models, and the constraints of compute hardware, must all be considered in the designing of the end-to-end inference steps.

• **Proposed approach:** To address Challenge 1, ICARUS includes a pure signal processing approach (`Pipeline 1`) as well as multiple high-accuracy neural network (NN) architectures for detecting DSSS signals within a baseline LTE using both IQ samples and CSP features (`Pipelines 2-4`). It is robust to different relative DSSS signal power levels compared to the baseline, ranging from -10dB to 10dB signal to interference ratio (SIR, defined as ratio of LTE to DSSS signal power). If DSSS is detected, ICARUS provides additional qualifying characteristics, such as whether BPSK or QPSK modulation is present. This hierarchical approach is depicted in Fig. 1, where we have trained models for the stages of anomaly detection (*Stage 1*) and modulation recognition (*Stage 2*), respectively. ICARUS addresses Challenge 2 by incorporating an algorithm for autonomously choosing the optimal `Pipeline (1-4)` given the hardware specifications and constraints, and thus, it eliminates the need for a human operator for this decision.

Generalizability of ICARUS: ICARUS can be used in more general settings involving other kinds of anomaly and baseline signals because the training of different versions of the system is not strongly dependent on the statistical nature of the signals, and because many different kinds of anomalous signals possess readily estimable cyclic (CSP) features, such as PSK, CPM, FSK, OFDM, and various generic TDMA signals. However, this paper's focus on DSSS as the anomaly is central to the basic problem of interest in that it can be successfully received by the anomaly receiver in spite of heavy cochannel interference imposed by the baseline signal, unlike most non-DSSS signals. The name 'ICARUS' reflects a design goal: Like its mythical Greek counterpart, ICARUS aims to optimize performance, here inference accuracy (in place of high altitude flight described in the myth). However, unlike its namesake, our software framework is aware of the environmental conditions, specifically computational constraints. Thus, it explicitly recognizes such hardware constraints and avoids an impractical outcome where the processing cannot be completed in time at all. Regrettably, a perilous outcome could not be avoided for the winged ICARUS, who ignored system limitations and flew too close to the sun.

• **Paper contributions and ICARUS design:**

(1) We propose a pure signal-processing based DSSS anomaly detector using domain knowledge of CSP and DSSS (`Pipeline 1`).

(2) We design a deep learning based NN architecture that can detect the DSSS anomaly within LTE baseline using only IQ samples (`Pipeline 2`). We also propose a second neural NN model that performs the same task but uses CSP features extracted from the signal (`Pipeline 3`).

(3) We design a fusion-based NN architecture, where the latent embeddings from the CSP and IQ NN models are fused together to serve as input to a subsequent NN inference model (`Pipeline 4`). We propose a hierarchical classifier that identifies the modulation scheme (*Stage 2 in Fig. 1*), BPSK or QPSK, following the DSSS prediction at *Stage 1*.

(4) We design an algorithm that selectively triggers specific parts of the ICARUS framework depending on available computing resources, resulting in autonomous selection of one of the `Pipelines 1-4`.

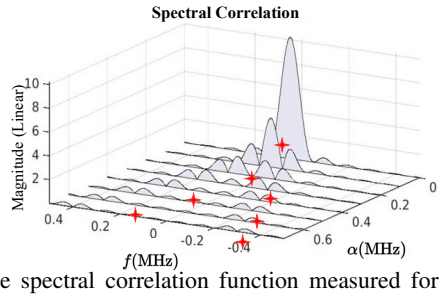


Fig. 3. The spectral correlation function measured for rectangular-pulse BPSK signal.

(5) We validate ICARUS on synthetic and OTA IQ samples collected from cellular LTE basestations and the NSF PAWR Platform for Open Wireless Data-driven Experimental Research (POWDER). We observe $98-100\%$ of anomaly- and $67-99\%$ modulation detection accuracy across these datasets.

II. Related Work and Background on CSP

Here, we survey the state-of-the-art for detecting the presence of underlay signals within a baseline signal. Benedetto *et al.* use pure signal processing for detecting unauthorized access in underlay satellite communication networks in [13]. Similarly, Hofmann *et al.* present a novel anomaly detector based on cyclostationary signal properties for low SINR conditions in low-earth orbit satellite communication links in [14]. Different from signal processing, [15] and [16] design ML-based detection using the power spectral density (PSD) and spectrogram of the received signal, respectively.

DSSS detection. Pure signal-processing-based methods for detecting the presence of such spread spectrum signals have been proposed in [17]. Recently, Zhang *et al.* demonstrate the use of wavelet decomposition to detect the DSSS signal under low SNR in a non-cooperative communication system in [18]. Liu *et al.* propose a non-cooperative compressive detection technique that uses random as well as designed measurement kernels for DSSS signals [19]. Similarly, [1] describes a single-class support vector machine (SVM) to detect a DSSS anomaly in the baseline using signal power information entropy.

While the above works make great strides in anomaly detection, to achieve near real-time operation, we need to consider the computational cost of the approaches. Different from the above works, ICARUS selects a combination of signal processing and ML steps for end-to-end inference, depending on both computational requirements of the chosen method and hardware availability. Furthermore, following the state-of-the-art [20], [21] in other domains, it takes the first step towards fusing IQ samples with CSP features as well as providing information about the *modulation type* of the detected DSSS signal, which are not covered in prior work.

Background on CSP. A cyclostationary signal has statistical parameters that vary with time, such as mean, variance, and higher-order moments [8]. These parameters can be defined for both the time- and frequency-domain representation of a signal, for which we have *temporal-* and *spectral moments*, respectively. The second-order spectral moment of a signal is also called the spectral correlation function (SCF). Fig. 3 shows the SCF surface of a BPSK signal. We consider four different parameters, {F, A, C, S}, extracted from the SCF, which we henceforth collectively refer to as *CSP features*. F is conventional spectral frequency (f in the plot), A is the cycle-frequency (α in the plot), C is the peak coherence magnitude for α, and S is the peak spectral correlation magnitude for α. Rigorous mathematical formulations of these features can be found in [7], [8], [17]. In summary, when we refer to *CSP features* in this paper, we imply this 4-tuple set {F, A, C, S} corresponding to selected peaks from the SCF.

III. Signal Types and System Model

A. Signal Notations

Baseline LTE signal. The LTE standard supports a variety of downlink configurations and duplex modes [22]. We consider frequency division duplex (FDD) LTE downlink signal transmission, which uses OFDMA modulation. We represent this baseband LTE signal using notation $s_{\texttt{LTE}}(t)$ and the received signal as $r_{\texttt{LTE}}(t) = s_{\texttt{LTE}}(t) + w(t)$, where $w(t)$ is the background noise.

DSSS anomaly signal. We represent a DSSS baseband signal as $s_{\texttt{DSSS}}(t)$, modulated with either BPSK or QPSK.

Anomalous baseline signal. The received baseband anomalous signal (DSSS is present within the baseline LTE) is represented as: $r_{\texttt{LTE+DSSS}}(t) = s_{\texttt{LTE}}(t) + s_{\texttt{DSSS}}(t) + w(t)$, where $w(t)$ is the background noise. In particular, the DSSS with BPSK and QPSK modulations are separately denoted as: $r_{\texttt{LTE+DSSS}_{\texttt{BPSK}}}(t)$ and $r_{\texttt{LTE+DSSS}_{\texttt{QPSK}}}(t)$, respectively. Hence, $r_{\texttt{LTE+DSSS}}(t) \in \{r_{\texttt{LTE+DSSS}_{\texttt{BPSK}}}(t), r_{\texttt{LTE+DSSS}_{\texttt{QPSK}}}(t)\}$.

B. Problem Formulation

We define the problems for anomaly detection and modulation recognition, separately as follows.

• **Anomaly detection.** We formulate the function for anomaly detection $\mathcal{A}(.)$ assuming $r_{\texttt{SIG}}(t) \in \{r_{\texttt{LTE}}(t), r_{\texttt{LTE+DSSS}}(t)\}$ as:

$$\mathcal{A}(r_{\texttt{SIG}}(t)) = 0 \quad \Rightarrow \text{LTE} \tag{1a}$$

$$= 1 \quad \Rightarrow \text{LTE+DSSS} \tag{1b}$$

• **Modulation recognition.** We denote the function for modulation classification as $\mathcal{M}(.)$, and $r_{\texttt{LTE+DSSS}}(t) \in \{r_{\texttt{LTE+DSSS}_{\texttt{BPSK}}}(t), r_{\texttt{LTE+DSSS}_{\texttt{QPSK}}}(t)\}$. Hence, we formulate the problem as:

$$\mathcal{M}(r_{\texttt{LTE+DSSS}}(t)) = 0 \quad \Rightarrow \text{BPSK} \tag{2a}$$

$$= 1 \quad \Rightarrow \text{QPSK} \tag{2b}$$

C. Overview: Model Training

The ICARUS system architecture consists of four components (see Fig. 4) as a modular solution:

1) CSP Computation: The first component extracts the CSP features from the received signal $r_{\texttt{SIG}}(t)$ (details in Section IV-A). The extracted features are then passed to `Pipelines 1, 3,` and `4` for *Stage 1* and `Pipelines 3` and `4` for *Stage 2*.

2) Signal Processing + ML Stage 1*: Anomaly Detection:* The first stage of the hierarchical classifier within ICARUS consists of `Pipelines 1-4`. We perform training for the `Pipelines 2-4` using the IQ samples and extracted CSP features. The CSP features are directly fed to `Pipeline 1` for conventional anomaly detection without involving ML. The details are discussed in Section IV-B.

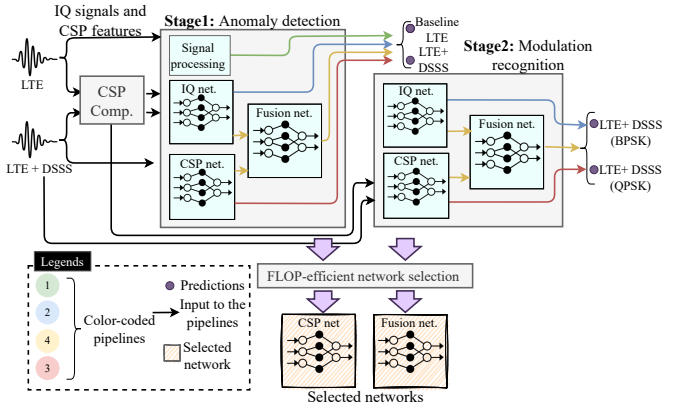


Fig. 4. The ICARUS training framework, details in Section III-C.

3) ML-only Stage 2*: Anomaly Modulation Recognition:* We use the IQ samples and extracted CSP features from the received signals $r_{\texttt{LTE+DSSS}}(t)$ for generating trained models for `Pipelines 2-4`. Additional details are in Section IV-C.

4) FLOP-efficient Pipeline Selection: Once the NN models are trained for all the pipelines, ICARUS can dynamically trigger one of the pipelines depending on available computing resources and expected FLOPS. An example of pipeline selection is given in Fig. 4, where the 'CSP network' (`Pipeline 3`) and 'fusion network' (`Pipeline 4`) are selected for anomaly detection and modulation recognition, respectively. The details are discussed in Section IV-D.

D. ICARUS Overview: Example of Inference

We show an example inference phase in Fig. 5, which is performed once the models are trained and incorporated. Note that the selected 'CSP network' (`Pipeline 3`) and 'fusion network' (`Pipeline 4`) from Fig. 4 are used for inference in Fig. 5. Depending on the available computation resources, these networks are used to predict whether DSSS anomaly is present within the baseline LTE or not. Once an anomaly is detected, the same IQ samples and CSP features are fed to the selected network of ML-only *Stage 2* for modulation recognition.

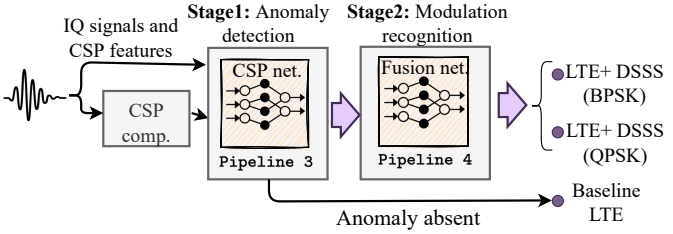


Fig. 5. The ICARUS inference framework. Here the input signal is either $r_{\texttt{LTE}}(t)$ or $r_{\texttt{LTE+DSSS}}(t)$.

IV. Detailed Design of the ICARUS Framework

A. Computing CSP Features

The anomalous baseline signal, i.e., LTE with DSSS present, exhibits discriminative cyclostationary properties relative to the baseline LTE signal. Let the IQ samples of $r_{\texttt{LTE}}(t)$ and $r_{\texttt{LTE+DSSS}}(t)$ be represented as $IQ_{\texttt{LTE}}$ and $IQ_{\texttt{LTE+DSSS}}$. $IQ_{\texttt{LTE}}$ and $IQ_{\texttt{LTE+DSSS}}$ are processed using CSP to generate low-dimensional feature matrices $CSP_{\texttt{LTE}}$ and $CSP_{\texttt{LTE+DSSS}}$. We represent one block from $IQ_{\texttt{LTE}}$ and $IQ_{\texttt{LTE+DSSS}}$ as $BL_{\texttt{LTE}}$ and $BL_{\texttt{LTE+DSSS}}$, respectively, with block length of BL_N. The number of blocks Num_{BL} in $IQ_{\texttt{LTE}}$ and $IQ_{\texttt{LTE+DSSS}}$ can be computed as $\frac{IQ_{\texttt{LTE}}}{BL_N}$ and $\frac{IQ_{\texttt{LTE+DSSS}}}{BL_N}$, respectively. The CSP features computed from the b^{th} block $BL^b_{\texttt{LTE}}$ can be represented as $CSP^b_{\texttt{LTE}} = \{(f_j, \alpha_j, c_j, s_j)\}_{j=1}^{N^{\texttt{LTE}}_{\texttt{CSP},b}}$, where $N^{\texttt{LTE}}_{\texttt{CSP},b}$ is the number of CSP features estimated for that block. Similarly, we denote the features for $BL^b_{\texttt{LTE+DSSS}}$ as: $CSP^b_{\texttt{LTE+DSSS}} = \{(f_j, \alpha_j, c_j, s_j)\}_{j=1}^{N^{\texttt{LTE+DSSS}}_{\texttt{CSP},b}}$, where $N^{\texttt{LTE+DSSS}}_{\texttt{CSP},b}$ is the variable number of CSP features estimated for the b^{th} block of $IQ_{\texttt{LTE+DSSS}}$.

B. Signal Processing + ML- Stage 1: Anomaly Detection

After computing the CSP features, both the IQ samples and CSP features are passed through the *Stage 1* of our hierarchical classifier. Here, we predict the presence of an anomaly using one of the 4 different pipelines.

`Pipeline 1`: *Pure signal processing.* In the first pipeline, we use only signal-processing methods to detect the presence of DSSS within the LTE baselines by processing the computed features $CSP_{\texttt{LTE}}$ and $CSP_{\texttt{LTE+DSSS}}$. Since we do not require any form of training, this approach can directly be used 'out of the box.' `Pipeline 1` consists of an application of the *strip spectral correlation analyzer* (SSCA) algorithm followed by a *cycle-frequency pattern recognizer* (CPR). The SSCA produces the {F, A, C, S} feature matrix and the CPR processes that matrix and outputs a decision.

• **SSCA.** The SSCA $f_{\texttt{SSCA}}(.)$ is a conventional time-smoothing method for estimating the SCF for all cycle-frequencies on a cycle-frequency grid with spacing equal to the native cycle-frequency resolution of all CSP methods, which is the reciprocal of the processed data length [23], [24]. Therefore, no cycle-frequency exhibited by the input data is missed due to a search-pattern deficiency. The SSCA itself produces the large set of point estimates of the spectral correlation function that cover the diamond-shaped CSP principal domain. To detect which of these millions of point estimates correspond to true cycle-frequencies exhibited by the data, the spectral correlation estimates are converted to correlation-coefficient estimates by the usual normalization of correlation by the geometric mean of the variances of the involved quantities. This processing step produces the spectral coherence, which can be conveniently thresholded with a single threshold across the principal domain.

A final processing step combines the detected cycle-frequencies across multiple values of spectral frequency f for each detected cycle-frequency α (see Fig. 3), leading in most cases to a small values of $N^{\texttt{LTE}}_{\texttt{CSP},b}$ and $N^{\texttt{LTE+DSSS}}_{\texttt{CSP},b}$ and generates {F, A, C, S} feature matrix.

• **CPR.** The design of the CPR requires knowledge of the expected patterns of CSP features, which depend on the probabilistic parameters (spectral moments) of the LTE and DSSS anomaly signals. Downlink LTE signals have relatively

weak cyclostationarity, but also the particular α values and the numbers of significant cycle-frequencies $N_{\mathtt{CSP},b}^{\mathtt{LTE}}$ for the b^{th} block depend on time as the LTE signal goes through various changes due to unpredictable user loads, framing, and movement of pilot subcarriers. The DSSS anomaly signals, on the other hand, have highly predictable and time-invariant cycle-frequencies that are harmonics of the code-repetition rate, including the chip rate and, in the case of BPSK, the doubled-carrier frequency plus such cycle-frequencies. Both kinds of cycle-frequencies will be found at the output of the SSCA when both the anomaly and LTE are present. The job of the CPR is to blindly find the cycle-frequency patterns that are present. It does this by looking for sets of cycle-frequencies with harmonic relations and joining them together.

Ideally the CPR would produce two decisions whenever both the anomaly and LTE are present: DSSS and LTE. Due to the time-variant and weak nature of the LTE cycle-frequencies, typically the output of the CPR with only LTE present will consist of no found patterns, whereas when the DSSS signal is present in isolation or is combined with LTE, the typical output would be DSSS or a generic cycle-frequency chain. In `Pipeline 1`, we decide that the DSSS anomaly has been detected when the DSSS or cycle-frequency-chain decision is made with a cycle-frequency chain that has elements exceeding 100 kHz, since the typical DSSS anomaly will have its strongest cycle-frequency that is equal to the chip rate, and this will be a large fraction of the occupied LTE bandwidth. That is, DSSS will produce long cycle-frequency chains with members with large α and LTE will produce at most small frequency chains with small α as the strongest LTE cycle-frequencies are well below 100 kHz.

We define the CPR domain knowledge as a function $f_{\mathtt{CPR}}(.)$ operating on the 4-tuple {F, A, C, S} matrix. Hence, in this case $\mathcal{A}(.)$ is defined as: $\mathcal{A}(.) = f_{\mathtt{CPR}}(f_{\mathtt{SSCA}}(.))$.

`Pipeline 2:` *ML using IQ samples.* The training data matrix for IQ is composed of the IQ samples $IQ_{\mathtt{LTE}}$ and $IQ_{\mathtt{LTE+DSSS}}$ from $N_t^{\mathtt{A}}$ baseline LTE and anomalous baseline LTE+DSSS signals, respectively. This data matrix is denoted as: $X_{\mathtt{IQ}}^{\mathtt{A}} \in \mathbb{R}^{N_t^{\mathtt{A}} \times d_0^{\mathtt{IQ}} \times 2}$. The $(d_0^{\mathtt{IQ}} \times 2)$ gives the dimensionality of the IQ samples where 2 accounts for the I and Q components. The set of the output labels are: $\mathcal{L}_{\mathtt{A}} = \{\mathtt{LTE}, \mathtt{LTE} + \mathtt{DSSS}\}$. We consider the label matrix $Y_{\mathtt{A}} \in \{0,1\}^{N_t^{\mathtt{A}} \times |\mathcal{L}_{\mathtt{A}}|}$ that represent the one-hot encoding for either DSSS present or absent.

The penultimate (second to last) layer of each neural network captures the latent representation of the input data [25]. We assume that the penultimate layer of this network has $d_{\mathtt{A}}^{\mathtt{IQ}}$ neurons. Hence, we represent the penultimate and ultimate layer transformations with $p_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}}$ and $f_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}}$, respectively, parameterized by weight vector $\theta_{\mathtt{A}}^{\mathtt{IQ}}$. The $p_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}}$ maps the input to a vector with dimension $d_{\mathtt{A}}^{\mathtt{IQ}}$. The $f_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}}$ maps the input to the one-hot encoded output. Formally, we define the penultimate and ultimate layers of the IQ network as:

$$\mathbf{z}_{\mathtt{IQ}}^{\mathtt{A}} = p_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}}(X_{\mathtt{IQ}}^{\mathtt{A}}), \qquad p_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}} : \mathbb{R}^{d_0^{\mathtt{IQ}} \times 2} \mapsto \mathbb{R}^{d_{\mathtt{A}}^{\mathtt{IQ}}} \tag{3a}$$

$$\mathbf{s}_{\mathtt{IQ}}^{\mathtt{A}} = \sigma(f_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}}(X_{\mathtt{IQ}}^{\mathtt{A}})), \qquad f_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}} : \mathbb{R}^{d_0^{\mathtt{IQ}} \times 2} \mapsto \mathbb{R}^{|Y_{\mathtt{A}}|} \tag{3b}$$

where $\mathbf{z}_{\mathtt{IQ}}^{\mathtt{A}}$ represents the extracted latent embeddings for input data $X_{\mathtt{IQ}}^{\mathtt{A}}$, $\sigma : \mathbb{R}^{|Y_{\mathtt{A}}|} \mapsto (0,1)^{|Y_{\mathtt{A}}|}$ signifies the Softmax activation, and $\mathbf{s}_{\mathtt{IQ}}^{\mathtt{A}}$ is the prediction score of the network. Overall the anomaly detection problem is solved using the IQ network by: $\mathcal{A}(.) = \arg\max \sigma(f_{\theta_{\mathtt{A}}^{\mathtt{IQ}}}^{\mathtt{IQ}}(.))$.

`Pipeline 3:` *ML using CSP features.* The training data matrix of CSP features $X_{\mathtt{CSP}}^{\mathtt{A}} \in \mathbb{R}^{N_t^{\mathtt{A}} \times d_0^{\mathtt{CSP}} \times 4}$ is composed of $N_t^{\mathtt{A}}$ $CSP_{\mathtt{LTE}}$ and $CSP_{\mathtt{LTE+DSSS}}$ feature matrices from baseline LTE and anomalous baseline signals, respectively. The dimension of the CSP features are $(d_0^{\mathtt{CSP}} \times 4)$, where 4 implies the {F, A, C, S} values. The representation of label matrix $Y \in \{0,1\}^{N_t^{\mathtt{A}} \times |Y_{\mathtt{A}}|}$ is same as earlier. The penultimate (with $d_{\mathtt{A}}^{\mathtt{CSP}}$ neurons) and ultimate layers of the CSP network are:

$$\mathbf{z}_{\mathtt{CSP}}^{\mathtt{A}} = p_{\theta_{\mathtt{A}}^{\mathtt{CSP}}}^{\mathtt{CSP}}(X_{\mathtt{CSP}}^{\mathtt{A}}), \qquad p_{\theta_{\mathtt{A}}^{\mathtt{CSP}}}^{\mathtt{CSP}} : \mathbb{R}^{d_0^{\mathtt{CSP}} \times 4} \mapsto \mathbb{R}^{d_{\mathtt{A}}^{\mathtt{CSP}}} \tag{4a}$$

$$\mathbf{s}_{\mathtt{CSP}}^{\mathtt{A}} = \sigma(f_{\theta_{\mathtt{A}}^{\mathtt{CSP}}}^{\mathtt{CSP}}(X_{\mathtt{CSP}}^{\mathtt{A}})), \qquad f_{\theta_{\mathtt{A}}^{\mathtt{CSP}}}^{\mathtt{CSP}} : \mathbb{R}^{d_0^{\mathtt{CSP}} \times 4} \mapsto \mathbb{R}^{|Y_{\mathtt{A}}|} \tag{4b}$$

where $\mathbf{z}_{\mathtt{CSP}}^{\mathtt{A}}$ are the extracted latent embeddings for input data $X_{\mathtt{CSP}}^{\mathtt{A}}$, σ is same as earlier, and $\mathbf{s}_{\mathtt{CSP}}^{\mathtt{A}}$ is the prediction score of the network. Hence, the anomaly detection problem is solved as: $\mathcal{A}(.) = \arg\max \sigma(f_{\theta_{\mathtt{A}}^{\mathtt{CSP}}}^{\mathtt{CSP}}(.))$.

`Pipeline 4:` *ML fusing IQ samples and CSP features.* In this pipeline, we leverage both the IQ samples and CSP features as input to a complex neural network, called a *fusion network*. The latent embeddings from the penultimate layers of both IQ and CSP networks are concatenated [25] to generate the combined latent feature matrix $\mathbf{z}^{\mathtt{A}} = [\mathbf{z}_{\mathtt{IQ}}^{\mathtt{A}}; \mathbf{z}_{\mathtt{CSP}}^{\mathtt{A}}] \in \mathbb{R}^{d_{\mathtt{A}}^{\mathtt{IQ}} + d_{\mathtt{A}}^{\mathtt{CSP}}}$. Layers that follow concatenation step form the fusion network $f_{\theta_{\mathtt{A}}^{\mathtt{FN}}}^{\mathtt{FN}}(.)$, intuitively assign higher weights to either IQ or CSP features depending on their relevance. Finally, the prediction vector is generated through a Softmax activation:

$$\mathbf{s}_{\mathtt{FN}}^{\mathtt{A}} = \sigma(f_{\theta_{\mathtt{A}}^{\mathtt{FN}}}^{\mathtt{FN}}(\mathbf{z}^{\mathtt{A}})), \qquad f_{\theta_{\mathtt{A}}^{\mathtt{FN}}}^{\mathtt{FN}} : \mathbb{R}^{d_{\mathtt{A}}^{\mathtt{IQ}} + d_{\mathtt{A}}^{\mathtt{CSP}}} \mapsto \mathbb{R}^{|Y_{\mathtt{A}}|} \tag{5}$$

In this pipeline, the anomaly detection problem is solved with: $\mathcal{A}(.) = \arg\max \sigma(f_{\theta_{\mathtt{A}}^{\mathtt{FN}}}^{\mathtt{FN}}(.))$.

C. ML-only Stage 2: Modulation Recognition

After the anomaly is detected within a signal, we propose the second stage of hierarchical classifier to detect the modulation techniques used in the DSSS. We use the `Pipelines 2-4` for this task.

`Pipeline 2:` *ML using IQ samples.* In this case, the data matrix of IQ samples $X_{\mathtt{IQ}}^{\mathtt{M}} \in \mathbb{R}^{N_t^{\mathtt{M}} \times d_0^{\mathtt{IQ}} \times 2}$ constitutes $N_t^{\mathtt{M}}$ IQ samples of anomalous signals $IQ_{\mathtt{LTE+DSSS}}$. The output labels are: $\mathcal{L}_{\mathtt{M}} = \{\mathtt{BPSK}, \mathtt{QPSK}\}$. The label matrix is $Y_{\mathtt{M}} \in \{0,1\}^{N_t^{\mathtt{M}} \times |\mathcal{L}_{\mathtt{M}}|}$ in the one-hot encoding representation to signify either BPSK or QPSK modulation schemes. The penultimate (with $d_{\mathtt{M}}^{\mathtt{IQ}}$ neurons) and ultimate layers of this IQ network are:

$$\mathbf{z}_{\mathtt{IQ}}^{\mathtt{M}} = p_{\theta_{\mathtt{M}}^{\mathtt{IQ}}}^{\mathtt{IQ}}(X_{\mathtt{IQ}}^{\mathtt{M}}), \qquad p_{\theta_{\mathtt{M}}^{\mathtt{IQ}}}^{\mathtt{IQ}} : \mathbb{R}^{d_0^{\mathtt{IQ}} \times 2} \mapsto \mathbb{R}^{d_{\mathtt{M}}^{\mathtt{IQ}}} \tag{6a}$$

$$\mathbf{s}_{\mathtt{IQ}}^{\mathtt{M}} = \sigma(f_{\theta_{\mathtt{M}}^{\mathtt{IQ}}}^{\mathtt{IQ}}(X_{\mathtt{IQ}}^{\mathtt{M}})), \qquad f_{\theta_{\mathtt{M}}^{\mathtt{IQ}}}^{\mathtt{IQ}} : \mathbb{R}^{d_0^{\mathtt{IQ}} \times 2} \mapsto \mathbb{R}^{|Y_{\mathtt{M}}|} \tag{6b}$$

where $\sigma : \mathbb{R}^{|Y_{\mathtt{M}}|} \mapsto (0,1)^{|Y_{\mathtt{M}}|}$ is the Softmax activation, $\mathbf{z}_{\mathtt{IQ}}^{\mathtt{M}}$ represents the latent embeddings and $\mathbf{s}_{\mathtt{IQ}}^{\mathtt{M}}$ is the prediction score of the network. The modulation recognition problem is solved as: $\mathcal{M}(.) = \arg\max \sigma(f_{\theta_{\mathtt{M}}^{\mathtt{IQ}}}^{\mathtt{IQ}}(.))$.

Pipeline 3: *ML using CSP features.* The data matrix of CSP features $X^{\mathtt{M}}_{\mathtt{CSP}} \in \mathbb{R}^{N^{\mathtt{M}}_t \times d^{\mathtt{CSP}}_0 \times 4}$ has $N^{\mathtt{M}}_t$ CSP features from the anomalous baseline signals $IQ_{\mathtt{LTE+DSSS}}$. The penultimate (with $d^{\mathtt{CSP}}_{\mathtt{M}}$ neurons) and ultimate layers of this IQ network as:

$$\mathbf{z}^{\mathtt{M}}_{\mathtt{CSP}} = p^{\mathtt{CSP}}_{\theta^{\mathtt{CSP}}_{\mathtt{M}}}(X^{\mathtt{M}}_{\mathtt{CSP}}), \qquad p^{\mathtt{CSP}}_{\theta^{\mathtt{CSP}}_{\mathtt{M}}} : \mathbb{R}^{d^{\mathtt{CSP}}_0 \times 4} \mapsto \mathbb{R}^{d^{\mathtt{CSP}}_{\mathtt{M}}} \tag{7a}$$

$$\mathbf{s}^{\mathtt{M}}_{\mathtt{CSP}} = \sigma(f^{\mathtt{CSP}}_{\theta^{\mathtt{CSP}}_{\mathtt{M}}}(X^{\mathtt{M}}_{\mathtt{CSP}})), \qquad f^{\mathtt{CSP}}_{\theta^{\mathtt{CSP}}_{\mathtt{M}}} : \mathbb{R}^{d^{\mathtt{CSP}}_0 \times 4} \mapsto \mathbb{R}^{|Y_{\mathtt{M}}|} \tag{7b}$$

where $\mathbf{z}^{\mathtt{M}}_{\mathtt{CSP}}$ and $\mathbf{s}^{\mathtt{M}}_{\mathtt{CSP}}$ hold the similar meaning as earlier. The solution through this pipeline is represented as: $\mathcal{M}(.) = \arg\max \sigma(f^{\mathtt{CSP}}_{\theta^{\mathtt{CSP}}_{\mathtt{M}}}(.))$.

Pipeline 4: *ML using IQ and CSP features.* The combined latent feature matrix $\mathbf{z}^{\mathtt{M}}$ is defined as: $\mathbf{z}^{\mathtt{M}} = [\mathbf{z}^{\mathtt{M}}_{\mathtt{IQ}}; \mathbf{z}^{\mathtt{M}}_{\mathtt{CSP}}] \in \mathbb{R}^{d^{\mathtt{IQ}}_{\mathtt{M}} + d^{\mathtt{CSP}}_{\mathtt{M}}}$. The fusion network $f^{\mathtt{FN}}_{\theta^{\mathtt{FN}}_{\mathtt{M}}}(.)$ is used to generate the prediction vector following a Softmax activation:

$$\mathbf{s}^{\mathtt{M}}_{\mathtt{FN}} = \sigma(f^{\mathtt{FN}}_{\theta^{\mathtt{FN}}_{\mathtt{M}}}(\mathbf{z}^{\mathtt{M}})), \qquad f^{\mathtt{FN}}_{\theta^{\mathtt{FN}}_{\mathtt{M}}} : \mathbb{R}^{d^{\mathtt{IQ}}_{\mathtt{M}} + d^{\mathtt{CSP}}_{\mathtt{M}}} \mapsto \mathbb{R}^{|Y_{\mathtt{M}}|} \tag{8}$$

Hence, the modulation recognition problem is solved as: $\mathcal{M}(.) = \arg\max \sigma(f^{\mathtt{FN}}_{\theta^{\mathtt{FN}}_{\mathtt{M}}}(.))$.

D. Algorithm for FLOP-efficient Pipeline Selection

While ICARUS strives to achieve faster inference from both the stages of hierarchical classifier, it also aims to maximize the accuracy of inference. Hence, ICARUS includes Algorithm 1 that dynamically selects one pipeline out of four for *Stage 1*, and out of three for *Stage 2*. In `Pipelines 2-4`, the higher-weighted neurons are more important for generating a prediction [26] from the the NN models. Hence, Algorithm 1 considers the viability of the `Pipelines 2-4`, by ranking the corresponding NN models depending on the number of important neurons they have to ensure the best performance [26], while meeting the threshold of the FLOPS supported by the device.

***Stage 1*:** Formally, in the *Stage 1* of ICARUS, `Pipelines 2-4`, information of CSP features (BL_N, Num_{BL}), and floating point operations (FLOPs) threshold $Thres_{FLOP}$ are fed to the Algorithm 1. The associated NN models are: $\mathcal{N}_{\mathtt{IQ}} = f^{\mathtt{IQ}}_{\theta^{\mathtt{IQ}}_{\mathtt{A}}}$, $\mathcal{N}_{\mathtt{CSP}} = f^{\mathtt{CSP}}_{\theta^{\mathtt{CSP}}_{\mathtt{A}}}$, and $\mathcal{N}_{\mathtt{FN}} = f^{\mathtt{FN}}_{\theta^{\mathtt{FN}}_{\mathtt{A}}}$. The importance vectors of those NN models are calculated (`Step 5`) as:

$$I^n_{\mathcal{N}} = \sum_{l=1}^{L_n} \sum_{i=1}^{|l|} \alpha_l ||\theta_i||^2_2 + \beta_l, \qquad n \in \{\mathtt{IQ}, \mathtt{CSP}, \mathtt{FN}\} \tag{9}$$

where l is the layer index, $|l|$ is the total number of neurons in layer l, θ_i is the weight of i^{th} neuron, $\alpha_l, \beta_l \in \mathbb{R}^{L_n}$ are learnable parameters, L_n is the total number of layers of n^{th} network, $||.||$ denotes l_2 norm. This importance vector reflects the number of important neurons a model has: the higher the value, the greater the probability of the NN model to perform more accurately [26]. Hence, we rank the $\mathcal{N}_{\mathtt{IQ}}$, $\mathcal{N}_{\mathtt{CSP}}$, and $\mathcal{N}_{\mathtt{FN}}$ depending on their $I_{\mathcal{N}}$. Finally Algo. 1 chooses the best ranked model, where the number of required FLOPs meets the FLOPS configuration ($Thres_{FLOP}$) of the given device. If none of the NN models of `Pipelines 2-4` satisfy the hardware constraint (`Step 14`), the required FLOPs for `Pipeline 1` is compared against the $Thres_{FLOP}$. In case `Pipeline 1` also fails to meet the $Thres_{FLOP}$ (`Steps 18-21`), the pipeline with the NN model of highest $I_{\mathcal{N}}$ is selected. The selected pipeline $\mathcal{P}_{Selected}$ then can be used for inference.

Algorithm 1: FLOP-efficient Pipeline Selection

1: **Inputs:** Pipelines `P1`, `P2`, `P3`, `P4`, block length BL_N, approximate number of blocks per signal Num_{BL}, and FLOPs count threshold $Thres_{FLOP}$
2: **Output:** The selected pipeline $\mathcal{P}_{Selected}$
3: $\mathcal{N}_{\mathtt{IQ}}$= model(`P2`), $\mathcal{N}_{\mathtt{CSP}}$= model(`P3`), $\mathcal{N}_{\mathtt{FN}}$= model(`P4`)
4: **for** each $n \in$ {`IQ`, `CSP`, `FN`} **do**
5: Initialize the importance vector $I^n_{\mathcal{N}} = \sum_{l=1}^{L_n} \sum_{i=1}^{|l|} \alpha_l ||\theta_i||^2_2 + \beta_l$
6: FLOPs count vector $\mathcal{F}^n_{\mathcal{N}}$ = Calculate_FLOPs($\mathcal{N}_n$).
7: $K_n = False$
8: **if** $\mathcal{F}^n_{\mathcal{N}} < Thres_{FLOP}$ **then**
9: $K_n = True$
10: **end if**
11: **end for**
12: $\mathcal{R}_{\mathcal{N}}$ = Rank the networks based on the importance vector $I_{\mathcal{N}}$.
13: $n = \arg\max \mathcal{R}_{\mathcal{N}}$ for $n \in$ {`IQ`, `CSP`, `FN`} and $K_n == True$
14: **if** $n ==$ NULL **then**
15: $\mathcal{F}_{BL_N}$ = Calculate_FLOPs(BL_N, Num_{BL})
16: **if** $\mathcal{F}_{BL_N} < Thres_{FLOP}$ **then**
17: $\mathcal{P}_{Selected}$ = `P1`
18: **else**
19: $n = \arg\max \mathcal{R}_{\mathcal{N}}$ for $n \in$ {`IQ`, `CSP`, `FN`}
20: $\mathcal{P}_{Selected}$ = Get_pipeline(n)
21: **end if**
22: **else**
23: $\mathcal{P}_{Selected}$ = Get_pipeline(n)
24: **end if**

***Stage 2*:** At this stage, the inputs to Algo. 1 are only `Pipelines 2-4` and $Thres_{FLOP}$. Here, $\mathcal{N}_{\mathtt{IQ}}$=$f^{\mathtt{IQ}}_{\theta^{\mathtt{IQ}}_{\mathtt{M}}}$, $\mathcal{N}_{\mathtt{CSP}}$=$f^{\mathtt{CSP}}_{\theta^{\mathtt{CSP}}_{\mathtt{M}}}$, and $\mathcal{N}_{\mathtt{FN}}$=$f^{\mathtt{FN}}_{\theta^{\mathtt{FN}}_{\mathtt{M}}}$. The `Steps 14-24` are skipped.

V. Datasets Collected for Validating ICARUS

A. Synthetic Dataset with MATLAB-generated Waveforms

We generate a dataset of synthetic LTE and DSSS waveforms using MATLAB R2021b and combine them in an emulated channel environment. We generate the LTE signal $s_{\mathtt{LTE}}(t)$ in FDD mode for downlink communication using MATLAB LTE Toolbox 3.6 and a pulse-shaped variable-parameter DSSS signal $s_{\mathtt{DSSS}}(t)$. We assume the background noise to be Gaussian distributed with IID components having zero mean and unit variance. We generate the received baseline LTE signal $r_{\mathtt{LTE}}(t)$ by adding the LTE signal and noise, and the received anomalous baseline signal $r_{\mathtt{LTE+DSSS}}(t)$ by also adding the DSSS signal. We create 408 frames of duration 80 ms each at sampling rates $\in \{7.68, 15.36, 30.72\}$ MHz, such that 120 frames contain $r_{\mathtt{LTE}}(t)$, while the remaining 288 frames contain $r_{\mathtt{LTE+DSSS}}(t)$. For LTE signals, we use bandwidth $\mathrm{BW}_{\mathtt{LTE}} \in \{5, 10, 15, 20\}$ MHz. Without loss of generality, we consider same center frequency for the DSSS and LTE signals. However, we consider different percentages of overlap ($\{25, 50, 75, 100\}\%$) of the DSSS signals with the baseline LTE signal, i.e. setting $\mathrm{OBW}_{\mathtt{DSSS}}$ equal to $(\frac{\%\mathrm{overlap}}{100} \times \mathrm{OBW}_{\mathtt{LTE}})$, where $\mathrm{OBW}_{\mathtt{DSSS}}$ and $\mathrm{OBW}_{\mathtt{LTE}}$ are occupied BWs of DSSS and LTE signals, respectively. $\mathrm{OBW}_{\mathtt{LTE}}$ equals $(0.9 \times \mathrm{BW}_{\mathtt{LTE}})$

and $\text{OBW}_{\texttt{DSSS}}$ equals $((1+\gamma) \times \text{BW}_{\texttt{DSSS}})$, with roll-off factor $\gamma = 0.5$ used for square-root raised-cosine pulse shaping of the DSSS signals. Furthermore, for the DSSS signals, we use BPSK and QPSK modulation with linear-feedback shift register-based PN sequences of length $2^M - 1$, where $M \in \{6, 7, 8, 9, 10\}$ is the number of registers. We consider signal to noise ratio of the LTE signal $\text{SNR}_{\texttt{LTE}} \in [0, 10]$ dB and the SIR $\in [0, 10]$ dB.

Remark 1. *This dataset is generated with precise control over SNR and SIR parameters of the signals since the signals/sources are synthetic.*

B. Indoor OTA-PAWR Dataset with srsLTE Waveforms

We collect this dataset from the NSF POWDER testbed [27] of PAWR platform using `srslte-otalab` profile [28], which provides resources for performing over-the-air (OTA) operation in their indoor lab. This collection emulates a private LTE network. We use one USRP X310 operating at 3.56 GHz center frequency as an srsLTE eNodeB for transmitting LTE FDD downlink signals. We use two USRP B210s operating at 3.56 GHz center frequency, one as an srsLTE UE and another for collecting the IQ samples of the downlink srsLTE signal. The range of $\text{SNR}_{\texttt{LTE}}$ is $[10.57, 15.94]$ dB. For creating the anomalous baseline signal, we generate a synthetic DSSS signal at the sampling rate of the collected srsLTE signal and add them together. The parameter values for this dataset are provided in Table I.

Remark 2. *This dataset is collected in an indoor environment with network traffic only from a single UE. We observe that the number of subcarriers in the captured srsLTE signal is significantly less than that in a commercial LTE signal, which is a representative case for private LTE networks.*

C. OTA-Cellular Dataset w/ Commercial Cellular Waveforms

In this dataset, also referred to as the 'OTA-Cellular' dataset, we capture ambient downlink OTA LTE signals present in the cellular PCS bands using USRP X310, after which we add either synthetic DSSS (BPSK/QPSK) or OTA-captured DSSS BPSK signals having the same sampling rate as the captured LTE. Parameter values are shown in Table I.

Remark 3. *This dataset captures the channel effects and is representative of general scenarios where cellular LTE signals are present from commercial network operators.*

The *OTA-PAWR* and *OTA-Cellular* datasets are realistic as they capture the *real wireless channel* artifacts and hardware impairments in the LTE signals.

VI. Experimental Analysis

A. Performance of ICARUS Anomaly Detection

NN architectures. The details of NN models $f^{\texttt{IQ}}_{\theta^{\texttt{IQ}}_{\texttt{A}}}(.)$, $f^{\texttt{CSP}}_{\theta^{\texttt{CSP}}_{\texttt{A}}}(.)$, and $f^{\texttt{FN}}_{\theta^{\texttt{FN}}_{\texttt{A}}}(.)$ (discussed in Section IV-B) for the *Stage 1* of the hierarchical classifier are shown in Fig. 6. We exploit categorical cross-entropy loss for training with a batch size of 32 for 100 epochs. We use Adam [29] as the optimizer with decay rate of $(0.9, 0.999)$, L2 penalty of 0.0001, and the learning rate of 0.0001. We use 80/20 train/test ratio for all the experiments using Pytorch.

Table I: The parameter settings for the different data collection setup.

Settings	**Synthetic** (Section V-A)	**OTA-PAWR** (Section V-B)	**OTA-Cellular** (Section V-C)
Platform	MATLAB (LTE Toolbox)	POWDER (Indoor lab) + MATLAB	USRPs (Indoor lab)
LTE Signal Type	Synthetic	srsLTE OTA	Captured OTA (Live)
DSSS Signal Type	Synthetic	Synthetic	Synthetic/Captured
$\text{BW}_{\texttt{LTE}}$ (MHz)	$\{5, 10, 15, 20\}$	$\{5, 10\}$	$\{5, 10, 15, 20\}$
%Overlap of DSSS	$\{25, 50, 75, 100\}$		$[8.60, 99.69]$
Registers in LFSR (M)	$\{6, 7, 8, 9, 10\}$		$\{6, 7, 8, 9, 10, 11\}$
Roll-off Factor (γ)	0.5		$[0.1, 1)$
Sampling Rates (MHz)	$\{7.68, 15.36, 30.72\}$	$\{5.76, 11.52\}$	$\{7.68, 15.36, 30.72\}$
Frame Duration (ms)	80	$\{79.21875, 80\}$	$\{4.267, 8.533, 17.067, 34.133, 68.267\}$
($\#\text{Frames}_{\texttt{LTE}}$, $\#\text{Frames}_{\texttt{LTE+DSSS}}$)	$(120, 288)$	$(500, 2000)$	$(2430, 2430)$
SIR (dB)	$[0, 10]$	$[-10, 10]$	$[-10, 10]$
Center freq. offset for DSSS (MHz)	0	$[-3.3715, 3.3387]$	$[-1.4816, 1.4994]$

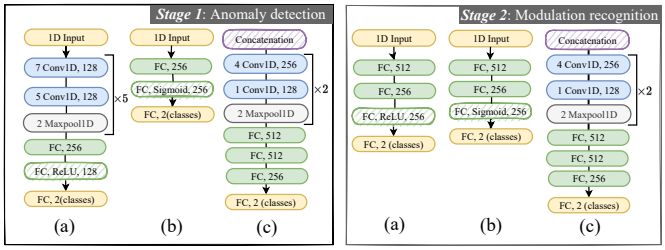


Fig. 6. Different optimal neural network architectures used for the two stages of hierarchical classifier using: (a) IQ samples (`Pipeline 2`), (b) CSP features (`Pipeline 3`), (c) both (`Pipeline 4`).

Table II: Performance of ICARUS *Stage 1*, anomaly detection.

Datasets	Block Lengths	Acc. (%) of `Pipelines` 1	2	3	4
Synthetic	131072	61.38	99.81	99.81	**99.81**
	262144	69.30	99.6	100	**100**
	524288	80.19	92.38	99.1	**99.1**
OTA-PAWR	131072	50.68	100	62.01	**100**
	262144	32.56	100	80.68	**100**
	524288	22.32	100	83.33	**100**
OTA-Cellular	131072	75.80	96.75	80.62	**98.95**
	262144	81.77	93.75	81.16	**97.96**
	524288	83.22	97.5	84.7	**98.25**

Block lengths. As mentioned in Section IV-A, CSP features are computed block-wise from the collected IQ samples. One *block* represents the chunk of IQ samples of a fixed length. In our experiment, we consider three block lengths BL_N: (a) 131072, (b) 262144, and (c) 524288 captured within $4-68\,ms$, to generate representative CSP features of the signals.

1) Performance of Different Pipelines: In the first set of experiments, we evaluate the *Stage 1* of the hierarchical classifier that detects DSSS anomalies. The performance of all the pipelines across different datasets is shown in Table II. `Pipeline 1` gives comparatively lower accuracy of 22%-83%, but does not require any prior training.

Observation 1. *We observe that fusion of both CSP and IQ features in the `Pipeline 4` gives better (for OTA-Cellular dataset) or equivalent (for Synthetic and OTA-PAWR datasets) performance (98-100%) than other NN-based pipelines (80-97%), across all datasets (see Table II).*

2) In-depth Study on `Pipelines 2-4`: Here, we perform more in-depth analysis of the performance for changing RF environment. We also study how the other existing ML approaches perform compared to `Pipeline 3` of ICARUS.

Table III: SIR-wise acc (%) of anomaly detection on the Synthetic dataset with block length 131072 and $\text{SNR}_{\text{LTE}} \in \{0, 5, 10\}$ dB.

LTE to DSSS Interference Ratio (SIR)	IQ data	CSP Features				Fusion
	P2	P3	Logistic Regression	Naïve Bayes	SVM	P4
0dB	100	99.81	90.56	99.81	99.81	100
5dB	99.43	99.81	88.1	99.81	98.67	99.81
10dB	55.87	73.86	65.09	72.45	73.77	73.86

Table IV: SNR-wise accuracy (%) of anomaly detection for 10dB SIR on the Synthetic dataset with block length 131072.

Baseline SNR_{LTE}	IQ data	CSP Features				Fusion
	P2	P3	Logistic Regression	Naïve Bayes	SVM	P4
0dB	53	59.09	58.19	57.06	57.62	59.09
5dB	54	81.81	74.01	81.92	81.92	81.81
10dB	55.68	84.65	71.18	79.66	83.61	84.65

We use the Synthetic dataset to conduct our experiments in different controlled SIR and SNR_{LTE} settings.

•**Varying SIR.** In Table III, we analyze how both the `Pipelines 2` and `3` perform with increasing SIR, where higher SIR values imply the signal strength for LTE is considerably higher than DSSS. The performance of ICARUS decreases with increasing SIR, because a stronger baseline LTE signal makes it difficult to detect the relatively low-power DSSS signal. Further, we compare our proposed NN based model with other ML based methods, such as logistic regression, SVM [30] and Naïve Bayes. We use a non-linear kernel radial basis function (RBF) for the SVM implementation to capture the non-linear boundaries between the labels. We observe that SVM and Naïve Bayes yield similar performance to the NN used in Pipeline 3. However, these methods do not infer the decision from the high dimensional IQ data.

Observation 2. *If ICARUS must predict anomalies using only CSP features, the SVM and Naïve Bayes suffice; they are as effective as NNs with lower complexity (see Table III).*

Observation 3. *We observe a sudden drop in performance of* `Pipelines 2-4` *and ML methods on changing SIR from 5dB → 10dB SIR, compared to 0dB → 5dB. This is because, on average, the DSSS power is less than the noise power for 10dB SIR, unlike the other SIR settings (see Table III).*

•**Varying SNR.** Next, we show results from an SNR-based study for all the pipelines and competing methods in Table IV. Increasing SNR_{LTE} while keeping SIR fixed implies decreasing the strength of background noise relative to the LTE signal strength, which in turn decreases it with respect to the DSSS signal strength as well. Our intuition is that capturing discriminative properties within DSSS signals through CSP features becomes easier in such cases.

Observation 4. *With increasing SNR at 10dB SIR, all the pipelines and competing methods show monotonic improvement (see Table IV).*

• **Comparison with the state-of-the-art.** We compare the performance of ICARUS with state-of-the-art techniques of DSSS detection [1] and [18], with results shown in Table V. We limit the comparison study to the above literature, as the other techniques differ from ours with respect to either: (a) spectrum of interest [14], [13], (b) anomaly (DSSS) of interest [15], (c) focus on different performance metrics [7], [17], [19]. It is evident that while state-of-the-art provides $\sim 100\%$ accuracy for some studies, it is limited to synthetic data or involves signal processing solutions without elapsed time considerations. We note however that ICARUS returns close to $\sim 100\%$ accuracy when validated on real OTA-indoor and cellular datasets with standard compliant waveforms.

Table V: Comparison of ICARUS with state-of-the-art techniques for anomaly detection on different signals.

Methods	Baseline Signal	Anomaly Signal	Architecture	Acc (%)	Datasets
Ma *et al.* [1]	FM	DSSS	Signal Processing	55-98	Synthetic
Zhang *et al.* [18]	-	DSSS	Signal Processing	10-100	Synthetic
ICARUS	LTE	DSSS	Fusion-based ML (P4)	98-100	Synthetic, OTA-indoor, OTA-cellular

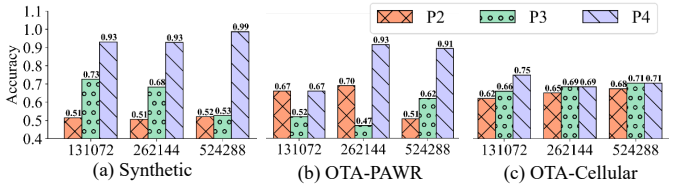


Fig. 7. Performance of ICARUS *Stage 2*, modulation recognition.

B. Performance of ICARUS Modulation Recognition

NN architectures. The details of the proposed NN models $f^{\texttt{IQ}}_{\theta^{\texttt{IQ}}_{\texttt{M}}}(.)$, $f^{\texttt{CSP}}_{\theta^{\texttt{CSP}}_{\texttt{M}}}(.)$, and $f^{\texttt{FN}}_{\theta^{\texttt{FN}}_{\texttt{M}}}(.)$ (discussed in Section IV-C) for the *Stage 2* of the hierarchical classifier are also shown in Fig. 6. We use similar train/test parameters as used for anomaly detection in Section VI-A.

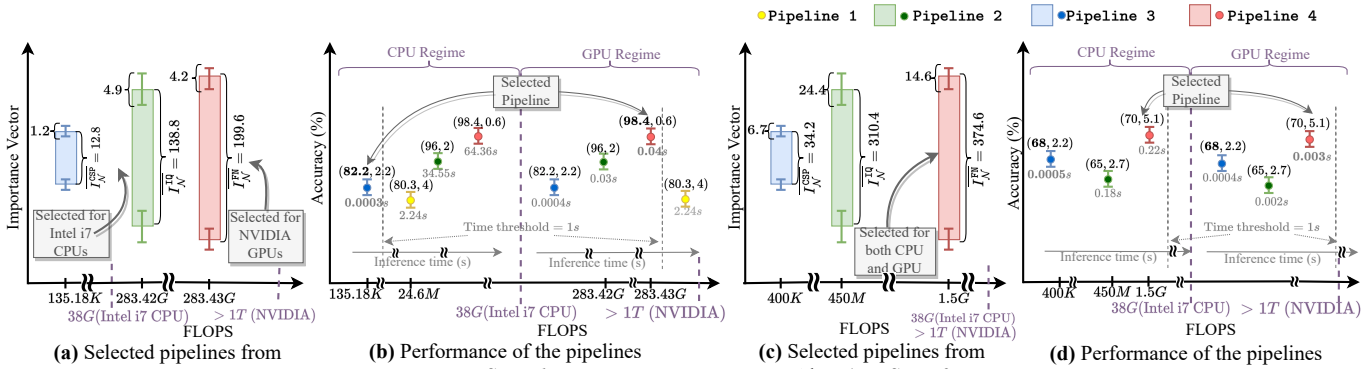
Performance of different pipelines. In this case, we only apply `Pipelines 2-4` on different datasets, shown in Fig. 7.

Observation 5. *We observe that fusion of both IQ and CSP features in* `Pipeline 4` *performs best (67%- 99%) across all datasets for recognizing the DSSS modulation (see Fig. 7).*

C. Analysis of ICARUS FLOP-efficient Pipeline Selection

We perform this set of experiments on OTA-Cellular dataset as it captures the most challenging RF scenarios of real-world.

Impact of compute-hardware FLOPS-constraints on inference time. We calculate the FLOPs count of the trained models for `Pipelines 2-4` by using *ptflops* python library. However, the FLOPs count of `Pipeline 1` is dominated by the cost of the SSCA calculation. Recall, in `Pipeline 1`, the SSCA is followed by a cycle-frequency pattern recognizer, which does not impact significantly the total FLOPs count [23], [24]. Overall, the required FLOPs counts for the `Pipelines 1-4` in *Stage 1* are: 24.6 MFLOPs, 283.42 GFLOPs, 135.18 KFLOPs, and 283.43 GFLOPs, respectively. Similarly, FLOP counts for the `Pipelines 2-4` in *Stage 2* are: 450 MFLOPs, 400 KFLOPs, and 1.5 GFLOPs, respectively. In Table VI, we show the inference times of *Stage 1* varies depending on the platform the pipeline is executed in, due to the different supported FLOPS in them. We consider three different computing platform for this study: (a) Intel(R) Core(TM) i7-7820HQ CPU @2.90GHz (38.05 GFLOPS), (b) NVIDIA 2080 Ti GPU (13.45 TFLOPS), and (c) NVIDIA A100 GPU (312 TFLOPS).



(a) Selected pipelines from Algo. 1 at *Stage 1*

(b) Performance of the pipelines at *Stage 1*

(c) Selected pipelines from Algo. 1 at *Stage 2*

(d) Performance of the pipelines at *Stage 2*

Fig. 8. The predicted best pipeline on a specific hardware from Algo. 1 (refer to (a) and (c)) has the best performance during the inference time (refer to (b) and (d)) for OTA-Cellular dataset. The mean and variance of the accuracies over all the three considered block lengths are presented in (b) and (d) in (X, Y) format. Similarly, the mean (over block lengths) of the importance vector (Eq. 9) in (a) and (c) is represented as $\overline{I_{\mathcal{N}}^{n}}, n \in \{\texttt{IQ}, \texttt{CSP}, \texttt{FN}\}$. The inference times in (b) and (d) are also averaged over the three block lengths. The time constraint is set to $1s$.

Observation 6. *In Table VI, the inference time does not scale when the FLOPs requirement of a model is lower, e.g.,* `Pipeline 3`*. However, we observe* $\sim 1672\times$ *and* $\sim 7.6\times$ *of speed up for a A100 GPU as compute-hardware than i7 CPU and 2080 Ti GPU, respectively, for* `Pipeline 4` *at Stage 1.*

Performance of pipeline selection Algo. 1. In this case, we validate the proposed FLOP-efficient pipeline selection Algo. 1 for both *Stage 1* and *Stage 2*. For tractable analysis, we consider only CPU (Intel i7) and GPU (NVIDIA A100) as the compute-hardware platforms, which we refer to as *CPU* and *GPU regimes*. In Fig. 8(a), we show the selected pipelines from the Algo. 1 are `Pipeline 3` and `Pipeline 4` for the CPU and GPU regimes, respectively, for *Stage 1*. However, for *Stage 2*, the Algo. 1 chooses `Pipeline 4` for both the regimes as it satisfies both regimes' FLOPS requirement with highest importance vector, see Fig. 8(c). Next, in Figs. 8(b) and 8(d), we run all the available pipelines in both the regimes to test which one gives the best performance while satisfying a timing threshold of $1s$. In Algo. 1, we consider the machine's FLOPS configuration as $Thres_{FLOP}$, and hence the time constraint is taken as $1s$. Note that when the FLOP-intensive `Pipelines 2` and `4` run in the CPU regime, they result in longer inference time. Hence, the same `Pipelines 3` and `4` are the best ones at *Stage 1* for the CPU and GPU regimes, respectively (see Fig. 8(b)). This demonstrates the efficacy of Algo. 1 for selecting the optimum pipeline depending on the FLOPS of the compute-hardware platform.

Observation 7. `Pipeline 3` *and* `4` *for Stage 1, and* `Pipeline 4` *for Stage 2 are selected through Algo. 1 as they best perform under the imposed timing constraints. This result is specific to the supported FLOPS of the compute-hardware (CPU or GPU), see Figs. 8(a) and (c).*

Observation 8. `Pipeline 4` *detects the anomaly within* $\sim 40ms$ *and predicts the modulation scheme of it within* $\sim 3ms$ *on an average in a GPU regime, while giving* $\sim 98\%$ *and* $\sim 70\%$ *accuracy, respectively (see Figs. 8(b) and (d)).*

Table VI: Inference times (s) of *Stage 1* on different computing platforms for the OTA-Cellular dataset.

Block Length	Intel i7 CPU (38.05 GFLOPS)				2080Ti GPU (13.45 TFLOPS)			A100 GPU (312 TFLOPS)		
	P1	P2	P3	P4	P2	P3	P4	P2	P3	P4
131072	0.97	68.80	0.0003	63.23	0.27	0.018	0.29	0.028	0.0004	0.035
262144	1.78	22.86	0.0003	20.97	0.28	0.036	0.28	0.025	0.00045	0.033
524288	3.96	11.99	0.0004	108.88	0.30	0.076	0.31	0.031	0.00047	0.05

VII. Conclusions

Underlay signals within a high-power baseline transmission can carry information, and are difficult to detect. ICARUS shows how detection of such hidden signals could be performed, especially if these anomalies exhibit noise-like properties. ICARUS incorporates the idea of using both CSP features and IQ samples to detect such anomalous transmission by using both signal processing and a 'black box' of machine learning with information fusion. We also propose an algorithm to dynamically select the appropriate pipeline to detect an anomaly and recognize its modulation scheme, under available hardware constraints. ICARUS is validated on diverse datasets to demonstrate improvement over state-of-the-art methods. Future scopes include the fusion of CSP features with privacy preserving representation of signal such as spectrogram and extension towards other computing platforms. The authors have provided public access to their data at [31].

Acknowledgements

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via [2021-2106240007]. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

REFERENCES

[1] S. Ma, Z. Sun, A. Ye, S. Huang, and X. Zhang, "Detection of anomaly signal with low power spectrum density based on power information entropy," in *Communications, Signal Processing, and Systems*, Q. Liang, W. Wang, X. Liu, Z. Na, M. Jia, and B. Zhang, Eds., Singapore, 2020, pp. 1517–1527.

[2] D. Chang, G. Bhat, U. Ogras, B. Bakkaloglu, and S. Ozev, "Detection mechanisms for unauthorized wireless transmissions," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 23, no. 6, 2018.

[3] Q. Feng, Y. Zhang, C. Li, Z. Dou, and J. Wang, "Anomaly detection of spectrum in wireless communication via deep auto-encoders," *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3161–3178, 2017.

[4] J. ho Bang, Y.-J. Cho, and K. Kang, "Anomaly detection of network-initiated lte signaling traffic in wireless sensor and actuator networks based on a hidden semi-markov model," *Computers Security*, vol. 65, pp. 108–120, 2017.

[5] Federal Aviation Administration, "Federal Register/Vol. 86, No. 234," 2021. [Online]. Available: https://www.govinfo.gov/content/pkg/FR-2021-12-09/pdf/2021-26777.pdf

[6] A. Napolitano and I. Perna, "Cyclic spectral analysis of the gps signal," *Digital Signal Processing*, vol. 33, pp. 13–33, 2014.

[7] W. Gardner and C. Spooner, "Signal interception: performance advantages of cyclic-feature detectors," *IEEE Transactions on Communications*, vol. 40, no. 1, pp. 149–159, 1992.

[8] G. B. Giannakis and V. Madisetti, "Cyclostationary signal analysis," in *Digital Signal Processing Handbook*. Citeseer, 1998, vol. 31, pp. 1–17.

[9] K. Youssef, L. Bouchard, K. Haigh, J. Silovsky, B. Thapa, and C. V. Valk, "Machine learning approach to rf transmitter identification," *IEEE Journal of Radio Frequency Identification*, vol. 2, no. 4, pp. 197–205, 2018.

[10] M. Belgiovine, K. Sankhe, C. Bocanegra, D. Roy, and K. R. Chowdhury, "Deep learning at the edge for channel estimation in beyond-5g massive mimo," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 19–25, 2021.

[11] S. Hanna, S. Karunaratne, and D. Cabric, "Open set wireless transmitter authorization: Deep learning approaches and dataset considerations," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 59–72, 2021.

[12] "NVIDIA A100 TENSOR CORE GPU," https://docs.nvidia.com/gameworks/content/developertools/desktop/analysis/report/cudaexperiments/kernellevel/achievedflops.htm, accessed: 2022-07-30.

[13] F. Benedetto, G. Giunta, and L. Pallotta, "Unauthorized access detection in underlay cognitive satellite networks," *IEEE Networking Letters*, vol. 3, no. 4, pp. 181–185, 2021.

[14] J. Hofmann, A. Knopp, C. M. Spooner, G. Minelli, and J. Newman, "Spectral correlation for signal presence detection and frequency acquisition of small satellites," *Aerospace*, vol. 8, no. 2, 2021.

[15] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, "Saife: Unsupervised wireless spectrum anomaly detection with interpretable features," in *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2018, pp. 1–9.

[16] N. Soltani, V. Chaudhary, D. Roy, and K. Chowdhury, "Finding Waldo in the CBRS Band: Signal Detection and Localization in the 3.5 GHz Spectrum," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2022.

[17] C. M. Spooner and A. N. Mody, "Wideband cyclostationary signal processing using sparse subsets of narrowband subchannels," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 162–176, 2018.

[18] X. Zhang, S. Li, and Z. Chen, "A dsss signal detection method based on wavelet decomposition and delay multiplication," in *IEEE Asia-Pacific Conference on Antennas and Propagation (APCAP)*, 2018, pp. 341–343.

[19] F. Liu, M. W. Marcellin, N. A. Goodman, and A. Bilgin, "Compressive detection of direct sequence spread spectrum signals," *Electronics Letters*, vol. 54, no. 24, pp. 1379–1381, 2018.

[20] B. Salehi, G. Reus-Muns, D. Roy, Z. Wang, T. Jian, J. Dy, S. Ioannidis, and K. Chowdhury, "Deep learning on multimodal sensor data at the wireless edge for vehicular network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7639–7655, 2022.

[21] D. Roy, Y. Li, T. Jian, P. Tian, K. R. Chowdhury, and S. Ioannidis, "Multi-modality Sensing and Data Fusion for Multi-vehicle Detection," *IEEE Transactions on Multimedia*, 2022.

[22] Evolved Universal Terrestrial Radio Access (E-UTRA), "Physical Channels and Modulations," 2019. [Online]. Available: document3GPPTS36.211V15.6.0

[23] R. Roberts, W. Brown, and H. Loomis, "Computationally efficient algorithms for cyclic spectral analysis," *IEEE Signal Processing Magazine*, vol. 8, no. 2, pp. 38–49, 1991.

[24] W. Brown and H. Loomis, "Digital implementations of spectral correlation analyzers," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 703–720, 1993.

[25] J.-M. Perez-Rua, V. Vielzeuf, S. Pateux, M. Baccouche, and F. Jurie, "MFAS: Multimodal Fusion Architecture Search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] T.-W. Chin, R. Ding, C. Zhang, and D. Marculescu, "Towards efficient model compression via learned global ranking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[27] "POWDER: Platform for Open Wireless Data-driven Experimental Research," *Computer Networks*, vol. 197, p. 108281, 2021.

[28] POWDER srslte-otalab profile, https://www.powderwireless.net/show-profile.php?profile=258a8000-3b4e-11ec-84f8-e4434b2381fc.

[29] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[30] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[31] "ICARUS Dataset," https://genesys-lab.org/ICARUS.