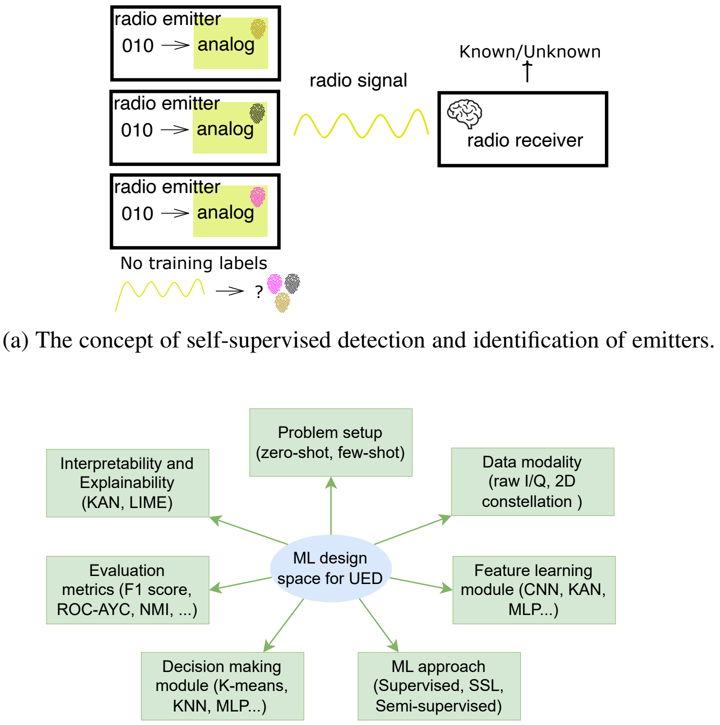
# Design Principles of Zero-Shot Self-Supervised Unknown Emitter Detectors

Mikhail Krasnov*†, Ljupcho Milosheski*†, Mihael Mohorčič*†, Carolina Fortuna*
* Jožef Stefan Institute, Ljubljana, Slovenia
†Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
Email: {mikhail.krasnov, ljupcho.milosheski, miha.mohorcic, carolina.fortuna}@ijs.si

***Abstract*—Robust situational awareness in contested wireless environments requires the ability to detect unauthorized or hostile emitters without prior knowledge of their signatures. Existing studies on Unknown Emitter Detection (UED) and identification are hindered by their reliance on labeled or proprietary datasets, unrealistic assumptions (e.g., all samples containing identical transmitted messages), or a lack of systematic evaluation across architectures and design dimensions. In this work, we identify the *design dimensions* of machine learning based UED and introduce a *structured workflow* for developing such systems. The workflow, suitable for guiding studies and end-to-end label-free UED designs, consists of four key components: (a) data modality, (b) feature learning module, (c) machine learning approach, and (d) decision-making module. Using this workflow, we examine the design principles of UED in two distinct transmission scenarios: same messages scenario (SMS) and different messages scenario (DMS). Our investigation yields several key findings. (a) Under realistic DMS, the 2D constellation data modality achieves up to a 20-percentage-point improvement in ROC-AUC compared to the conventional raw I/Q representation. (b) Kolmogorov-Arnold Networks (KANs) provide interpretable representations while achieving performance comparable to CNNs in both scenarios (c) for DMS, the best-performing configuration combines SVD-initialized KANs with a deep clustering approach and the 2D constellation modality, improving ROC-AUC by up to 20 percentage points over standard KANs with identical workflow components. (d) through analysis of the decision-making module, we determine the optimal number of clusters for environments with varying numbers of known emitters.**

***Index Terms*—emitter detection, self-supervised, machine learning, zero-shot, feature extraction**


radio emitter
010 → analog
radio emitter
010 → analog
radio emitter
010 → analog
radio signal
Known/Unknown
radio receiver
No training labels
→ ?
Problem setup (zero-shot, few-shot)
Interpretability and Explainability (KAN, LIME)
Data modality (raw I/Q, 2D constellation )
ML design space for UED
Evaluation metrics (F1 score, ROC-AYC, NMI, ...)
Feature learning module (CNN, KAN, MLP...)
Decision making module (K-means, KNN, MLP...)
ML approach (Supervised, SSL, Semi-supervised)

(a) The concept of self-supervised detection and identification of emitters.

(b) Design space for building deep learning model for detection and identification of emitters.

Fig. 1: The design space for ML detection and identification of known/unknown emitters.

## I. Introduction

The management and operation of wireless communication networks increasingly depend on situational and spectrum awareness to dynamically adapt to environmental variations and user/application-imposed demands. Spectrum awareness and network security are envisioned as part of the main research directions [1] for the future Open Radio Access Networks (O-RAN) [2], but the same techniques can also be used in interference management, regulatory compliance and spectrum optimization. Together with the development of modulation and technology classification [3], they contribute to the creation of efficient, spectrum-aware, wireless communication networks [4]. One of the key challenges related to spectrum awareness lies in the reliable detection and identification of emitters within the communication range, on the conceptual level depicted in Figure 1a. It relies on robust signal processing techniques and increasingly leverages advanced machine learning approaches. In this respect, the research community distinguishes between two complementary processes within the broader context of devices recognition: the Specific Emitter Identification (SEI) and the Unknown Emitter Detection (UED).

SEI focuses on recognizing and classifying known emitters under a closed-set assumption based on unique, device-specific signal alterations such as transient behaviors, hardware-induced imperfections, or modulation patterns allowing precise attribution of transmissions to individual sources [5]. This makes SEI particularly well suited for network security applications [6]. The aim is to distinguish emitters that are allowed to operate in a certain wireless network, in which case they are transmitting useful information and can be referred to as transmitters. In contrast, UED [7] is working under open-set assumption and it aims to identify signals that deviate from any known emitter profile, detecting and isolating novel, spoofed, or adversarial sources that fall outside the established feature space of the known emitters. In network security application, UED is used to detect and reject or flag unrecognized signals to prevent their misclassification by SEI. As such, these processes form a closed-loop detection and identification system whereby SEI provides fine-grained, instance-level identification within a known feature space used for training, whereas UED ensures robustness and adaptability by flagging outliers and prompting model updates or human review for inclusion of novel devices, together maintaining situational awareness and security in dynamic or contested radio environments.

Most of the existing approaches are focusing on the closed-set classification problem. Such approaches consider that all the emitters are known and labeled transmission data is available for each of them for developing supervised classification models [8], [9], reflecting the authentication use case of known devices. Some of these models achieved remarkable performance in distinguishing known devices [10]–[12], even when thousands of such devices [6] exist in the network. However, in real-world operating scenarios with the rising volume and types of wireless devices, new transmitters could appear regularly in the network, which could pose significant security and management challenges. In such open-set setup there is little or no labeled data available for training, making the supervised learning for model development unsuitable. The open-set challenges are divided into two distinct subsets: few-shot learning [13]–[15] and zero-shot learning [16]. Few-shot learning follows the supervised evaluation protocol but operates under severe label constraints, with the number of labeled samples ranging from just a few (1–20) [14] to several hundred [17]. In contrast, zero-shot learning aims to detect emitters that are entirely absent during model training [18], [19].

We notice that, overall, the approach to developing a system capable of learning to distinguish emitters based on device-specific signal alterations largely depends on the application and systematic consideration of all the existing design choices, as visualized in Figure 1b. The problem set-up considering zero-shot and few shot learning is one dimension. Other dimensions relate to data modality, feature learning, ML approach, etc. The selection of data modality may significantly improve the final performance, but it also influences the complexity of the feature learning module. The design of the feature learning module influences many of the other design choices, such as performance, complexity, explainability, and interpretability. The existing model architectures such as Convolutional Neural Networks (CNN) [16], transformers [10], Long Short-Term Memory (LSTM) recurrent neural networks [20] and/or combinations with Multilayer Perceptron (MLP) [21] have been already proven in related domains and they usually provide satisfactory performance [22]. However, designing a dedicated model for the radio signal specifics and predefined use cases can lead to significant complexity reduction [23].

The choice of the neural architecture for feature learning also affects its explainability and interpretability. For the deep learning models, there are several explainability approaches, such as Local Interpretable Model-agnostic Explanations (LIME) [24], SHapley Additive exPlanations) (SHAP) [25], and saliency maps [26], whereas the issue of model interpretability remains largely unaddressed. This is primarily due to the complexity and opacity of the deep learning modules employed, typically including large CNNs or self-attention-based architectures [10], which make it difficult to understand or explain model decisions, while also lacking interpretability. So far, the interpretable models are the classic machine learning (ML) approaches, such as Support Vector Machine (SVM), and logistic regression [27], which are significantly outperformed by the deep learning models, while interpretability is overlooked in most of the recent works. However, due to the potential use in security-related applications, such as device authentication, interpretability of the detection and identification of emitters remains an important design choice.

For zero-shot set-ups, Self-Supervised Learning (SSL) [28] techniques are the most suitable ML approaches. SSL represents a family of machine learning methods, which create supervision signal without using ground truth labels but through data manipulation techniques such pseudo-labeling, augmentations, reconstruction etc. By formulating and solving pretext tasks derived directly from the data, SSL enables the development of models that generalize well without relying on manual labeling. This makes it particularly well-suited for open-set detection and identification of emitters [17], [21], [29], closely aligning with realistic deployment conditions, where the diversity of emitters and the dynamic nature of the spectrum make comprehensive labeling impractical.

In this work, we identify the *design dimensions* of machine learning based UED as depicted in Figure 1b: 1) problem set-up, 2) data modality, 3) feature learning, 4) ML approach, 5) decision making, 6) evaluation metrics and 7) explainability and interpretability. Focusing on realistic environments where no prior information is available, we consider a *zero-shot* set-up that inherently requires a SSL approach. Consequently, we introduce a *structured workflow* for developing such systems. The workflow, suitable for guiding studies for end-to-end label-free UED designs. Using this workflow, we examine the design principles of UED in two distinct transmission scenarios: same messages scenario (SMS) and different messages scenario (DMS) enabled by two distinct datasets [37], [38]. Throughout the study we consider two possible data modalities, three different feature learning and three different SSL approaches evaluated through several metrics including also explainability and interpretability considerations. For replicability purposes, the scripts are available as open source[1]. We summarize our contributions as follows:

- We formalize a structured, self-supervised workflow for zero-shot UED on unlabeled data, consisting of four key components: a) data modality, b) feature learning module, c) ML approach, and d) decision-making module.

[1] https://github.com/sensorlab/ZeroUED

TABLE I: Comparison of related works on SEI and UED tasks with focus on semi-supervised, unsupervised and self-supervised learning approaches

| Ref. | Problem Setup | Data Modality | ML Approach | Feature Extractor | Interpret / Explain | Supervision Setup | Results / Notes |
|---|---|---|---|---|---|---|---|
| [30] | Self-supervised SEI/UED | Raw I/Q | CL | CNN | DMM level | No. of training emitters | 93% F1 |
| [15] | Self-supervised SEI | Raw I/Q | CL (Viewmaker) | CNN | None | Few-shot fine-tuning | 83% ACC |
| [31] | Unsupervised SEI | Raw I/Q | DC | CNN | DMM level | Unsupervised | 83% F1 |
| [14] | Masked AE (Reconstruction) | Raw I/Q | AE | CNN | None | Few-shot fine-tuning | 83% ACC |
| [13] | Hybrid Few-shot SEI | Raw I/Q | Hybrid (AE + CL) | CNN | None | Few-shot fine-tuning | 90% ACC |
| [32] | Few-Shot SEI | Raw I/Q | Assymetric MAE | CNN | None | Few-shot fine-tuning | 93% ACC |
| [33] | Few-Shot SEI | Raw I/Q | SimCLR | CNN | None | Few-shot fine-tuning | 70% ACC |
| [34] | Few-Shot SEI | Raw I/Q | SimSiam CL | CNN | None | Few-shot fine-tuning | 80% ACC |
| [35] | Few-Shot SEI | Raw I/Q | CL | CNN | None | Few-shot fine-tuning | 95% ACC |
| [36] | Few-Shot SEI | 2D constellation | SimCLR | CNN | None | Few-shot fine-tuning | 90% ACC |
| **Ours** | Self-supervised zero-shot UED | Raw I/Q, 2D constellation | CL, DC, AE | CNN, KAN | DMM and FE levels | Unsupervised | 85% F1, 94% ACC |

**CL** – Contrastive Learning, **DC** – Deep Clustering, **AE** – Autoencoder, **DMM** – Decision-Making Module, **FE** – Feature Extractor

- Through the lens of the proposed workflow, we investigate the design principles of designing a an UED on two transmission scenarios: Same Messages Scenario (SMS) and Different Messages Scenario (DMS), two data modalities, two neural architectures, including KAN designed for interpretability, and three SSL ML approaches.
- During our investigation, we find the following. a) Under the realistic DMS scenario, 2D constellation data modality shows up to 20 p.p. ROC-AUC metric improvement compared to conventional raw I/Q representation. b) Kolmogorov-Arnold networks (KANs) are interpretable and have performance comparable to CNNs, a convenient alternative that suffers from ”black-box” design, across SMS and DMS. c) We observe that for DMS, the best performing configuration is SVD initialized KAN with deep clustering approach and 2D constellation data modality. This configuration improves ROC-AUC performance up to 20 p.p. compared to the standard KAN with the same other parts of the workflow in DMS. d) Investigating the decision-making module, we determine the optimal number of clusters for environments with different numbers of known emitters.

This rest of the paper is organized as follows. Sec. II summarizes the related work, Sec. III provides the background and problem statement, while Sec. IV elaborates on the design choices of the emitter detection and identification framework. Sec. V describes the evaluation methodology and Sec. VI analyzed the results. Finally, Sec. VII concludes the paper.

## II. Related work

When focusing on semi-supervised and unsupervised deep learning approaches for detection and identification of emitters the number of related studies is rather low. Table I summarizes the related works focused on SEI and UED tasks based on semi-supervised and unsupervised deep learning approaches, structured along the design space presented in Figure 1b. As shown, prior works are predominantly focusing on few-shot self-supervised SEI using raw I/Q data with Contrastive Learning (CL) [15], [33]–[35] or Auto Encoder (AE) [13], [14], [32], [39], with CNN-based feature extractors being the most common backbone. Besides the necessity for labels, the interpretability of these approaches, if supported at all, is limited to the decision-making stage, often via distance-based metrics, while their feature extractors remain opaque. There is also only one work for self-supervised SEI utilizing 2D constellation data modality [36], which is also designed for few-shot set-up limiting its applications, whereby the authors do not demonstrate either theoretically nor experimentally the need of this transformation. In this work, we investigate the design space of approaches for self-supervised UED without using labels during the training, which is a more challenging problem. We prove theoretically and experimentally that in completely label-less case the scenarios of signal transmission greatly affect the correct design choice. We show the advantage of using 2D constellation data modality when transmitted messages are different.

Despite differences in methodology, only two studies [30], [31] are fully unsupervised, requiring no labels at any stage. In [30], as part of the study, authors demonstrate 91% accuracy in a zero-shot UED task using a contrastive learning framework, underscoring the effectiveness of SSL approaches in overcoming the labeling bottleneck. However, this work uses a proprietary non-public dataset, which makes the results non-replicable. Authors of [31], on the other hand, use publicly available WiSig dataset [38], but it only contains signals transmitting the same messages, making the setup less realistic. They achieve an F1 score between 80% and 87% on self-supervised UED task depending on the number of emitters in the training set versus all emitters in the testing set. All above-mentioned approaches lack interpretability of the features learning module.

In this work, we aim to fill the identified gaps by conducting experiments on two publicly available datasets [37], [38] supporting also the investigation of the impact of transmitting same and different messages. We are focusing on a zero-shot self-supervised open-set UED task and in contrast to related works we introduce interpretability both at the embedding and feature-extractor levels, providing the first systematic performance-versus-interpretability comparison across Deep Clustering (DC),

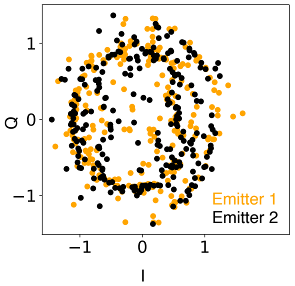


Fig. 2: Yellow dots represent an arbitrary signal from the first emitter from the WiSig Dataset [38], and black dots stand for the arbitrary signal from the second Emitter. The signals carry the same message and differ only by environmental noise and hardware imperfections.

Contrastive Learning (CL), and Auto Encoder (AE) paradigms.

## III. Background Assumptions and Problem Statement

In this section, we mathematically describe device-specific signal alterations that are exploited by SEI and UED tasks and formulate the problem statement of zero-shot self-supervised unknown emitter detection.

### A. RF Fingerprint

In this work, we focus on detection and identification of emitters using Orthogonal Frequency Division Multiplexing (OFDM) modulation scheme. In this modulation scheme, each transmitted symbol corresponds to a complex-valued point in the I/Q (In-phase/Quadrature) constellation diagram, which represents the signal's amplitude and phase characteristics. Figure 2 shows an example constellation diagram of the OFDM modulation scheme with the axes representing the I and Q components of signal points. Each OFDM signal consists of clean modulated message and distortions caused by hardware imperfections and channel noise. While channel noise is affected by environment conditions, hardware imperfections, often referred as radio frequency (RF) fingerprint, are caused by tiny imperfections of individual emitter analog components, making them emitter specific. In this study, we explore these hardware imperfections for distinguishing known and unknown emitters. Simplifying, RF fingerprint [40] can be modeled as:

$$x(t) = y(t) \cdot e^{j(2\pi(f+\Delta f)t+\phi(t))}, \tag{1}$$

where $\Delta f$ and $\phi(t)$ are frequency and phase offsets in signal $x(t)$ caused by hardware imperfections. Treating them as small perturbations, we can apply first order Taylor decomposition to divide signal into clean and corrupted components:

$$\begin{aligned} &x(t) = y(t) \cdot e^{j2\pi ft + j(2\pi\Delta ft+\phi(t))} \\ &x(t) \approx y(t) \cdot e^{j2\pi ft} + y(t) \cdot e^{j2\pi ft} j(2\pi\Delta ft + \phi(t)) \\ &x(t) \approx x_{symbols}(t) + r_{emitter}(t) \\ &|r_{emitter}(t)| \ll |x_{symbols}(t)| \end{aligned}, \tag{2}$$

where $x_{symbols}(t)$ represents clean signal without hardware imperfections and $r_{emitter}(t)$ stands for the RF fingerprint part. Figure 2 illustrates the distinctions created by the fingerprint $r_{emitter}(t)$ for two measured transmissions of the same symbols from the WiSig Dataset [38] by two different emitters, where the yellow dots represent Emitter 1 and the black dots represent Emitter 2.

Clearly, the challenge in distinguishing $r_{emitter}(t)$ in the case of known transmitted messages differs from the case of more general unknown transmitted messages, as we show in this work by considering two datasets. The physical (PHY) layer of transceiver is responsible for sending the data over the wireless medium by modulating it onto a carrier signal. It consists of a preamble, a header and a payload. The preamble and header have fixed length, whereby the first also has a fixed repeating sequence of bits used for synchronization while the second contains varying control information like the transmission rate and payload length. These are followed by a varying payload of the length specified in a header containing a medium access control (MAC) frame with the actual data and a trailer with error-checking information. Taking this into account, we consider in this work two signal receiving scenarios: *Same Messages Scenario* for the cases when only the preamble symbols are considered and *Different Messages Scenario* when all received symbols are considered.

### B. Same Messages Scenario (SMS)

If only looking at a preamble, a signal with the same content is transmitted by every emitter in the network. The preamble signal coming from the $i^{\text{th}}$ emitter, using Eq. 2, can be described as:

$$x_i(t) = x_{symbols}(t) + r_{emitter,i}(t), \tag{3}$$

where $x_{symbols}(t)$ is the constant preamble signal, and $r_{emitter,i}$ is the $i^{\text{th}}$ emitter specific signal. Then, the difference between the preamble signals from different emitters is higher than the variation inside each emitter:

$$\begin{aligned} &x_i(t) - x_i'(t) \approx 0 \\ &x_i(t) - x_j(t) = r_{emitter,i}(t) - r_{emitter,j}(t) \neq 0. \end{aligned} \tag{4}$$

This makes the self-supervised task easier because of linearly separable classes.

### C. Different Messages Scenario (DMS)

In a more generic use case when we do not know the exact technology and protocol used, we may only have access to some random part of the received frame, which means that the clean signal $x_{symbols}$ is also unknown. The signal can now be expressed as:

$$x_i(t) = \hat{x}_{symbols}(t) + r_{emitter,i}(t), \tag{5}$$

where $\hat{x}_{symbols}(t)$ is random. Thus, the variation within the emitter becomes the same as differences between emitters because the emitter specific part represents a small perturbation compared to the clean symbol as stated in Eq. 2 and expressed as follows:

$$\begin{aligned} &x_i(t) - x_i'(t) \approx \hat{x}_{symbols}(t) - \hat{x}'_{symbols}(t) \\ &x_i(t) - x_j(t) \approx \hat{x}_{symbols}(t) - \hat{x}''_{symbols}(t) \end{aligned}, \tag{6}$$

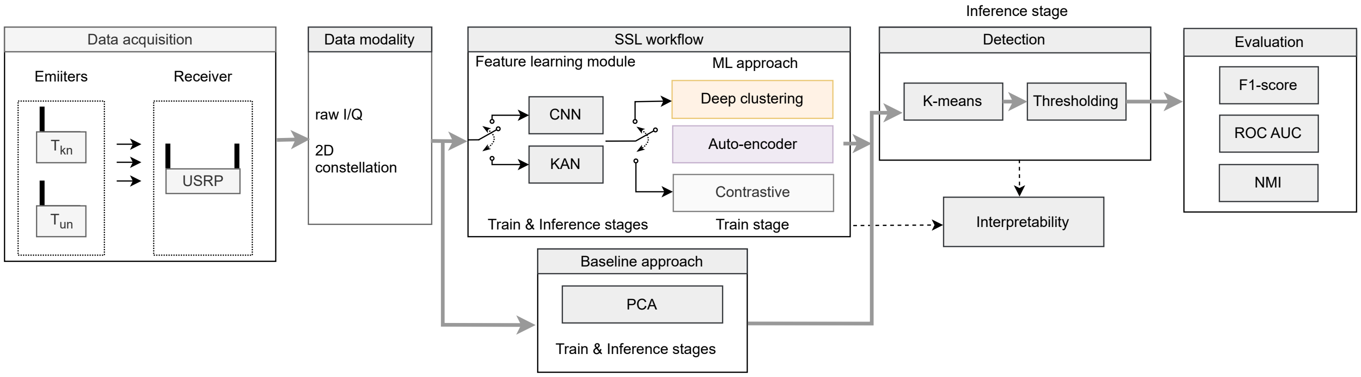


Fig. 3: Formal self-supervised ML based UED workflow.

where $\hat{x}_{symbols}(t), \hat{x}'_{symbols}(t), \hat{x}''_{symbols}(t)$ are random signals. This more reasonable assumption makes self-supervised task much harder.

### D. Problem Statement

In this work, we formulate the problem of UED as a *zero-shot learning task* within a *self-supervised learning setting*. We develop an approach that extracts the RF fingerprint part from signals, as shown in Eq. 2, grouping identical emitters and separating different ones in the feature space. The objective is to learn a composite function

$$\Phi : X_{\text{in}} \to L, \quad \text{where } L = \{\text{known } (0), \text{unknown } (1)\} \tag{7}$$

that maps raw RF signal samples into corresponding decisions indicating whether a signal originates from a previously seen or an unseen emitter. This is done without access to ground-truth labels during training and under the constraint that the number of emitters in the training set is not known apriori.

The composite function $\Phi(X_{\text{in}})$ from Eq. (7) can be decomposed into two modular sub-functions as per Eq. (8): the embedding function $F$, which transforms raw RF data into a projection (i.e. embedding) in a discriminative feature space, and the detection function $M$, which determines whether an embedded sample belongs to a known or an unknown emitter.

$$\text{L} = \Phi(X_{\text{in}}) = M(F(X_{\text{in}})) \tag{8}$$

*1) Embedding Function:* Let $F$ be the embedding function trained in a self-supervised manner to project input signals into a latent space $F : X_{\text{in}} \to \mathbb{R}^d$, where $X_{\text{in}}$ denotes the input signal domain, and $\mathbb{R}^d$ is the latent feature space of dimension $d$. The output embeddings should cluster tightly for samples from the same emitter while remaining well-separated for different emitters. In other words, the latent space has to represent fingerprint components of signals from Eq. 2.

*2) Detection Function:* The detection function $M$ operates over the embedded space to determine whether a given sample is associated with a known or unknown emitter $M : \mathbb{R}^d \to L$.

## IV. Design of the UED workflow

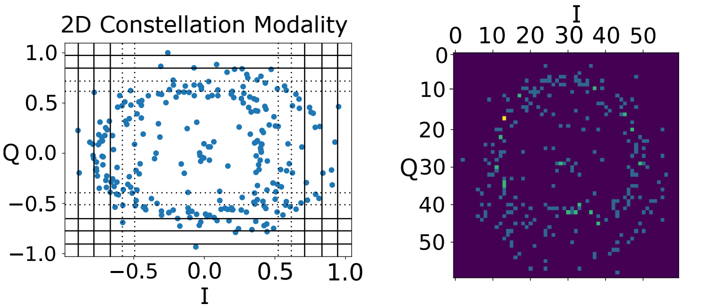
To address the problem identified in Sec. III-D, we formalize the structured workflow for ML-based UED as depicted in Figure 3. The most important components of this workflow are: a) data modality, b) feature learning module (FLM), c) machine learning approach and d) decision making module. Data modality component transforms the input signal to the chosen representation: raw I/Q or 2D constellation. The FLM and machine learning approach represent the self-supervised learning block. The selected FLM: KAN or CNN is trained by one of the machine learning approaches: DC,CL, AE in a self-supervised manner. Principal component analysis (PCA) is a prominent baseline in unsupervised learning, but it does not fit within a feature learning model-ML approach scheme. Thus, we treat it as an individual feature-learning module with a specialized training procedure. The decision making module is the final stage of the ML workflow, where a classification decision is made based on extracted features. The other components include evaluating metrics and interpreting the FLM and DMM. While not being directly involved, they are essential for the performance measuring and analysis of the system. The following subsections provide a detailed design description of each component.

### A. Transmitting devices

In this study, devices transmit signals following two scenarios: the Same Messages Scenario as described in Sec. III-B and Different Messages Scenario as per Sec. III-C. The first scenario, in which all emitters transmit the same content all the time, is considered in the case where one has access to the preamble. The second scenario with different content is more realistic but also more challenging. The particular scenario affects the best design choices of Unknown Emitter self-supervised zero-shot detectors.

### B. Data Modality

In the receiver, the RF signal is discretized with the same time step: $x[i] = x(i\delta t)$, where $x[i]$ is a complex value denoting I and Q components. The way it is processed to be fed to the features learning module $F$ is referred to as a data modality, an important part of the design space. In this work, we investigate two types



(a) 2D constellation Modality construction from Raw I/Q with Grid Size $K = 60$

(b) 2D constellation Modality Single Channel Image with Grid Size $K = 60$

Fig. 4: 2D constellation Modality Description

of data modalities: raw I/Q data and a 2D constellation. The raw I/Q is the convenient choice for OFDM data; each observation $x[i] = x[i]^I + jx[i]^Q \in \mathbb{C}^N$ is represented as a two-channel input: $X_{\text{in}}^{0i} = x^I[i], \quad X_{\text{in}}^{1i} = x^Q[i]$. In this study, we argue that the raw I/Q data modality produces poor quality under the realistic scenario, assuming different transmitting messages as described in Sec. III-C because of the prevalence of semantic information $x_{symbols}(t)$ over the fingerprint information $r_{emittter}(t)$ according to Eq. 5. To address this issue, one can transform the data in a time-invariant manner:

$$T(X_{\text{in}}) = T(\sigma(X_{\text{in}})), \sigma(X_{\text{in}})^{ij} = X_{\text{in}}^{i,\sigma(j)}, \tag{9}$$

where $T$ is a transformation operation over raw I/Q data, and $\sigma$ is a permutation operation over the time axis. The $T(X_{\text{in}})$ tensor does not contain any semantic information, because all messages have the same representation, assuming that transmitted signals are random. A straightforward way to provide the time-invariant property is to represent raw I/Q samples as a 2D constellation data. Firstly, the data is scaled to $[-1, 1]$ range:

$$X_{\text{in norm}} = \frac{X_{in}}{max(|X_{in}|)}. \tag{10}$$

Secondly, the procedure counting I/Q observations over cells created by the grid on a 2D plane is applied as depicted in the Figure 4a:

$$\begin{aligned} T(X_{\text{in}})^{i,j} = \#\{k : \frac{X_{\text{in norm}}^{0k} + 1}{2} \in [i\epsilon, (i+1)\epsilon], \\ \frac{X_{\text{in norm}}^{1k} + 1}{2} \in [j\epsilon, (j+1)\epsilon]\}/N, \\ \text{where } i, j \in [0, K-1], \epsilon = \frac{1}{K}, \end{aligned} \tag{11}$$

where $K$ is the grid size and $N$ is the number of I/Q observations. The resulting tensor looks like a heat map shown in Figure 4b.

### C. Feature Learning Module

The FLM $F$ is a core part of self-supervised zero-shot unknown emitter detectors. Based on the available literature [15], [30], Convolutional Neural Networks (CNNs) based architectures are employed for FLM s. To provide a more in-depth exploration of design principles of this component, we also study new, interpretable KAN based extractors and PCA baselines.

With respect to CNNs, we investigate 1D and 2D architectures for raw I/Q and 2D constellation data modalities, respectively. The CNN-1D two-channel network, adopted from [41], consists of four blocks, each containing convolution, batch normalization, and max pooling operations. 2D CNN is similar to CNN-1D but with 2D spatial operations. However, CNN architectures are not interpretable, as their decisions are not understandable by humans.

As a novel, interpretable type of architecture, we investigate Kolmogorov-Arnold Networks (KANs) [42] in the design space to bring interpretability to unknown emitter detectors. The single KAN layer can be described as an MLP layer, where activations are replaced with learnable functions and fixed weights on the edges. The learnable activation functions are often drawn from the k-order spline functions [43], which can be parameterized as linear combinations of basic B-splines. These linear coefficients are then optimized during the training process. We develop a single-layer Efficient KAN version as:

$$F^{(j)}(x_1, ..x_N) = \sum_{i}^{N} \psi_{ij}(x_i), \tag{12}$$

where $j \in [1, N^*]$ is a coordinate of the feature space, $(x_1, ..., x_N)$ is an input tensor $X_{in}$ divided into individual nodes, and $\psi_{ij}$ are functions which are represented as:

$$\psi_{ij}(x) = w_{ij}^{(b)} silu(x) + w_{ij}^{(s)} \sum_{k=1}^{G+4} \gamma_{i,j}^{(k)} \phi_{i,j}^{(k)}(x), \tag{13}$$

where $\phi_{i,j}^{(k)}(x)$ are fourth-order B-splines, $silu$ is Sigmoid Linear Unit (SiLU) activation function and $G$ is a grid size.

For CNNs and KANs, we also investigate SVD initialized modifications of those architectures. Let $F$ be a FLM (CNN or KAN), which transforms the input data to $d$-dimensional features space. We initialize a linear layer with SVD weights with the number of latent components $d$ and combine this linear layer with the initial FLM $F$:

$$F'(x) = \frac{F(x)}{10} + L(x), \tag{14}$$

where $L$ is the linear layer initialized with SVD weights. Under conditions of sparse and limited data, like in 2D constellation data modality, this modification aims to provide additional knowledge of the linear nature of the RF fingerprint (see Eq. 2) and boost the performance of a complex self-supervised approach, which might degrade under the conditions as mentioned earlier [44], [45]. This transformation does not move the KAN model out of KAN architectures, because adding linear functions to cubic splines produces cubic splines.

We also investigate PCA - a prominent baseline in unsupervised learning. It builds a linear transformation that extracts low-dimensional, most informative features based on their contributions to total variation.

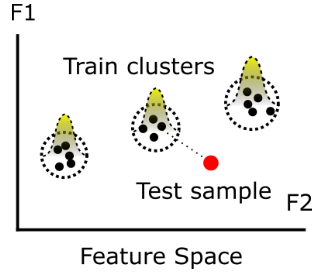


Fig. 5: Decision Making on a test sample. Three circles represent training clusters with probability densities equal to zero outside the circle. The red point represents a test sample, which, in this case, is an outlier and is predicted as *unknown*. $F1, F2$ are axes of demonstrative 2D features space.

### D. Machine Learning Approach

According to Sec. II, there are three main ML approaches used in self-supervised SEI/UED: Deep Clustering, Auto Encoder, and Contrastive Learning. These approaches are used to train the FLM and are typically followed by a decision making module at inference time. In this paper, we investigate the design dimension of the ML approach by evaluating the performance trade-offs of the three selected approaches.

*1) Deep Clustering:* DC [46] is an ML approach in self-supervised learning, which is employed in [31]. The training procedure consists of two alternating stages. During the first stage, features are extracted from the training set using the current state of FLM and clustered using the K-means algorithm, assigning cluster numbers (pseudo-labels) to training samples. At the second stage FLM is updated using the classification objective with those pseudo-labels. The clustering results with label assignment are used solely for the FLM $F$'s training objective and are not reused thereafter for inference or decision making.

*2) Auto Encoders:* AE [28] is a classical self-supervised ML approach, used in [14]. It learns the FLM through introducing a decoder network and a reconstruction objective. The loss function used is the mean square error (MSE) between the input and its reconstruction. In the AE approach, learnable decoders are required. We use CNN with deconvolutions as the decoder for the CNN architecture and the KAN layer as the decoder for the KAN architecture.

*3) Contrastive learning:* We investigate the SimCLR [47] CL approach similar to [30], while [15], [33], [35] also employ CL but in slightly different set-ups. For each sample, two augmented views are generated via random transformations. Views of the sample are treated as positive pairs, and others are treated as negative pairs. Features are extracted from augmented samples using FLM, and a cross-entropy-like loss computed with objective to classify positive and negative pairs.

### E. Decision Making Module

DMM is the final stage of ML based UED as depicted in Figure 3. It has to be independent of the choice of a specific FLM or ML approach. There are several widely used DMMs like K-means based [48], KNN based [30], MLP-based [31], etc. As we solve the SSL problem of outliers detection, we investigate the cluster-based DMM [48], which is a robust, universal, and interpretable method for outlier detection. After the training, we extract the training features using a trained FLM $F$ and cluster them into $C$ clusters with the K-means algorithm. During inference, we extract features from the test sample and identify the nearest cluster. Then, we compare the distance from the test sample to the nearest cluster center with the distances of the train samples in this cluster. If a test sample lies in the right $\alpha\%$ quantile of the train distribution, it is marked as unknown. This concept is illustrated in Figure 5.

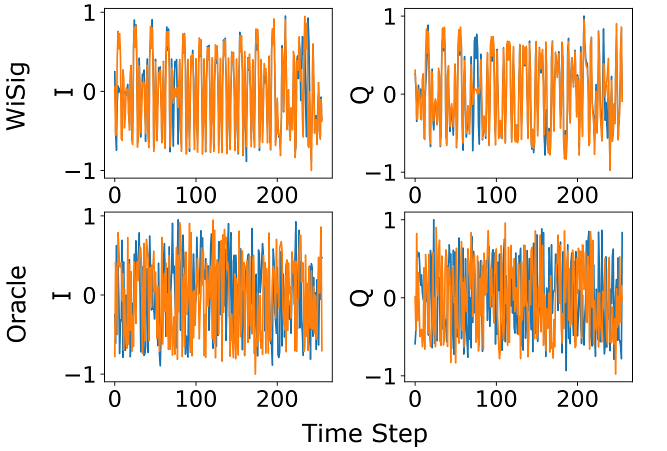


Fig. 6: WiSig and ORACLE datasets comparison. The first column shows I components, and the second column represents Q components of the I/Q samples. Different colors refer to different signal traces from the datasets.

Formally, if $d_i$ is a distance from the test sample to the nearest cluster center and $\{d'_j\}_{j=1}^{N_{tr}}$ are distances from the same cluster center to train samples lying in it, then the classification score $s_i$ is computed as:

$$s_i = \frac{\#\{j : d_i > d_j\}}{N_{tr}}. \tag{15}$$

If $s_i > \alpha$, then the test sample is labeled as unknown. Classification scores from Eq. (15) and the indices of nearest clusters are further used for metric computation.

## V. Methodology

In this section, we describe the datasets used for evaluation, followed by the evaluation metrics and experiment setups.

### A. Datasets

As mentioned in Sec. IV-A, we leverage the ML-based UED workflow under two scenarios to study the design principles of zero-shot self-supervised detection. For the study we rely on the WiSig [38] and ORACLE [37] datasets designed for the same and different messages scenarios, respectively, sharing similar indoor environment conditions.

*1) WiSig:* The WiSig dataset specifically includes signals captured from six distinct WiFi transmitting devices, recorded over a period of four consecutive days in indoor conditions. For each device, $1,000$ signals were recorded per day, resulting in a well-structured dataset designed to facilitate emitter identification tasks. Each signal trace consists of 256 consecutive I/Q samples, capturing a short time window of the emitted waveform. The upper line in Figure 6 shows the I and Q components of two signal traces from the WiSig dataset; the repeating patterns indicate that the messages of the two signal traces are the same.

*2) ORACLE:* The ORACLE dataset was also collected in indoor conditions. It consists of 16 emitters, each containing two long signals of 40 million I/Q observations recorded at a distance of 20 meters. Those signals are sliced into short traces of 256 I/Q samples each, similar to those in the WiSig dataset. The number of samples per emitter is reduced to match the WiSig dataset, i.e., $4,000$. The bottom line in Figure 6 shows I and Q components of two signal traces from the ORACLE dataset; different patterns indicate that the messages of two signal traces are different.

### *B. Metrics*

For evaluation in terms of clustering quality and detection performance, we use three complementary metrics, i.e. the area under the Receiver Operating Characteristic curve (ROC-AUC), Normalized Mutual Information (NMI), and F1-score. We calculate these metrics using predictions from DMM described in Sec. IV-E.

The *ROC-AUC* metric measures the ability of the model to discriminate between classes across varying decision thresholds. A higher AUC value indicates better separability between known and unknown samples. Unlike F1, ROC-AUC provides a threshold-independent evaluation of detection quality. For ROC-AUC calculation, classification scores from DMM are used with $\{known(0), unknown(1)\}$ labels in the test dataset.

The *NMI* metric quantifies the agreement between predicted cluster assignments and ground-truth labels. It measures how much information is shared between the predicted and actual label distributions, where $0$ indicates no mutual information and $1$ denotes a perfect clustering alignment. NMI is calculated between assigned clusters' indices from DMM and emitters' IDs.

*F1-score* metric represents the harmonic mean between precision and recall, defined as $F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$. It balances false positives and false negatives, making it an effective indicator of detection reliability in both closed- and open-set conditions. In our context, the F1-score is computed for the binary classification task of distinguishing between known and unknown samples. We obtain predictions via the procedure from DMM with $\alpha = 5\%$. Along with performance metrics, the numbers of learnable parameters and floating point operations (FLOPs) are calculated for each FLM.

### *C. Experiments*

In the following, we outline the experiment setups for the two scenarios, i.e., with all emitters transmitting the same messages or different messages.

*1) Same Messages Scenario (SMS):* For the SMS scenario described in Sec. III-B, we employ the WiSig dataset described in Sec. V-A1. We evaluate both CNN-1D and KAN architectures, each producing feature embeddings of size 20, with DMM configured to use 80 clusters. The four-layer CNN-1D model follows the architecture proposed in [41], while the KAN model is implemented with a grid size of 10, input dimensionality of 512, and spline order of 4. For the AE paradigm, we employ a 1D CNN-based encoder and decoder. The former is a convenient CNN model with reducing image size and increasing number of channels, and the latter is constructed by replacing convolutional layers with deconvolutional layers and max-pooling operations with up-sampling, following the design in [49]. The KAN-based AE uses a single-layer KAN architecture comprising 20 input nodes and 512 output nodes. In the Contrastive Learning framework, Gaussian noise with a mean of zero and a standard deviation of 0.05 is added to input samples as a data augmentation technique along with random rotations and amplifying similarity to [30].

For evaluation, we employ a cross-validation procedure in which, iteratively, one of the six emitters is designated as the unknown test emitter. In each iteration, the training dataset consists of signals from the remaining five emitters, collected across all days and using the first 80% of samples from each day. The test dataset includes data from all six emitters, assembled from the remaining $20\%$ of samples per day. Within this test set, the selected emitter is labeled as *unknown*, while the other five are treated as *known*.

Both the CNN-1D and KAN architectures are trained for 100 epochs on the training dataset under three learning paradigms: Deep Clustering, Auto-Encoding, and Contrastive Learning with a learning rate $10^{-3}$ and an Adam optimizer. During training, evaluation metrics are computed on the current test dataset every 15 epochs. For each evaluation step, metrics are averaged across the six cross-validation iterations. The epoch corresponding to the maximal average performance is then selected, and the metrics at this epoch are reported as final results. This approach follows the standard practice in unsupervised and open-set learning, where the final epoch may not represent the best model state due to non-monotonic training dynamics [47], [50]. A similar evaluation strategy was previously adopted in [51].

Additionally, we evaluate the PCA with feature dimensionality of 20, varying the number of clusters in DMM among $\{10, 40, 80, 120, 160, 200, 240\}$, as well as KAN-based models with feature sizes $\{2, 3, 5, 10, 20\}$, using the same cross-validation procedure.

*2) Different Messages Scenario (DMS):* For the DMS scenario described in Sec. III-C, we use the ORACLE dataset described in Sec. V-A2. We perform the experiments using two data modalities: raw I/Q data and 2D constellation data (Figure 4 and Eq. 11) with a grid size of $G = 60$. The approaches and models for raw I/Q modulation are the same as for the SMS as explained in Sec. V-C1. For the 2D constellation modality, we evaluate a four-layer 2D CNN, structurally similar to the previously described CNN-1D variant, along with KANs and their SVD-initialized counterparts from Sec. IV-C. For the AE approach, a 2D CNN decoder is utilized, which is also similar to a CNN-1D Decoder. All feature extractors produce embeddings of size 20, and DMM operates with 200 clusters. For KAN-based architectures, the input layer consists of 3600 nodes. In the contrastive learning setup, data augmentations include rotations of 0, $\pi$, $\frac{\pi}{2}$, and $\frac{3\pi}{2}$ radians, as well as additive Gaussian noise with a standard deviation of 0.05. Each FLM is evaluated under three learning paradigms: Deep Clustering, Auto-Encoding, and Contrastive Learning. We also run the PCA method with feature size 20 across $\{10, 40, 80, 120, 160, 200, 240\}$ numbers of clusters in DMM. For the evaluation, we use a scheme similar to Sec. V-C1. The only difference is that we randomly shuffle the emitters beforehand and consequently choose two emitters as unknown, making a total of five folds.

## VI. Results

In this section, we analyze aspects of designing the various components of the workflow illustrated in Figure 3. The results are organized as a discussion of the design choices illustrated in Figure 1b. Table II summarizes the performance of KAN and CNN architectures across three self-supervised learning paradigms. The *Approach* and *FE Module* columns specify the ML approach discussed in Sec. IV-D and FLM from Sec. IV-C, respectively. The columns *ROC-AUC*, *NMI*, and *F1* report the corresponding evaluation metrics (in percentages), whereas *Params* and *FLOPs* columns indicate the number of parameters and the number of floating-point operations required to process a single sample. Tables III and IV follow the same organization as Table II, but report results for the ORACLE dataset with different data modalities.

### A. The impact of data modality

As can be seen from Tables II, III, and IV, raw I/Q data modality shows high performance in the SMS scenario. It demonstrates severely degraded performance in the DMS scenario, whereas using the 2D constellation data modality significantly improves performance in the DMS setup due to the time-invariant representation property described in Sec. IV.

As shown in Table II for the SMS scenario, *PCA, KAN trained with the AE* approach and *CNN-1D trained with the DC* approach achieve the performance of (ROC-AUC = 98.7%, NMI = 56.2%, F1 = 84.8%), (ROC-AUC = 99.3%, NMI = 58.9%, F1 = 88.0%) and (ROC-AUC = 95.4%, NMI = 51.5%, F1 = 76.0%), respectively, which shows high quality of cluster assignments in DMM measured by NMI, and well-shaped clusters reflected by the ROC-AUC and F1 metrics.

The same methods in the DMS setup on the raw I/Q data modality shown in Table III perform significantly worse with (ROC-AUC = 42.1%, NMI = 4.9%, F1 = 3.4%), (ROC-AUC = 50.1%, NMI = 2.1%, F1 = 10.2%) and (ROC-AUC = 50.1%, NMI = 2.4%, F1 = 10.1%) for *PCA, KAN trained by AE* and *CNN-1D trained by DC* methods, respectively. This can be attributed to the prevalent semantic information in samples, so self-supervised approaches tend to separate messages instead of emitters.

The results in Table IV for using the 2D constellation data modality in the DMS scenario for the same methods indicate notable performance improvement across the three metrics: (ROC-AUC = 60.1%, NMI = 34.3%, F1 = 17.1%), (ROC-AUC = 70.6%, NMI = 41.1%, F1 = 38.7%) and (ROC-AUC = 65.8%, NMI = 43.3%, F1 = 24.1%). This confirms that the 2D constellation data modality reduces the presence of semantic information, allowing self-supervised approaches to better separate emitters. Overall, 2D constellation data modality shows significant improvement in performance under DMS due to the mitigation of different messages difference as described in Sec. IV-A.

### B. The impact of Feature Learning Module

Table II shows that in the SMS setup, the KAN architecture slightly but consistently outperforms CNN-1D across all learning paradigms (AE, DC, and CL). PCA achieves comparable performance while requiring nearly an order of magnitude fewer parameters and FLOPs, highlighting its efficiency as a lightweight baseline. In the DMS scenario using the raw I/Q data modality, Table III, the KAN-based models exhibit weaker cluster assignment quality compared to the CNN-based counterparts, whereas PCA underperforms both deep learning approaches across all metrics. For the 2D constellation data modality (Table IV), KAN initialized with SVDweights and trained via DC achieves the best performance, substantially surpassing other architectures. CNN-2D models exhibit stable behavior with SVD initialization when trained using CL. PCA continues to provide a strong baseline with competitive clustering quality while maintaining the lowest computational cost. Comparing CNN-1D and KAN models for the raw I/Q data modality, the latter are several times more demanding in both parameters and FLOPs. However, for the 2D constellation data modality, KANs are more effective in terms of FLOPs.

Specifically, in the SMS scenario from Table II, the KAN-based AE model achieves the best overall performance, reaching (ROC-AUC = 99.3%, NMI = 59%, F1 = 88%), compared to the CNN-1D variant with (ROC-AUC = 99.2%, NMI = 57%, F1 = 88%). A similar trend is observed for DC and CL paradigms, where KAN variants consistently outperform their CNN counterparts by on average 2–4 percentage points in ROC-AUC, NMI and F1 metrics. In the same table, PCA achieves (ROC-AUC = 99%, NMI = 56%,F1 = 85%), closely matching deep learning models while using only 10k parameters and 0.2M FLOPs, requiring approximately 8–30 times fewer computational resources than CNN or KAN architectures.

In the DMS scenario from Table III, the distinction between models diminishes considerably, with all deep learning approaches performing close to the chance level: the best ROC-AUC does not exceed $(51.3\pm1.7)\%$ (CL CNN-1D ), and F1 values remain around 10%. Here, KAN architectures perform slightly worse in clustering quality (i.e., NMI = 0.5% for DC KAN) compared to CNN models (i.e., NMI = 2.4% for DC CNN-1D ). At the same time, PCA drops significantly to (ROC-AUC = 42%, NMI = 5%, F1 = 3%), confirming its inability

TABLE II: SMS: WiSig dataset results with the raw I/Q data modality.

| Approach | FE Module | Modality | F.Size | Clusters | ROC-AUC % | NMI % | F1 % | Params | FLOPs |
|---|---|---|---|---|---|---|---|---|---|
| DC | CNN-1D | raw I/Q | 20 | 80 | 95.4±1.9 | 51.5±0.8 | 76.0±6.2 | 80K | 1M |
| DC | KAN | raw I/Q | 20 | 80 | 97.0±2.5 | 55.4±0.9 | 77.5±8.7 | 143K | 6M |
| AE | CNN-1D | raw I/Q | 20 | 80 | 99.2±0.5 | 57.4±0.7 | 87.6±1.2 | 80K | 1M |
| AE | KAN | raw I/Q | 20 | 80 | 99.3±0.7 | 58.9±0.8 | 88.0±1.3 | 143K | 6M |
| CL | CNN-1D | raw I/Q | 20 | 80 | 97.6±1.1 | 51.4±1.1 | 82.7±4.9 | 80K | 1M |
| CL | KAN | raw I/Q | 20 | 80 | 97.8±1.3 | 55.9±1.7 | 83.8±3.8 | 143K | 6M |
| PCA | PCA | raw I/Q | 20 | 80 | 98.7±1.1 | 56.2±0.9 | 84.8±4.1 | 10k | 0.2M |

TABLE III: DMS: ORACLE dataset results with the raw I/Q data modality.

| Approach | FE Module | Modality | F.Size | Clusters | ROC-AUC % | NMI % | F1 % | Params | FLOPs |
|---|---|---|---|---|---|---|---|---|---|
| DC | CNN-1D | raw I/Q | 20 | 200 | 50.1±2.2 | 2.4±0.8 | 10.1±1.2 | 80K | 1M |
| DC | KAN | raw I/Q | 20 | 200 | 49.5±0.9 | 0.5±0.1 | 9.1±0.6 | 143K | 6M |
| AE | CNN-1D | raw I/Q | 20 | 200 | 50.7±3 | 9.4±5.4 | 9.8±3.3 | 80K | 1M |
| AE | KAN | raw I/Q | 20 | 200 | 50.1±0.6 | 2.1±0.7 | 10.2±0.7 | 143K | 6M |
| CL | CNN-1D | raw I/Q | 20 | 200 | 51.3±1.7 | 10.6±1.5 | 12.1±2.7 | 80K | 1M |
| CL | KAN | raw I/Q | 20 | 200 | 49.4±0.5 | 1.2±0.6 | 10.1±0.5 | 143K | 6M |
| PCA | PCA | raw I/Q | 20 | 200 | 42.1±2.3 | 4.9±0.3 | 3.4±1.3 | 10k | 0.2M |

TABLE IV: DMS: ORACLE dataset results with the 2D constellation data modality. Rows shaded according to ROC-AUC score (green = best, yellow = medium, unshaded = low).

| Approach | FE Module | Modality | F.Size | Clusters | ROC-AUC % | NMI % | F1 % | Params | FLOPs |
|---|---|---|---|---|---|---|---|---|---|
| DC | KAN SVD Init | 2D constellation | 20 | 200 | 71.5±5.1 | 48.9±1.7 | 41.1±8.1 | 720K | 1.8M |
| AE | KAN SVD Init | 2D constellation | 20 | 200 | 52.3±2.9 | 21.9±1.3 | 10.3±2.1 | 720K | 1.8M |
| CL | KAN SVD Init | 2D constellation | 20 | 200 | 64.2±6.2 | 33.7±0.8 | 26.7±9.7 | 720K | 1.8M |
| DC | CNN-2D SVD Init | 2D constellation | 20 | 200 | 65.8±6.5 | 43.3±2.7 | 24.1±6.1 | 180K | 11M |
| AE | CNN-2D SVD Init | 2D constellation | 20 | 200 | 51.1±3.2 | 0.5±0.1 | 15.6±2.1 | 180K | 11M |
| CL | CNN-2D SVD Init | 2D constellation | 20 | 200 | 66.2±6.4 | 34.2±0.9 | 29.5±9.6 | 180K | 11M |
| DC | KAN | 2D constellation | 20 | 200 | 48.5±0.09 | 5.3 ±0.5 | 8.7±5.3 | 720K | 1.8M |
| AE | KAN | 2D constellation | 20 | 200 | 49.7±1.2 | 6.1±0.6 | 11.9±1.6 | 720K | 1.8M |
| CL | KAN | 2D constellation | 20 | 200 | 57.1±4.5 | 34.9±1.4 | 15.8±5.4 | 720K | 1.8M |
| DC | CNN-2D | 2D constellation | 20 | 200 | 49.6±1.1 | 2.4 ±0.6 | 9.7±1.2 | 180K | 11M |
| AE | CNN-2D | 2D constellation | 20 | 200 | 49.7±1.2 | 6.1±0.6 | 11.9±1.6 | 180K | 11M |
| CL | CNN-2D | 2D constellation | 20 | 200 | 66.5±8.2 | 33.6±0.9 | 29.7±9.5 | 180K | 11M |
| PCA | PCA | 2D constellation | 20 | 200 | 60.1±5.2 | 34.3±0.8 | 17.1±0.6 | 72k | 0.1M |

to separate emitters when signal variability is dominated by the message content.

Finally, in the 2D constellation data modality from Table IV, KANs initialized with SVD weights and trained by DC demonstrate a clear advantage, achieving (ROC-AUC = 71%, NMI = 41%, F1 = 39%), which surpasses all other architectures by at least 4–10 percentage points across metrics. In contrast, the best CNN-2D model trained with CL reaches (ROC-AUC = 66%, NMI = 34%, F1 = 30%) while PCA achieves (ROC-AUC = 60%, NMI = 34%, F1 = 17%), maintaining competitive performance despite minimal complexity. Collectively, these results confirm that KAN architectures benefit substantially from SVD-based initialization under DC, and that PCA remains an efficient and interpretable baseline delivering strong performance relative to its computational footprint.

Figure 7 compares KAN and CNN performance on the WiSig dataset across KAN feature sizes and learning paradigms. In all cases, KANs exhibit a clear monotonic improvement as the feature dimensionality increases, most notably under the Deep Clustering (DC) and Contrastive Learning (CL) schemes, where ROC-AUC rises from approximately 85% to above 97%. NMI improves by nearly 10 percentage points when moving from 2 to 20 features. AE-based KANs display smaller yet consistent gains, maintaining superior low-dimensional performance compared to other paradigms. Across all metrics, KAN models surpass the CNN baseline (dashed gray line) once the feature dimensionality exceeds 5, confirming that the adaptive spline-based functional representation in KANs scales more effectively with representational capacity than convolutional filters. These results suggest that KANs can utilize higher-dimensional embeddings to enhance discriminative clustering and improve open-set separability. In contrast, CNNs reach saturation at lower feature sizes due to their limited nonlinearity and fixed receptive-field structure. Summing up, Kolmogorov-Arnold networks (KANs) are interpretable and have performance comparable to CNNs, a convenient alternative that suffers from ”black-box” design, across SMS and DMS.

### *C. The impact of the machine learning approach*

In Table II, it is evident that in the SMS scenario with the WiSig dataset and raw I/Q data modality, the AE learning approach achieves the best performance, followed by CL. However, AE, DC, and CL learning approaches evaluated in the DMS setup using the raw I/Q data modality do not show significant differences, as shown in Table III, with PCA performing the worst. Finally, Table IV shows that on the sparse 2D constellation data modality, the DC approach is the best.

Quantitatively, the results in Table II confirm the superior performance of the AE approach in the SMS setup using raw

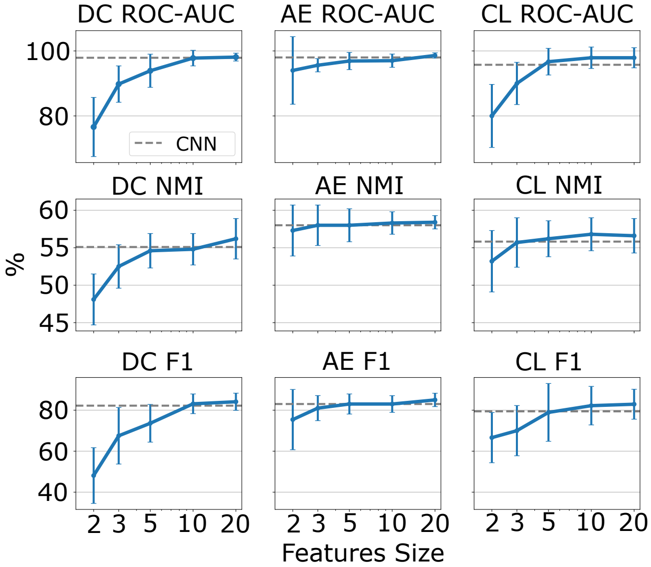


Fig. 7: KANs vs baseline CNN performance on WiSig Dataset. Blue lines represent the average value, blue ticks represent the error bars, and dashed gray lines show the baseline CNN performance.

I/Q data. The KAN-based AE achieves the highest average scores across all metrics, with (ROC-AUC = 99%, NMI = 59%, F1 = 88%), surpassing the best CL KAN configuration (ROC-AUC = 98%, NMI = 56%, F1 = 84%). These results indicate that AE captures the signal structure most effectively when message content remains consistent. While PCA provides a strong linear baseline, CL lags slightly behind, with (ROC-AUC = 98%, NMI = 56%, F1 = 84%) for CL and (ROC-AUC = 99%, NMI = 56%, F1 = 85%) for PCA.

In contrast, Table III demonstrates that for the DMS scenario under the raw I/Q data modality, the overall performances of AE, DC, and CL approaches are nearly indistinguishable, with ROC-AUC values clustered around 50%, NMI between 0.5% and 10%, and F1 scores between 9% and 12%.

Finally, Table IV shows that for the sparse 2D constellation data modality, the DC approach achieves the best overall results, particularly for the KAN models initialized with SVD weights, reaching (ROC-AUC = 71%, NMI = 41%, F1 = 39%). CL ranks second with (ROC-AUC = 66%, NMI = 34%, F1 = 30%), while PCA, although simpler, remains competitive at (ROC-AUC = 60%, NMI = 34%, F1 = 17%). Summarizing, for DMS, the best performing configuration is SVD initialized KAN with deep clustering approach and 2D constellation data modality.

### D. The impact of the number of clusters in DMM

Figure 8 shows the DMM's sensitivity to the number of selected clusters using the PCA baseline and measured with ROC-AUC, NMI, and F1 metrics on the ORACLE and WiSig datasets. The results reveal that ROC-AUC and F1 reach a plateau, with only minor changes once the number of clusters becomes sufficiently large, while NMI consistently decreases. The decrease in the NMI metric does not mean the decrease in cluster assignment [52], because NMI also depends on the number of train emitters. These results indicate that the number of clusters required to reach stability is dataset-dependent. For the WiSig dataset with six emitters, stability is reached at approximately 80 clusters, whereas for the ORACLE dataset with 16 emitters, the stable region begins at approximately 200 clusters.

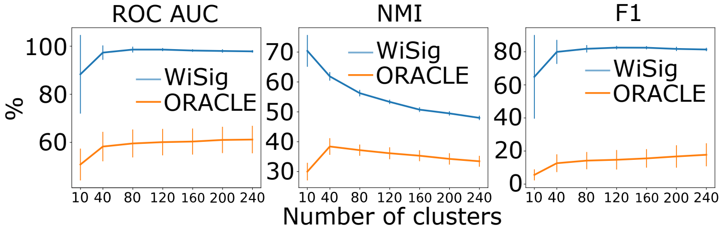


Fig. 8: Sensitivity of the DMM to the number of clusters. Without loss of generality, we use the PCA baseline with feature size 20.

### E. Interpretability evaluation

Ultimately, we assess the implementation approaches to the proposed concept in terms of interpretability. CNNs incorporate several techniques for interpreting model decisions. One widely used method is LIME, which approximates the model's behavior around a specific input sample using a linear model:

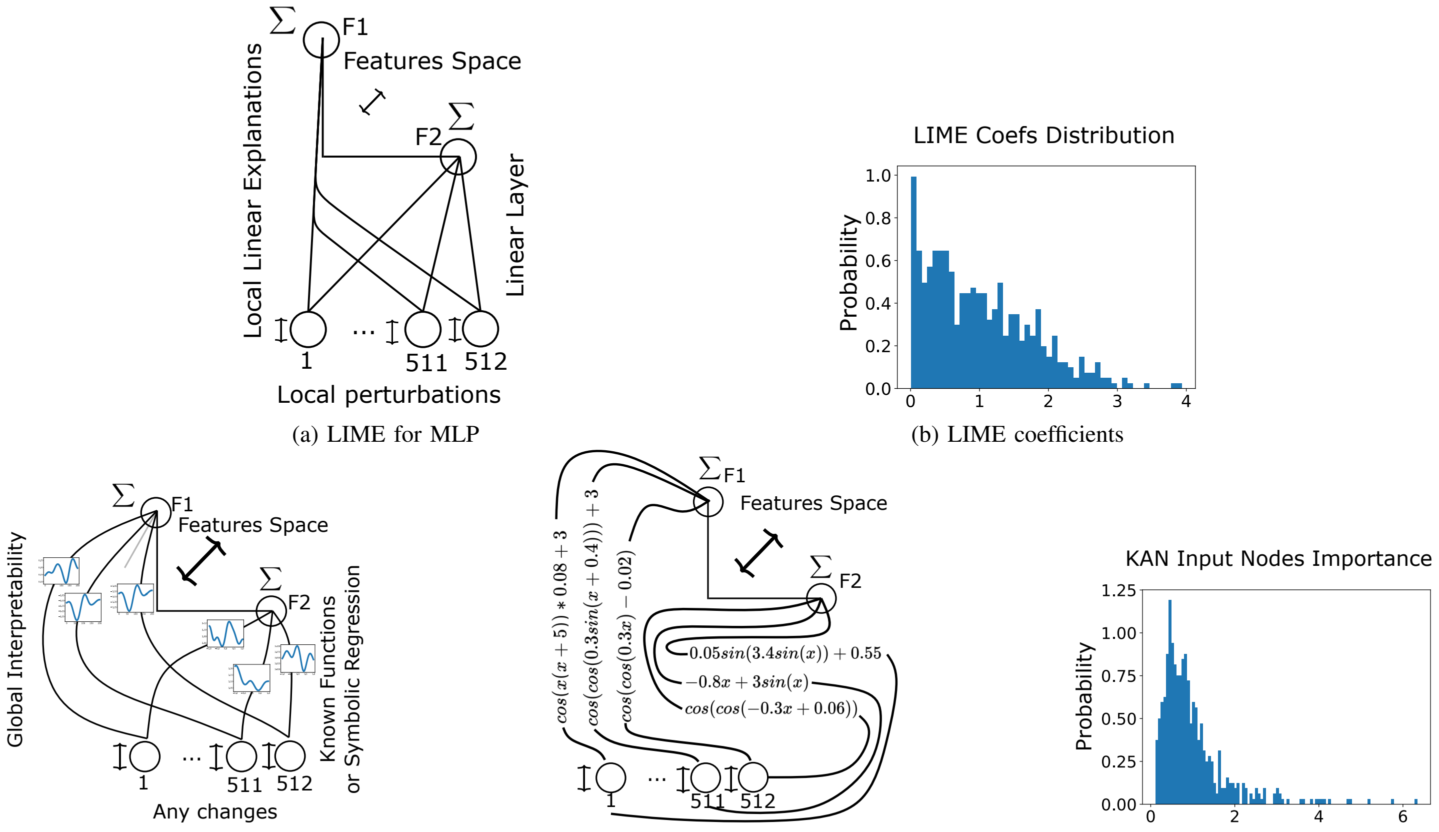
$$M(\delta x + x) \approx \mathbf{W}\delta x + M(x), \quad \text{where } \frac{\|\delta x\|}{\|x\|} \ll 1. \tag{16}$$

Here, $x$ is the input sample, and the matrix $\mathbf{W}$ captures the local linear approximation of the model. LIME can further be used to analyze how specific inputs influence the sample's position in the embedding space, or serve as a locally fully explainable method, especially after pruning non-essential weights from $\mathbf{W}$.

For the WiSig dataset, this concept is illustrated in Figure 9a, where a linear layer approximates the CNN's behavior in the vicinity of a data point. The corresponding distribution of weights from these locally fitted linear models is shown in Figure 9b. However, due to the inherent non-linearity of CNNs, such local interpretability methods cannot generalize across the entire input space. In contrast, Kolmogorov–Arnold Networks provide *global interpretability*. As depicted in Figure 9c, each component of the 2D embedding space is connected to the inputs via known spline functions. The most important connections (black lines) can be approximated using symbolic regression, while less important ones are shown in gray. The possible symbolic regression of spline functions is shown in Figure 9d. This structure enables tracing how input perturbations affect internal representations. The distribution of input importance across KANs is illustrated in Figure 9e, providing a more holistic view of feature influence.

## VII. Conclusions

This paper presents a comprehensive analysis of the design space for Unknown Emitter Detection. We explore the key design dimensions with respect to SMS and DMS. Through the lens of the ML workflow, we investigate the impact of (a) data modality,



(a) LIME for MLP

(b) LIME coefficients

(c) KAN concept

(d) KAN with symbolic regresssion

(e) KAN input nodes importance

Fig. 9: Model interpretability insights for WiSig dataset

(b) feature learning module, (c) machine learning approach, and (d) decision-making module. Our theoretical and experimental results reveal several key insights: 1) Under realistic DMS conditions, using a 2D constellation data modality improves performance by up to 20 p.p. in ROC-AUC compared to raw I/Q data. 2) KAN provides an interpretable architecture and performs comparably to CNNs, a "black-box" alternative, across both SMS and DMS settings. 3) For DMS, the best-performing configuration combines an SVD-initialized KAN with a deep clustering approach and 2D constellation data. This setup outperforms a standard KAN (with all other workflow components held constant) by up to 20 p.p. in ROC-AUC. 4) In analyzing the decision-making module, we identify the optimal number of clusters required for environments with varying numbers of known emitters. In summary, the results for UED demonstrate that incorporating KAN-based feature extractors and interpretable representations at both the Decision Making Module and Feature Learning Module enables the proposed model to generalize effectively to unseen emitters and varying signal conditions, a capability not previously shown in the literature.

## References

[1] M. Polese, L. Bonati, S. D'oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, 2023.

[2] O. Alliance, "O-ran use cases and deployment scenarios," *White Paper, Feb*, 2020.

[3] L. Milosheski, G. Cerar, B. Bertalanič, C. Fortuna, and M. Mohorčič, "Self-supervised learning for clustering of wireless spectrum activity," *Computer Communications*, vol. 212, pp. 353–365, 2023.

[4] C. Ouyang, Y. Liu, H. Yang, and N. Al-Dhahir, "Integrated sensing and communications: A mutual information-based framework," *IEEE Communications Magazine*, vol. 61, no. 5, pp. 26–32, 2023.

[5] L. Ding, S. Wang, F. Wang, and W. Zhang, "Specific emitter identification via convolutional neural networks," *IEEE communications letters*, vol. 22, no. 12, pp. 2591–2594, 2018.

[6] J. Robinson, S. Kuzdeba, J. Stankowicz, and J. M. Carmack, "Dilated causal convolutional model for rf fingerprinting," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 0157–0162.

[7] S. Apfeld and A. Charlish, "Recognition of unknown radar emitters with machine learning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 4433–4447, 2021.

[8] L. J. Wong, W. C. Headley, S. Andrews, R. M. Gerdes, and A. J. Michaels, "Clustering learned cnn features from raw i/q data for emitter identification," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 26–33.

[9] H. Han, W. Li, Z. Feng, G. Fang, Y. Xu, and Y. Xu, "Proceed from known to unknown: Jamming pattern recognition under open-set setting," *IEEE wireless communications letters*, vol. 11, no. 4, pp. 693–697, 2022.

[10] Y. Luo, X. Chen, N. Ge, W. Feng, and J. Lu, "Transformer-based device-type identification in heterogeneous iot traffic," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5050–5062, 2022.

[11] S. Basak, S. Rajendran, S. Pollin, and B. Scheers, "Combined rf-based drone detection and classification," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 111–120, 2021.

[12] J. Wang, B. Zhang, J. Zhang, N. Yang, G. Wei, and D. Guo, "Specific emitter identification based on deep adversarial domain adaptation," in *2021 4th International Conference on Information Communication and Signal Processing (ICICSP)*. IEEE, 2021, pp. 104–109.

[13] L. Xu, W. Shi, X. Fu, H. Xu, Y. Wang, B. Adebisi, and G. Gui, "Few-shot specific emitter identification method using rotation feature decoupling for secure 6g," in *2023 IEEE 23rd International Conference on Communication Technology (ICCT)*. IEEE, 2023, pp. 490–494.

[14] K. Huang, J. Yang, H. Liu, and P. Hu, "Deep learning of radio frequency fingerprints from limited samples by masked autoencoding," *IEEE Wireless Communications Letters*, 2022.

[15] C. Liu, X. Fu, Y. Wang, L. Guo, Y. Liu, Y. Lin, H. Zhao, and G. Gui, "Overcoming data limitations: A few-shot specific emitter identification method using self-supervised learning and adversarial augmentation," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 500–513, 2023.

[16] J. Robinson and S. Kuzdeba, "Novel device detection using rf fingerprints," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2021, pp. 0648–0654.

[17] Y. Peng, C. Hou, Y. Zhang, Y. Lin, G. Gui, H. Gacanin, S. Mao, and F. Adachi, "Supervised contrastive learning for rff identification with limited samples," *IEEE Internet of Things Journal*, 2023.

[18] Y. Dong, X. Jiang, H. Zhou, Y. Lin, and Q. Shi, "Sr2cnn: Zero-shot learning for signal recognition," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2316–2329, 2021.

[19] X. Wen, C. Cao, Y. Li, and Y. Sun, "Drsn with simple parameter-free attention module for specific emitter identification," in *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2022, pp. 192–200.

[20] A. Al-Shawabka, P. Pietraski, S. B. Pattar, F. Restuccia, and T. Melodia, "Deeplora: Fingerprinting lora devices at scale through deep learning and data augmentation," in *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2021, pp. 251–260.

[21] J. Stankowicz and S. Kuzdeba, "Unsupervised emitter clustering through deep manifold learning," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2021, pp. 0732–0737.

[22] J. Robinson and S. Kuzdeba, "Riftnet: Radio frequency large population classification," in *2021 IEEE 18th Annual Consumer Communications and Networking Conference (CCNC)*. IEEE, 2021.

[23] L. Milosheski, M. MohorčIč, and C. Fortuna, "Spectrum sensing with deep clustering: Label-free radio access technology recognition," *IEEE Open Journal of the Communications Society*, 2024.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[25] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[27] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, 2018.

[28] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 1, pp. 857–876, 2021.

[29] X. Zhang, Y. Huang, M. Lin, Y. Tian, and J. An, "Transmitter identification with contrastive learning in incremental open-set recognition," *IEEE Internet of Things Journal*, 2023.

[30] X. Hao, Z. Feng, R. Liu, S. Yang, L. Jiao, and R. Luo, "Contrastive self-supervised clustering for specific emitter identification," *IEEE Internet of Things Journal*, vol. 10, no. 23, pp. 20 803–20 818, 2023.

[31] M. Krasnov, L. Milosheski, M. Mohorčič, and C. Fortuna, "Novel devices identification with deep clustering," in *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*. IEEE, 2025, pp. 1–6.

[32] Z. Yao, X. Fu, L. Guo, Y. Wang, Y. Lin, S. Shi, and G. Gui, "Few-shot specific emitter identification using asymmetric masked auto-encoder," *IEEE Communications Letters*, vol. 27, no. 10, pp. 2657–2661, 2023.

[33] Z. Wu, F. Wang, and B. He, "Specific emitter identification via contrastive learning," *IEEE Communications Letters*, vol. 27, no. 4, pp. 1160–1164, 2023.

[34] X. Zha, T. Li, Z. Qiu, and F. Li, "Cross-receiver radio frequency fingerprint identification based on contrastive learning and subdomain adaptation," *IEEE Signal Processing Letters*, vol. 30, pp. 70–74, 2023.

[35] M. Sun, J. Teng, X. Liu, W. Wang, and X. Huang, "Few-shot specific emitter identification: A knowledge, data, and model-driven fusion framework," *IEEE Transactions on Information Forensics and Security*, 2025.

[36] B. Liu, H. Yu, J. Du, Y. Wu, Y. Li, Z. Zhu, and Z. Wang, "Specific emitter identification based on self-supervised contrast learning," *Electronics*, vol. 11, no. 18, p. 2907, 2022.

[37] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "Oracle: Optimized radio classification through convolutional neural networks," in *IEEE INFOCOM 2019-IEEE conference on computer communications*. IEEE, 2019, pp. 370–378.

[38] S. Hanna, S. Karunaratne, and D. Cabric, "Wisig: A large-scale wifi signal dataset for receiver and channel agnostic rf fingerprinting," *IEEE Access*, vol. 10, p. 22808–22818, 2022.

[39] J. Bai, Y. Lian, Y. Wang, J. Ren, Z. Xiao, H. Zhou, and L. Jiao, "An interpretable explanation approach for signal modulation classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–13, 2024.

[40] L. Xie, L. Peng, J. Zhang, and A. Hu, "Radio frequency fingerprint identification for internet of things: A survey," *Security and Safety*, vol. 3, p. 2023022, 2024.

[41] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[42] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.

[43] C. De Boor and C. De Boor, *A practical guide to splines*. springer New York, 1978, vol. 27.

[44] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *International conference on learning representations*, 2017.

[45] Q. Fournier and D. Aloise, "Empirical comparison between autoencoders and traditional dimensionality reduction methods," in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. IEEE, 2019, pp. 211–214.

[46] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[47] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[48] A. Smiti, "A critical overview of outlier detection methods," *Computer Science Review*, vol. 38, p. 100306, 2020.

[49] S. Chen, J. Yu, and S. Wang, "One-dimensional convolutional auto-encoder-based feature learning for fault diagnosis of multivariate processes," *Journal of Process Control*, vol. 87, pp. 54–67, 2020.

[50] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 132–149.

[51] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *European conference on computer vision*. Springer, 2014, pp. 393–409.

[52] A. Mahmoudi and D. Jemielniak, "Proof of biased behavior of normalized mutual information," *Scientific Reports*, vol. 14, no. 1, p. 9021, 2024.