Open Set Wireless Transmitter Authorization: Deep Learning Approaches and Dataset Considerations

Samer Hanna, *Student Member, IEEE*, Samurdhi Karunaratne, *Student Member, IEEE*,
and Danijela Cabric, *Fellow, IEEE*

***Abstract*—Due to imperfections in transmitters' hardware, wireless signals can be used to verify their identity in an authorization system. While deep learning was proposed for transmitter identification, existing work has mainly focused on classification among a closed set of transmitters. Malicious transmitters outside this closed set will be misclassified, jeopardizing the authorization system. In this article, we formulate the problem of recognizing authorized transmitters and rejecting new transmitters as open set recognition and anomaly detection. We consider approaches based on one and several binary classifiers, multiclass classifiers, and signal reconstruction. We study how these approaches scale with the required number of authorized transmitters. We propose using a known set of unauthorized transmitters to assist the training and study its impact. The evaluation procedure takes into consideration that some transmitters might be more similar than others and nuances these effects. The authorization's robustness against temporal changes in fingerprints is also evaluated as a function of the approach and the dataset structure. When using 10 authorized and 50 known unauthorized WiFi transmitters from a publicly accessible testbed, we were able to achieve an outlier detection accuracy of 98% on the same day test set and 80% on the different day test set.**

***Index Terms*—Transmitter identification, deep learning, open set recognition, authorization, physical layer authentication, RF fingerprint.**

I. Introduction

With the growth in the number of wireless connected devices, securing them has become more challenging. Unlike wired communications, a wireless network is accessible by any device with sufficient transmit power. This makes *authentication*, the process of verifying the identity of devices, challenging. After authentication, devices are granted access, the process known a *authorization*. While cryptographic methods are used for authentication, many devices like Internet-of-Things (IoT) devices don't possess the energy nor computational power to run them, leading to many authentication based attacks [1].

Physical Layer Authentication (PLA) leverages the dynamics of physical layer attributes to address these challenges and to enhance wireless security [2]. While active PLA typically requires changes in transmitters, passive PLA is performed on the receiver side, making it more practical. Passive PLA uses RF fingerprints; combining channel state information (CSI) and transmitter hardware fingerprints to authenticate devices. Transmitter fingerprints result from the imperfections in their RF chain components like ADCs, power amplifiers, etc. The interaction between these non-idealities makes signals from identical transmitters exhibit unique characteristics typically modeled as carrier frequency offset, IQ imbalance, among others [3].

While there has been many approaches for using RF fingerprints based on handcrafted features [4]–[16], it was shown to be highly dependent on the quality of the receiver hardware [17] and requires manual feature engineering which are protocol dependent. For these reasons, recently, there has been wide interest is using deep learning approaches to address this problem [18]–[28]. Deep learning has the ability to learn a richer set of features from raw IQ samples leading to improved performance over manually selected features, as has been demonstrated in [18].

While previous deep learning work in this area has addressed many of the challenges of RF fingerprinting, this body of work has posed the problem as a *closed set* classification which assumes a known set of transmitters, except for [27] and our prior work [28]. No matter how large the set is, if any new unseen transmitter gets within communication range, its signal will get misclassified leading to security vulnerabilities. This calls for *open set* approaches which are capable of rejecting signals from unseen transmitters. While [27] proposed their own approach to address this problem, rejecting samples from a new distribution is not a novel problem for the machine learning community. A plethora of approaches have already been proposed for similar problems in computer vision, natural language processing, intrusion detection, etc. Two problems are most relevant; openset classification [29]: classifying among known classes and rejecting unseen classes, and anomaly detection [30]: identifying abnormal samples. Instead of reinventing the wheel, we aim to adapt the most prominent approaches for these problems and evaluate their performance. Unlike other domains, transmitter authorization has its own challenges and requirements: (1) RF fingerprints arise from random channel and hardware variations, hence, generalizable conclusions can not be derived from single point

Manuscript received May 1, 2020; revised August 31, 2020 and October 26, 2020; accepted November 27, 2020. Date of publication December 8, 2020; date of current version March 8, 2021. This work was supported in part by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The associate editor coordinating the review of this article and approving it for publication was G. Ding. *(Corresponding author: Samer Hanna.)*

The authors are with the Electrical and Computer Engineering Department, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: samerhanna@ucla.edu; samurdhi@ucla.edu; danijela@ee.ucla.edu).

evaluations (2) the number of authorized transmitters is a system requirement that can vary significantly (3) the ease of collecting data (compared to image classification, for instance) raises questions about how to construct a training dataset, and (4) the robustness of the approach against time varying fingerprints needs to be evaluated.

In our previous work [28], which we extend in this work, we started investigating a few approaches from the existing openset recognition literature. In this work, our contributions can be summarized as follows:

- We formulate the problem of rejecting signals from unseen transmitters as both an openset recognition problem and an anomaly detection problem. We consider 5 deep learning approaches to the problem and evaluate their performance; (1) Binary classification using "Disc" (2) Multiclass classification using "DClass" (3) A binary classifier for each transmitter using "OvA" (4) Evaluating signal reconstruction errors using "AutoEnc" (5) Analyzing classifier's activations using "OpenMax"
- We discuss several considerations for network architecture design for transmitter authorization. We show that making minor changes to the neural network architecture and data labeling strategy yields a conceptually different approach with different performance. We show that classification within a closed set is not always an indicator for performance in an open set.
- We compare the performance of the considered approaches with respect to the number of authorized transmitters required by the system. We propose using a set of known unauthorized transmitters and show its benefits.
- We show that the results obtained are dependent on the choice of transmitters and time of evaluation. Then we address this by showing results in terms of statistics of multiple transmitter choices using data captures on same day as training and different day.

II. Related Work

Physical Layer Authentication (PLA) can be classified as active or passive. Active PLA typically overlays a tag over the message used for authentication, thus requiring changes to the physical layer of the transmitters [31], [32]. Passive PLA on the other hand uses the channel state information and the RF fingerprint due to hardware imperfections to identify transmitters [3], requiring no change to transmitter signals, and hence is easier to apply. Approaches for passive PLA either use a set of handcrafted features or use deep learning to learn the features.

1) *Handcrafted Feature PLA:* In these works, a set of manually designed features are extracted from the signals, which are then used for distinguishing transmitters according to some procedure. A variety of features were considered as transmitter fingerprints in the literature [33]. These features include transient ones like the patterns at the start of packets [34], and steady-state ones like carrier frequency offset [35], IQ imbalance, sampling frequency offset or a combination of these features [4]–[10]. Other works have used channel state information (CSI) as features. This kind of approach has been considered for SISO [11]–[14] and massive MIMO [15] communications. Combining CSI with transmitter fingerprints has also been proposed [16].

As for the procedure for distinguishing transmitters using the features, some works used traditional methods like a distance metric among features [8], [9], [34], and hypothesis testing [13]. Other works have used machine learning applied to the features; K-nearest neighbor (KNN) [4], support vector machines (SVM) [4], and neural networks [9] were proposed for classification. For authentication, Gaussian mixture models were also used [5], [11], [12].

2) *Deep Learning Based PLA:* In contrast to handcrafted features, deep learning approaches are able to extract features from the high dimensional signals without requiring manual feature engineering. Deep learning became popular due to its superior performance in computer vision [36]. It was successfully applied in many problems in wireless communications like modulation classification [28], [37], [38], detecting anomalies in spectrum utilization [39], [40], spectrum sensing [41], and backscatter signal detection [42].

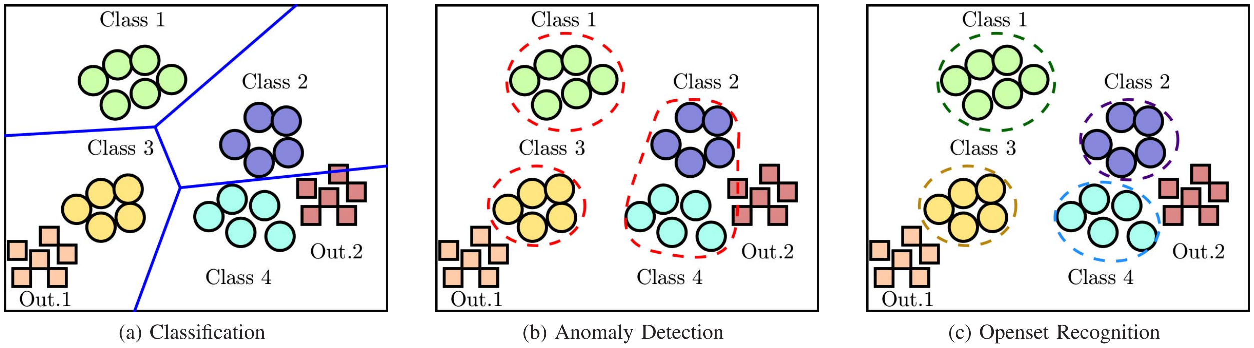
Due to the better performance of deep learning compared to handcrafted features in transmitter identification [18], it has gained widespread interest [18]–[26]. Some of these works fall under the category of active PLA, requiring changes in the transmitters, while others are passive.

In active approaches, modifications are intentionally added to the signal to improve classification. A protocol inserting IQ imbalance and DC offset impairments to improve the RF fingerprints was proposed in [43], [44]. FIR filtering was also considered in [45] to optimize RF fingerprints. This work requires modification in the transmitter side, which is not always feasible. The work considering passive PLA focused on the data representation, the network type, or studying the impact of a specific transmitter characteristic.

a) *Data representation:* The work in [21] has compared different data representations like wavelet transform and Short Time Fourier transform while in [46], the authors considered recurrence plots. Applying the Hilbert-Huang transform to the signal and deep residual networks were proposed in [47]. Higher order statistics like bispectrum were proposed in [48].

b) *Network architecture:* In [26], authors compared different types of neural networks and machine learning techniques. CNNs and RNNs to classify IoT devices over a wide range of SNR in [49]. In [24], multiple CNN architectures were tested on indoor and outdoor data with a focus on cognitive radio applications. Complex neural networks were proposed in [22] using convolutional and recurrent architectures. The effect of a dynamic channel on deep learning RF fingerprinting along with the type of data was the focus of the work in [50]. In [19], the authors have considered a multisampling neural network using LOS and NLOS datasets. Denoising autoencoders were also proposed for the same problem [51]. Adversarial learning was adapted [52] in to detect rogue transmitters.

c) *Transmitter characteristic:* Some works used deep learning while focusing on a specific type of impairment. The effect of power amplifier nonlinearity and signal type on classification performance was studied in [25]. In [53], CNNs



(a) Classification (b) Anomaly Detection (c) Openset Recognition

Fig. 1. Known classes are depicted as circles and outlier classes as squares. Classifiers would mistakenly label outliers. Anomaly detector rejects outliers but cannot distinguish among the known classes. Openset classification classifies among known classes and rejects outliers.

were used to learn IQ imbalance as a modulation-independent way of transmitter identification.

The main limitation of this body of work is that it focuses on classification among a closed set of known transmitters. To the best of the authors' knowledge, two works have considered the problem of using deep learning for transmitter authorization that generalizes to unseen transmitters. The first work has proposed a novel approach for outlier detection that works on a per-packet basis [27]. The classifying neural network is applied to slices of the packet and statistics of the slices predictions are compared to a threshold. This approach is discussed further later in this work in Section V. In their work, several datasets using WiFi and ADS-B were considered using 50, 250, and 500 devices. The data is said to have been captured "in the wild" with no further details provided. We only mention their results most similar to our work; using 50 authorized WiFi devices, they were able to detect new devices with an accuracy of 73% at the cost of a drop in classification accuracy from 63% to 43%.

The second work that has considered this problem was our previous work [28], which is extended in this work. In [28], three approaches based on open set recognition were contrasted using a dataset consisting of WiFi preambles captured in a publicly accessible wireless testbed [54]. We have considered the effects of the number of authorized transmitters and demonstrated the benefit of using known outliers in training. Using 40 authorized transmitters, we were able to identify new devices from authorized devices with an accuracy of 84%. The improved results obtained in our work can be partially attributed to relying on the packet preambles, as discussed in Section VI.

Compared to [28], in this work, we consider two additional approaches which are OpenMax [55] and autoencoders [30] and the latter is shown to improve transmitter authorization with fewer transmitters. The results are presented as probability of false alarm and probability of detection providing more insights on the obtained performance. We further explore considerations for neural network architecture design with regard to outlier detection and how it differs from classification. The dataset used is expanded to include more transmitters, captured over a period of five days, increasing the confidence of our results and exploring temporal generalization.

III. Classification, Openset Recognition, and Anomaly Detection

In this section, we highlight the difference between classification, openset recognition, and anomaly detection. A closed set classifier determines boundaries that separate a pre-defined finite set of classes, shown as colored circles in Fig. 1(a). As such, for samples from new classes (shown as squares in Fig. 1(a)) the classifier will predict the nearest class. This poses a grave security risk for a wireless authentication system. Since, it is impossible to train a classifier on all the transmitters in the world, an approach that generalizes to signals from new unseen transmitters is needed.

We consider two formulations to address this limitation. The first one is posing the problem as an anomaly detection [30]. Anomaly detection aims to identify instances which are different from the normal. Hence, it finds boundaries around the seen classes (considered as normal cases), as shown in Fig. 1(b). The limitation, however, is that it treats all authorized transmitters as a single class. An overestimation of the determined boundaries could lead to errors in outlier detection as illustrated for outlier 2 in Fig. 1(b). Open set classification [29] takes an approach similar to anomaly detection while additionally classifying among the known samples. Hence, it isolates each class on its own as shown in Fig. 1(c).

Classifying among transmitters using only received signals in a robust manner is a challenging problem on its own. Extending it to open set poses even more challenges. Unlike approaches using handcrafted features [5], [11], [12], [33], which use well separated features like CSI, in order to leverage the power of deep learning, we use raw IQ samples. The challenge for the deep neural networks is to learn features which separate the known classes from the unknown classes, for which no training samples are available.

IV. System Model and Problem Formulation

We consider a finite set of authorized transmitters given by $A = \{A_1, A_2, \dots, A_{|A|}\}$ that are allowed to send data to a receiver R, where $|A|$ is the size of the set A. When some transmitter T sends a set of symbols $\boldsymbol{x}$, the signal received by R is $f_T(\mathbf{x}, t)$. The function f_T represents the time variant RF fingerprint, which captures the transmitter hardware fingerprints and wireless channel effects. Since the channel depends

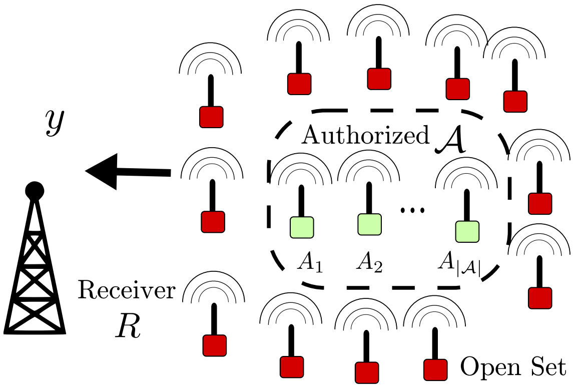


Fig. 2. Signal y is received by receiver R. We want to determine if it was sent by an authorized transmitter in the set A or a new unseen transmitter.

on the environment surrounding the transmitters, it is more prone to temporal variation.

The authorization problem can be formulated as shown in Fig. 2: receiver R receives a signal $\boldsymbol{y}$ from some transmitter T and wants to determine whether the transmitter T belongs to the authorized set or not without decoding $\boldsymbol{y}$. This can be formulated as the following hypothesis test:

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{y} = f_T(\mathbf{x}, t), T \in A \\ \mathcal{H}_1 &: \mathbf{y} = f_T(\mathbf{x}, t), T \notin A \quad \forall \quad t \end{aligned} \tag{1}$$

Here, $\mathcal{H}_0$ corresponds to an authorized transmitter and $\mathcal{H}_1$ corresponds to an outlier.

Additionally, in cases where each authorized transmitter has different privileges, we might be interested in classifying it within the authorized set, which can be formulated as finding $\hat{A}$ that is most likely to have generated $\boldsymbol{y}$, defined as

$$\hat{A} = \underset{T}{\operatorname{argmax}} \Pr(f_T(\mathbf{x}, t) = \mathbf{y}|\mathbf{y}), \qquad T \in A, \quad \forall \ t \tag{2}$$

While the anomaly detection problem addresses only problem (1), the open set problem addresses both (1) and (2). Since classification has been studied extensively in the literature, our main focus in this work is on the results of outlier detection when using either formulation.

To improve outlier detection, we propose using an additional class of *known* outlier transmitters $K = \{K_1, K_2, \dots, K_{|\mathcal{K}|}\}$, where $K \not\subset A$. Samples from transmitters in K will be used during training to assist the outlier detector to differentiate between authorized and non-authorized transmitters. In practice, samples from the set K can be obtained by capturing data from a finite number of non-authorized transmitters. For evaluating performance on unseen transmitters, we will use a set of *unknown* outlier transmitters $\mathcal{O}$. All signals from transmitters in $\mathcal{O}$ are reserved only for the test set.

V. Machine Learning Approaches

In this section, we discuss machine learning approaches to address this problem. An approach consists of a neural network, followed by an output processing stage to decide whether a signal is an outlier or not. The neural networks are used as function approximators [56]. They a map an input $\boldsymbol{y}$ to an output $\boldsymbol{z}$ using parameters θ, such that $\mathbf{z} = g_\theta(\mathbf{y})$. The parameters θ are learned by applying stochastic gradient descent on labeled training data. For this work, the input is the raw IQ samples and the output and its interpretation differs from one approach to the other. As for the processing stage, for some approaches, it involves modifying a threshold to control the sensitivity to outliers.

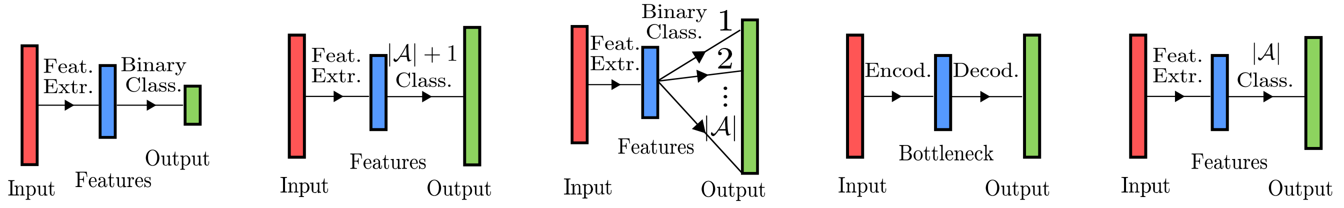
1) *Discriminator (Disc):* This the most intuitive approach for outlier detection. It consists of a binary classifier that outputs a decision on whether the signal is an outlier or no. While simple, the main limitation of this method is its complete reliance on the known outlier set for training. In terms of architecture, the discriminator has as single scalar output z as shown in Fig. 3(a). z is generated by a sigmoid function and takes a value between 0 and 1. The labels for authorized transmitters and outliers are $l = 0$ and $l = 1$, respectively. A threshold γ is used to make a decision with $\mathcal{H}_1$ declared if $z > \gamma$; $\mathcal{H}_0$ is declared otherwise.

2) *Discriminating Classifier (DClass):* This approach detects outliers and classifies transmitters within the authorized set. It consists of a multiclass classifier having $|A|+1$ outputs. The first $|A|$ outputs correspond to the authorized class and the last class corresponds to outliers. This classifier is expected to perform better than Disc, as individually labelling transmitters should help it extract better features. To train this network, we also need known outliers similar to Disc. A signal is classified as an outlier if the maximum activation corresponds to the last class; it is considered authorized otherwise. For this architecture, it is not straightforward to design an adjustable threshold as in Disc.

3) *One vs All (OvA):* OvA consists of multiple binary classifiers, one for each authorized transmitter. While the Disc network can serve as a binary classifier, reusing Disc increases the network size due to repeating the feature extractor. A better way to implement OvA is shown in Fig. 3(c). In this approach, all $|A|$ binary classifiers share the same feature extractor similar to what was proposed in [57]. Unlike Disc, OvA does not require a known set of outliers as long as $|A| \geq 2$, since for binary classifier i, signals from all transmitters $j \neq i$ are considered outliers. The output of this network will be a vector $\boldsymbol{z}$ of $|A|$ real numbers such that $\mathbf{0} \leq \mathbf{z} \leq \mathbf{1}$, where $\mathbf{0}$ and $\mathbf{1}$ are the vectors of all-zeros and all-ones, respectively. Following the notation in [57], the labels for a sample from authorized transmitter A_i will have $l_i = 1$ and $l_j = 0 \ \forall j \neq i$ and a known outlier, if used, will have $\boldsymbol{l} = \mathbf{0}$.

The decision is based on $|A|$ thresholds given by $\boldsymbol{\gamma}$, where element γ_i is the threshold for z_i. Each binary classifier i declares a sample to belong to its class if $z_i > \gamma_i$. A signal is declared to be an outlier (corresponding to $\mathcal{H}_1$), if all discriminators declare the signal to be not within their class ($\mathbf{z} \leq \boldsymbol{\gamma}$), and to be within the authorized set (corresponding to $\mathcal{H}_0$) otherwise.

4) *OpenMax (OpMx):* OpenMax is a popular approach for openset recognition [55]. It consists of modifying a trained classifier having $|A|$ softmax outputs. This modification is based on the statistical analysis of activations of authorized transmitters. It works as follows; the output activation vector $\mathbf{v}$, obtained prior to softmax, is processed to generate a



(a) Disc (b) DClass (c) OvA (d) AutoEnc (e) OpenMax

Fig. 3. High level architecture of the proposed methods. Autoencoder consists of an encoder and a decoder. The remaining ones consists of a feature extractor followed by one or more classifiers.

modified activations vector $\mathbf{v}'$ having $|\mathcal{A}|+1$ outputs, with the additional output corresponding to outliers. The modified activation vector is given by

$$v_i' = \begin{cases} v_i \omega_i, & i \in \{1, \dots, |\mathcal{A}|\} \\ \sum_{i=1}^{|\mathcal{A}|} v_i(1-\omega_i) & i = |\mathcal{A}|+1 \end{cases} \qquad (3)$$

where ω_i represents our confidence in the membership of the given sample to class *i*. The concept behind calculating the vector $\boldsymbol{\omega}$ is that the activation vectors of samples belonging to the same class are similar, while those belonging to unseen classes are different from all classes. This is implemented by calculating the mean activation vector $\bar{\mathbf{v}}_i$ for each class *i* using the training set. The distance $d_i(\mathbf{v}) = \|\bar{\mathbf{v}}_i - \mathbf{v}\|$ represents the similarity of the sample generating vector $\mathbf{v}$ to class i. On the training set, for the correctly classified samples, the distance of each sample belonging to a class *i* is calculated. Using extreme value theorem [58], the tail of the distribution is calculated by fitting the τ samples with the largest d_i to a Weibull distribution having parameters (m_i, η_i). Then, for the α classes having the highest activations, $\boldsymbol{\omega}_i$ is calculated by evaluating the probability of belonging to the tail of distribution of *i* using

$$\omega_i = 1 - R_\alpha(i) \times \text{WeibullCDF}(d_i(\mathbf{v}), (m_i, \eta_i)) \qquad (4)$$

where $R_\alpha(i) = \frac{\alpha - i}{\alpha}$ is a calibrator with parameter α and $\text{WeibullCDF}(x, (m, \eta)) = \exp(-(x/\eta)^m)$, as explained in [55], [59]. After calculating $\mathbf{v}'$, uncertain outputs are rejected if the confidence is below some threshold ϵ. This is done by applying the softmax function to $\mathbf{v}'$ to obtain the vector *z*. Then we calculate $i = \text{argmax}(\mathbf{z})$; the sample is considered an outlier if $i = |\mathcal{A}|+1$ or $z_i < \epsilon$. Since OpenMax is based on a classifier, it does not benefit from known outliers in training.

The approach for transmitter authorization in [27] consists of modifying a classifier similar to OpenMax. However, their approach only uses the maximum value of the softmax output, while OpenMax uses the entire activation vector. Hence, it uses more information from the neural network output.

5) AutoEncoder (AutoEnc): Autoencoders are commonly used for anomaly detection [30]. They detect anomalies by reconstructing their input and evaluating the reconstruction error. They consist of an encoder mapping the data into a smaller dimension, the bottleneck, followed by a decoder trying to reconstruct the original input. During training, autoencoders learn the distribution of the training data. When the autoencoder processes anomalous data, it generates a higher reconstruction error, which can be used to detect anomalies.

TABLE I
FEATURES OF APPROACHES

Approach	Samples from $\mathcal{K}$	Adjustable Threshold	Classifies $\mathcal{A}$
Disc	Necessary	Yes	No
DClass	Necessary	No	Yes
OvA	Optional	Yes	Yes
OpMx	Unsupported	No	Yes
AutoEnc	Unsupported	Yes	No

During training, the objective of the autoencoder is to reduce the mean squared error, $\text{MSE} = \|\mathbf{y} - \hat{\mathbf{y}}\|$ where $\boldsymbol{y}$ is the input and $\hat{\mathbf{y}}$ is the output of the autoencoder. A signal is considered an outlier if this error is bigger than some threshold γ.

To summarize, the proposed approaches are either based on classifiers (Disc, DClass, OvA, OpMx) or uses signal reconstruction (AutoEnc). The classifier-based approaches share a neural network-based feature extractor and differ only in the output labels and the last layers activation function. Some of the classifier-based approaches, beside outlier detection, classify the authorized signal among the set A. As for the known outlier set K, it is necessary for the training of some approaches, can be used to improve the performance of others, and cannot be used in some other approaches. Table I provides a high level comparison of approaches.

For the approaches which have an adjustable threshold, a tight threshold would lead to signals from authorized transmitters being mistakenly rejected (high probability of false alarm P_{FA}) and a loose threshold would fail to recognize many outlier signals (low probability of detection P_D). The desired value of P_{FA} can be obtained by empirically calibrating the thresholds on the training set. The method for setting the thresholds along with any other hyperparameters used in this work is discussed in the Appendix.

VI. DATASET

The dataset was captured using off-the-shelf WiFi modules (Atheros 5212, 9220, and 9280) as transmitters and a software defined radio (USRP N210) as a receiver. The capture was performed in the Orbit testbed grid [54]. Orbit testbed grid consists of 400 nodes arranged in a 20 × 20 grid with a separation of one meter. The receiver was chosen near the center of the grid and 163 nodes surrounding it were used as transmitters. An image of the testbed along with the location of the transmitters and receivers are shown in Fig. 4.

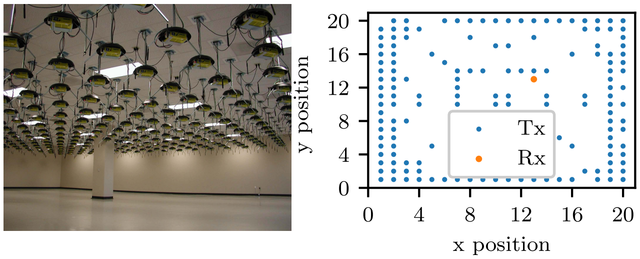


Fig. 4. Image of orbit testbed. The participating nodes are plotted.

The capture was done over IEEE 802.11g Channel 11 which has a center frequency of 2462 MHz and a bandwidth of 20 MHz. Captures were taken using a sampling rate of 25 Msps, over a duration of one second. After the IQ capture was complete, the packets were extracted using energy detection. The number of packets captured during this period for each transmitter varied according to the WiFi rate control and their total number is over 300,000.

While it is possible to use the entire packet payload for training, this would make the data contained in each slice different. In our previous work [25], we have shown that using the slices with the same data leads to a better performance than using slices containing random data. This was also verified in [50]. Hence, from each captured WiFi packet, we used the first 256 IQ samples containing the preamble. The IQ samples were normalized to have a unity average magnitude without any further preprocessing. For transmitter authorization in [27], the entire packet payload was used for training and inference was done on slices of a packet, which are combined to obtain one decision per packet. While this method leads to having more data, it makes the learning task of the neural network harder, as was previously demonstrated in [25], [50]. Since in this work we only consider preambles and due to the similarity of their approach to OpenMax, we don't consider their approach in our evaluation.

As was pointed out in recent works [60], [61], the fingerprints learned by neural networks can be dependent on the channel and not the transmitter. This causes significant degradation in the recognition performance if the testing data was captured on a different day than the training capture. To this end, using the previous setup, we made five data captures over 5 different days. The data from the first capture is the one used in our previous work [28], and contained data from less transmitters. Also, it was made two months earlier than the remaining captures. The data from the last day was kept exclusively for testing.

VII. Network Architectures

In this section, we describe the architecture of the proposed networks. The design of neural networks (NN) plays a crucial role in the performance achieved. While there are many variants of networks, in this work we use feed forward neural networks. They are typically built using convolutional and dense neural layers [56, II]. While there is no systematic way to design NN, there are known guidelines for optimizing their hyperparameters [56, 11.4].

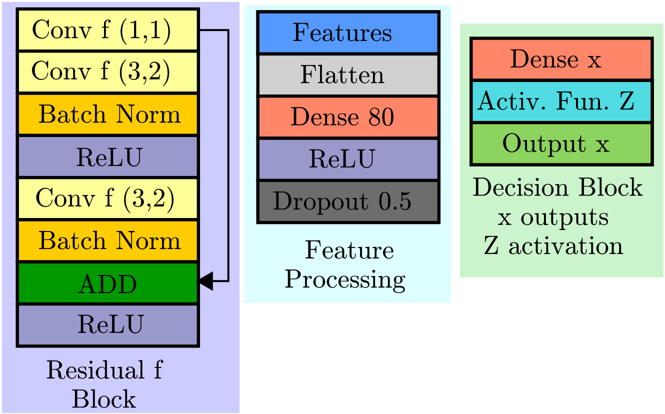


Fig. 5. The residual, feature processing, and decision blocks used as building blocks for the neural networks considered in this work.

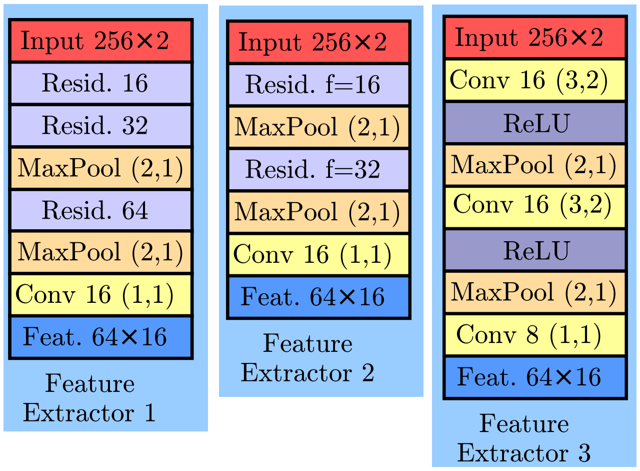


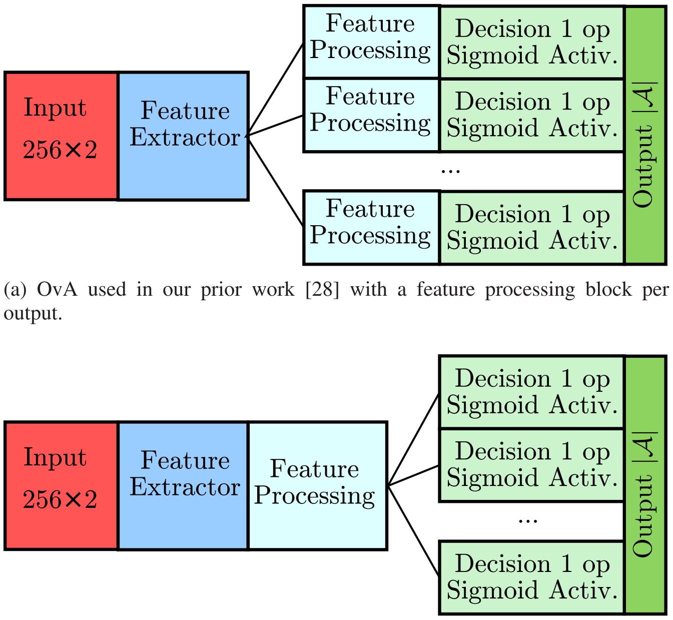
Fig. 6. The three different architectures compared in this work. Feature extractor 2 was eventually selected. The residual block is described in Fig. 5.

Since the feature extractor is an essential component for Disc, DClass, OvA, and OpMx, we consider several alternatives for it and compare to the architecture used in our previous work [28]. All the architectures in this work are built using the blocks shown in Fig. 5; the residual block which has f filters [36], the feature processing block, and the decision block using activation function Z to generate a given number x outputs. The values of f, Z, and x are specified when the block is used. For all the figures used in this section, we use a color code; each type of layer (convolutional, dense, etc) is given a unique color, and the background color of each modular block is used as a layer color when this block is used.

A. Architecture Comparison

For choosing the best architecture and contrasting to the architectures in our previous work, we use OvA as a benchmark using data from 10 authorized transmitters ($|A| = 10$) collected on the same 4 days and no known outliers. Details for the network training is deferred for later in this article.

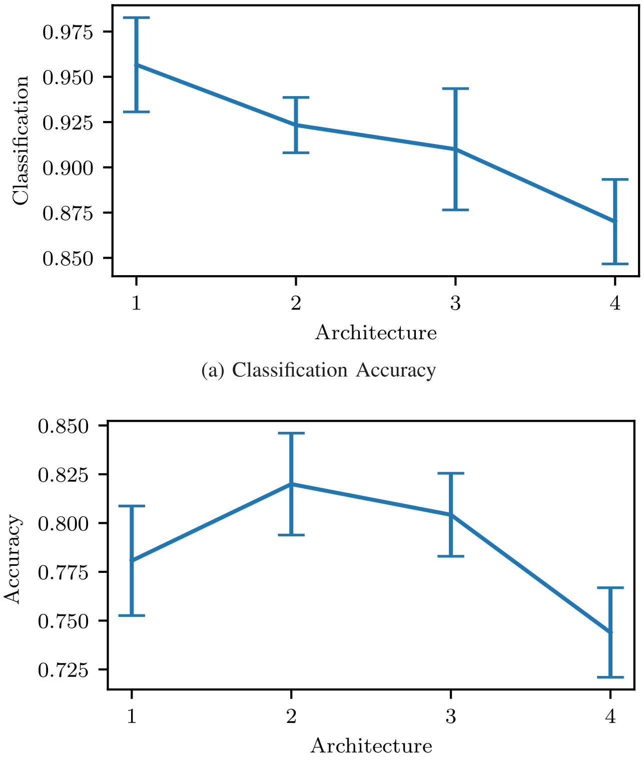
We consider 4 architectures of OvA. OvA 1 (from [28]) which has $|A|$ Feature Processing block and OvA 2, which uses the same Feature Extractor 1 with a common Feature


Input
256×2
Feature Extractor
Feature Processing
Decision 1 op Sigmoid Activ.
Feature Processing
Decision 1 op Sigmoid Activ.
...
Feature Processing
Decision 1 op Sigmoid Activ.
Output |A|
Input
256×2
Feature Extractor
Feature Processing
Decision 1 op Sigmoid Activ.
Decision 1 op Sigmoid Activ.
...
Decision 1 op Sigmoid Activ.
Output |A|

(a) OvA used in our prior work [28] with a feature processing block per output.

(b) OvA used in this work having a shared feature processing block. Several feature extractors were compared and eventually Feature Extractor 2 was used.

Fig. 7. We considered two OvA architectures, the first having a feature processing block for each output, and the second using a common one. The feature extractors are described in Fig. 6 and the feature processing and decision blocks in Fig. 5 .



(a) Classification Accuracy

(b) Outlier Detection Accuracy

Fig. 8. The average performance of the OvA architectures from Table II are shown as a blue solid line. The error bars represent the standard deviation due to training 10 repetitions of the same network using the same data.

TABLE II
OvA Network Sizes

OvA	Description	# params.
1	Feat. Ext. 1 + 10 Feat. Process.	890,170
2	Feat. Ext. 1 + Common Feat. Process.	152,170
3	Feat. Ext. 2 + Common Feat. Process.	99,754
4	Feat. Ext. 3 + Common Feat. Process.	46,226

Processing blocks as shown in Fig. 7. Moving from OvA 2 to OvA 3 and OvA 4, we use smaller feature extractors 1,2, and 3 respectively which are described in Fig. 6. The summary of each architecture and the number of trainable parameters are shown in Table II from which we see that from OvA 1 to OvA 4 the network gets smaller.

Due to the random initialization of the weights, along with randomness in batch division during training, the same network trained using the same data can give different results. To have confidence on the significance of our results, each network is trained from scratch using the same data for ten repetitions and the statistics of the results are shown. Fig 8(a) shows the classification results of the authorized nodes. The larger networks perform better, as expected; by having more learning capacity, the networks are better at distinguishing between transmitters. But looking at the accuracy of outlier detection in Fig. 8(b) for OvA 1 and 2, we see a rather surprising result; the largest network actually performs worse than some of the smaller networks. Since we want the network to generalize to new transmitters, we want each binary classifier of OvA to learn only the characteristics of its designated transmitter, while rejecting any other transmitter. But once the learning capacity of the per transmitter branches increases beyond a certain point, it starts to learn the characteristics of the remaining transmitters. Although this improves classification and minimizes training and validation loss, it does not generalize well to outlier detection.

As we decrease the capacity of the common feature extractor in OvA 2 to 4, the performance of both classification and outlier detection degrades. Since this shared block extracts features and does not make a decision, the more learning capacity it has, the better the results. Although feature extractor 1 performs about 1% better, as a design choice, we use feature extractor 2 for the rest of this work because of its smaller network size. For the remaining of this work, we use OvA 3, having the common feature processing architecture.

B. Architecture Description

The architectures for Disc, DClass, and OpMx consist of Feature Extractor 2 followed by a feature processing block and a decision block with 1 sigmoid, $|\mathcal{A}|+1$ softmax, and $|\mathcal{A}|$ softmax respectively. Disc was provided with a larger feature processing network having an additional Dense network with 100 neurons after the flattening to be comparable in size to the other networks. The autoencoder architecture is shown in Fig. 9. It consists of encoder with a bottleneck consisting of 32 samples followed by a decoder which reconstructs the signal.

The number of trainable parameters of each network is shown in Table III. The network sizes of OvA, DClass, and

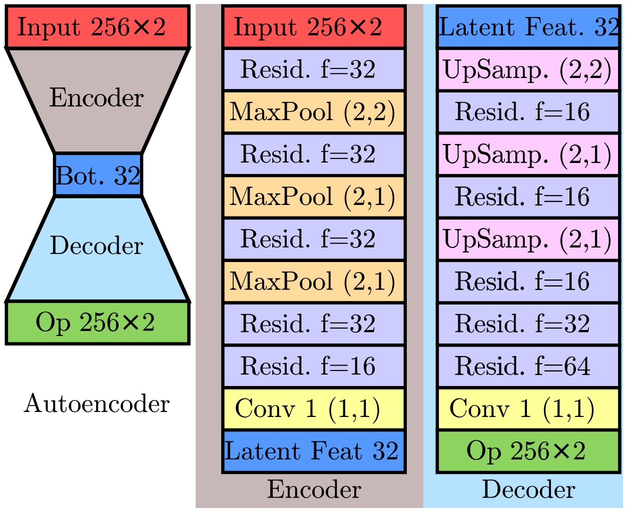


Fig. 9. The architecture of the autoencoder. The residual block is described in Fig. 5.

TABLE III
NETWORK SIZE

Net	# of trainable prams.
OvA	98944 + 81 $\lvert\mathscr{A}\rvert$
Disc	127,605
DClass	99,025 + 81 $\lvert\mathscr{A}\rvert$
AutoEnc	109,362
OpMx	98944 + 81 $\lvert\mathscr{A}\rvert$

OpMx scale with $|\mathcal{A}|$.[1] AutoEncoder and Disc have a fixed number of parameters for any value of $|\mathcal{A}|$. Notice that OvA and OpMx have an identical number of parameters while DClass differs by only an additional 81 parameter.

From an architecture perspective, Disc, OvA, OpMx, and DClass are very similar. They share the same feature extractor and feature processing blocks, which constitute most of their neural network. The difference between these methods come from the type of activation function, data labeling, and post processing performed. These differences lead to conceptual differences in the approach as we have discussed, leading to different characteristics as we summarized in Table I, and will lead to significantly different performance.

VIII. Evaluation Procedure

In this section, we describe the evaluation procedures used through out this work. We discuss the evaluation metrics used, steps to avoid the dependence on the specific division of the dataset to different sets ($\mathcal{A}$, $\mathcal{K}$, $\mathcal{O}$), and the way we evaluate how the approach generalizes across time.

As stated earlier, for many of these networks, there is a threshold which defines the trade-off between detection and false alarm. This trade-off is typically represented by the Receiver Operating Characteristic (ROC) curve relating the probabilities of false alarm and detection. To compactly visualize the results, we consider the area under the ROC curve (AUC) calculated using integration, which measures performance in a manner independent of the threshold [62]. The procedure for calculating the ROC curve for each approach is described in the Appendix. Still, when the network is deployed, we have to choose a threshold. Given a specific threshold-set, we calculate the accuracy of correctly identifying outliers over a balanced test set, such that random guessing would yield a 50% accuracy. In that case, the accuracy is the average of the performance on the authorized samples given by $1 - P_{FA}$, and the outliers given by P_D. The thresholds are chosen to provide a high value of accuracy according to the procedures discussed in the Appendix. We use the accuracy instead of the F1 score since it is an easier metric to interpret when the dataset is balanced. Classification accuracy results included for the authorized samples are evaluated for a balanced version of the test set having the same number of the samples from each authorized transmitter, where any trivial or random guess would yield an accuracy of $1/|\mathcal{A}|\%$. Note that as stated earlier, not all methods have an adjustable threshold (as summarized in Table I) and hence won't have a corresponding AUC result.

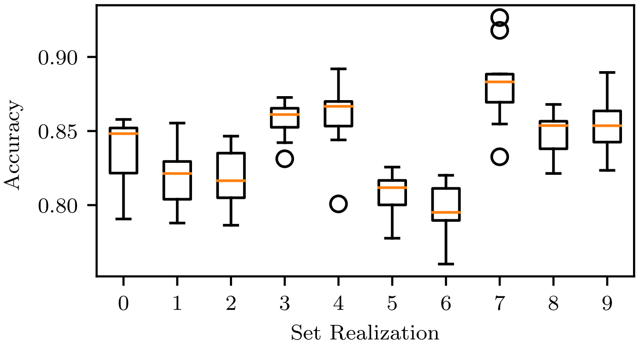
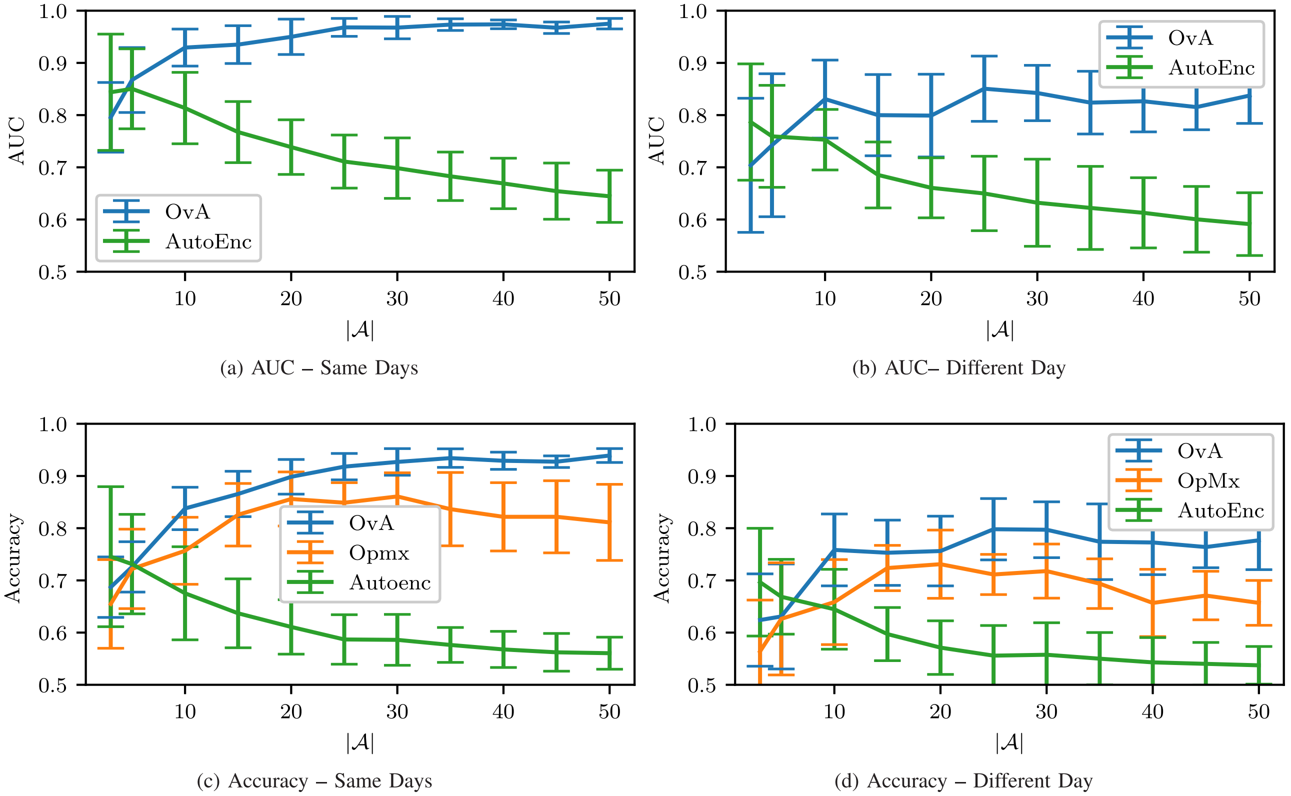


Fig. 10. Box plot of the outlier detection accuracy of OvA is shown for several realizations of the sets. For each set realization, a network is trained for 10 repetitions. The center line represents the median, the box represents the first and third quartiles, and the whiskers represent the range, except for outlier points which are represented as circles.

Unlike with classification, where typically all transmitters available are used, outlier detection involves dividing the transmitters into sets of authorized transmitters, outliers, and possibly known outliers. Since RF fingerprints are random, some transmitters are more similar than others, and a comprehensive evaluation cannot be done using only one realization of the sets. This adds another source of variability to our results, besides the inherent randomness in training neural networks. To demonstrate this, we train OvA using 10 authorized nodes and evaluate it using 63 outlier nodes picked randomly from the 163 transmitters. Ten random realizations of these sets are compared and for each we train 10 repetitions. The results are shown as a box plot in Fig. 10, from which we can see up to 9% difference in the median due to the different realizations of the sets. This is more significant than the difference between the first and third quartiles due to training randomness which did not exceed 3%. Based on these findings, our evaluation considers multiple realization of the sets while only considering one repetition.

[1] In our previous work [28], since we repeated the feature processing block for OvA, the network size increased with $|\mathcal{A}|$ at a much higher rate.



(a) AUC – Same Days

(b) AUC– Different Day

(c) Accuracy – Same Days

(d) Accuracy – Different Day

Fig. 11. Average outlier detection performance of several approaches as we change $|A|$. Error bars represent the standard deviation for different realizations of the sets.

As for assessing the ability of our network to generalize through time, we create two test sets: a same-day test set which was captured on the same days as the training set and the different-day test set, which was captured on a different day than the training data.

Based on the previous considerations, we describe our evaluation procedure. For certain values of $|A|$, $|\mathcal{K}|$, and $|\mathcal{O}|$, we randomly partition the dataset to A, K, and $\mathcal{O}$ to form 10 realizations of $\{A, K, \mathcal{O}\}$. All approaches are evaluated using the same 10 realizations and the results are shown in terms of mean and standard deviation. For the training data and the same-day test data, we use only samples from the captures made on the first four days, while the last day capture is entirely left for different day testing.

Given a realization of A, K, and $\mathcal{O}$, for training and validation, we use 70% of the samples belonging to A, and all the samples belonging K, from the same day data. The combination of this data is split into 80% for training and 20% for validation. The same day test set contains all samples from $\mathcal{O}$ and the remaining 30% of A. For different realizations of the sets, the dataset can get highly imbalanced. To avoid degenerate solutions, where the network always predicts the class with the majority of samples, the training loss is weighted based on class frequency. As for the different day test set, it is obtained by combining all samples from A and $\mathcal{O}$ from the last day capture.

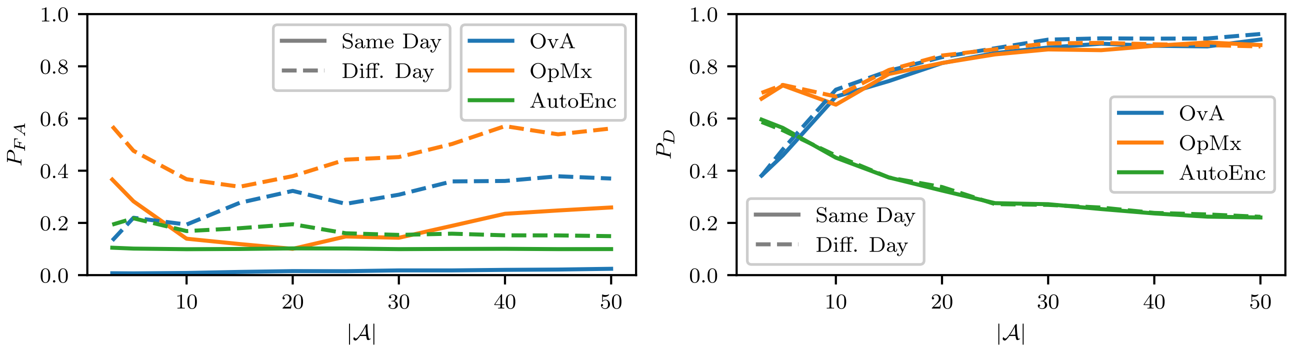
The training used 10 epochs using the ADAM optimizer with a learning rate of 0.001. The weights corresponding to the epoch which produced the lowest validation loss are kept. Data was first normalized, then augmented by adding noise with a variance of 0.01 and applying a uniform random phase shift. Cross-Entropy was used as the loss function for Disc, DClass, OvA, and OpMx with classes weighted depending on the number of samples of each class. AutoEnc used MSE loss.

IX. Transmitter Set Sizes Evaluation

In this section, we explore the effect of changing the size of the required authorized set A, and evaluate the effect of having a known outlier set K and its size on the ability of the network to distinguish authorized signals from outliers. Throughout this section, we used $|\mathcal{O}| = 63$ for the evaluation.

A. Authorized Set

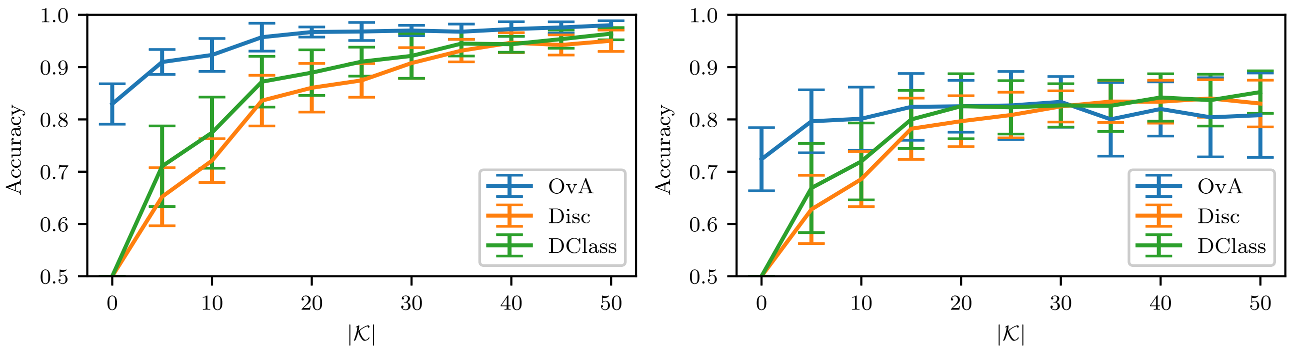
We start the evaluation by considering no known outliers, i.e., $|\mathcal{K}| = 0$. We want to know how large the set A has to be for good outlier detection and what performance can be achieved thereof. Results are shown for AUC and accuracy in Fig. 11(a) and Fig. 11(c) for the same-day test. For OvA, we see that as we increase the number of authorized nodes, the average AUC increases and its standard deviation decreases, showing less dependence on the set realization. The accuracy, shown in Fig. 11(c), follows the same trend, and we are able to achieve accuracies above 90% on the average when $|A|$ is more than 20. The reason behind this pattern is that as $|A|$ increases, each binary classifier has more signals from other transmitters, helping it learn better its designated transmitter without memorizing others, leading to better generalization.



(a) Probability of False Alarm

(b) Probability of Detection

Fig. 12. Average outlier detection performance of several approaches as we change $|\mathcal{A}|$. Error bars are omitted for clarity. Solid lines represent same days test, and dashed line represent different day test..



(a) Accuracy – Same Days

(b) Accuracy – Different Day

Fig. 13. Average outlier detection performance of several approaches as we change $|\mathcal{K}|$. Error bars represent the standard deviation for different realizations of the sets.

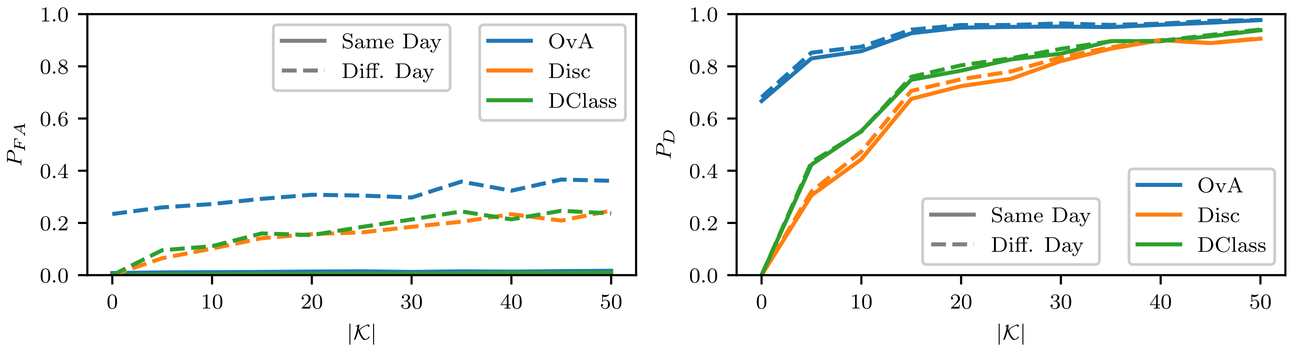
This interpretation is supported by the observation that the improvement in accuracy is due to the decrease in P_D for the same P_{FA} as shown in Fig. 12(b) and Fig. 12(a), respectively.

As for autoencoders, the trend seems to be reversed in Fig. 11(a) and Fig. 11(c). As $|\mathcal{A}|$ increases, both the accuracy and AUC decrease. Autoencoders generate a compressed representation of their input by memorizing their distribution. For small $|\mathcal{A}|$, this is the distribution of the authorized transmitters. As $|\mathcal{A}|$ increases, they learn the distribution of signals in general, independent of the transmitter. Hence, they reconstruct signals for unknown transmitters with low MSE and fail to detect them. This is verified by looking at the decreasing P_D curve for AutoEnc in Fig. 12(b), while P_{FA} is almost constant in Fig. 12(a). This trend coincides with our visualization in Fig. 1(b).

For OpMx, the accuracy increases until $|\mathcal{A}| = 20$ and then slightly decreases. This fluctuation is mostly attributed to P_{FA}, as shown in Fig. 12(a). The results of OpMx depend on the value of the activation vectors (AV) with respect to tail distributions and the uncertainty threshold ϵ. The key is understanding that the modified activations $\mathbf{v}'$ calculated using (3) reduces the AV of the top α classes. After calculating the output $\mathbf{z} = \mathrm{softmax}(\mathbf{v}')$, the maximum $z_i = \max\{\mathbf{z}\}$ is thresholded using ϵ. For small $|\mathcal{A}|$, classification for authorized is highly confident, leading to AV belonging to the tails of other classes, pushing z_i below ϵ, and leading to false alarms. As $|\mathcal{A}|$ becomes larger, this confidence decreases, leading to an improvement in P_{FA}. However, the similarity between AVs of different classes decreases, pushing them to the tail of other distributions as $|\mathcal{A}|$ increases beyond a certain point, leading to an increase in P_{FA}.

In comparison, we see that for small $|\mathcal{A}|$, namely $|\mathcal{A}| = 3$, AutoEnc gives the highest accuracy on average because it is able to capture the distributions of small number of transmitters. At $|\mathcal{A}| = 5$, all methods are equally as good. As $|\mathcal{A}|$ increases, OvA gives the best performance because each branch uses all the data to learn its transmitter without memorizing the other transmitters.

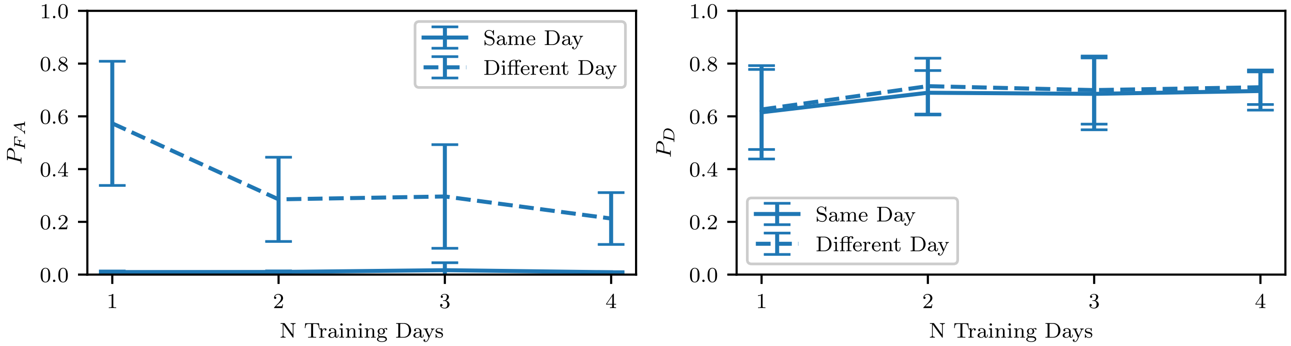
In Fig. 11(d), we plot the accuracy for a different day test. The plots follow the same trends, but the accuracy of OvA and OpMx drop by about 15% while AutoEnc drops by only 5%. The reason behind this drop is clearer by inspecting the P_{FA} and P_D separately shown as dashed lines in Fig. 12. The P_{FA} and P_D curves represent the same information as confusion matrices—applied to binary classification—in a more compact manner. From Fig 12(b), we see that the performance in detecting the new transmitters is almost unaffected. The drop in accuracy is a result of the failure to identify authorized transmitters as shown in the P_{FA} curves in Fig 12(a). This is reasonable since any change on unseen transmitters should not have any effect, unlike changes in the learned transmitters. For OvA, we see that as we increase $|\mathcal{A}|$, P_{FA} increases, since identifying transmitters from data captures on different days becomes even more challenging as we increase the number



(a) Probability of False Alarm

(b) Probability of Detection

Fig. 14. Average outlier detection performance of several approaches as we change $|\mathcal{K}|$. Error bars are omitted for clarity. Solid lines represent same days test, and dashed line represent different day test.



(a) Probability of False Alarm

(b) Probability of Detection

Fig. 15. Average outlier detection performance against the number of training days. Error bars represent the standard deviation for different set realizations. Solid lines represent same days test, and dashed line represent different day test..

of transmitters. The smaller gap on different day test using AutoEnc is explained by the encoders ability to learn more general features of the signal compared to OvA and OpMx, which leads to an overall smaller drop in accuracy, which also causes the lower P_D.

B. Known Set

We expect that seeing more known outliers would help the network differentiate the authorized transmitters from the outliers. To show this, we evaluate the performance of the approaches that support having K as input, as a function of $|\mathcal{K}|$ given $|\mathcal{A}| = 10$. The accuracy curves are shown in Fig. 13(a). As stated earlier, at $|\mathcal{K}| = 0$, DClass and Disc don't have any outlier samples for training and predict everything as authorized. From Fig. 13(a), we see that the accuracy of all methods improve as we increase the number of known outliers. We also note that OvA is performing noticeably better than the others. This can be understood by realizing that in OvA, each binary classifier sees more samples to reject, the known outliers and the samples from other authorized transmitters. Thus, it is able to isolate its class better. DClass and Disc, on the other hand, only learn to reject samples from K. This is further supported by looking at the curves for P_{FA} and P_D shown in Fig. 14(a) and Fig. 14(b), where we see that the accuracy improvement is due to the probability of detection. DClass slightly outperforms Disc because the labels of A help it extract better features compared to Disc. So it can be concluded that even if we are not interested in classifying among the nodes in A, including these labels in training improves the outlier detection performance. Fig. 13(b) shows the accuracy curves when using the data from a different day for testing. Again, we see the same trend from the previous section, both AUC and accuracy drop by about 15% for all methods due to the degradation of P_{FA}.

X. Dataset Training Days Evaluation

In this section, we evaluate the effect of the dataset construction on the ability of the proposed approaches to generalize over time. While developing methods to counter temporal variation in RF fingerprints is not the main focus of this article and has been discussed in other works [60], we study its effect on transmitter authorization. For our evaluation, we only consider the OvA architecture with $|\mathcal{A}| = 10$ and $|\mathcal{K}| = 0$. We built four datasets, where dataset i contains the data captures on dates prior to and including day i. The network was trained according to the same procedures discussed earlier and the results are shown in Fig. 15. From Fig. 15(a), we see that as the number of days included in training increases, P_{FA} decreases. On the other hand, P_D is almost unaffected. The larger improvement from 1 day capture to two day captures is explained by the fact that the capture on day 1 was two months earlier than the remaining captures. During this long period, the transmitter fingerprints changed more significantly. The remaining captures were done on consecutive days, during

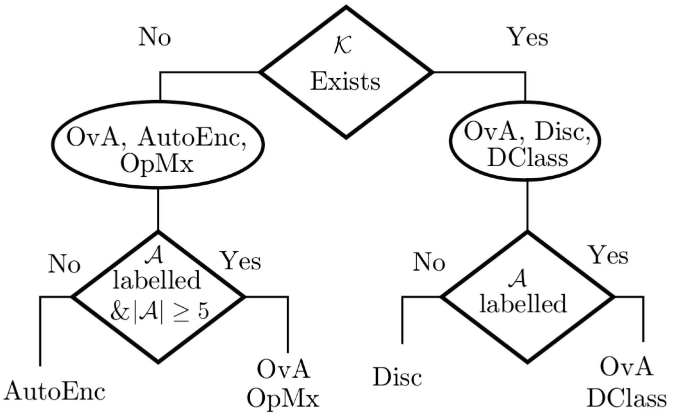


Fig. 16. A tree diagram summarizing the feasible architectures as a function of the dataset. The networks are ordered so that the one yielding the better performance comes first.

which fingerprints had less severe changes, and hence smaller improvements to P_{FA}. Hence, a simple way to improve the robustness against temporal variation is to collect data from the authorized transmitters over an extended period of time. Still, more sophisticated approaches are needed to close the gap between same day and different day testing.

XI. Summary

The results we obtained can be summarized as follows. *Regarding the dataset:* It is better to label the authorized transmitters even if we are not interested in classifying among them, as it enables us to use openset methods (OvA, OpMx, DClass) which outperform the anomaly detection methods (Disc, AutoEnc) in many cases. Furthermore, the data for authorized transmitters should ideally be collected over a span of multiple days; as we have demonstrated, the drop in outlier detection accuracy is due to misclassification of authorized signals, i.e., P_{FA}, and can be reduced by collecting the data over multiple days.

Regarding the approach: The dataset structure; whether it is labeled or not, and the sizes of $\mathcal{A}$ and $\mathcal{K}$, determines which approaches are feasible or are better. If we have no data from known outliers and no labels, our only option is AutoEnc. Without known outliers and with the availability of labels, for $|\mathcal{A}| \leq 5$, ignoring the labels and using AutoEnc is a better choice driven by its superior performance for small $|\mathcal{A}|$; if $|\mathcal{A}| > 5$, OvA gives the best performance. If we have a pretrained classifier, then OpMx is the best option. If we have $\mathcal{K}$, we can use OvA, Disc, and DClass. Without labels, we are limited to Disc. If we have labels, OvA is typically better than DClass with tunable thresholds. This is summarized in Fig. 16.

Regarding the network architecture: We have shown that if we have labeled data, the performance on classifying the transmitters does not necessarily correlate with the performance on outlier detection. A known outlier set is recommended to optimize the architecture.

Eventually, if we are able to collect the data as we wish (having labels, data for authorized transmitters captured over multiple days, and with data for known outliers) the best approach is OvA. When using 10 authorized transmitters and 50 known outliers, it yielded an outlier detection accuracy of 98% on the same day test set and 80% on the different day test set.

XII. Conclusion

In this article, we have considered the problem of transmitter authorization using RF fingerprints captured from raw IQ samples. Since this problem has been scarcely investigated in the wireless domain, we performed a comprehensive evaluation of the most prominent machine learning approaches from the openset recognition and the anomaly detection literature, as applied to our problem definition. The dependence of the evaluation results on the choice of transmitters was demonstrated and a simple strategy was proposed to reduce it. We have also shown that the performance of a given neural network model on closed set classification is not an indicator of its performance in outlier detection, indicating the need for architectures specifically designed for this problem. Also, we demonstrated that minor change in network architecture and data labeling can lead to a significantly different approaches. Using a known outlier set was proposed and was shown to improve the outlier detection accuracy. While classification based OvA gives the highest accuracy in most cases, it is outperformed by reconstruction based AutoEnc for small number of authorized. This opens the door for hybrid approaches combining classification and reconstruction. We also pointed out that the temporal variation of fingerprints is an open problem for transmitter authorization.

Appendix
Parameter Selection

For each approach, we describe how the ROC curve is calculated. We also state how a specific threshold is chosen to calculate the outlier detection accuracy along with choices of hyperparameters. Since the evaluation is to be done over multiple realizations, manual tuning is not possible and we provide a systematic way to set these values.

1) *Discriminator (Disc):* In Disc, we only have one threshold to make a decision. Ideally, we want the threshold to be as low as possible without falsely rejecting authorized transmitters. This can be done by adapting the threshold to tightly fit the predictions of authorized signals in the training set. We follow the approach proposed in [57], where the predicted output of the sigmoid for the correctly classified authorized training samples $\bar{z}_0$ (having labels equal to 0) is concatenated with its negative $-\bar{z}_0$ (to make the distribution symmetric around zero) and fit to a Gaussian distribution having mean 0. The standard deviation σ of these samples is calculated and a threshold of 3σ would allow the majority of authorized transmitters to be accepted. To deal with degenerate cases having large standard deviation, the threshold is set to $\gamma = \min(0.5, 3\sigma)$ in practice. As for obtaining the ROC curve to calculate the AUC, the value of γ is scanned from 0 to 1.

2) *One vs All (OvA):* OvA has $|\mathcal{A}|$ thresholds given by $\boldsymbol{\gamma}$. While it is possible to use one common threshold, we use multiple thresholds designed according to the same method of

Gaussian fitting used in Disc to calculate the accuracy as it yields better results. As for obtaining the ROC curve for the AUC calculation, we consider one single threshold γ scanned from 0 to 1 such that $\boldsymbol{\gamma} = \gamma \mathbf{1}$.

3) *OpenMax (OpMx):* As for the parameters, the tail size used to calculate the Weibull distribution is $\tau = 10$, $\alpha = \min(\lfloor |\mathcal{A}|/3 \rfloor, 5)$, and ϵ was chosen to be the 95% quantile of the maximum activation in the training data. To obtain these values, we started by considering the values proposed in [55] which was optimized for their dataset. However the performance of these values varied drastically as we varied the authorized set and from one realization to the other. After running several experiments and analyzing the activation values, we found that these parameters gave the best performance.

4) *AutoEncoder (AutoEnc):* We chose γ to be the 90% quantile of the mean squared error of the training data. The ROC curves used for calculating the AUC are obtained by scanning the value of γ from 0 to the maximum MSE.

References

[1] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.

[2] X. Wang, P. Hao, and L. Hanzo, "Physical-layer authentication for wireless security enhancement: Current challenges and future developments," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 152–158, Jun. 2016.

[3] W. Wang, Z. Sun, S. Piao, B. Zhu, and K. Ren, "Wireless physical-layer identification: Modeling and validation," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2091–2106, Sep. 2016.

[4] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, 2008, pp. 116–127.

[5] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, "Device fingerprinting to enhance wireless security using nonparametric Bayesian method," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1404–1412.

[6] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Fingerprinting Wi-Fi devices using software defined radios," in *Proc. 9th ACM Conf. Security Privacy Wireless Mobile Netw. (WiSec)*, Darmstadt, Germany, 2016, pp. 3–14.

[7] Y. Ren, L. Peng, W. Bai, and J. Yu, "A practical study of channel influence on radio frequency fingerprint features," in *Proc. IEEE Int. Conf. Electron. Commun. Eng. (ICECE)*, Dec. 2018, pp. 1–7.

[8] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid RF fingerprint extraction and device classification scheme," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 349–360, Feb. 2019.

[9] B. Chatterjee, D. Das, and S. Sen, "RF-PUF: IoT security enhancement through authentication of wireless nodes using *in-situ* machine learning," in *Proc. IEEE Int. Symp. Hardw. Orient. Security Trust (HOST)*, Washington, DC, USA, Apr. 2018, pp. 205–208.

[10] X. Zhou, A. Hu, G. Li, L. Peng, Y. Xing, and J. Yu, "Design of a robust RF fingerprint generation and classification scheme for practical device identification," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, Jun. 2019, pp. 196–204.

[11] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2571–2579, Jul. 2008.

[12] N. Gulati, R. Greenstadt, K. R. Dandekar, and J. M. Walsh, "GMM based semi-supervised learning for channel-based authentication scheme," in *Proc. IEEE 78th Veh. Technol. Conf. (VTC Fall)*, Sep. 2013, pp. 1–6.

[13] A. Weinand, M. Karrenbauer, R. Sattiraju, and H. D. Schotten, "Application of machine learning for channel based message authentication in mission critical machine type communication," Nov. 2017. [Online]. Available: arXiv:1711.05088.

[14] L. Senigagliesi, M. Baldi, and E. Gambi, "Statistical and machine learning-based decision techniques for physical layer authentication," Sep. 2019. [Online]. Available: arXiv:1909.07969.

[15] P. Zhang, T. Taleb, X. Jiang, and B. Wu, "Physical layer authentication for massive MIMO systems with hardware impairments," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1563–1576, Mar. 2020.

[16] H. Fang, X. Wang, and L. Hanzo, "Learning-aided physical layer authentication as an intelligent process," Aug. 2018. [Online]. Available: arXiv:1808.02456.

[17] S. U. Rehman, K. Sowerby, and C. Coghill, "Analysis of receiver front end on the performance of RF fingerprinting," in *Proc. IEEE 23rd Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2012, pp. 2494–2499.

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

[19] J. Yu, A. Hu, G. Li, and L. Peng, "A robust RF fingerprinting approach using multi-sampling convolutional neural network," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6786–6799, Aug. 2019.

[20] S. Gopalakrishnan, M. Cekic, and U. Madhow, "Robust wireless fingerprinting via complex-valued neural networks," May 2019. [Online]. Available: arXiv:1905.09388.

[21] G. Baldini, C. Gentile, R. Giuliani, and G. Steri, "Comparison of techniques for radiometric identification based on deep convolutional neural networks," *Electron. Lett.*, vol. 55, no. 2, pp. 90–92, 2019.

[22] I. Agadakos, N. Agadakos, J. Polakis, and M. R. Amer, "Deep complex networks for protocol-agnostic radio frequency device fingerprinting in the wild," Sep. 2019. [Online]. Available: arXiv:1909.08703.

[23] Q. Wu *et al.*, "Deep learning based RF fingerprinting for device identification and wireless security," *Electron. Lett.*, vol. 54, no. 24, pp. 1405–1407, Nov. 2018.

[24] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018.

[25] S. S. Hanna and D. Cabric, "Deep learning based transmitter identification using power amplifier nonlinearity," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Feb. 2019, pp. 674–680.

[26] K. Youssef, L.-S. Bouchard, K. Z. Haigh, H. Krovi, J. Silovsky, and C. P. V. Valk, "Machine learning approach to RF transmitter identification," Nov. 2017. [Online]. Available: arXiv:1711.01559.

[27] A. Gritsenko, Z. Wang, T. Jian, J. Dy, K. Chowdhury, and S. Ioannidis, "Finding a 'new' needle in the haystack: Unseen radio detection in large populations using deep learning," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Nov. 2019, pp. 1–10.

[28] S. Hanna, S. Karunaratne, and D. Cabric, "Deep learning approaches for open set wireless transmitter authorization," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC) (IEEE SPAWC)*, May 2020, pp. 1–5.

[29] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," Jul. 2019. [Online]. Available: arXiv:1811.08581.

[30] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," Jan. 2019. [Online]. Available: arXiv:1901.03407.

[31] N. Xie and S. Zhang, "Blind authentication at the physical layer under time-varying fading channels," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 7, pp. 1465–1479, Jul. 2018.

[32] Z. Gu, H. Chen, P. Xu, Y. Li, and B. Vucetic, "Physical layer authentication for non-coherent massive SIMO-based industrial IoT communications," Jan. 2020. [Online]. Available: arXiv:2001.07315.

[33] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 94–104, 1st Quart., 2016.

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

[35] J. Yu, A. Hu, and L. Peng, "Blind DCTF-based estimation of carrier frequency offset for RF fingerprint extraction," in *Proc. 8th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2016, pp. 1–6.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Dec. 2015. [Online]. Available: arXiv:1512.03385.

[37] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over the air deep learning based radio signal classification," Dec. 2017. [Online]. Available: arXiv:1712.04578.

[38] S. Hanna, C. Dick, and D. Cabric, "Combining deep learning and linear processing for modulation classification and symbol decoding," Jun. 2020. [Online]. Available: arXiv:2006.00729.

[39] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, "Unsupervised wireless spectrum anomaly detection with interpretable features," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 637–647, Sep. 2019.

[40] A. Toma *et al.*, "AI-based abnormality detection at the PHY-layer of cognitive radio by learning generative models," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 21–34, Mar. 2020.

[41] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, “Deep CM-CNN for spectrum sensing in cognitive radio,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2306–2321, Oct. 2019.

[42] C. Liu, Z. Wei, D. W. K. Ng, J. Yuan, and Y.-C. Liang, “Deep transfer learning for signal detection in ambient backscatter communications,” Sep. 2020. [Online]. Available: arXiv:2009.05231.

[43] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, “ORACLE: Optimized radio classification through convolutional neural networks,” Dec. 2018. [Online]. Available: arXiv:1812.01124.

[44] K. Sankhe *et al.*, “No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 165–178, Mar. 2020.

[45] F. Restuccia *et al.*, “DeepRadioID: Real-time channel-resilient optimization of deep learning-based radio fingerprinting algorithms,” Apr. 2019. [Online]. Available: arXiv:1904.07623.

[46] G. Baldini, R. Giuliani, and F. Dimc, “Physical layer authentication of Internet of Things wireless devices using convolutional neural networks and recurrence plots,” *Internet Technol. Lett.*, vol. 2, no. 2, p. e81, 2019.

[47] Y. Pan, S. Yang, H. Peng, T. Li, and W. Wang, “Specific emitter identification based on deep residual networks,” *IEEE Access*, vol. 7, pp. 54425–54434, 2019.

[48] L. Ding, S. Wang, F. Wang, and W. Zhang, “Specific emitter identification via convolutional neural networks,” *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2591–2594, Dec. 2018.

[49] H. Jafari, O. Omotere, D. Adesina, H. Wu, and L. Qian, “IoT devices fingerprinting using deep learning,” in *Proc. MILCOM IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2018, pp. 1–9.

[50] C. Morin, L. Cardoso, J. Hoydis, J.-M. Gorce, and T. Vial, “Transmitter classification with supervised deep learning,” May 2019. [Online]. Available: arXiv:1905.07923.

[51] J. Yu *et al.*, “Radio frequency fingerprint identification based on denoising autoencoders,” Jul. 2019. [Online]. Available: arXiv:1907.08809.

[52] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, “RFAL: Adversarial learning for RF transmitter identification and classification,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 783–801, Jun. 2020.

[53] L. J. Wong, W. C. Headley, and A. J. Michaels, “Emitter identification using CNN IQ imbalance estimators,” Aug. 2018. [Online]. Available: arXiv:1808.02369.

[54] D. Raychaudhuri *et al.*, “Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols,” in *Proc. Wireless Commun. Netw. Conf.*, vol. 3, 2005, pp. 1664–1669.

[55] A. Bendale and T. Boult, “Towards open set deep networks,” Nov. 2015. [Online]. Available: arXiv:1511.06233.

[56] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[57] L. Shu, H. Xu, and B. Liu, “DOC: Deep open classification of text documents,” Sep. 2017. [Online]. Available: arXiv:1709.08716.

[58] S. Kotz and S. Nadarajah, *Extreme Value Distributions: Theory and Applications*. Singapore: World Sci., 2000.

[59] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, “Classification-reconstruction learning for open-set recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4011–4020.

[60] M. Cekic, S. Gopalakrishnan, and U. Madhow, “Robust wireless fingerprinting: Generalizing across space and time,” Feb. 2020. [Online]. Available: arXiv:2002.10791.

[61] A. Al-Shawabka *et al.*, “Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2020, p. 10.

[62] Y. Sun, A. K. C. Wong, and M. S. Kamel, “Classification of imbalanced data: A review,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, Jun. 2009.

Samer Hanna (Student Member, IEEE) received the B.Sc. degree in electrical engineering and the M.Sc. degree in engineering mathematics from Alexandria University, Alexandria, Egypt, in 2013 and 2017, respectively. He is currently pursuing the Ph.D. degree with the University of California at Los Angeles, Los Angeles, CA, USA. His research interests include the applications of machine learning in wireless communications and coordinated communications using unmanned aerial vehicles.

Samurdhi Karunaratne (Student Member, IEEE) received the B.Sc. degree in computer engineering from the University of Peradeniya, Sri Lanka, in 2019. He is currently pursuing the M.S./Ph.D. degree with the Cognitive Reconfigurable Embedded Systems Lab, University of California at Los Angeles. His research interests include optimization algorithms and the applications of machine-learning techniques in wireless communications.

Danijela Cabric (Fellow, IEEE) received the M.S. degree in electrical engineering from the University of California at Los Angeles (UCLA) in 2001, and the Ph.D. degree in electrical engineering University of California at Berkeley in 2007. She is a Professor of Electrical and Computer Engineering with UCLA. Her research interests are millimeter-wave communications, distributed communications and sensing for Internet of Things, and machine learning for wireless networks co-existence and security. She received the Samueli Fellowship in 2008, the Okawa Foundation Research Grant in 2009, the Hellman Fellowship in 2012, the National Science Foundation Faculty Early Career Development (CAREER) Award in 2012, and the Qualcomm Faculty Award in 2020. She served as an Associate Editor for IEEE Transactions on Cognitive Communications and Networking, IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing, and *IEEE Signal Processing Magazine*, and an IEEE ComSoc Distinguished Lecturer.