

Open Set RF Fingerprinting using Generative Outlier Augmentation

Samurdhi Karunaratne, Samer Hanna, and Danijela Cabric
*Electrical and Computer Engineering Department,
University of California, Los Angeles*
samurdhi@ucla.edu, samerhanna@ucla.edu, danijela@ee.ucla.edu

Abstract—RF devices can be identified by unique imperfections embedded in the signals they transmit called RF fingerprints. The closed set classification of such devices, where the identification must be made among an authorized set of transmitters, has been well explored. However, the much more difficult problem of open set classification, where the classifier needs to reject unauthorized transmitters while recognizing authorized transmitters, has only been recently visited. So far, efforts at open set classification have largely relied on the utilization of signal samples captured from a known set of unauthorized transmitters to aid the classifier learn unauthorized transmitter fingerprints. Since acquiring new transmitters to use as known transmitters is highly expensive, we propose to use generative deep learning methods to emulate unauthorized signal samples for the augmentation of training datasets. We develop two different data augmentation techniques, one that exploits a limited number of known unauthorized transmitters and the other that does not require any unauthorized transmitters. Experiments conducted on a dataset captured from a WiFi testbed indicate that data augmentation allows for significant increases in open set classification accuracy, especially when the authorized set is small.

Index Terms—Transmitter Identification, Deep Learning, Open set recognition, authorization, physical layer authentication

I. INTRODUCTION

With billions of new Internet of Things (IoT) devices added each year, the task of securing IoT networks has become more important by the day. However, traditional cryptography based authentication systems are typically not suited for authenticating such devices due to their limited power and computational constraints. To address this, passive Physical Layer Authentication (PLA) has been proposed since it requires little to no work on the part of the transmitter [1]. Here, the authenticator uses channel state information and fingerprints due to hardware impairments to identify transmitters. Recently, research on passive PLA that uses deep learning techniques has been gaining momentum. Since deep learning based classifiers tend to extract more salient features, these approaches have been shown to outperform others which use handcrafted features, reaching markedly higher accuracy [2].

Most such authentication systems have been modeled as closed set classifications, where the classification decision is one of a known set of authorized transmitters. However, it is

obvious that such a model is ill-suited for an authentication system which needs to discriminate among the authorized set while rejecting unauthorized transmitters—since any unauthorized transmitter not included in training would be misclassified as authorized. Therefore, open set classification, which can detect unseen transmitters, has recently been proposed in this regard [3] [4].

There are two primary methods in which openset classification problems can be solved using deep learning [5]. The first one is discriminative model-based methods. Here, authorized data (signal samples from authorized transmitters) and known unauthorized data (signal samples from some known unauthorized transmitters) may be used to train a specialized neural network architecture. Subsequently an accompanying detection method allows for the detection of outliers while detecting authorized classes. To this end, we have previously tested architectures like One vs All (OvA) and methods like OpenMax and demonstrated that, for example, using a higher number of known unauthorized transmitters during training leads to better performance [4]. However, in this paper, our focus is on the second approach to open set classification: instance-generation based methods [5]. As the name implies, here generative models are used to synthetically create samples from the unknown unauthorized data that could be used to augment the training dataset. Then discriminative model-based open set methods or even closed set methods could be used on the augmented training data.

Data augmentation within the umbrella of RF fingerprinting is not an entirely new concept—for example, in [6] data with many simulated channel and noise variations were augmented to train a more channel-resilient RF fingerprinting classifier. However, to the best of our knowledge, instance-generation based data augmentation has not been applied in the literature regarding RF authentication. Yet generative models and methods have been proposed for data augmentation in other domains [5]. Inspired by this past work, we adapt some of these methods to RF fingerprinting while presenting our own methods. Our contributions include two types of outlier generation based on the type of information available during training:

- 1) Supervised outlier generation: Some signal samples from a limited number of known unauthorized transmitters are available during training.

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.

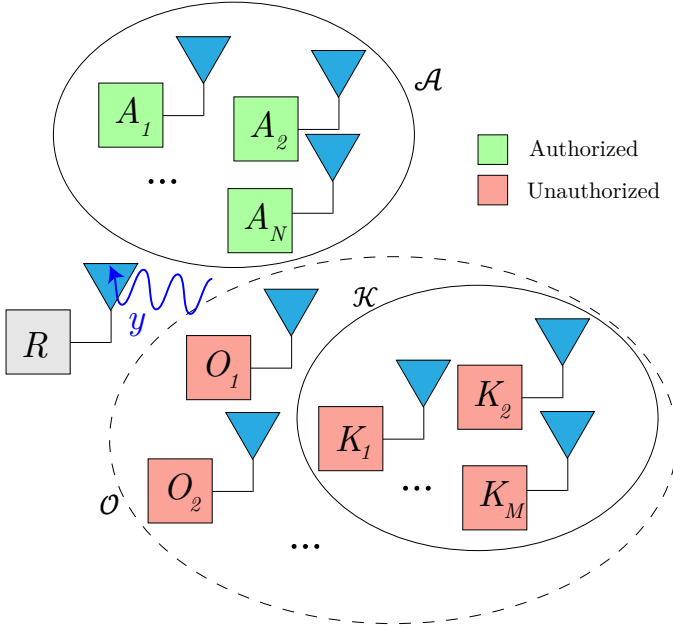


Fig. 1: System model: R must determine whether the received signal y originated from an unauthorized transmitter in \mathcal{O} , and if not, from which one of the authorized transmitters in \mathcal{A} . Some unauthorized transmitters \mathcal{K} may be known to R .

- 2) “Blind” outlier generation: Only authorized signal samples are available during training.

In both cases, the goal is to emulate signal samples from the set of unauthorized transmitters.

The rest of the paper is organized as follows: we start by formulating the problem in Section II. Section III discusses the dataset and evaluation method used for approaches presented throughout the paper. Supervised outlier generation is presented in Section IV while Blind outlier generation is presented in Section V. Section VI concludes the paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a finite set of authorized transmitters given by $\mathcal{A} = \{A_1, A_2, \dots, A_{|\mathcal{A}|}\}$ that are authorized to send data to a receiver R , where $|\mathcal{A}|$ is the size of the set \mathcal{A} . When a transmitter T sends a set of symbols x , the signal received is $f_T(x)$. The function f_T models the transmitter fingerprint determined by the variability of its circuits and also includes the effects of the channel. The authentication problem can be formulated as shown in Fig. 1: receiver R receives a signal y from some transmitter T and should determine whether the transmitter T belongs to the authorized set or not, based on y . This can be formulated as the following hypothesis testing:

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

However, in this paper, we are additionally interested in classifying the transmitter within the authorized set, in cases

where each authorized transmitter has different privileges. In such a setting, the open set authorization problem could be written as

$$\begin{aligned} \mathcal{H}_0 : y = f_T(x), T = A_1 \\ \mathcal{H}_1 : y = f_T(x), T = A_2 \\ \dots \\ \mathcal{H}_{|\mathcal{A}|-1} : y = f_T(x), T = A_{|\mathcal{A}|} \\ \mathcal{H}_{|\mathcal{A}|} : y = f_T(x), T \notin \mathcal{A} \end{aligned} \quad (2)$$

To improve the outlier detection, an additional set of known outliers $\mathcal{K} = \{K_1, K_2, \dots, K_{|\mathcal{K}|}\}$, where $\mathcal{K} \not\subset \mathcal{A}$, may be used. So typically during training, a captured set of signal samples $\mathcal{X} = \{X_1, X_2, \dots, X_{|\mathcal{A}|}\}$ corresponding to transmitters in \mathcal{A} and a similar set of signal samples $\mathcal{Y} = \{Y_1, Y_2, \dots, Y_{|\mathcal{K}|}\}$ corresponding to transmitters in \mathcal{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 \mathcal{K} can be obtained by capturing data from a finite number of non-authorized transmitters.

During supervised outlier generation, the task is to use the information in \mathcal{Y} (and possibly \mathcal{X}) to generate more signal samples from \mathcal{O} , which is the theoretically infinite set of unauthorized transmitters ($\mathcal{K} \subset \mathcal{O}$). The task in blind outlier generation is to use only the information encoded in \mathcal{X} to generate more signal samples from \mathcal{O} . The set of such samples generated from \mathcal{O} , using either method, is denoted by $\hat{\mathcal{Y}}$.

III. DATASET AND EVALUATION METHOD

We introduce the dataset and outlier detector classifier architecture for evaluating the proposed approaches.

A. Dataset

The dataset was captured on the Orbit testbed [7]—each one of the 71 transmitters was one 3 off-the-shelf WiFi modules (Atheros 5212, 9220, and 9280) and the receiver was a software defined radio (USRP N210).

Each transmitter was allowed to transmit over Channel 11, which has a center frequency of 2462 MHz and a bandwidth of 20 MHz, and captures were taken at a rate of 25 Msps for one second. After the IQ capture was complete, energy detection was used to detect the start and end of packets and extract them. Due to WiFi rate control, the number of packets obtained from each transmitter varied between 200 and 1500 packets with a mean of 800 packets. From each packet, we used the first 256 IQ samples, containing the preamble, without any synchronization or further preprocessing, as the signal sample.

B. Outlier detector architecture and evaluation

In [8], we explored several neural network architectures that could be used for a problem like Eq. 2 such as DClass and OvA. In this paper, for our evaluations, we chose to use the OvA (One-versus-All) architecture, most importantly because it could be used even when there are no known outliers during training. Simply put, the OvA architecture consists of a feature extractor followed by $|\mathcal{A}|$ binary classifiers in parallel, where the i -th binary classifier decides (0 or 1) whether the input

signal is from A_i or not. The feature extractor architecture used for all experiments conducted in this paper is given in Fig. 3.

\mathcal{A} , \mathcal{K} and \mathcal{O} will be chosen randomly, subject to the constraints specified for each evaluation—however, when comparing a non-augmented dataset vs. the corresponding augmented dataset, the same \mathcal{A} , \mathcal{K} and \mathcal{O} will be kept. For chosen \mathcal{A} , \mathcal{K} and \mathcal{O} , the dataset split will be as follows: for training and validation, we use 70% of the samples belonging to \mathcal{A} , and all the samples belonging \mathcal{K} . The shuffled combination of this data is split into 80% for training and 20% for validation. The test set contains all samples from \mathcal{O} and the remaining 30% of \mathcal{A} . The evaluation metric is the prediction accuracy of the OvA classifier on this test set.

IV. SUPERVISED OUTLIER GENERATION

The idea behind supervised outlier generation is simple: we could train a generative model on signal samples collected from a few known outliers (\mathcal{Y}), and use that generative model to generate more outlier samples. We chose Variational autoencoders (VAEs) [9] as our preferred generative model due to their relative ease of training. A VAE is an unsupervised neural network architecture that consists of an encoder network E which transforms the input data into a latent space and a decoder network D which maps the encoded data back into the input space. By learning parameters related to the underlying data distribution, for any standard normal z , it is able to take $D(z)$ as a sample from that distribution.

The most straightforward approach is to train a VAE on all the data available in \mathcal{Y} . Intuitively, if we then isolate the decoder D , it should be possible to emulate as many known outlier samples as we want. To test this approach, we first created a VAE with the architecture highlighted in Fig. 2 and performed the following experiment: we pick a fixed set of 10 authorized transmitters \mathcal{A} . Then on 5 different occasions, we pick \mathcal{K} randomly from \mathcal{O} such that $|\mathcal{K}| \in \{5, 10, 15, 20, 25\}$. These values for $|\mathcal{A}|$ and $|\mathcal{K}|$ were picked since we seek to evaluate the effect of $|\mathcal{K}|$ on outlier detection performance as $|\mathcal{A}|$ is kept constant—any value for $|\mathcal{A}|$ and any set of values for $|\mathcal{K}|$ not too close together should give similar observations for what follows. For each such \mathcal{K} , a VAE was trained on the corresponding \mathcal{Y} and its decoder was then used to generate more samples from \mathcal{K} . We then train an OvA classifier with the individually labelled \mathcal{X} as authorized and \mathcal{Y} and $\hat{\mathcal{Y}}$ as the outliers. In our experiments it was interesting to note that while the testing accuracy of the trained OvA as $|\hat{\mathcal{Y}}|$ is increased rises initially, beyond a certain number of generated samples, there is no appreciable increase. This makes sense because the finite set \mathcal{K} and the finite number of captured signal samples \mathcal{Y} only expose a certain amount of information about \mathcal{O} that could be exploited by generating more samples from \mathcal{K} . Therefore we fixed $|\hat{\mathcal{Y}}| = 7500$ and evaluated the OvA testing accuracy when the training data is augmented with the generated outliers and compared it against the non-augmented OvA performance. From Fig. 4a it is clear that when $|\mathcal{K}|$ is increased, the effect of augmentation is greater. While many reasons are possible,

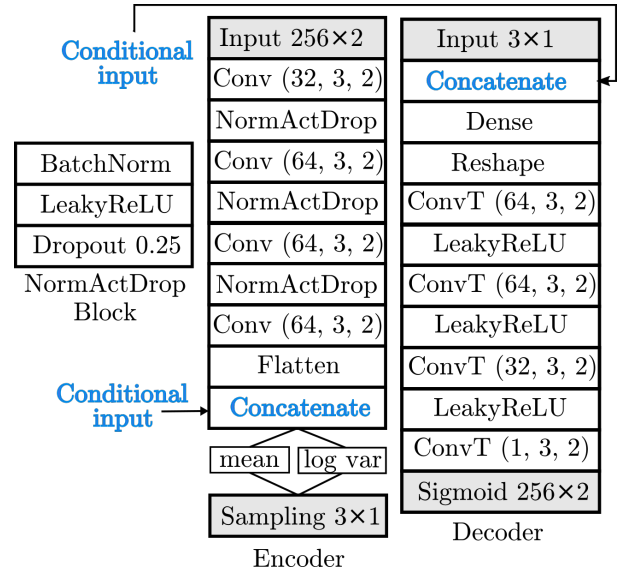


Fig. 2: Neural network architectures used for the standard VAE and the conditional VAE. The components in blue are only present in the conditional VAE. Conv (x, y, z) is a 2D Convolutional layer with x filters of kernel size y and stride ($z, 1$). ConvT represents a transposed version of Conv.

it is immediately not clear why this trend is observed—so we experiment further.

Note that when trying to generate samples from \mathcal{K} by training a standard VAE on \mathcal{Y} as above, we do not know from which transmitters in \mathcal{K} these generated signals samples are drawn from. For example, if 1000 samples are generated and $|\mathcal{K}| = 5$, 800 of those could be from K_1 and 200 could be from K_3 , or all 1000 could be from K_4 . To avoid this disproportionality during the sample generation, we tested another type of VAE called a conditional VAE or CVAE. The CVAE is simply an extension on the standard VAE in which there is a conditional or categorical input to the decoder indicating the class of the sample to be generated, as is clear from Fig. 2. This allows us to specify the particular transmitter from which samples are to be generated after the CVAE is trained. Similar to the VAE, for each $|\mathcal{K}| \in \{5, 10, 15, 20, 25\}$ we generated 7500 samples but this time, $7500/|\mathcal{K}|$ samples were generated from each known outlier in \mathcal{K} . Then the same OvA classifier as for the standard VAE was trained—the testing accuracy observed for different $|\mathcal{K}|$ is given in Fig. 4b. We can see that now roughly the same level of accuracy increase is seen for all $|\mathcal{K}|$ —when $|\mathcal{K}|$ is large, the probability that generated samples will be picked from a small subset of is small, while that same probability is large when $|\mathcal{K}|$ is small. So using a CVAE has an equalizing effect by allowing generated samples to be proportionately chosen. This is why we see the augmentation accuracy gap increase with $|\mathcal{K}|$ for the standard VAE while roughly staying the same for the CVAE. Note that the non-augmented curves in Fig. 4a and Fig. 4b are different due to the randomness in which \mathcal{K} is selected.

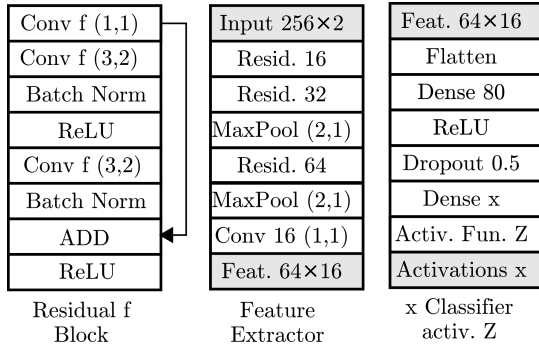
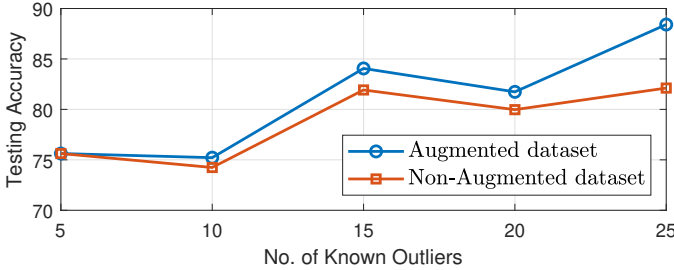
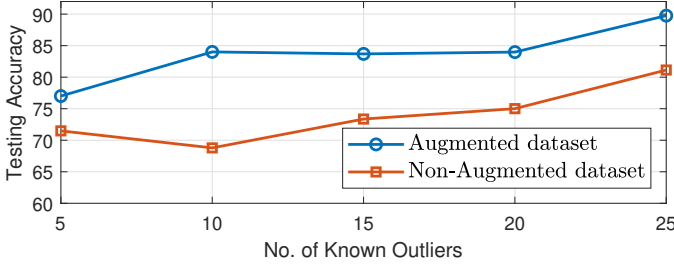


Fig. 3: Detailed architecture of the feature extractor (made of residual blocks with f filters), and a classifier with x outputs.



(a) Variation of testing accuracy with $|\mathcal{K}|$ for VAE



(b) Variation of testing accuracy with $|\mathcal{K}|$ for CVAE

Fig. 4: Variation of testing accuracy with $|\mathcal{K}|$ for VAE and CVAE.

V. BLIND OUTLIER GENERATION

While supervised outlier generation is somewhat straightforward, it requires the need to acquire additional hardware in the form of known unauthorized transmitters. This is an expensive task that becomes infeasible in many situations. So a more important, and more challenging task is to use just the information available in \mathcal{X} to emulate signal samples from \mathcal{O} : blind outlier generation.

A. Ellipsoidal method

Our approach to this task is rooted in visualizing the authorized set in some latent space, which is essentially a dimensionality-reduced space. Any signal $x \in X$ is typically a high-dimensional vector ($256 \times 2 = 512$ -dimensional in the dataset above), so this dimensionality reduction allows for correlated features to be merged and the dimensionality of the vector space to be made tractable. Here, we will use an *autoencoder* to map the authorized signal samples to a latent

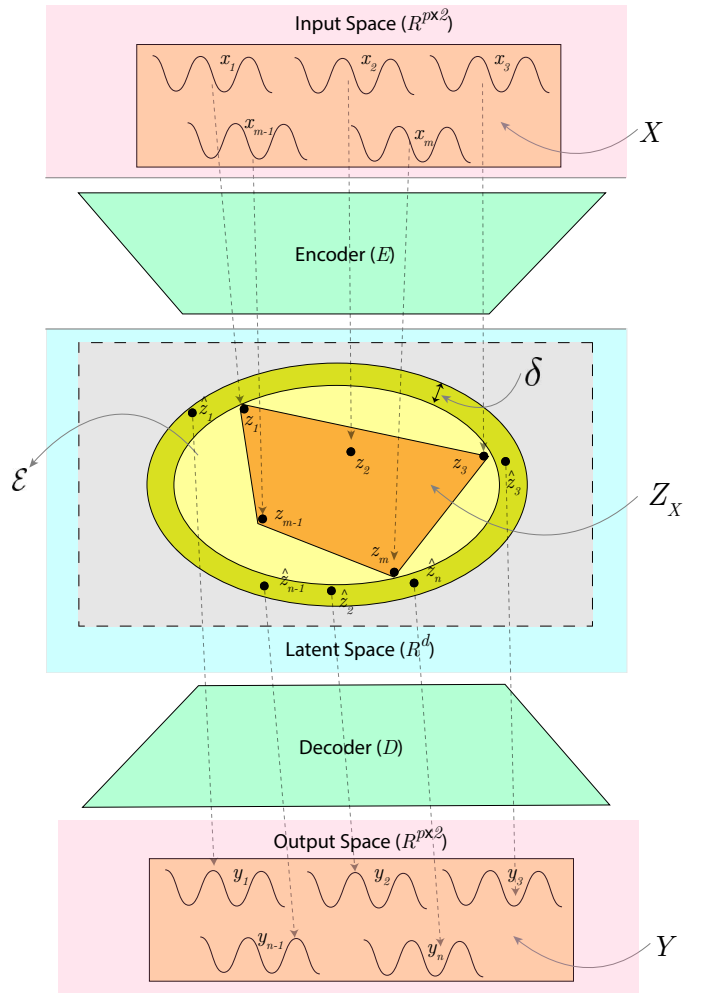


Fig. 5: Ellipsoidal method: Using an encoder to map authorized samples x_i to the latent space z_i . Random points sampled in the shell are mapped using the decoder to recover generated outlier samples y_j

space (as will become evident, a VAE is unnecessary as we do not require any sampling in this case). Since the size of the encoder output is usually designed to be much smaller than the size of the input, the encoder essentially extracts a latent space representation of the input.

This process is visually depicted by Fig. 5 — we start off with some authorized signal samples $X = \{x_1, x_2, \dots, x_m\}$ that we map to their corresponding latent space representations $Z_X = \{z_1, z_2, \dots, z_m\}$ using the encoder, i.e. $z_i = E(x_i)$. Note that the region in latent space occupied by Z_X could be some arbitrary shape. The key insight behind our approach to blind outlier generation is to conjecture that in latent space: i) outliers are still wireless signals so they should be “close” to authorized signal samples within the latent space; and ii) outliers are clearly not authorized signals so they should be clearly separated from the region of space occupied by the authorized signals.

Now assume we were able to find the minimum volume

ellipsoid \mathcal{E} enclosing Z_X . Then, outlier signals should be located in some region of the latent space that is separate from but not too far away from the region occupied by the authorized signals. This suggests that if we take some shell of uniform thickness δ around \mathcal{E} , points within this region should correspond to outlier signals.

To generate outlier samples in the signal space, we could simply generate random points within the shell and use the decoder D to map them into their signal space counterparts. This process is visualized in Fig. 5.

To practically apply this method, we first need a method to find the minimum volume ellipsoid \mathcal{E} . We can write

$$\mathcal{E} = \{z \in Z_X \mid \|Az + b\|_2 \leq 1\} \quad (3)$$

i.e. \mathcal{E} is the inverse image of the Euclidean unit ball under an affine mapping, and the variables are $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ where A could be assumed to be positive definite without loss of generality.

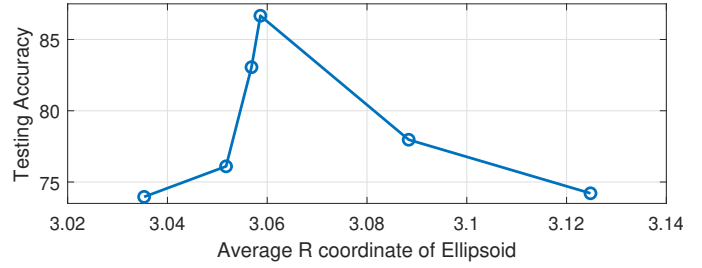
Since the volume of \mathcal{E} is proportional to $\det A^{-1}$, this can be formulated as the following convex optimization problem, which can be solved numerically

$$\begin{aligned} & \text{minimize}_{A \in S^n, b \in \mathbb{R}^n} \quad \log \det A^{-1} \\ & \text{subject to} \quad \|Az_i + b\|_2 \leq 1 \quad i = 1, \dots, m \end{aligned} \quad (4)$$

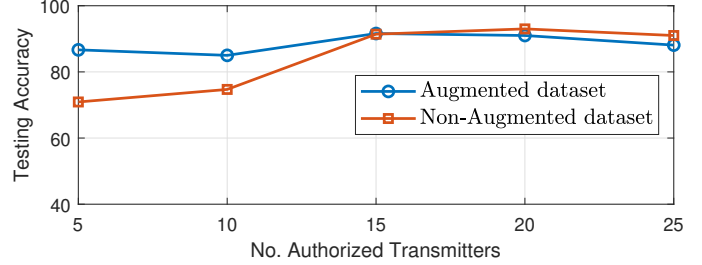
To evaluate this method, we performed the following experiment. 30 transmitters were first selected from the 71-transmitters in the dataset as the set of outliers ($|\mathcal{O}| = 30$). Then for $|\mathcal{A}| \in \{5, 10, 15, 20, 25\}$, a set of $|\mathcal{A}|$ transmitters were randomly selected as the authorized set \mathcal{A} in 5 separate occasions. Note that similar to the VAE methods in Section IV, even though it is technically possible to use this method to generate an infinite number of outliers, practically we see almost no improvement in performance after a certain number of outlier samples are generated. So as we did in Section IV, $|\hat{\mathcal{Y}}| = 7500$ outlier samples were generated and subsequently used to augment the training dataset and train an OvA classifier. The testing accuracy corresponding to outlier samples that were generated for different values of δ for $|\mathcal{A}| = 5$ is given in Fig. 6a. We can clearly see that there is an ideal value for the shell thickness δ , that is not too small and not too large, when the generated samples represent the best outliers. To generate Fig. 6b, the same δ that was ideal for $|\mathcal{A}| = 5$ was used for all $|\mathcal{A}|$ when generating $\hat{\mathcal{Y}}$. In Fig. 6b, we can see that especially when $|\mathcal{A}|$ is low, significant gains (>15%) in testing accuracy could be obtained by augmenting the dataset.

B. Optimization method

The ellipsoidal method provided a visually intuitive method for outlier generation. However, it required the empirical tuning of a parameter δ . So we try to approach the same problem with a similar intuition about outlier signals but now formulate the constraints of the problem differently. As we had previously remarked, the outlier signals should correspond to wireless signals. So if $D(z)$ is an outlier, the encoding $E(x)$



(a) Sweet spot for δ : Variation of testing accuracy as thickness of the ellipsoid shell is tuned for $|\mathcal{A}| = 5$



(b) Variation of testing accuracy against no. of authorized transmitters

Fig. 6: The results of using the ellipsoidal method to generate outliers

of an authorized signal x should be close to z in latent space. For $D(z)$ to be an outlier, z should satisfy,

$$\begin{aligned} & \text{minimize}_z \quad \|E(x) - z\| \\ & \text{subject to} \quad D(z) \notin \mathcal{A} \end{aligned} \quad (5)$$

Here, $\|\cdot\|$ could be some norm measure like the Euclidean distance. However the constraint $D(z) \notin \mathcal{A}$ is less straightforward. To solve this problem and find some optimal z for some input x , we need to integrate this constraint numerically. Authors in [10] suggested using a closed set classifier $C_{\mathcal{A}}$ with $|\mathcal{A}|$ outputs, which has learned to differentiate the known classes, as a judge of whether $D(z)$ is an outlier or not. They argued that if $D(z)$ is an outlier, the logits of the classifier prediction $C_{\mathcal{A}}(D(z))_i$, should be low for all classes i . Inspired by this, we suggest using a closed set classifier $C_{\mathcal{A}+1}$ with $|\mathcal{A}| + 1$ outputs as the judge of whether $D(z)$ is an outlier instead. With this definition, an ideal outlier would result in a softmax output of $[0 \ 0 \ \dots \ 0 \ 1]$ meaning that we can formulate the constraint in terms of a loss function and integrate in to the objective function:

$$z^* = \min_z \|E(x) - z\| + \text{loss} \left([0 \ \dots \ 0 \ 1], C_{\mathcal{A}+1}(D(z)) \right) \quad (6)$$

Using an $|\mathcal{A}| + 1$ output classifier as the judge instead of an $|\mathcal{A}|$ output classifier has one added benefit—once we have found some set of initial outliers $\mathcal{O}^{(0)}$, we can re-train the initial judge $C_{\mathcal{A}+1}^{(0)}$ using those outliers as unauthorized samples to get a better judge $C_{\mathcal{A}+1}^{(1)}$ that could be used in a subsequent iteration to generate even better outliers. Note that during its initial training $C_{\mathcal{A}+1}$ does not have training samples for the $|\mathcal{A}| + 1$ -th class, so this process should help it become a

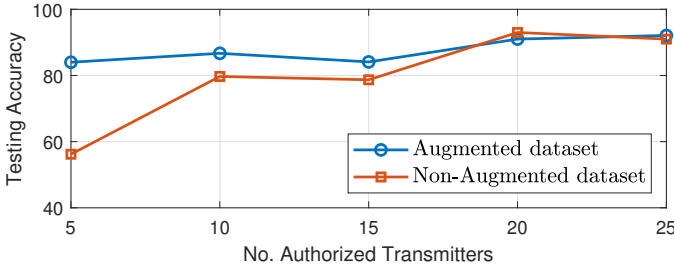


Fig. 7: Variation of testing accuracy against no. of authorized transmitters for the optimization method

better judge as iterations progress. However, due to the self-supervised nature of this procedure, it could be sensitive to the quality of initial training of $C_{\mathcal{A}+1}^{(0)}$. This method is highlighted in Algorithm 1 in detail.

Algorithm 1: Blind outlier generation

Input : Set of authorized signals X ;
Output : Samples from the set of outliers \mathcal{O} ;

- 1 Train encoder network $E(x)$, decoder network $D(z)$ and $|\mathcal{A}| + 1$ closed set classifier $C_{\mathcal{A}+1}^{(0)}$;
- 2 **for** $i \in \{0, 1, \dots, N-1\}$ **do**
- 3 **initialize** $\mathcal{O}^{(i)} = \emptyset$;
- 4 **for** $x \in X$ **do**
- 5 $z^* = \min_z \|z - E(x)\| +$
 $\text{loss}([0 \ 0 \ \dots \ 0 \ 1], C_{\mathcal{A}+1}^{(i)}(D(z)))$;
- 6 $\mathcal{O}^{(i)} = \mathcal{O}^{(i)} \cup \{D(z^*)\}$
- 7 **end**
- 8 $C_{\mathcal{A}+1}^{(i+1)} = \text{train}(C_{\mathcal{A}+1}^{(i)}, \mathcal{X} \cup \mathcal{O}^{(i)})$;
- 9 **end**
- 10 $\mathcal{O} = \mathcal{O}^{(N-1)}$;

To evaluate this approach, we follow a similar approach to the ellipsoidal method. The architectures for the E and D networks are the same as that in Fig. 2, while cross-entropy was used as the loss function in Eq. 6. The results in Fig. 7 are almost identical to that of the ellipsoidal method in Fig. 6b, except that we are able to see an improvement in testing accuracy even at $|\mathcal{A}| = 15$, which was not seen with the ellipsoidal method. The crucial benefit is that there is no tuning of an extra hyperparameter (δ) necessary in this case. However, the ellipsoidal method is still a valid method to consider if proper tuning of δ can be done, as solving Eq. 6 through gradient descent is computationally expensive. For example, generating 10^3 outlier samples with the optimization method took $30\times$ the time it took for the ellipsoidal method. Note that the non-augmented curves in Fig. 6b and Fig. 7 are slightly different due to the randomness in which \mathcal{A} is selected.

One reason for the degradation in the augmentation accuracy gap in Fig. 7 as $|\mathcal{A}|$ increases is that the outlier detection accuracy of $C_{\mathcal{A}+1}$ decreases as $|\mathcal{A}|$ increases, due to there being no supervision for the outliers during the initial training of $C_{\mathcal{A}+1}$ (it is an $|\mathcal{A}| + 1$ -class classifier). So $C_{\mathcal{A}+1}$ becomes worse at judging the constraint $D(z) \notin \mathcal{A}$ in Eq. 5, affecting the quality of the outliers thereby produced.

VI. CONCLUSION

In this paper, we considered the instance-generation based method for the open set classification problem in RF fingerprinting. Outlier instances were generated in two primary settings according to the information used during training; supervised outlier generation, where signal samples from a limited number of known outliers are used to generate more outlier samples; and blind outlier generation where outlier samples must be generated by using only authorized samples available during training. For supervised outlier generation, we considered VAEs and CVAEs and showed that CVAEs allows for a more uniform performance improvement when $|\mathcal{K}|$ is varied by enabling the proportionate generation of outliers from each known outlier in \mathcal{K} . For blind outlier generation we first presented the ellipsoidal method, which was a geometrically intuitive way. Although it gave satisfactory results, it required the tuning of an extra hyperparameter. So another optimization-based method was presented that achieved the same-level of results but did not require any tuning. In both cases, significant performance improvements were seen for small $|\mathcal{A}|$. This provides evidence that generative outlier augmentation can replace the need to acquire transmitters as known outliers when building RF authentication systems.

REFERENCES

- [1] W. Wang, Z. Sun, S. Piao, B. Zhu, and K. Ren, "Wireless Physical-Layer Identification: Modeling and Validation," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 2091–2106, Sept. 2016.
- [2] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep Learning Convolutional Neural Networks for Radio Identification," *IEEE Communications Magazine*, vol. 56, pp. 146–152, Sept. 2018.
- [3] 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 *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, (Newark, NJ, USA), pp. 1–10, IEEE, Nov. 2019.
- [4] S. Hanna, S. Karunaratne, and D. Cabric, "Open Set Wireless Transmitter Authorization: Deep Learning Approaches and Dataset Considerations," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, pp. 59–72, Mar. 2021. Conference Name: IEEE Transactions on Cognitive Communications and Networking.
- [5] C. Geng, S.-j. Huang, and S. Chen, "Recent Advances in Open Set Recognition: A Survey," *arXiv:1811.08581 [cs, stat]*, July 2019. arXiv: 1811.08581.
- [6] N. Soltani, K. Sankhe, J. Dy, S. Ioannidis, and K. Chowdhury, "More Is Better: Data Augmentation for Channel-Resilient RF Fingerprinting," *IEEE Communications Magazine*, vol. 58, pp. 66–72, Oct. 2020. Conference Name: IEEE Communications Magazine.
- [7] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremono, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3, pp. 1664–1669, IEEE, 2005.
- [8] S. Hanna, S. Karunaratne, and D. Cabric, "Deep Learning Approaches for Open Set Wireless Transmitter Authorization," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, May 2020. ISSN: 1948-3252.
- [9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, May 2014. arXiv: 1312.6114.
- [10] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open Set Learning with Counterfactual Images," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), vol. 11210, pp. 620–635, Cham: Springer International Publishing, 2018. Series Title: Lecture Notes in Computer Science.