AFRL-RY-WP-TP-2022-0145

NOVELTY DETECTION FOR RF WAVEFORMS WITH ENSEMBLED CONTRASTIVE LEARNING

Edward Reehorst
RF Electronic Warfare Branch
Spectrum Warfare Division

APRIL 2022
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
April 2022	Dissertation	12 April 2022	12 April 2022

4. TITLE AND SUBTITLE

NOVELTY DETECTION FOR RF WAVEFORMS WITH ENSEMBLED CONTRASTIVE LEARNING

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
N/A	N/A	N/A
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
N/A	N/A	N/A

6. AUTHOR(S)

Edward Reehorst

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

EO/IR Components Branch
Aerospace Components & Subsystems Division
Air Force Research Laboratory, Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command, United States Air Forces

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'SREPORT NUMBER(S)
Air Force Research Laboratory, Sensors Directorate Wright-Patterson Air Force Base, OH 45433-7320 Air Force Materiel Command, United States Air Forces	AFRL/RYWE	AFRL-RY-WP-TP-2022-0145

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

PAO case number AFRL-2022-1700, Clearance Date 12 April 2022. This document is required to be shared with PhD candidate's dissertation committee in order to complete the PhD program. This is a work of the U.S. Government and is not subject to copyright protection in the United States. Report contains color.

14. ABSTRACT

We consider the problem of identifying whether an observed radio-frequency (RF) waveform belongs to a known class of waveforms or is unfamiliar. For example, this problem arises in spectrum monitoring of communication or radar signals, where there is a need to detect illegal or unknown transmitters operating in the range of the receiver. When the known waveforms are described by a dataset rather than a mathematical model, our problem is an instance of \novelty detection" from the eld of machine learning. In this dissertation proposal, we describe research on novelty detection for RF waveforms that leverages deep neural networks and recent advances from the eld of contrastive learning. One of our main contributions is a method for ensembling several detection scores, which we call shift-ensembled novelty detection (SEND). Our technique is shown to increase detection performance over existing novelty detectors on communication and radar datasets.

15. SUBJECT TERMS

crsytal growth, heterostructure(s), hydride vapor phase epitaxy, nonlinear optical materials, optical conversion

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
Unclassified	Unclassified	Unclassified	SAR	21

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Michael Wharton	

Novelty Detection for RF Waveforms with Ensembled Contrastive Learning

a Candidacy Exam Proposal by
Edward Reehorst

Abstract

We consider the problem of identifying whether an observed radio-frequency (RF) waveform belongs to a known class of waveforms or is unfamiliar. For example, this problem arises in spectrum monitoring of communication or radar signals, where there is a need to detect illegal or unknown transmitters operating in the range of the receiver. When the known waveforms are described by a dataset rather than a mathematical model, our problem is an instance of "novelty detection" from the field of machine learning. In this dissertation proposal, we describe research on novelty detection for RF waveforms that leverages deep neural networks and recent advances from the field of contrastive learning. One of our main contributions is a method for ensembling several detection scores, which we call shift-ensembled novelty detection (SEND). Our technique is shown to increase detection performance over existing novelty detectors on communication and radar datasets.

Publications

- E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. on Computational Imaging*, vol. 5, pp. 52–67, Mar. 2019
- R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, "Plug and play methods for magnetic resonance imaging," *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 105–116, 2020
- S. Liu, E. Reehorst, P. Schniter, and R. Ahmad, "Free-breathing cardiovascular MRI using a Plug-and-Play method with learned denoiser," in *Proc. IEEE Internat. Symp. on Biomedical Imaging*, Apr. 2020
- M. Wharton, E. T. Reehorst, and P. Schniter, "Compressive SAR image recovery and classification via CNNs," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, pp. 1081–1085, 2019

1 Introduction

Radio-frequency (RF) waveform identification is a difficult engineering problem with many open research areas standing between the current state-of-the-art and the deployment of real-world systems. These open areas of research include detecting dynamic signals, detecting signals in crowded spectrum, and detecting signal classes that were not included in the training set. In this work, we focus on the latter case, known as novelty detection.

If we have a simple mathematical model for the approved modulation types, it may be possible to design a closed-form detector. However, these techniques are not able to capture the complexity of modern radar and communication systems. For this reason, there has been significant research into utilizing data-driven approaches for waveform identification, such as deep neural networks (DNNs) [5, 6].

One possible application of novelty detection in RF waveforms would be to monitor spectrum for non-approved modulation types. In this application, we would have a list of approved types of modulations with any other type of modulation being illegal. It would be near-impossible to train a binary DNN classifier for this problem because we would need to construct a dataset with every non-approved type of modulation, which would be prohibitively large and ever changing as new modulation types are developed. In this example, we want to be able to train a DNN-based detector to flag illegal modulation types, but with a training set made-up exclusively of approved waveforms.

In this proposal, we will focus on a data-driven approach to novelty detection. We assume we have access to a training set $X = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i$ is the ith training signal and $y_i \in \{1 \dots K\}$ is the corresponding class label. We denote the number of known classes as $K \geq 1$. We will call a signal coming from the known class(es) an inlier and any other signal a novelty. Our goal is to learn a function from the training set that can distinguish between an inlier and a novelty. This problem is different from binary classification, because we do not have any training examples of novelties in the training set.

In our formulation of the problem we allow the number of known classes K to be any positive integer. If $K = 1$, this problem is called one-class classification [7]. In one-class classification we are only given training examples from a single class and must use that data to learn a function to distinguish between signals from that class and any novelties. When $K > 1$, we will call the problem multi-class novelty detection [8]. In this problem, the training data will be labeled and come from multiple classes. These problems are illustrated in Fig. 1.

Almost all novelty detection methods score samples with an "inlier score." The higher this score, the more confident the model is that the test sample is an inlier. In order to perform novelty detection they threshold this score. Any test sample with a score below the threshold is classified as a novelty, and anything with a score higher than the threshold is classified as an inlier. A significant contribution of our work is to develop a principled way to combine, or "ensemble", multiple scores. We find that our technique is compatible with several popular score functions, and that the ensembled score gives better performance than the individual scores themselves.

Recently, Tack et al. [9] introduced Contrastive Shifting Instances (CSI). CSI is a novelty detection technique, developed for image datasets, that utilizes a feature encoder trained on the self-supervised contrastive loss. CSI also utilizes "shifting transforms", which are transforms designed to semantically change a sample, which they use to augment training, and ensemble their technique over multiple realizations of the test sample. CSI will be discussed in more detail in the background section below, but for now it suffices to say that, first, CSI as originally propsed is not compatible with RF waveform data and, second, CSI uses multiple scores but combines them in a heuristic manner.

Figure 1: Example plots of (left) one-class classification and (right) multi-class novelty detection problems.

We propose a dissertation along these lines:

- Develop a self-supervised learning method for RF waveform data based on CSI.
- Develop SEND, an ensembled novelty detection strategy based on neural network encodings. This task has multiple parts:
  - Develop an inlier score ensembling technique based on detection theory.
  - Develop similarity and shifting transforms for waveform data.
  - Develop one-class and multi-class implementations of SEND.

2 Background

2.1 Low-Dimensional Novelty Detection

In low dimensions, novelty detection is essentially a solved problem. Classical machine learning methods such as Mahalanobis distance [10], nearest neighbor distance [11], kernel density estimation [12], one-class support vector machine (OC-SVM) [13], and support vector data description [14] are techniques that perform well in low-dimensional settings. Later, we will utilize the Mahalanobis distance, defined as

$$s_{Mah}(\boldsymbol{x}) = (\boldsymbol{x} - \widehat{\boldsymbol{\mu}})^T \widehat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}), \tag{1}$$

where $\widehat{\boldsymbol{\mu}}$ is the sample mean and $\widehat{\boldsymbol{\Sigma}}$ is the sample covariance matrix of the training set X. Mahalanobis distance can be interpreted as the negative log likelihood of a Gaussian distribution fit to the training data. We will also use the nearest-neighbor distance

$$s_{NN}(\boldsymbol{x}) = \min_{\boldsymbol{x}' \in X} d(\boldsymbol{x}, \boldsymbol{x}'), \tag{2}$$

where $d(\cdot, \cdot)$ is an arbitrary distance function such as ℓ_2 distance or cosine distance.

These methods assume a notion of similarity between data-points defined as some distance function, whether that is ℓ_2 distance, Mahalanobis distance, or some kernel function, such as the radial basis function. These assumptions work well in low dimensions but break down in high dimensions due to the curse of dimensionality.

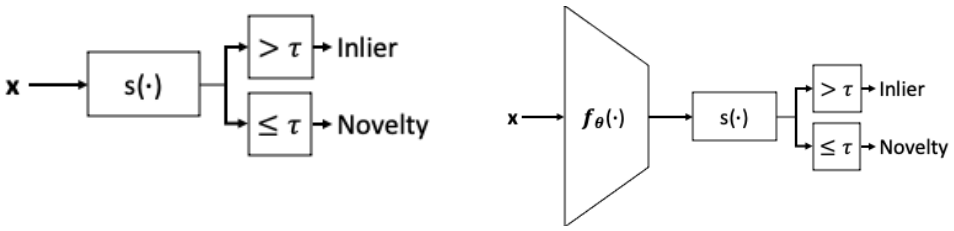

Figure 2: Diagrams of classic, low-dimensional (left) and deep, high-dimensional (right) novelty detection. Deep novelty detection utilizes a deep neural network to extract features before evaluating the score function $s(\cdot)$.

2.2 High-Dimensional Novelty Detection

Novelty detection in high-dimensional signals is an active area of research. Most novelty detection solutions for high-dimensional signals utilize a deep neural network to encode the signal into a lower-dimensional space, where low-dimensional novelty detection approaches can be used [15]. Novelty detection is then performed by thresholding an inlier score. We will call this feature encoder $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$, parameterized by $\boldsymbol{\theta}$, and the score function $s(\cdot)$. Fig. 2 shows the difference between classic, low-dimensional novelty detection and deep, high-dimensional novelty detection.

For the multi-class novelty detection problem, a common solution is to utilize a DNN classification network as the feature extractor $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$. A popular inlier score is the maximum of the softmax output of that DNN classifier. This technique is called softmax-thresholding [16]. It is popular due to its simplicity, but often performs poorly due to DNN classifiers being overconfident on samples far from the training set [17]. More sophisticated novelty detection techniques utilize the penultimate features of a classification DNN as $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$. Andrew et al. [18] introduce a technique where a OC-SVM is trained on the penultimate features of a DNN classifier to provide the inlier score. A similar technique is applied to radar waveform data in [19]. In Lee et al. [20], the penultimate layer of a DNN classifier is used as the features and scored by the class-dependent Mahalanobis distance

$$s_{Mah}(\boldsymbol{x}) = -\min_{j=\{1\ldots K\}} (\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_j)^T \widehat{\boldsymbol{\Sigma}}_j^{-1} (\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_j), \tag{3}$$

where $\widehat{\boldsymbol{\mu}}_j$ is the sample mean of $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ for class j and $\widehat{\boldsymbol{\Sigma}}_j$ is the sample covariance matrix for class j, both evaluated on the training set X.

Label smoothing [21] is a technique used to improve calibration and generalization of DNN classifiers. It is shown in [22] that label smoothing results in penultimate features being more compact. More recently, label-smoothing has been shown to assist with novelty detection performance [23]. During training, the target label distribution is smoothed by taking a weighted average of the true label, y, and a uniform distribution

$$\boldsymbol{p}^{LS}(y) = (1-\alpha)\boldsymbol{e}_y + \frac{\alpha}{K}, \tag{4}$$

where $\boldsymbol{e}_i$ is the unit vector with a 1 in the ith element and all other elements being zero. The label smoothing loss is the cross-entropy between this target distribution and the distribution defined by the output of the DNN classifier

$$\mathcal{L}_{LS}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} -p^{LS}(y_i)_k \log(\widehat{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)_k). \tag{5}$$

Note that if $\alpha = 0$, this loss simplifies to the standard cross-entropy loss.

These classifier-based methods for novelty detection are popular, but there are several downsides to using features from a classification network for novelty detection. The first is that we need class-labels to train the classifier. This means this technique cannot be used for one-class classification problems. The second downside is that a DNN classifier will learn features that are good for classification, but may learn to ignore features important to novelty detection [24]. Thus a better option for learning features is self-supervised learning.

2.3 Self-Supervised Learning

Self-supervised learning [25] is where DNN feature encoders are trained without using class-labels. The first way to do this is to train the DNN to perform an auxiliary task on the data. One example is to train the DNN to classify the rotation in images that have been rotated 0, 90, 180, or 270 degrees [26]. Since natural images usually have some notion of "up," it is possible to train a DNN to classify these rotations. In learning the ability to classify rotations, the DNN learns to encode semantically important feautres.

Another approach to self-supervised learning is contrastive learning [27]. In contrastive learning, DNNs are trained so that the encoding of semantically similar images are "close" in some metric, like cosine or ℓ_2 distance. This technique requires us to have some notion of similarity. For supervised problems, the notion of similarity could be if two samples come from the same class [28]. However, in many problems we do not have this notion of similarity built into the data. A recent implementation of contrastive learning, Simple framework for Contrastive Learning of visual Representations (SimCLR) [29], solves this problem.

In SimCLR, the authors take any batch of samples as dissimilar to one another and use random augmentations of the same sample to create pairs of similar samples. We will refer to these random augmentations as "similarity" transforms, because they produce samples that are semantically similar to the input. We will denote similarity transforms with $T_j(\cdot)$ where the index j captures different possible random augmentations. When working with images, the similarity transforms would be some combination of random crops, gray-scaling, color jitter, etc. We will develop similarity transformations for use with RF waveform data below. SimCLR utilizes the normalized temperature cross-entropy (NT-Xent) loss [30]. Given a sample $\boldsymbol{x}$, a sample, $\boldsymbol{x}_+$, that is "similar" to $\boldsymbol{x}$, and a batch of images, X_-, dissimilar to $\boldsymbol{x}$, the NT-Xent loss is given by

$$\mathcal{L}_{NT-Xent}(\boldsymbol{x}, \boldsymbol{x}_+, X_-) = -\log \frac{\exp(\text{sim}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_+))/\tau)}{\sum_{\boldsymbol{x}_- \in X_-} \exp(\text{sim}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_-)/\tau) + \exp(\text{sim}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_+))/\tau)} \tag{6}$$

where $\text{sim}(\cdot, \cdot)$ is some similarity metric, such as cosine similarity and $\tau > 0$ is a tunable normalizing temperature. This loss rewards features that encode similar samples to be close in terms of $\text{sim}(\cdot, \cdot)$, while punishing dissimilar samples from being close in terms of $\text{sim}(\cdot, \cdot)$.

The next contribution of SimCLR is the addition of a "head" network that takes the output of the encoder $\boldsymbol{f}_{\boldsymbol{\theta}}$ as its input. The head network is made up of two fully-connected layers with a ReLU activation in between. We will denote the SimCLR head evaluated on $\boldsymbol{x}$ as $\boldsymbol{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$. SimCLR shows that by encouraging "like" images to be close in cosine similarity, given by $\text{sim}(\boldsymbol{z}, \boldsymbol{z}') = \frac{\boldsymbol{z}^T \boldsymbol{z}'}{\|\boldsymbol{z}\| \|\boldsymbol{z}'\|}$, the encoding of $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$ becomes better for classification.

For a batch of size n_b, $B = \{\boldsymbol{x}_i\}_{i=1}^{n_b}$, they augment the batch with two transforms, $\widetilde{\boldsymbol{x}}_{i,1} = \boldsymbol{T}_1(\boldsymbol{x}_i)$ and $\widetilde{\boldsymbol{x}}_{i,2} = \boldsymbol{T}_2(\boldsymbol{x}_i)$ for all $i \in \{1 \dots n_b\}$. We will denote the augmented batch as $\widetilde{B} = \{\widetilde{\boldsymbol{x}}_{i,1}\}_{i=1}^{n_b} \cup$

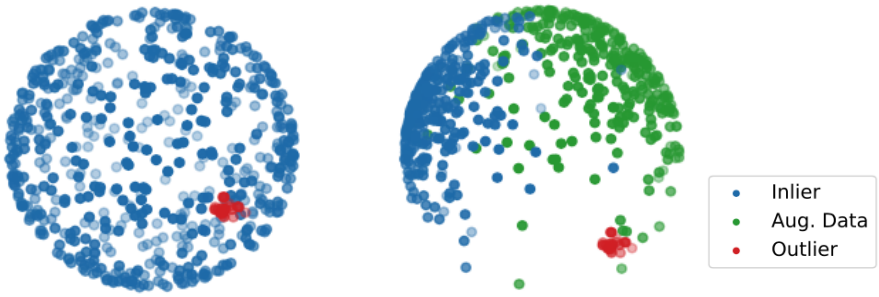

Figure 3: Distribution of features where encoders are trained with (left) normal SimCLR and (right) SimCLR augmented with shifting transforms. We can see that by utilizing shifting transforms, novelty detection becomes easier. Figure from [31]

.

$\{\widetilde{\boldsymbol{x}}_{i,2}\}_{i=1}^{n_b}$. The SimCLR loss can be written as

$$\mathcal{L}_{SimCLR}(B) = \frac{1}{2n_b}\sum_{i=1}^{n_b}\sum_{j=1}^{2} -\log \frac{\exp(\text{sim}(\boldsymbol{z}_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}_{i,j}), \boldsymbol{z}_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}_{i,3-j}))/\tau)}{\sum_{(i',j')\neq(i,j)}\exp(\text{sim}(\boldsymbol{z}_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}_{i,j}), \boldsymbol{z}_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}_{i',j'})/\tau)}. \tag{7}$$

For our experiments we set $\tau = 0.5$. Each sample is randomly transformed twice and its matching pair is used as the similar sample for the contrastive loss, while all other samples in the batch are the negative, or unlike, samples.

2.4 Self-Supervised Learning for Novelty Detection

One problem with using SimCLR features for novelty detection is that the loss will try to spread feature encodings of the inliers throughout the feature space. This problem is illustrated in the left plot of Fig. 3. Sohn et al. [31] discuss that this problem can be alleviated by reducing the batch size, and utilizing "shifting" augmentation transforms. They augment the training with these shifting augmentations and use them as dissimilar samples during SimCLR training. This has the affect of not filling the space with only inliers. The effects of this technique can be seen in right plot of Fig. 3. A shifting transform should significantly change the data so it is semantically different from the unshifted version. This is in contrast to the similarity transforms that attempt to not change a sample semantically. For example, [31] use 90, 180, and 270 degree rotations for their experiments using images. A 90 degree rotation of a natural image will look very different from an image that is right side up.

In parallel work, Tack et al. [9] introduce Contrastive Shifting Instances (CSI), which is another self-supervised novelty detection technique that utilizes shifting transforms. We define a set of shifting transforms $\mathcal{S} = \{\boldsymbol{S}_j\}_{j=1}^{n_t}$ and a shifted version of batch B as $B_{\boldsymbol{S}} = \{\boldsymbol{S}(\boldsymbol{x}_t)\}_{t=1}^{n_b}$. Now we can define the contrastive-CSI loss,

$$\mathcal{L}_{con-CSI} = \mathcal{L}_{SimCLR}\left(\bigcup_{j=1}^{n_t} B_{\boldsymbol{S}_j}\right). \tag{8}$$

Samples evaluated on different shifting transforms are treated as negative, or unlike, samples within the contrastive loss. To use images as an example, a 90 degree rotated image of a dog would be a negative sample to the upright version of the dog. The shifting transformations also allow

us to evaluate a test point multiple times. We can augment a test point $\boldsymbol{x}$ with $\mathcal{S}$, giving us $\{\boldsymbol{S}_1(\boldsymbol{x}), \boldsymbol{S}_2(\boldsymbol{x}), \dots \boldsymbol{S}_{n_t}(\boldsymbol{x})\}$. This augmentation will allow us to evaluate a single novelty score n_t times.

The second contribution of CSI is the addition of a shift-classification head. CSI also uses an additional network head that acts on the features $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$. This layer has an output dimension equal to the number of shifts n_t. We will denote the logits of this layer as $\boldsymbol{\ell}_{\boldsymbol{\theta}}(\cdot)$. This shift-classification head is trained with cross entropy loss

$$\mathcal{L}_{con-shi} = \frac{1}{2n_b n_t} \sum_{j=1}^{n_t} \sum_{t=1}^{n_b} \sum_{l=1}^{2} -\log \text{softmax}(\boldsymbol{\ell}_{\boldsymbol{\theta}}(\boldsymbol{S}_j(\widetilde{\boldsymbol{x}}_{t,l})))_j \tag{9}$$

$$\text{softmax}(\boldsymbol{\ell})_j = \frac{\exp(\ell_j)}{\sum_{i=1}^{n_t} \exp(\ell_i)}. \tag{10}$$

These losses are combined to make the CSI loss

$$\mathcal{L}_{CSI} = \mathcal{L}_{con-CSI} + \lambda_{shi} \mathcal{L}_{con-shi}, \tag{11}$$

where $\lambda_{shi} > 0$ trades off between the two losses. The CSI authors found that setting $\lambda_{shi} = 1$ worked well. We will also use this setting in our experiments.

Finally, CSI introduces several inlier scores that work well with SimCLR/CSI trained encoders. They develop a feature norm score,

$$s_{norm}(\boldsymbol{x}) = \|\boldsymbol{z}(\boldsymbol{x})\|. \tag{12}$$

This score is based on the observation that features of inlier samples have a larger norm than features of novelties.

They also develop a nearest neighbor cosine similarity score. This score utilizes the features from the training set as a dictionary. The nearest neighbor cosine similarity score can be written as

$$s_{cos}(\boldsymbol{x}, X) = \max_{\boldsymbol{x}' \in X} \text{sim}(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x}')), \tag{13}$$

where X is the training set. SimCLR and CSI are training the feature encodings of semantically similar samples to be close in terms of cosine similarity. Therefore, we assume inliers, which should be semantically similar to something in the training set, should have an encoding that is close, in terms of cosine similarity, to a sample from the training set.

In CSI, these scores are combined in a product to form what they call the contrastive score.

$$s_{con}(\boldsymbol{x}, X) = s_{norm}(\boldsymbol{x}) s_{cos}(\boldsymbol{x}, X). \tag{14}$$

This product is one way the CSI authors combine scores, but there is no justification for it, other than empirical success. CSI also utilizes the shift classifier head logits $\boldsymbol{\ell}_{\boldsymbol{\theta}}(\cdot)$ with the shift score

$$s_{shi}(\boldsymbol{x}, j) = \ell_{\boldsymbol{\theta}}(\boldsymbol{x})_j \tag{15}$$

where $\ell_{\boldsymbol{\theta}}(\cdot)_j$ is the logit corresponding to the jth shift. Inliers are more likely to have the shift correctly identified by the shift-classifier than novelties.

CSI combines s_{con} and s_{shi} by performing a weighted sum over the n_t shifts

$$s_{CSI}(\boldsymbol{x}) = \sum_{j=1}^{n_t} \left[\lambda_j^{con} s_{con}(\boldsymbol{S}_j(\boldsymbol{x}), \boldsymbol{S}_j(X)) + \lambda_j^{cls} s_{cls}(\boldsymbol{S}_j(\boldsymbol{x}), j) \right] \tag{16}$$

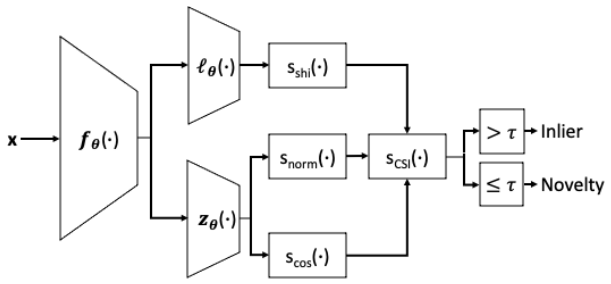

Figure 4: CSI uses two head networks, a SimCLR head, $\boldsymbol{z}_{\boldsymbol{\theta}}(\cdot)$, and a shift-classifier head, $\boldsymbol{\ell}_{\boldsymbol{\theta}}(\cdot)$, which both utilizes a common feature encoder network, $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$. These networks are jointly trained to minimize $\mathcal{L}_{CSI}$. The final CSI score performs ensembling over the shift score, norm score, and cos score.

where λ_js are weights chosen to balance the scores,

$$\lambda_j^{con} = \frac{n}{\sum_{i=1}^{n} \|\boldsymbol{z}_{\boldsymbol{\theta}}(\boldsymbol{S}_j(\boldsymbol{x}_i))\|} \tag{17}$$

$$\lambda_j^{cls} = \frac{n}{\sum_{i=1}^{n} \ell_{\boldsymbol{\theta}}(\boldsymbol{S}_j(\boldsymbol{x}_i))_j}. \tag{18}$$

Again, the authors of CSI give no justification for using this weighted sum to combine scores other than empirical success. We are motivated by the empirical success of CSI's ensembling to develop a technique that is more grounded in detection theory. CSI is summarized in Fig. 4. In our work, we use the CSI loss to train our one-class models and we use the CSI score as a state-of-the-art ensembling benchmark.

3 Contrastive Learning for RF Waveform Data

3.1 Similarity Transforms

Based on the success of self-supervised novelty detection techniques on images, we want to develop self-supervised learning for RF waveform data. In this section we will discuss our efforts to adapt SimCLR to work with radar and communication waveform data. First, we must develop new similarity transforms (denoted as $\boldsymbol{T}(\cdot)$ above). The goal of these transformations is to transform the original signal into a signal that is stylistically different but semantically similar to the original. We want our SimCLR model to take two transformations of the same sample and output two encodings that are close with respect to cosine similarity. In order to design similarity transforms for RF waveforms, we started with the radar pulse measurement model from [32]. The SIDLE dataset waveforms are modeled as a pulse train given by

$$y(t) = \sum_{p=1}^{P} A_p x(t/\alpha - t_p) \exp(j\omega_0 t) + w(t) \tag{19}$$

where $x(t)$ is the base radar pulse, $j = \sqrt{-1}$, A_p is the pulse amplitude, α is a time scaling parameter, t_p is a pulse-dependent time-shift, ω_0 is the carrier frequency, and $w(t)$ is additive noise. With this knowledge of the measurement model, we can design several similarity transformations. We will try using the following transformations: adding noise, random crop (in time), and random phase rotation.

Table 1: Performance of Linear classifier trained on SimCLR features for SIDLE (left) and GNU-Radio PSK signals (right).

Transform	Error rate(%)
Phase + Crop	23.722
Noise	0.033
Noise + Crop	0.033
Noise + Phase	**0.022**
Noise + Phase + Crop	**0.022**

Transform	Error rate(%)
Phase + Crop	37.550
Noise	32.033
Noise + Crop	**31.317**
Noise + Phase	32.283
Noise + Phase + Crop	33.100

The noise transformation is simply adding Gaussian noise to the original, noiseless waveform $\boldsymbol{x}$,

$$\boldsymbol{T}_{noise}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{w}, \tag{20}$$

where $\boldsymbol{w}$ is a realization of Gaussian random noise. We note that the training data is noiseless. The SNR for every sample in the augmented batch is drawn independently from a uniform random distribution. The bounds of this distribution are -12 to 12 dB for the radar waveforms and -20 to 20 dB for the communication waveforms. By using a range of noise levels during training, the feature encoder learns to be agnostic to noise level.

The random crop transformation is a uniformly random crop that is 100 samples smaller than the initial window. Radar and communication signals from an uncontrolled transmitter will occur at unknown timing. Therefore, the original sample from the training set and a time-shifted version of that sample are just as likely to be seen by our receiver.

The random phase angle rotation transform is given by

$$\boldsymbol{T}_{phase}(\boldsymbol{x}) = \boldsymbol{x}\exp(j\phi), \tag{21}$$

where ϕ is the phase angle uniformly drawn between 0 and 2π. This transform takes advantage of the phase mismatch between the transmitter and receiver. If $\boldsymbol{x}$ is a valid waveform, than $\boldsymbol{T}_{phase}(\boldsymbol{x})$ is valid for any $\phi \in [0, 2\pi)$.

In our experiments we will use a composition of these transforms. When a batch is drawn during training, each transformation is performed on every sample with an independently drawn transform realization. For example, if we are using the noise and phase transformations, every sample will get an independent random draw of the phase rotation and the noise realization.

To evaluate these self-supervised training methods, we train a model with the SimCLR loss (7) using different combinations of the above transformations. We then train a linear classifier on the encoding $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot)$ and report the error rate of this classifier on a hold out test set. Details, such as learning rate schedule and epochs, are the same as our novelty detection experiments, which are described in the experiments section below.

Results from the SimCLR experiments can be seen in Table 1. For the SIDLE data, noise + phase and noise + phase + crop are tied for best performance. For the GNURadio data, noise + crop is the best. We notice is that the phase + crop combination does the worst for both SIDLE and GNURadio data. This shows that adding noise is a powerful similarity transform when training a SimCLR method on RF waveform data. We also notice that the random phase rotation boosts performance on the SIDLE data, and the addition of the random crop boosts performance on the GNURadio data. Therefore we will use all three similarity transformations (noise + phase + crop) for SimCLR models for the rest of the paper.

3.2 Shifting Transforms

We now consider shifting transforms for RF waveforms. As discussed above, the shifting transform's goal is to semantically change the signal. We will develop these transforms based on (19), intuition about radar and communication signals, and observations of the types of transformations that worked well for images in CSI.

In CSI, a 0, 90, 180, and 270 degree rotation operator was shown to work well for images. For 1D signals, the analogous operation would be to "time-reverse" the signal. For time-reverse, we have the number of transforms $n_t = 2$, where $\boldsymbol{S}_1(\cdot)$ returns the original signal and $\boldsymbol{S}_2(\cdot)$ returns the time-reversed signal.

In preliminary work, we experimented with additional shifts, including quantization, modulation, and a composition of multiple shifts. The time-reverse shift was chosen due to better performance in these preliminary experiments. Since those experiments were performed, the developed novelty detection technique has changed significantly. We plan to systematically evaluate other shifting transforms before completing this dissertation.

4 Proposed Novelty Detection Technique

4.1 Ensembling of Inlier Scores

We are motivated by CSI's observation that novelty detection performance can be improved by ensembling several scores. As discussed above, CSI uses both a weighted sum and a product in order to combine inlier scores. The CSI ensembling method works well for some datasets (e.g., their results are state-of-the-art for CIFAR-10), but we find that their strategy does not perform well in general. Motivated by the partial success of CSI's ensembling technique we set out to develop an improved ensembling strategy.

Let us assume we have n_s different inlier scores $\{s_i(\cdot)\}_{i=1}^{n_s}$. Using these scores, we want to perform a hypothesis test [33], where the H_0 hypothesis is that the test sample is an inlier and the H_1 hypothesis is that the test sample is a novelty.

The first challenge to creating an ensembling technique is that different scores have very different distributions. To illustrate this, we plot histograms of the norm and cos scores evaluated on inliers from the SIDLE radar waveform dataset. In the left plot of Fig. 5, we can see that the distributions of the norm and cos scores are very different. In order to make the inlier distributions of the scores more comparable, we will use the quantile transform [34], which will allow us to transform the inlier distribution of each score to be approximately unit Gaussian. Note that we do not have access to the novelty distribution when designing our approach. The quantile transform for the ith score is defined as

$$q_i(s) = \Phi^{-1}(\widehat{F}_i(s)) \tag{22}$$

$$\widehat{F}_i(s) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}(s \leq s_i(\boldsymbol{x}_t)), \tag{23}$$

where $\Phi^{-1}(\cdot)$ is the inverse Gaussian CDF, $\widehat{F}_i(\cdot)$ is the empirical CDF of score i evaluated on the training set, and $\mathbb{1}(\cdot)$ is the indicator function that has a value of 1 if the statement is true, and 0 otherwise. The center plot of Fig. 5 shows the same norm and cos scores after processing by the quantile transform.

We model the quantile transformed score using the random variable

$$r_i = q_i(s_i(\boldsymbol{x})). \tag{24}$$

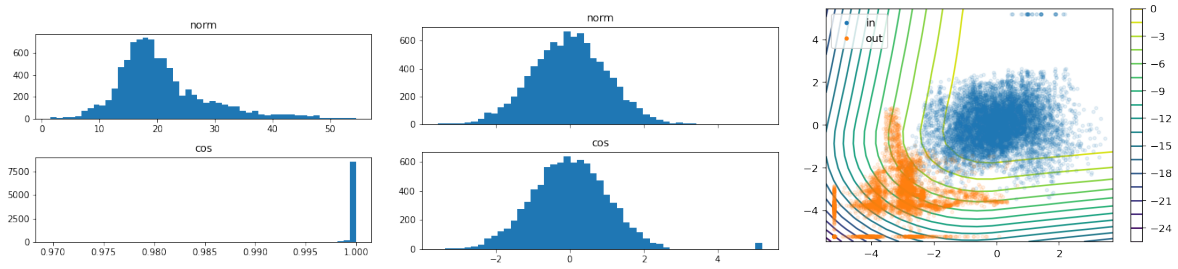

Figure 5: Left: Raw norm and cos scores evaluated on the inliers of the SIDLE dataset. Center: Quantile transformed versions of the scores. Right: Plot of inliers and novelties of norm score (x-axis) and cos score (y-axis). Contours show the value of $\log GLRT$.

This allows us to define conditional inlier and novelty distributions,

$$p(r_i|H_0) = p(q_i(s_i(\boldsymbol{x}))|\boldsymbol{x} \text{ is an inlier}) \tag{25}$$
$$p(r_i|H_1) = p(q_i(s_i(\boldsymbol{x}))|\boldsymbol{x} \text{ is a novelty}). \tag{26}$$

The quantile transform gives us

$$p(r_i|H_0) \approx \mathcal{N}(r_i; 0, 1). \tag{27}$$

When we vectorize r_is into the vector $\boldsymbol{r}$, we will approximate the distribution for the inliers as jointly Gaussian,

$$p(\boldsymbol{r}|H_0) \approx \mathcal{N}(\boldsymbol{r}; \boldsymbol{0}, \boldsymbol{\Sigma}), \tag{28}$$

where $\boldsymbol{\Sigma}$ is the covariance of $\boldsymbol{r}$ under H_0. This is an approximation because the quantile transform only enforces that every entry in $\boldsymbol{r}$ is marginally Gaussian. However, we can see in the right plot of Fig. 5 that the transformed inliers have an approximately joint Gaussian distribution. We can approximate the covariance with the sample covariance, $\widehat{\boldsymbol{\Sigma}}$, computed on samples of $\boldsymbol{r}$ from the training set.

If we had full knowledge of the distributions $p(\boldsymbol{r}|H_0)$ and $p(\boldsymbol{r}|H_1)$, we would be justified in performing novelty detection with a likelihood ratio test (LRT)

$$LRT(\boldsymbol{r}) = \frac{p(\boldsymbol{r}|H_0)}{p(\boldsymbol{r}|H_1)}. \tag{29}$$

This is because a detection algorithm based on thresholding the LRT is Neyman-Pearson optimal, meaning that the probability of detection is maximized for a fixed false-alarm rate [33]. Unfortunately, we do not have access to distribution $p(\boldsymbol{r}|H_1)$, due to there being no novelties in the training set. Our approach will be to parameterize $p(\boldsymbol{r}|H_1)$ and formulate a composite hypothesis test [35].

We will assume that the inlier score has been designed such that most inliers have a higher score than most novelties. Since the quantile transform is monotonic, the quantile transformed score, r_i, will also have this characteristic. Furthermore, the novelty distribution $r_i|H_1$ will have a negative mean. Other than this negative mean property, we know very little about the novelty distribution. For the sake of tractability, we will assume homogeneity between novelties and inliers. Intuitively, we anticipate the correlations among the inlier scores will mirror correlations among the novelty scores.

This gives us

$$p(\boldsymbol{r}|H_0) \approx \mathcal{N}(\boldsymbol{r}; \mathbf{0}, \widehat{\boldsymbol{\Sigma}}) \tag{30}$$

$$p(\boldsymbol{r}|H_1, \boldsymbol{\alpha}) \approx \mathcal{N}(\boldsymbol{r}; \boldsymbol{\alpha}, \widehat{\boldsymbol{\Sigma}}), \tag{31}$$

where $\boldsymbol{\alpha}$ is the negative, unknown mean of the novelty distribution and $\widehat{\boldsymbol{\Sigma}}$ is the shared covariance matrix, estimated from the empirical covariance matrix of the training set. As long as $n_t > 1$, the LRT based on these assumptions will be dependent on the direction of $\boldsymbol{\alpha}$. This means that a uniformly most powerful test does not exist.

However, we can formulate a generalized likelihood ratio test (GLRT) [33],

$$GLRT(\boldsymbol{r}) = \frac{p(\boldsymbol{r}|H_0)}{\max_{\boldsymbol{\alpha}:\alpha_i \leq 0} p(\boldsymbol{r}|H_1, \boldsymbol{\alpha})}. \tag{32}$$

It can be shown that the $\boldsymbol{\alpha}$ that maximizes the denominator is a projection of $\boldsymbol{r}$ onto the negative quadrant. We will define this projection as $\boldsymbol{r}^-$, whose ith element is given by

$$r_i^- = \begin{cases} r_i & r_i < 0 \\ 0 & r_i \geq 0 \end{cases} \tag{33}$$

We can write the $\log GLRT$ in terms of $\boldsymbol{r}$ and $\boldsymbol{r}^-$,

$$\log GLRT(\boldsymbol{r}) = \log p(\boldsymbol{r}|H_0) - \log p(\boldsymbol{r}|H_1, \boldsymbol{\alpha} = \boldsymbol{r}^-) \tag{34}$$

$$= -\frac{1}{2}\boldsymbol{r}^T \widehat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{r} + \frac{1}{2}(\boldsymbol{r} - \boldsymbol{r}^-)^T \widehat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{r} - \boldsymbol{r}^-) \tag{35}$$

$$= -\frac{1}{2}[\boldsymbol{r}^-]^T \widehat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{r}^- + \boldsymbol{r}^T \widehat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{r}^- \tag{36}$$

$$= (\boldsymbol{r} - \tfrac{1}{2}\boldsymbol{r}^-)^T \widehat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{r}^-. \tag{37}$$

Quantile transformed norm and cos scores, $\boldsymbol{r}$, for both inliers and novelties, are plotted together along with contours of the $\log GLRT$ score in the right plot of Fig. 5. From this plot, we can see that the contours of the $\log GRLT$ score do a good job of separating the inliers and novelties. It is clear from the figure that thresholding the $\log GLRT$ score provides better separation of the inliers and novelties than thresholding either score individually.

4.2 Shift Ensembled Novelty Detection

Now we will put together our shifting transforms and our ensembling strategies to form our novelty detection framework, called Shift-Ensembled Novelty Detection (SEND). Let us say we have n_s score functions, indexed by i, and n_t shifting transforms, indexed by j. We want to evaluate our n_s scores functions on our n_t transforms, giving a total ensemble of size $n_s n_t$. For SEND, we will define a version of the quantile transform indexed by i and j,

$$q_{ij}(s) = \Phi^{-1}(\widehat{F}_{ij}(s)) \tag{38}$$

$$\widehat{F}_{ij}(s) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}(s \leq s_i(\boldsymbol{S}_j(\boldsymbol{x}_t))) \tag{39}$$

where $\boldsymbol{S}_j(\cdot)$ is the jth shifting transform and $\widehat{F}_{ij}$ is the sample CDF for score function i and shift j on the inlier training data $\{x_t\}_{t=1}^n$. Next, we combine the $\log GLRT$ score and the shifting transform to define the SEND score,

$$s_{SEND}(\boldsymbol{x}) = (\boldsymbol{r}(\boldsymbol{x}) - \tfrac{1}{2}\boldsymbol{r}^-(\boldsymbol{x}))^T \widehat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{r}^-(\boldsymbol{x}) \tag{40}$$

$$\boldsymbol{r}(\boldsymbol{x}) = \begin{bmatrix} q_{1,1}(s_1(S_1(\boldsymbol{x}))) \\ \vdots \\ q_{1,n_t}(s_1(S_{n_t}(\boldsymbol{x}))) \\ q_{2,1}(s_2(S_1(\boldsymbol{x}))) \\ \vdots \\ q_{n_s,n_t}(s_{n_s}(S_{n_t}(\boldsymbol{x}))) \end{bmatrix}, \tag{41}$$

where $\widehat{\boldsymbol{\Sigma}}$ is the empirical covariance matrix of training samples of $\{\boldsymbol{r}(\boldsymbol{x}_i)\}_{i=1}^n$.

Next, we will discuss possible combinations of feature encoders and score functions, but SEND is compatible with any combination of novelty detection techniques that provide inlier scores.

1. We first introduce our one-class novelty detection strategy, One-Class SEND (OC-SEND). For this approach we will use a DNN trained using the CSI loss given by (11). We will ensemble $s_{norm}(\cdot)$, $s_{cos}(\cdot)$, and $s_{shift}(\cdot)$ with $s_{SEND}(\cdot)$. The norm and cos scores will utilize features from the SimCLR head ($\boldsymbol{z}_{\boldsymbol{\theta}}(\cdot)$) and the shift score will utilize the shift-classifier head ($\boldsymbol{\ell}_{\boldsymbol{\theta}}(\cdot)$). The shifting transform used for both training and testing will be time-reverse. This gives us $n_s = 3$ and $n_t = 2$ for a total ensemble of six scores. This implementation of SEND does not require access to the class-labels, so it is compatible with the one-class classification problem.

2. Next we will introduce our Multi-Class SEND (MC-SEND) technique. For MC-SEND we will train a classifier to learn the joint class-transform label (y, j), where y is the class label and j is the shift label. If we have n_t shifts, we turn a K-class problem into a n_tK-class problem. This means the final layer of the neural network is changed to have an output of dimension n_tK. We experiment with using cross-entropy as well as label-smoothing with $\alpha = 0.1$. We will denote these losses as Joint-CE and Joint-LS. With this trained classifier we can evaluate the joint class-dependent Mahalanobis distance

$$s_{joint-Mah}(\boldsymbol{x}) = -\min_{(k,j)} (\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_{kj}) \widehat{\boldsymbol{\Sigma}}_{kj}^{-1} (\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_{kj}), \tag{42}$$

where $\widehat{\boldsymbol{\mu}}_{kj}$ and $\widehat{\boldsymbol{\Sigma}}_{kj}$ are the sample mean and sample covariance of joint class (k, j).

For the MC-SEND ensemble we will utilize both this joint-label classification network and the self-supervised network used in OC-SEND. We ensemble $s_{joint-Mah}(\cdot)$, evaluated on the penultimate features of the joint-label classifier, $s_{norm}(\cdot)$ and $s_{cos}(\cdot)$ evaluated on the features of the SimCLR head ($\boldsymbol{z}_{\boldsymbol{\theta}}(\cdot)$), and $s_{shift}(\cdot)$ evaluated on the shift-classifier head ($\boldsymbol{\ell}_{\boldsymbol{\theta}}(\cdot)$). Our experiments in the next section show that the addition of the joint-Mahalanobis score increases performance over the base OC-SEND technique.

5 Experiments

5.1 Datasets

For our experiments we use the SIDLE radar waveform dataset used in [19] and a communication dataset constructed using GNURadio [36], which is an altered version of the RadioML10a dataset

[6]. The SIDLE dataset is made up of radar waveforms of various types. The waveforms are either padded or cropped to a length of 5000 samples. This padding or cropping is done randomly, so that the location of the waveform may not be centered in the window. Noise is added at training time so that we can get different noise realizations of the same signal. SNR levels are chosen uniformly random between -12 and 12 dB. We used classes 1-10 (excluding class 6) of the SIDLE dataset as inlier classes, while classes 11-18 were treated as the novelty classes.

Our GNURadio communication dataset simulates various communication modulation types. We use similar parameters to the RadioML10a dataset, except we choose to make our signals length 1 024, in order to use similar techniques to SIDLE. We use the GNURadio dynamic channel model, which gives us a fading channel with random timing, frequency, and phase offsets. Noise is added at training time with SNRs randomly chosen from between -20 and 20 dB. We use the PSK modulations (BPSK, QPSK, 8PSK) as inliers and the analog modulations (AM-DSB, AM-SSB, WBFM) as novelties. For both datasets, train/test splits are selected by uniformly sampling 10% of each class for the test set.

5.2 Evaluation

We will use the area under the receiver operating curve (AUC) and detection probability (DPR) at a fixed false alarm rate to evaluate the proposed techniques. We use AUC because it is a threshold independent metric of novelty detection performance. We also report DPR at a fixed false alarm rate because it is more tangible than AUC and gives a specific point on the receiver operating curve. DPR will be evaluated at 1% false alarm rate on the SIDLE dataset and 5% on the GNURadio dataset. We report results for a higher false alarm rate for GNURadio because that detection problem is harder than the SIDLE radar waveforms problem.

5.3 Model Training

For the neural network architecture we will use an adapted version of Resnet-18 [37]. We will use the adaptation technique specified in [5]. Waveforms are complex valued, so we use complex multiplications in the convolutional layers. Batch norm is computed separately for the real and imaginary components. The convolutional kernel size of layer 1 is set to 11, while all others are set to 9. We also convert 2D operations (for images) to 1D operations (for time series).

The self-supervised network is trained to minimize $\mathcal{L}_{SimCLR}$ or $\mathcal{L}_{CSI}$. The joint-label classifier network is trained on the cross entropy or label-smoothing loss with joint class-shift labels. The training samples for the joint-label classifier network are augmented with the same similarity transforms as the self-supervised network. We train the networks for 100 epochs on the SIDLE data and 300 epochs on the GNURadio data. Optimization is performed by the LARS [38] optimizer. The learning rate is scheduled using cosine annealing [39] with initial learning rate of 0.1, a max learning rate of 1.0, 1 epoch of warmup, and an ending learning rate of 10^{-6}.

For all experiments, the batch size was set to $128/n_t$, which makes the augmented batch have a size of 128. We note that the training time scales with n_t. Training a model with $n_t = 2$ takes twice as long as training a model with $n_t = 1$.

For our experiments, we split the training set into two subsets. The first is the "dictionary," which is used for the cosine nearest neighbor score and Mahalanobis distance, and the second will be used for the sample CDF (39) for the quantile transform. We use 10% of the training set for the dictionary set and the remaining 90% for fitting the quantile transform. We will implement the quantile transforms with the python sklearn [40] package, which approximates the sample CDF (39) with 10 000 bins.

Table 2: Novelty detection performance for various techniques on the SIDLE and GNURadio waveform datasets. Area under the receiver operating curve is labeled as AUC and detection probability is labeled DPR. DPR is evaluated at 1% false alarm rate for the SIDLE data and 5% false alarm rate for the GNURadio data.

	SIDLE		GNURadio	
Loss / Inlier Score	AUC	DPR	AUC	DPR
SimCLR / Con [9]	0.9563	92.51	0.2026	1.43
CSI / CSI [9]	0.9769	76.76	0.4243	1.03
CSI / OC-SEND (ours)	**0.9997**	**99.88**	**0.8456**	**56.70**
Cross Entropy / Mah. [20]	0.9944	89.33	0.7812	23.43
Cross Entropy / OC-SVM [18, 19]	0.9749	22.50	0.7676	39.45
Joint-CE+CSI / MC-SEND (ours)	0.9997	99.93	0.8629	60.42
Joint-LS+CSI / MC-SEND (ours)	**0.9998**	**99.99**	**0.8700**	**62.62**

5.4 Baseline Methods

We will compare our methods to multiple baseline novelty detection methods. For the one-class classification problem we will compare our methods to the CSI [9] score. We note that this score utilizes the same DNN model and base inlier scores (norm, cos, and shift) as OC-SEND, but combines the scores differently. We will also compare to the contrastive score evaluated on a SimCLR model.

We will also use two baseline multi-class novelty detection methods, that utiize the class-labels during training. Both methods use a the penultimate layer of a DNN classifier, trained using cross-entropy, as the feature encoder. The first utilizes the class dependent Mahalanobis distance evaluated on the penultimate layer of the DNN classifier [20]. The second uses a class-dependent OC-SVM, trained on the penultimate features of the DNN classifier [18]. We note that this is similar to the method used in [19].

5.5 Novelty Detector Performance

We now evaluate our developed novelty detection techniques. We provide novelty detection performance for various feature encoder/score function pairs on the SIDLE and GNURadio datasets in Table 2. The table lists the training loss used to train the model(s) and the inlier score function used. The tables are broken into two sections. The top section includes methods that are one-class classifiers, which means they do not use any class-labels. These methods could be used when class-labels are unavailable or for datasets with a single class. The bottom section displays methods that utilize the class-labels during training. These methods are limited to solving the multi-class novelty detection problem.

Our experiments show that our SEND framework significantly benefits novelty detection performance. For the one-class classification methods, OC-SEND is the best on both datasets in both DPR and AUC. For the multi-class novelty detection methods, the Joint-LS / MC-SEND pair performs best. These results show the strong benefit of utilizing the SEND methods to ensemble multiple inlier scores. We also see that MC-SEND performs better than OC-SEND. This result is not surprising since MC-SEND is able to use the joint classifier features as well as the self-supervised CSI features. These results also show that label-smoothing boosts performance of MC-SEND. In every metric the Joint-LS version of MC-SEND beats the Joint-CE version.

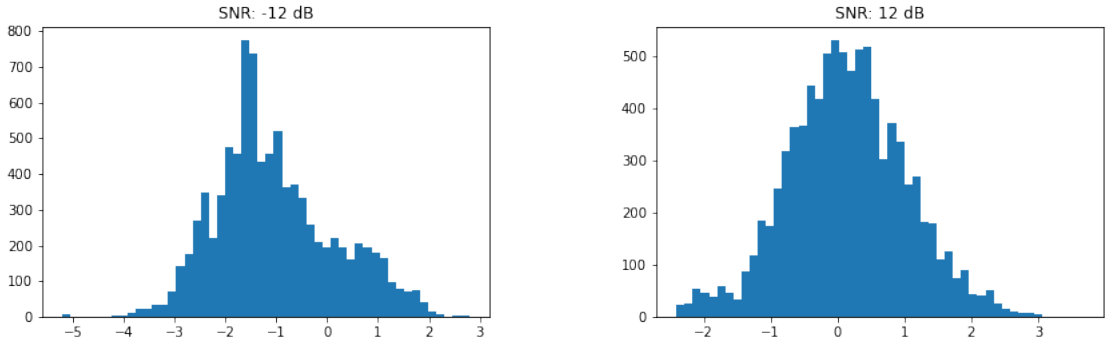

Figure 6: Histogram plots of quantile-transformed norm score when SNR is set to -12 dB (left) and 12 dB (right).

6 Remaining Work to Finish Dissertation

The proposed dissertation will include additional work. As discussed above, we plan to systematically evaluate other shifting transforms for RF waveform data. Additional transforms will include modulation, quantization, and compositions of multiple shifting transforms. Compositions of multiple transforms would allow us to increase n_t. For example if we compose time-reverse with modulation we would double the number of shifts to $n_t = 4$, which would then double the number of inlier scores in the SEND ensemble. Our hope is that this increase in ensemble size leads to improved performance.

We also plan to incorporate a SNR dependent quantile transform. We are motivated by the fact that score distributions are different for different SNR levels. Fig. 6 shows the norm score, after being processed by the quantile transform, evaluated on signals with SNRs of -12 and 12 dB. We can see that the low-SNR score distribution skews negative, while the high-SNR distribution skews positive. Our current ensembling strategy does not capture this behavior, and treats all SNR levels the same. We believe that by conditioning these distributions on the SNR, we may be able to increase performance of the ensembling strategy. Our plan is to use multiple ranges of SNR when evaluating the sample CDF on the training set. This will give us multiple quantile transforms corresponding to these different SNR bins. This would allow us to enforce

$$p(r_i|H_0, \text{SNR}) = \mathcal{N}(r_i; 0, 1), \tag{43}$$

across all SNR levels. This is in contrast to the current implementation, where (30) holds for the distribution of r_i over all SNRs, but, as we can see from Fig. 6, does not hold when we look at specific SNR levels. At test time, we will use signal energy to estimate the SNR of a test sample and choose the corresponding quantile transform with the appropriate SNR range. This should result in the quantile transform being more consistent across SNR levels, which should lead to better performance. A SNR-dependent quantile transform would also allow the same threshold to give the same false-alarm rate over the entire SNR range. From Fig. 6, we can see that low SNR samples will likely have a higher false alarm rate than high SNR signals, for a fixed threshold.

References

[1] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. on Computational Imaging*, vol. 5, pp. 52–67, Mar. 2019.

[2] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, "Plug and play methods for magnetic resonance imaging," *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 105–116, 2020.

[3] S. Liu, E. Reehorst, P. Schniter, and R. Ahmad, "Free-breathing cardiovascular MRI using a Plug-and-Play method with learned denoiser," in *Proc. IEEE Internat. Symp. on Biomedical Imaging*, Apr. 2020.

[4] M. Wharton, E. T. Reehorst, and P. Schniter, "Compressive SAR image recovery and classification via CNNs," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, pp. 1081–1085, 2019.

[5] M. Wharton, A. Pavy, and P. Schniter, "Phase-Modulated radar waveform classification using deep networks," *arXiv:2102:07827*, Feb. 2021.

[6] T. J. O'Shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. GNU Radio Conf.*, vol. 1, 2016.

[7] P. Perera, P. Oza, and V. M. Patel, "One-class classification: A survey," *arXiv:12101.03064*, 2021.

[8] P. Perera and V. M. Patel, "Deep transfer learning for multiple class novelty detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 11544–11552, 2019.

[9] J. Tack, S. Mo, J. Jeong, and J. Shin, "Csi: Novelty detection via contrastive learning on distributionally shifted instances," in *Proc. Neural Information Processing Systems Conf.*, vol. 33, pp. 11839–11852, 2020.

[10] R. Gnanadesikan and J. R. Kettenring, "Robust estimates, residuals, and outlier detection with multiresponse data," *Biometrics*, pp. 81–124, 1972.

[11] V. Hautamaki, I. Karkkainen, and P. Franti, "Outlier detection using k-nearest neighbour graph," 2004.

[12] E. Parzen, "On estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[13] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, July 2001.

[14] D. Tax and R. Duin, "Support vector data description," *Machine Learning*, vol. 54, pp. 45–66, 2004.

[15] L. Ruff, J. R. Kauffmann, and R. A. Vandermeulen, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, pp. 756–795, 2021.

[16] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Internat. Conf. on Learning Representations*, 2017.

[17] M. Hein, M. Andriushchenko, and J. Bitterwolf, "Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 41–50, 2019.

[18] J. Andrews, T. Tanay, E. J. Morton, and L. D. Griffin, "Transfer representation-learning for anomaly detection," in *Proc. Internat. Conf. on Machine Learning*, 2016.

[19] R. V. Chakravarthy, H. Liu, and A. M. Pavy, "Open-set radar waveform classification: Comparison of different features and classifiers," in *Proc. IEEE Radar Conf.*, pp. 542–547, 2020.

[20] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proc. Neural Information Processing Systems Conf.*, 2018.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[22] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?," in *Proc. Neural Information Processing Systems Conf.*, (Vancouver, B.C.), Dec. 2019.

[23] D. Bahri, H. Jiang, Y. Tay, and D. Metzler, "Label smoothed embedding hypothesis for out-of-distribution detection," *arXiv:2102.05131*, 2021.

[24] J. Winkens, R. Bunel, A. G. Roy, R. Stanforth, V. Natarajan, J. R. Ledsam, P. MacWilliams, P. Kohli, A. Karthikesalingam, S. Kohl, T. Cemgil, S. M. A. Eslami, and O. Ronneberger, "Contrastive training for improved out-of-distribution detection," *arXiv:2007.05566v1*, July 2020.

[25] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 4037–4058, 2021.

[26] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve robustness and uncertainty," in *Proc. Neural Information Processing Systems Conf.*, 2019.

[27] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[28] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proc. Neural Information Processing Systems Conf.*, 1994.

[29] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive leanring of visual representations," in *Proc. Internat. Conf. on Machine Learning*, pp. 1597–1607, July 2020.

[30] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Neural Information Processing Systems Conf.*, 2016.

[31] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister, "Learning and evaluating representations for deep one-class classification," in *Proc. Internat. Conf. on Learning Representations*, 2021.

[32] B. Rigling and C. Roush, "ACF-Based classification of phase modulated waveforms," in *Proc. IEEE Radar Conf.*, pp. 287–291, 2010.

[33] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer, 2nd ed., 1994.

[34] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics*, vol. 19, pp. 185–193, Sept. 2003.

[35] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 3rd ed., 1991.

[36] GNU Radio Website, accessed January 2021.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[38] Y. You, B. Ginsburg, and I. Gitman, "Large batch training of convolutional networks," *arXiv:1708:03888v3*, 2017.

[39] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rate," in *Proc. SPIE, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, May 2019.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.