2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC) | 978-1-6654-9455-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/SPAWC51304.2022.9834032

# Learning to Detect with Constant False Alarm Rate

Tzvi Diskin, Uri Okun and Ami Wiesel

***Abstract*—We consider the use of machine learning for hypothesis testing with an emphasis on target detection. Classical model-based solutions rely on comparing likelihoods. These are sensitive to imperfect models and are often computationally expensive. In contrast, data-driven machine learning is often more robust and yields classifiers with fixed computational complexity. Learned detectors usually provide high accuracy with low complexity but do not have a constant false alarm rate (CFAR) as required in many applications. To close this gap, we propose to add a term to the loss function that promotes similar distributions of the detector under any null hypothesis scenario. Experiments show that our approach leads to near CFAR detectors with similar accuracy as their competitors.**

***Index Terms*—hypothesis testing, deep learning**

## I. Introduction

The deep learning revolution has led many to apply machine learning methods to classical problems in all fields including statistics. Examples range from estimation [1–6] to detection [7–12]. Deep learning is a promising approach for deriving high accuracy and low complexity alternatives when classical solutions are intractable. To facilitate this switch, we must ensure that the learned solutions are accurate but also satisfy other classical requirements. In this paper, we focus on learning detectors for composite hypothesis testing. We claim that current solutions deliver on their promises but lack the Constant False Alarm Rate (CFAR) requirement which is critical in many applications. To close this gap, we provide a framework for learning accurate CFAR detectors.

Detection theory begins with simple hypothesis testing where a detector must decide between two fully specified distributions. The classical solution is the Likelihood Ratio Test (LRT) which is optimal in terms of maximizing the detection probability subject to a false alarm constraint. Composite hypothesis testing is a more challenging setting where the hypotheses involve unknown deterministic parameters. A CFAR detector is invariant to these parameters and has identical false alarm probabilities as long as no target is present. This allows the user to set the thresholds a priori. A popular approach is the Generalized Likelihood Ratio Test (GLRT) which can be interpreted as estimating the unknown parameters and then plugging them into a standard LRT. GLRT performs well when the number of unknowns is relatively small and is asymptotically CFAR. On the other hand, it is generally sub-optimal in finite sample settings and may be computationally expensive.

Data-driven classifiers are the machine learning counterpart to model-based detectors. In simple settings with no unknown deterministic parameters, it is well known that the optimal Bayes classifier converges to the LRT with a specific false alarm rate. More advanced classifiers can also maximize the cumulative detection rate over a wide range of false alarms, also known as the area under the curve (AUC) [13, 14].

There is a growing body of works on using machine learning for target detection. In the context of hyperspectral imagery, [11] introduced the use of support vector machines (SVM). Deep neural networks (DNN) were proposed in [8]. An important ingredient of these works is the use of an artificial training set where real noise data is augmented with synthetically planted targets. In the context of radar detection, SVMs were considered in [10]. Specific CFAR radar detectors were developed by relying on CFAR features [15–17].

The main contribution of this paper is a framework for learning CFAR detectors denoted by CFARnet. The framework is general purpose and can be applied to arbitrary composite hypothesis testing problems. CFARnet is based on adding a penalty function to the optimization which decreases the distances between the distributions under different parameter values. To optimize CFARnet we rely on empirical and differentiable distances that have recently become popular in unsupervised deep learning. Our numerical experiments, show that the resulting networks are approximately CFAR while paying a negligible price in terms of accuracy.

The CFAR notion is closely related to the topics of "fairness" and "out of distribution (OOD)" generalization which have recently attracted considerable attention in the machine learning literature, e.g., [18, 19]. CFAR can be interpreted as a fairness property with respect to the unknown deterministic parameters. The closest work is [20] which also enforces "equalized odds" using a distance between distributions. A main difference is that CFAR is a one-sided fairness property and requires equal rates only in the null hypothesis. Algorithmically, [20] compares the high dimensional joint distribution of the predictions and the unknown parameters, whereas we only consider the scalar distribution of the predictions. This makes our method significantly cheaper in terms of computational complexity. Indeed, we rely on a simple kernel based distance and do not require sophisticated adversarial networks.

## II. Background on statistical distances

We begin with a brief background on statistical distances.

**Definition 1.** *Let $X \sim p(X)$ and $Y \sim p(Y)$ be two random variables. A statistical distance $d(X;Y)$ is a function that satisfies $d(X;Y) \geq 0$ with equality if and only if $p(X) = p(Y)$.*

A distance which has recently become popular is the Max-

imum Mean Discrepancy (MMD) [21]:

$$d_{\mathrm{MMD}}(X;Y) = \mathbf{E}[k(X,X')] + \mathbf{E}[k(Y,Y')] - 2\mathbf{E}[k(X,Y)] \quad (1)$$

where $X$ and $X'$ are independent and identically distributed (i.i.d.), and so are $Y$ and $Y'$. The function $k(\cdot,\cdot)$ is a characteristic kernel over a reproducing kernel Hilbert space, e.g., the Gaussian Radial Basis Function (RBF).

Recent advances in deep generative models allow us to optimize distances as MMD in an empirical and differentiable manner. For this purpose, we need to represent each distribution using a small dataset. Let $\{X_i\}_{i=1}^N$ and $\{Y_i\}_{i=1}^N$ be i.i.d. realizations of $X$ and $Y$, respectively. Then, an empirical version of the MMD can be used where the expetations in (1) are replaced by their empirical estimates. More advanced metrics can be obtained using the tools of generative adversarial networks (GANs). In this paper, we only deal with distances between scalar random variables and simple MMD distances suffice.

## III. Problem formulation

We consider a binary hypothesis test. Let $\boldsymbol{x}$ be an observed random vector whose distribution $p(\boldsymbol{x};\mathbf{z})$ depends on an unknown deterministic parameter $\mathbf{z}$. The value of $\mathbf{z}$ defines two possible hypotheses

$$\begin{aligned} y=0: &\quad \mathbf{z}\in\mathcal{Z}_0 \\ y=1: &\quad \mathbf{z}\in\mathcal{Z}_1. \end{aligned} \quad (2)$$

The goal is to design a detector $\hat{y}(\boldsymbol{x}) \in \{0,1\}$ as a function of $\boldsymbol{x}$ that will identify the true hypothesis $y \in \{0,1\}$. Performance is measured in terms of probability of correct detection, also known as True Positive Rate (TPR):

$$P_{\mathrm{TPR}}(\mathbf{z}) = P(\hat{y}(\boldsymbol{x})=1; y=1) \quad (3)$$

and probability of false alarm, also known as False Positive Rate (FPR):

$$P_{\mathrm{FPR}}(\mathbf{z}) = P(\hat{y}(\boldsymbol{x})=1; y=0) \quad (4)$$

In practice, the user typically provides a false alarm constraint $P_{\mathrm{FPR}} \leq \alpha$ that must be satisfied and the goal is to maximize $P_{\mathrm{TPR}}$.

It is standard to consider detectors of the form

$$\hat{y}(\boldsymbol{x}) = \begin{cases} 0 & T(\boldsymbol{x}) < \gamma \\ 1 & T(\boldsymbol{x}) \geq \gamma \end{cases} \quad (5)$$

where $T(\boldsymbol{x})$ is a function of the measurements and $\gamma$ is a threshold. This structure allows users to tune $P_{\mathrm{FPR}}$ by adjusting the threshold. Performance is usually visualized using the Receiver Operating Characteristic (ROC) which plots the TPR as a function of the FPR. In signal processing applications, users are often interested in a region of very low FPRs, e.g., $10^{-1}-10^{-3}$ and the goal is to maximize the TPR probabilities in this area.

A main challenge in detection theory are the unknown parameters under the null hypothesis $y=0$. The false alarm probability (FPR) is generally a function of these parameters and cannot be controlled without their knowledge. Therefore, it is often preferable to restrict the attention to CFAR detectors.

**Definition 2.** *A detector $T(\boldsymbol{x})$ is CFAR if its distribution is invariant to the value of $\mathbf{z} \in \mathcal{Z}_0$.*

As we will review below, many classical detectors are CFAR or asymptotically CFAR. With the growing trend of switching to machine learning, the goal of this paper is to introduce a framework for learning CFAR detectors.

## IV. Classical Likelihood based Detectors

In this section, we provide a short background on classical detectors based on likelihood ratios. In the simple case where $\mathcal{Z}_0 = \{\mathbf{z}_0\}$ and $\mathcal{Z}_1 = \{\mathbf{z}_1\}$ are singletons, hypothesis testing has an optimal solution known as the Likelihood Ratio Test (LRT) due to Neyman Pearson [22, p. 65]. LRT theory states that the optimal detector for maximizing detection subject to a given false alarm probability is:

$$T_{\mathrm{LRT}}(\boldsymbol{x}) = 2\log\frac{p(\boldsymbol{x};\mathbf{z}=\mathbf{z}_1)}{p(\boldsymbol{x};\mathbf{z}=\mathbf{z}_0)} \quad (6)$$

and the threshold $\gamma$ is chosen to satisfy the false alarm (FPR) constraint.

The more realistic scenario is composite hypotheses testing where one or both of the hypotheses involve unknown parameters and there is no simple and optimal solution. A popular heuristic is the Generalized Likelihood Ratio Test (GLRT) that estimates the unknown parameters using the Maximum Likelihood technique and plugs them into the LRT detector:

$$T_{\mathrm{GLRT}}(\boldsymbol{x}) = 2\log\frac{\max_{\mathbf{z}\in\mathcal{Z}_1} p(\boldsymbol{x};\mathbf{z})}{\max_{\mathbf{z}\in\mathcal{Z}_0} p(\boldsymbol{x};\mathbf{z})} \quad (7)$$

Setting the threshold to ensure a fixed $P_{\mathrm{FPR}}$ is non trivial. Fortunately, under regular conditions, GLRT is asymptotically CFAR and thus its threshold can be set for all values of the unknown parameters simultaneously.

GLRT is probably the most popular solution to composite hypothesis testing. It gives a simple recipe that performs well under asymptotic conditions. Its main downsides are that it is sensitive to deviations from its theoretical model, it is generally sub-optimal under finite sample settings and that it may be computationally expensive. Both the nominator and denominator of the GLRT involve optimization problems that may be large scale, non-linear and non-convex. Therefore, there is an ongoing search for robust and low cost alternatives.

## V. Machine Learning for Detection

In this section, we explain the use of machine learning for hypothesis testing. The starting point to any data-driven learning is a training set. Hypothesis testing relies on a probabilistic model $p(\boldsymbol{x};\mathbf{z})$ and we need to use this model in order to generate a synthetic dataset. Hybrid settings involving a mixture of real and artificial samples are also common. For example, it is standard to plant synthetic targets on real noise samples [11]. Due to space limitations, we leave these hybrid extensions for the journal version of this paper.

The main challenge in generating data is that $y \in \{0,1\}$ and $\mathbf{z}$ are not random variables but deterministic parameters without any prior distribution. A natural heuristic is to assume uniform fake priors, e.g., choose half of labels as $y=0$ and half as $y=1$ and assume that $\mathbf{z}$ is uniformly distributed on its corresponding domains. For each $y_i$ and $\mathbf{z}_i$, we then generate a measurement $\boldsymbol{x}_i$ according to the true $p(\boldsymbol{x};\mathbf{z}_i)$ and obtain a synthetic dataset

$$\mathcal{D}_N = \{\boldsymbol{x}_i, \mathbf{z}_i, y_i\}_{i=1}^N. \tag{8}$$

Next, a class of possible detectors $\mathcal{T}$ is chosen in order to tradeoff expressive power with computational complexity in test time. The class is usually a fixed differentiable neural network architecture. In our context, it also makes sense to reuse existing ingredients from classical detector as non-linear features or internal sub-blocks.

Finally, the learned detector is defined as the minimizer of an empirical loss function

$$\min_{\hat{T}\in\mathcal{T}} \frac{1}{N}\sum_{i=1}^{N} L(\hat{T}(\boldsymbol{x}_i), y_i). \tag{9}$$

where $L(\cdot;\cdot)$ is a classification loss function. Ideally, we would like to minimize the zero-one loss which corresponds to the average probability of error. Practically, for efficient optimization, a smooth and convex surrogate loss, as the hinge or cross entropy functions, is preferable. The overall procedure for learning a detector is summarized in Algorithm 1.

**Algorithm 1** Fitting detectors for $p(\boldsymbol{x};\mathbf{z})$

- Choose $p^{\text{fake}}(y)$.
- Choose $p^{\text{fake}}(\mathbf{z};y)$.
- For each $i=1,\cdots,N$:
  - Generate $y_i$.
  - Generate $\mathbf{z}_i$ given $y_i$.
  - Generate $\boldsymbol{x}_i$ given $\mathbf{z}_i$.
- Solve $\min_{\hat{T}\in\mathcal{T}} \frac{1}{N}\sum_{i=1}^{N} L(\hat{T}(\boldsymbol{x}_i), y_i)$.

Learned detectors have strong theoretical guarantees in simple testing problems where $y$ uniquely defines $\mathbf{z}$. With a sufficiently expressive class of detectors and a large enough training set, minimizing the zero-one loss leads to the Bayes optimal detector. It is identical to the LRT with the threshold

$$\gamma = \frac{p^{\text{fake}}(y=0)}{p^{\text{fake}}(y=1)}. \tag{10}$$

Minimizers of Bayes risk consistent surrogates, as the hinge loss, also asymptotically converge to this LRT [23]. Thus, any simple LRT can be approximated by tuning $p^{\text{fake}}(y)$ to achieve a desired $P_{\text{FPR}}$.

To approximate LRT for a wide range of $P_{\text{FPR}}$ it is preferable to minimize the AUC. There are many works in the machine learning literature on fitting classifiers with this goal through an increased computational complexity [13, 14]. There are also variants that focus on partial regimes in the ROC [24].

To evaluate the machine learning approach to detection we turned to numerical experiments. We completed a wide range of simulations in different target detection scenarios comparing the classical solutions with different learned detectors based on various loss functions. We examined both simple and composite settings, with and without hidden variables. The conclusions were that learned detectors usually perform similarly to the their corresponding (G)LRTs. In some experiments, some detectors had a small advantage but the differences were not uniform nor significant. Our conclusion is therefore that simple classifiers with wide priors on the unknown parameters are practically sufficient for achieving an optimal ROC in most problems. On the negative side, most of the experiments showed that the learned detectors were not CFAR and resulted in significantly different false alarm rates for different values of $\mathbf{z} \in \mathcal{Z}_0$. To close this gap, in the next section we propose a framework for learning CFAR detectors.

## VI. Learning CFAR detectors

In this section, we introduce CFARnet, a framework for designing CFAR detectors using machine learning. CFARnet introduces two modifications to Algorithm 1. First, we augment the classification loss with a penalty function that ensures similar distributions for all values of $\mathbf{z} \in \mathcal{H}_0$. Second, in order to optimize this penalty, we generate multiple $\{\boldsymbol{x}_{ij}\}_{j=1}^M$ for each $\mathbf{z}_i$ and use them to empirically approximate the CFAR penalty.

The main idea is adding a penalty to the objective function that promotes a CFAR. The penalty is defined as a sum of distance functions between the distributions of $\hat{T}$ under different values of $\mathbf{z}$:

$$R(\hat{T}) = \sum_{\mathbf{z},\tilde{\mathbf{z}}\in\mathcal{Z}_0} d\left(\hat{T}(\boldsymbol{x});\hat{T}(\tilde{\boldsymbol{x}})\right) \tag{11}$$

where $d(\cdot;\cdot)$ is any statistical distance as detailed in Section II. The distributions of $\hat{T}(\boldsymbol{x})$ and $\hat{T}(\tilde{\boldsymbol{x}})$ are implicitly defined through

$$\begin{aligned} \boldsymbol{x} &\sim p(\boldsymbol{x};\mathbf{z}) \\ \tilde{\boldsymbol{x}} &\sim p(\boldsymbol{x};\tilde{\mathbf{z}}). \end{aligned} \tag{12}$$

Clearly, any CFAR test must satisfy $R(\hat{T})=0$.

To minimize a loss with a CFAR penalty, we need to represent each distribution using a small dataset. For each $\mathbf{z}_i$, we synthetically generate multiple observations $\boldsymbol{x}_{ij}$ for $j=1,\cdots,M$. Similarly, for each $\tilde{\mathbf{z}}_i$ we compute multiple $\tilde{\boldsymbol{x}}_{ij}$. We then plug these into the empirical distances as detailed in Sec. II:

$$\hat{R}(\hat{T}) = \sum_{i,\tilde{i}} \hat{d}\left(\{\hat{T}(\boldsymbol{x}_{ij}\}_{j=1}^M); \{\hat{T}(\tilde{\boldsymbol{x}}_{\tilde{i}j})\}_{j=1}^M\right) \tag{13}$$

Altogether, we recommend to use a hyper-parameter $\alpha > 0$ that trades off the importance of the classification accuracy versus the CFAR penalty. The overall procedure for learning a CFAR detector is summarized in Algorithm 2.

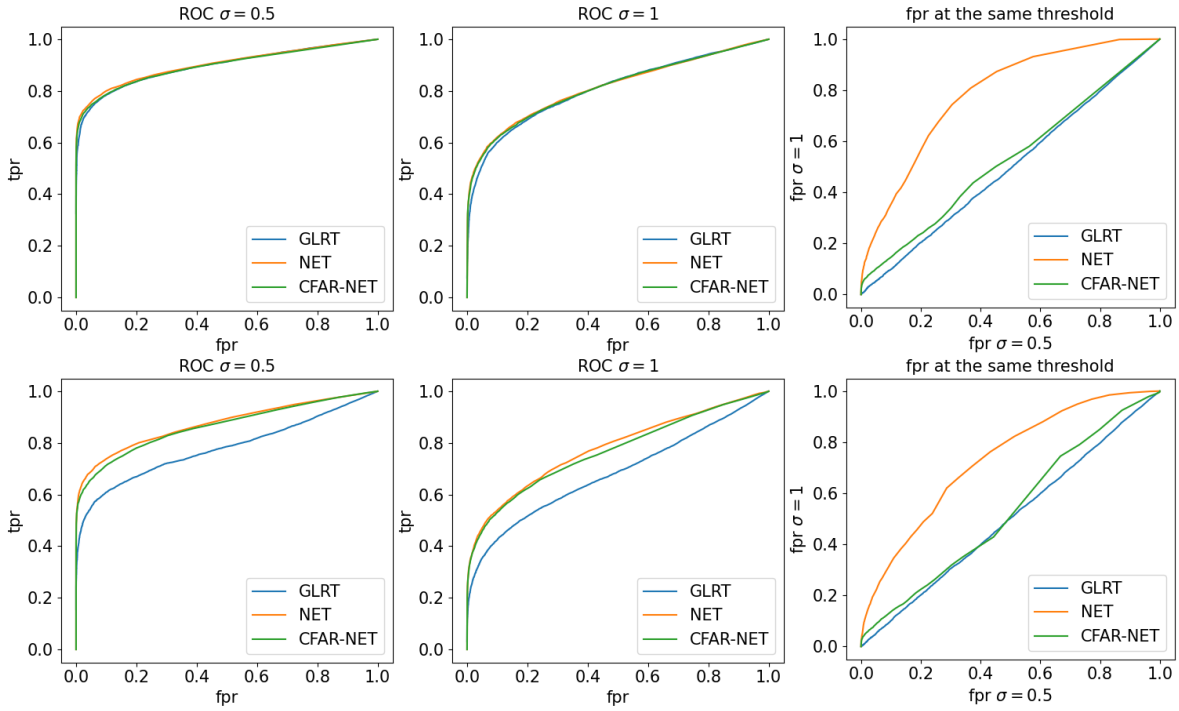


Fig. 1. Performance graphs in terms of FPR, TPR and thresholds for $\sigma \in \{0, 1\}$. Top row is with Gaussian noise and indeed the Gaussian GLRT is best both in terms of ROC and CFAR. NET succeeds in achieving its ROC but is not CFAR. CFAR-NET is both accurate and CFAR. Bottom row is with non-Gaussian noise and the learned detectors beat the Gaussian GLRT in accuracy. CFAR-NET is near CFAR with a slight decrease in accuracy.

**Algorithm 2** Fitting CFAR detectors for $p(\boldsymbol{x}; \mathbf{z})$

- Choose $p^{\text{fake}}(y)$.
- Choose $p^{\text{fake}}(\mathbf{z}; y)$.
- For each $i = 1, \cdots, N$:
  Generate $y_i$.
  Generate $\mathbf{z}_i$ given $y_i$.
  For $j = 1, \cdots, M$:
  Generate $\boldsymbol{x}_{ij}$ given $\mathbf{z}_i$.
- Solve
  $\min_{\hat{T} \in \mathcal{T}} \frac{1}{N} \sum_{i=1}^{N} L(\hat{T}(\boldsymbol{x}_i), y_i) + \alpha \hat{R}(\hat{T})$.

## VII. Numerical Experiments

In this section, we demonstrate the advantages of CFARnet via numerical experiments. We consider a basic yet realistic target detection scenario in which both the target amplitude and the noise scaling are unknown:

$$\boldsymbol{x} = A\mathbf{1} + \sigma \boldsymbol{n} \tag{14}$$

where $\mathbf{1}$ is a target vector of ones, $\boldsymbol{n}$ is a random vector with i.i.d. noise variables, and $\mathbf{z} = [A, \sigma]$ are deterministic unknown parameters

$$-1 \le A \le 1, \qquad 0.5 \le \sigma \le 1. \tag{15}$$

The goal is to decide between

$$\begin{aligned} y = 0: &\quad A = 0 \\ y = 1: &\quad A \neq 0. \end{aligned} \tag{16}$$

We compare three detectors:

- GLRT: assuming Gaussian noise, the classical GLRT has a simple closed form solution $T_{\text{GLRT}} = (\boldsymbol{x}^T \mathbf{1})^2 / (\boldsymbol{x}^T \boldsymbol{x})$ and is known to be CFAR.
- NET: a learned neural network as in Algorithm 1. We choose a uniform fake prior for the unknown parameters in (15). The architecture is based on four non-linear features: the sample mean of $\boldsymbol{x}$, its sample variance and robust versions of the two based on the median. These features are passed through a fully connected neural network, and are optimized to minimize a cross entropy loss using PyTorch.
- CFAR-NET: a learned neural network as in Algorithm 2. Architecture and implementation are all identical to NET. Loss is cross entropy with an MMD CFAR penalty with parameter $\alpha = 1$.

The first experiment considers Gaussian noise. In the first row of Fig. 1, we plot the two ROCs for different values of $\sigma$. To examine the CFAR property, we also plot the FPRs under different parameters. As expected, it is easy to see that the Gaussian GLRT performs well and is CFAR. NET provides similar accuracy as illustrated in its ROC but is non-CFAR and results in significantly different FPR when we change $\sigma$. On the other hand, CFAR-NET is both accurate and near CFAR.

The second experiment is more challenging and considers non-Gaussian noise. The setting is identical as before except for the noise distribution

$$p(n_k) = (1 - \epsilon) N(0, 1) + \epsilon N(0, 100) \tag{17}$$

where $\epsilon = 0.1$. There is no simple GLRT for this setting. The results are provided in the second row of Fig. 1. In this case, the Gaussian GLRT is no longer optimal and the two learned detectors provide a significantly better ROC. In terms of CFAR, GLRT is still invariant to the nuisance parameter, but the FPR of NET is dependent on its value. As promised, CFAR-NET is both accurate and CFAR.

## ACKNOWLEDGMENT

The authors would like to thank Yoav Wald and Yiftach Beer for fruitful discussions and helpful insights in the initial stages of this project.

## REFERENCES

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[2] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 39–56, 2020.

[3] L. Gabrielli, S. Tomassetti, S. Squartini, and C. Zinato, "Introducing deep machine learning for parameter estimation in physical modelling," in *Proceedings of the 20th International Conference on Digital Audio Effects*, 2017.

[4] V. Dua, "An artificial neural network approximation based decomposition approach for parameter estimation of system of ordinary differential equations," *Computers & chemical engineering*, vol. 35, no. 3, 2011.

[5] R. Dreifuerst and R. W. Heath Jr, "SignalNet: A low resolution sinusoid decomposition and estimation network," *arXiv preprint arXiv:2106.05490*, 2021.

[6] T. Diskin, Y. C. Eldar, and A. Wiesel, "Learning to estimate without bias," *Preprint arXiv:2110.12403*, 2021.

[7] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[8] L. Girard, V. Roy, P. Giguère, and T. Eude, "Deep neural network training using synthetic signatures for rare target detection in SWIR hyperspectral imagery," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pp. 4420–4423, IEEE, 2021.

[9] A. Brighente, F. Formaggio, G. M. Di Nunzio, and S. Tomasin, "Machine learning for in-region location verification in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2490–2502, 2019.

[10] D. de la Mata-Moya, M. P. Jarabo-Amores, J. M. de Nicolás, and M. Rosa-Zurera, "Approximating the Neyman–Pearson detector with 2C-SVMs. application to radar detection," *Signal Processing*, vol. 131, pp. 364–375, 2017.

[11] A. Ziemann, M. Kucer, and J. Theiler, "A machine learning approach to hyperspectral detection of solid targets," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXIV*, vol. 10644, p. 1064404, International Society for Optics and Photonics, 2018.

[12] J. Theiler, S. Matteoli, and A. Ziemann, "Bayesian detection of solid subpixel targets," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pp. 3213–3216, IEEE, 2021.

[13] A. Herschtal and B. Raskutti, "Optimising area under the ROC curve using gradient descent," in *Proceedings of the twenty-first international conference on Machine learning*, p. 49, 2004.

[14] U. Brefeld, T. Scheffer, *et al.*, "AUC maximizing support vector learning," in *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning*, 2005.

[15] C.-H. Lin, Y.-C. Lin, Y. Bai, W.-H. Chung, T.-S. Lee, and H. Huttunen, "DL-CFAR: A novel cfar target detection method based on deep learning," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1–6, IEEE, 2019.

[16] J. Akhtar and K. E. Olsen, "A neural network target detector with partial CA-CFAR supervised training," in *2018 International Conference on Radar (RADAR)*, pp. 1–6, IEEE, 2018.

[17] J. Akhtar, "Training of neural network target detectors mentored by SO-CFAR," in *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 1522–1526, IEEE, 2021.

[18] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[19] Y. Wald, A. Feder, D. Greenfeld, and U. Shalit, "On calibration and out-of-domain generalization," *arXiv preprint arXiv:2102.10395*, 2021.

[20] Y. Romano, S. Bates, and E. Candes, "Achieving equalized odds by resampling sensitive attributes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 361–371, 2020.

[21] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[22] S. Kay, *Fundamentals of Statistical Signal Processing: Detection theory*. Fundamentals of Statistical Si, Prentice-Hall PTR, 1998.

[23] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[24] H. Narasimhan and S. Agarwal, "A structural SVM based approach for optimizing partial AUC," in *International Conference on Machine Learning*, pp. 516–524, PMLR, 2013.