Naval Research Laboratory
Washington, DC 20375-5320

NRL/5340/MR—2023/5

Neural Network Constant False Alarm Rate Detection

KEVIN WAGNER

HUY VINH LE

Advanced Radar Systems Branch
Radar Division

July 25, 2023

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
25-07-2023	NRL Memorandum Report	October 1, 2019 – September 30, 2023

4. TITLE AND SUBTITLE

Neural Network Constant False Alarm Rate Detection

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62114N

6. AUTHOR(S)

Kevin Wagner and Huy Vinh Le

5d. PROJECT NUMBER

5e. TASK NUMBER
EW-114-013

5f. WORK UNIT NUMBER
6B72

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

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/5340/MR--2023/5

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
One Liberty Center
875 N. Randolph Street, Suite 1425
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)

ONR via NRL base program

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

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

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This document presents an approach to augmenting and improving cell averaging constant false alarm detectors with a neural network based approach. The simulated results show that the neural network is able to perform as well as the optimal detector and remove the losses typically associated with estimating the noise power in standard approaches. A feasible implementation of the neural network detector is also presented. Preliminary results demonstrate that the neural network requires the same inputs as the cell averaging constant false alarm rate detector and can work across a range of unknown input signal to noise ratios.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Kevin Wagner
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	U	12	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			(202) 404-4760

This page intentionally left blank.

CONTENTS

FIGURES

EXECUTIVE SUMMARY

This document presents an approach to augmenting and improving cell averaging constant false alarm detectors with a neural network based approach. The simulated results show that the neural network is able to perform as well as the optimal detector and remove the losses typically associated with estimating the noise power in standard approaches. A feasible implementation of the neural network detector is also presented. Preliminary results demonstrate that the neural network requires the same inputs as the cell averaging constant false alarm rate detector and can work across a range of unknown input signal to noise ratios.

This page intentionally left blank

NEURAL NETWORK CONSTANT FALSE ALARM RATE DETECTION

1. INTRODUCTION

In this work, we demonstrate the ability of a two-layer feed-forward neural network to perform constant false alarm rate detection of a point target in Gaussian noise. In particular, the results show that with this architecture, it is possible to match the predicted optimal detection performance [1] without knowing the input signal to noise ratio. This result suggests that the neural network is able to learn the same behavior as the optimal detector.

The optimal detector requires knowledge of the signal to noise ratio to set the threshold to meet a predefined false alarm rate. In practice, the signal to noise ratio is not known a priori and therefore approaches such as the cell-averaging constant false alarm rate (CA-CFAR) [2] are used. These approaches attempt to estimate the signal to noise ratio by estimating the noise in a region near the cell under test (CUT) and forming a ratio of the CUT power to the local noise estimate. For example, L cells prior to and after the CUT are used to estimate the noise power.

The size of L is typically chosen under two competing interests. The first interest, leads to large values of L to ensure a good statistical estimate of the noise. The second interest, pushes L to be small to avoid corrupting the noise estimate with neighboring targets. Due to the finite size of L, detectors such as the CA-CFAR and other variants suffer a performance loss relative to the optimal detector. This is commonly referred to as the CFAR loss [3]. The CFAR loss is a function of the number of training cells L. For instance, it is known that at probability of detections of 0.9 and probability of false alarm of 10^{-6} with twenty training samples, the CFAR loss is approximately 1.5 dB relative to the optimal detector [3].

In summary, we develop and present a practical implementation of a neural network architecture which can directly replace the CA-CFAR and provide near optimal detection performance for a given false alarm rate under a specific interference and radar scattering model. Thus, the CFAR loss is effectively removed without knowing the input SNR which is typically required to achieve the optimal detection probability.

2. RANGE-ONLY SIGNAL PROCESSING

For simplicity, we consider the one-dimensional range-only processing case. These methods can be extended to the two-dimensional range and Doppler case. In the one-dimensional range-only case, the data can treated as a complex valued time series.

Let us assume we have $2L + 1$ range-only samples. Let x_i where $i \in \{1, 2, \ldots, 2L + 1\}$ represent the i^{th} sample. The cell under test is assumed to be the center sample given by $x_{\text{CUT}} = x_{L+1}$. The sliding window for processing range-only samples is depicted in Figure 1. The boxes with blue diagonal lines denote an arbitrary number of guard cells. Guard cells are typically used to ensure that range side-lobes of the returned signal are not present in algorithms that estimate the noise power.

Manuscript approved July 24, 2023.

Fig. 1—Sliding Window Example

3. HYPOTHESIS TESTING

We begin by framing the radar detection problem as form of hypothesis testing [3]. For any cell under test measurement, x_{CUT}, there only exists two hypotheses $y \in \{0, 1\}$ where:

- $y = 0$ implies the null hypothesis $\mathcal{H}_0$: x_{CUT} is drawn from interference only
- $y = 1$ implies the alternative hypothesis $\mathcal{H}_1$: x_{CUT} consists of interference plus a target echo

3.1 Interference Only

It is assumed that the samples x_i have been passed through a square law detector. For all $i \in \{1, 2, \ldots, 2L+1\}$, the inputs to the square law detector are i.i.d with real and imaginary parts given by, w^R and w^I, respectively. It is assumed that w^R and w^I are both zero mean Gaussian random variables with variance σ_w^2. Then the input to the square law detector can be modeled as

$$x_i = \frac{1}{\sqrt{2}}(w^R + jw^I). \tag{1}$$

3.2 Interference plus Target: Swerling 0

In the Swerling 0 case [4], the target present hypothesis, $\mathcal{H}_1$, assumes the cell under test input to the square law detector has the following form:

$$x_{\text{CUT}} = \frac{1}{\sqrt{2}}(w^R + jw^I) + \alpha_{\text{SNR}} e^{j2\pi\phi} \tag{2}$$

where α_{SNR} is the signal to noise ratio of the input signal and ϕ is the phase of the input signal. It is assumed that the phase is unknown and is drawn from a uniform distribution in the interval $[0, 2\pi]$. It is assumed that the samples $x_i \; \forall \; i \in \{1, 2, \ldots, 2L+1 | i \neq L+1\}$ are given by equation (1).

4. NEURAL NETWORK TRAINING AND TESTING

4.1 Neural Network Data Generation

The methodology used to train and test the neural networks is presented next. For each SNR, an equal number of target absent and target present range-only data samples are generated. Let $\mathbf{x}^m_{\mathcal{H}_0}(\alpha_{\text{SNR}}) = [x_1, x_2, \ldots, x_{2L+1}]$ represent the m^{th} random draw under the null hypothesis, $\mathcal{H}_0$, for $m \in \{1, 2, \ldots, M\}$ and for a given SNR, α_{SNR}. Note the null hypothesis is not a function of the signal SNR and this notation is adopted to track the corresponding null samples for a given SNR.

In similar fashion, let $\mathbf{x}^m_{\mathcal{H}_1}(\alpha_{\text{SNR}}) = [x_1, x_2, \ldots, x_{2L+1}]$ represent the m^{th} random draw under the alternative hypothesis, $\mathcal{H}_1$, for $m \in \{1, 2, \ldots, M\}$ and for a given SNR. Balanced data sets are used for the training, validation, and testing phases.

Next, the M null and M alternative hypothesis samples can be concatenated into complex valued matrices of size $M \times 2L + 1$ samples, which are represented by $\mathbf{X}_{\mathcal{H}_0}(\alpha_{\text{SNR}}) \in \mathbb{C}^{M\times 2L+1}$ and $\mathbf{X}_{\mathcal{H}_1}(\alpha_{\text{SNR}}) \in \mathbb{C}^{M\times 2L+1}$, respectively.

Next, let P represent the total number of SNRs under consideration. We assume that the SNRs have been discretized such that

$$\alpha_{\text{SNR}} = (p-1)\delta_\alpha + \alpha_{min} \forall\ p \in \{1, 2, \ldots, P\} \tag{3}$$

where α_{min} and α_{max} represent a minimum and maximum SNR the neural network is to be trained on and δ_α is the SNR step size given by

$$\delta_\alpha = \frac{\alpha_{max} - \alpha_{min}}{P}. \tag{4}$$

For the P signal to noise ratios of interest, we can concatenate the matrices $\mathbf{X}_{\mathcal{H}_0}(\alpha_{\text{SNR}})$ and $\mathbf{X}_{\mathcal{H}_1}(\alpha_{\text{SNR}})$ along the first dimension to form the matrices

$$\begin{aligned}\mathbb{X}_{\mathcal{H}_0} &\in \mathbb{C}^{MP\times 2L+1}\\ \mathbb{X}_{\mathcal{H}_1} &\in \mathbb{C}^{MP\times 2L+1}.\end{aligned}$$

The labels for the supervised learning approach are given by

$$\begin{aligned}\mathbb{Y}_{\mathcal{H}_0} &= \mathbf{0}^{MP\times 1}\\ \mathbb{Y}_{\mathcal{H}_1} &= \mathbf{1}^{MP\times 1}.\end{aligned}$$

Finally, the regressors and labels given to the neural network for both the null and alternative hypothesis are stacked to form

$$\begin{aligned}\mathbb{X} &= [\mathbb{X}_{\mathcal{H}_0}; \mathbb{X}_{\mathcal{H}_1}] \in \mathbb{C}^{2MP\times 2L+1}\\ \mathbb{Y} &= [\mathbb{Y}_{\mathcal{H}_0}; \mathbb{Y}_{\mathcal{H}_1}] \in \mathbb{R}^{2MP\times 1}.\end{aligned}$$

4.2 Neural Network Architecture and Training

The neural network architecture used in this work is given in Table 1. Note that the focus of this work is on the feasibility of the approach and therefore hyperparameter optimization is not performed. The sizes used in the input layer correspond to typical neural network architectures which expect images and correspond to the following dimensions: (image height, image width, image channels, sample number). Therefore, the first

layer has a size of $2L + 1 \times 1 \times 2 \times 1$ consisting of the $2L + 1$ complex-valued range samples. The real and imaginary parts of the match filter output are mapped to separate image channels. Next, a fully connected layer with 1024 nodes, followed by a rectified linear unit layer, followed by a second fully connected layer with 256 nodes, and a rectified linear unit layer are used. Next, a fully connected layer with 2 nodes is used, followed by a softmax layer.

Layer Name	Number	Size
Input	1	$(2L + 1) \times 1 \times 2 \times 1$
Fully Connected	2	$2(2L + 1) \times 1024$
Relu	3	
Fully Connected	4	1024×256
Relu	5	
Fully Connected	6	256×2
Softmax	7	2×1

Table 1—Neural Network Architecture

Next, the neural network is trained with a 90% -10% split in training and validation data on the input 2MP regressors and labels. During the training and validation split process, care is taken to ensure that the data sets remain balanced in both the training and validation data across all SNRs.

Next, the neural network is trained using stochastic gradient descent with momentum [5] and a cross-entropy metric [6].

4.3 Threshold Selection for Desired False Alarm Rate

For any given input sample of size $2L + 1 \times 1 \times 2 \times 1$, the neural network's final softmax layer outputs a two dimensional vector of class scores which sum to 1. The class scores are represented here by $[(1 - s), s)]$. The first dimension of the class scores, $(1 - s)$, can be interpreted as the likelihood of the null hypothesis, and the second dimension, s, can be interpreted as the likelihood of the alternative hypothesis. During the training process, the scores have been tuned to give equal weight to both false alarms and missed detections. This is the outcome of using balanced data sets and using cross-entropy loss. The value of s can be used to make an estimate of labels, $\hat{y} \in \{0, 1\}$, for any input sample. For example, a common threshold used to make this decision is

$$\hat{y} = 0 \text{ if } s < 0.5$$
$$\hat{y} = 1 \; o.w.$$

In order to find a threshold which will result in the desired false alarm rate, a receiver operating characteristic (ROC) curve [7] can be calculated using the training data and known labels. In this implementation, the value s is used to explicitly calculate the threshold value for various false alarm rates.

A pseudo algorithm for this process in shown in Algorithm 1. The algorithm begins by choosing a desired false alarm rate P_{FA}. Next, the output threshold, T, is initially set to zero and a threshold step size is selected,

$\delta_T > 0$. The algorithm proceeds by empirically calculating the number of false alarms and estimates the current false alarm rate as given by

$$\hat{P}_{FA} = \frac{\sum_{\alpha_{\text{SNR}}} \sum_{m}^{M} \delta_{\hat{y}_m=1} \; \forall \; \mathbf{x}_{\mathcal{H}_0}^{m}(\alpha_{\text{SNR}})}{MP}. \tag{5}$$

where $\delta_{\hat{y}_m=1} = 1$ if $\hat{y}_m = 1$ and zero otherwise.

The empirical probability of false alarm shown in (5) is the ratio of the number of flagged detections from the set of regressors $\mathbb{X}_{\mathcal{H}_0}$ relative to the total number of samples in this set MP. Next, the threshold is incremented until the empirically estimated false alarm rate is less than the desired false alarm rate. At which point, the current threshold is returned.

Algorithm 1 Algorithm for Empirically Calculating Threshold

```
Select P_FA
Initialize Threshold T = 0
Compute Empirical P̂_FA
while Empirical P̂_FA > P_FA do
    Set T = T + δT
    Compute Empirical P̂_FA
end while
return T
```

5. PERFORMANCE COMPARISON ALGORITHMS AND METRICS

5.1 Algorithms

5.1.1 CA-CFAR Detector

As a baseline, the performance of a CA-CFAR is used as a comparison against the performance of the neural network. The input to the CA-CFAR algorithm are the complex samples given in (2) after they had been passed through a square law detector $|x|^2$.

The cell-averaging CFAR was introduced by Finn and Johnson [8] in 1968. The algorithm behind the CA-CFAR is simple in that it computes a threshold based on an estimate of the average interference power in the reference window. The CA-CFAR is designed to operate in a target and interference environment with the following assumptions:

1. The interference in the leading windows, lagging windows, and in the CUT are i.i.d.

2. With a target return present in the CUT, the leading and lagging windows do not contain returns from other targets that bias the threshold estimate

In this processor, the composite interference statistic, $\hat{g}$ is calculated by taking the average over the measurements in the reference cells. The composite interference statistic is given by equation (6) [1]:

$$\hat{g} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{6}$$

where $N = 2L$ is the total number of reference cells and x_i is the sample in each training cell.

The CFAR constant α, for a CA-CFAR, is given by equation (7) [1]:

$$\alpha = N[P_{FA}^{-1/N} - 1] \tag{7}$$

where P_{FA} is the desired false alarm rate.

Using the CFAR constant, α, and the average of both the leading and lagging samples, $\hat{g}$, the detection threshold, T, is given by equation (8) [1]:

$$T = \alpha \hat{g} \tag{8}$$

After estimating the threshold, a detection decision, $\mathcal{H}_j$ for $j \in \{0, 1\}$, can be made for the CUT.

$$\mathcal{H}_j = \begin{cases} j = 1 & \text{if } CUT \geq T, \\ j = 0 & \text{if } CUT < T \end{cases} \tag{9}$$

5.1.2 Optimal Detector

If the SNR is known a priori, the closed form representation of the probability of detection P_D as a function of the probability of false alarm P_{FA} is given by

$$P_D = Q_M \left(\sqrt{2\alpha_{\text{SNR}}}, \sqrt{-2 \ln(P_{FA})} \right) \tag{10}$$

where Q_M is Marcum's Q function [3]. This formulation is specifically for the Swerling 0 case.

5.1.3 Neural Network CFAR

The neural network CFAR detector (NN-CFAR) is described in Chapter 4.

5.2 Metrics

In order to compare the algorithms, ROC curves are generated and plotted. In particular, for a set of SNRs or for a given SNR, we plot the P_D vs. P_{FA} when using the test data. Two methods are used to generate the ROC curves.

In the first method, the test labels are used to empirically calculate the number of false alarms and detections as a function of a variable threshold.

In the second method, the training data is used in conjunction with Algorithm 1 to estimate a threshold for a particular false alarm rate for both the CA-CFAR and NN-CFAR algorithms.

The first method is commonly used to compare two classifiers for a given test data set. The second method is a practical implementation of the CA-CFAR and NN-CFAR. In both cases, the classifiers have no a prior knowledge of the test labels.

6. SIMULATION RESULTS

In this section, we present simulation results for the Swerling 0 case for three different algorithms: the CA-CFAR, NN-CFAR and optimal detector.

In this simulation, we chose a total of $P = 10$ SNRs to train on with an SNR step size of $\delta_\alpha = 1$. The minimum and maximum SNRs are given by $\alpha_{min} = 6$ and $\alpha_{max} = 15$, respectively. There is a total of $M = 100,000$ null hypotheses and $M = 100,000$ alternative hypotheses per SNR. This results in a total of $2MP = 2,000,000$ training and validation samples. These are split, 90%-10%, into training and validation samples.

A separate set of test data is subsequently generated with $M_{test} = 1,000,000$ null hypotheses and $M_{test} = 1,000,000$ alternative hypotheses per SNR.

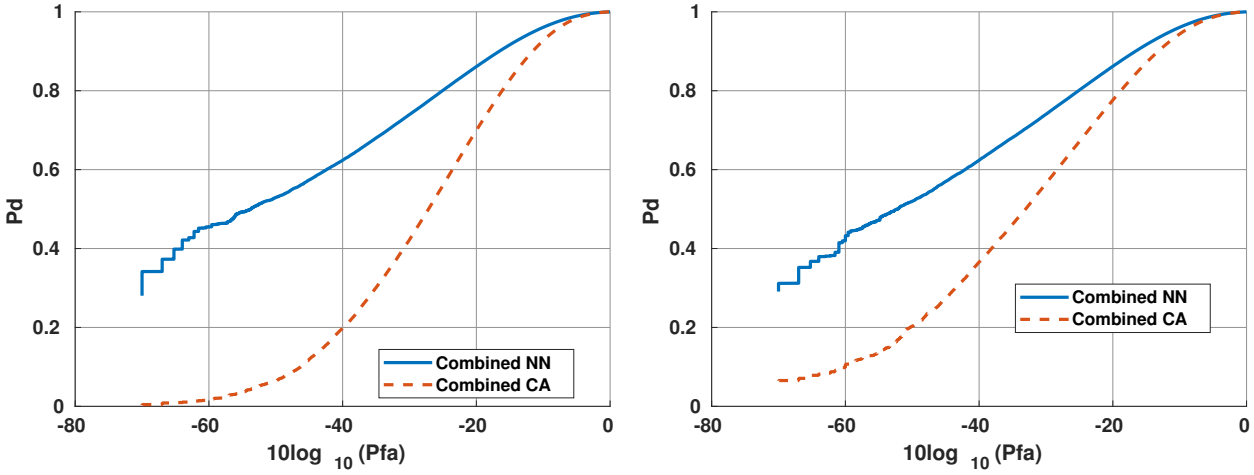
Two neural networks are trained for separate values of $L = 5$ and $L = 10$ across the ten SNRs. The results are presented next.

In Figure 2a and Figure 2b, the CA-CFAR and NN-CFAR ROC curves are plotted when using the test data across all SNRs, for $L = 5$ and $L = 10$, respectively. The CA-CFAR is plotted as a red dashed line and denoted as "Combined CA" in the legend. Likewise, the NN-CFAR is plotted as a blue line and denoted by "Combined NN". The word "Combined" references the fact that all SNRs are used in the test data simultaneously to assess performance. Subsequently, performance for each individual SNR will be presented. Here, we can see that in both cases, the neural network has a higher probability of detection across all probabilities of false alarm, indicating that the NN-CFAR has superior detection performance.

In similar fashion, we plot the ROC curves for each SNR value: SNR = 6 dB in Figure 3, SNR = 7 dB in Figure 4, SNR = 8 dB in Figure 5, SNR = 9 dB in Figure 6, SNR = 10 dB in Figure 7, SNR = 11 dB in Figure 8 , SNR = 12 dB in Figure 9, SNR = 13 dB in Figure 10, SNR = 14 dB in Figure 11, and SNR = 15 dB in Figure 12.

In each of these figures, the optimal performance is plotted in yellow, the NN-CFAR's ROC evaluation is plotted in blue, the CA-CFAR's ROC evaluation is plotted in a red dashed line, the NN-CFAR with pre-computed threshold from the training data is purple with "x" markers, and the CA-CFAR with pre-computed threshold is green with "x" markers.

First, we note that the two methods for computing the ROC's performance results in similar performance for both the CA-CFAR and NN-CFAR, indicating that our SNR specific thresholds generalize across all our SNRs of interest. Secondly, we note that the NN-CFAR achieves near optimal performance across all our SNRs of interest. There are some divergence between the optimal theory and the NN-CFAR. However, this is likely due to insufficient sample support at smaller false alarm rates. Recall that the test data set used $M = 1,000,000$ null hypotheses and $M = 1,000,000$ alternative hypotheses for each SNR. This value of M would result in only a single false alarm at a value of $P_{FA} = 10^{-6}$, making assessing the performance here statistically unreliable [9]. The fact that the NN-CFAR outperforms the optimal detector in Figure 7a warrants further understanding, but it is suspected that with increased samples at this SNR, the two curves will overlap.



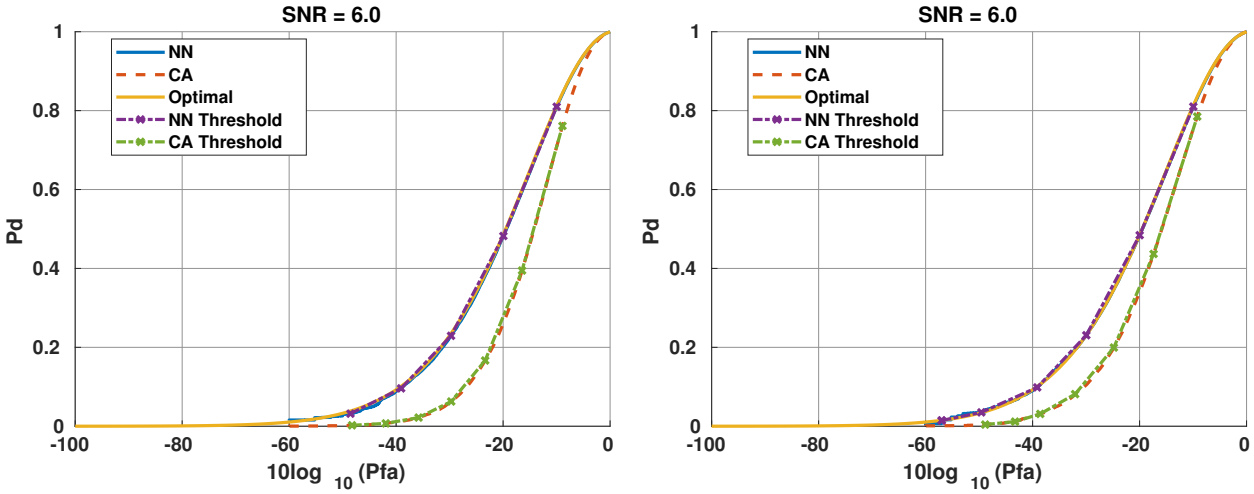
(a) $L = 5$ (b) $L = 10$

Fig. 2—ROC Comparison: All SNRs

7. CONCLUSION

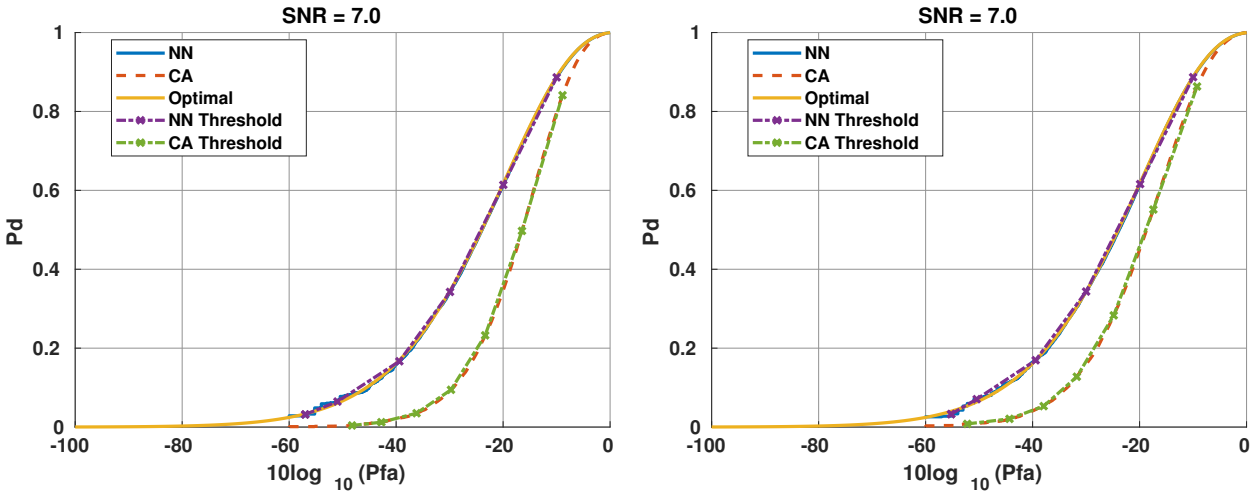
In this work, a neural network CFAR detector is proposed and compared to the optimal theory as well as the common CA-CFAR algorithm. It is shown via simulation that the NN-CFAR is able to achieve the optimal predicted performance without express knowledge of the signal to noise ratio and offers increased detection performance at a given SNR and probability of false alarm relative to the standard CA-CFAR algorithm. This result can be interpreted as the NN-CFAR's ability to overcome the noise estimate loss and reduce the CFAR loss of the CA-CFAR algorithm.

While these preliminary results are promising, further research is required to understand the current results and extend them to typical operating scenarios. For instance, generalizing this approach to other scattering phenomenon and clutter scenarios is warranted. Ideally, a single NN-CFAR capable of working in a wide variety of interference and signal return environments is desired. The ability to operate in the range-Doppler domain and range-Doppler-spatial domain is also desired. Generalizing to the case when a secondary target



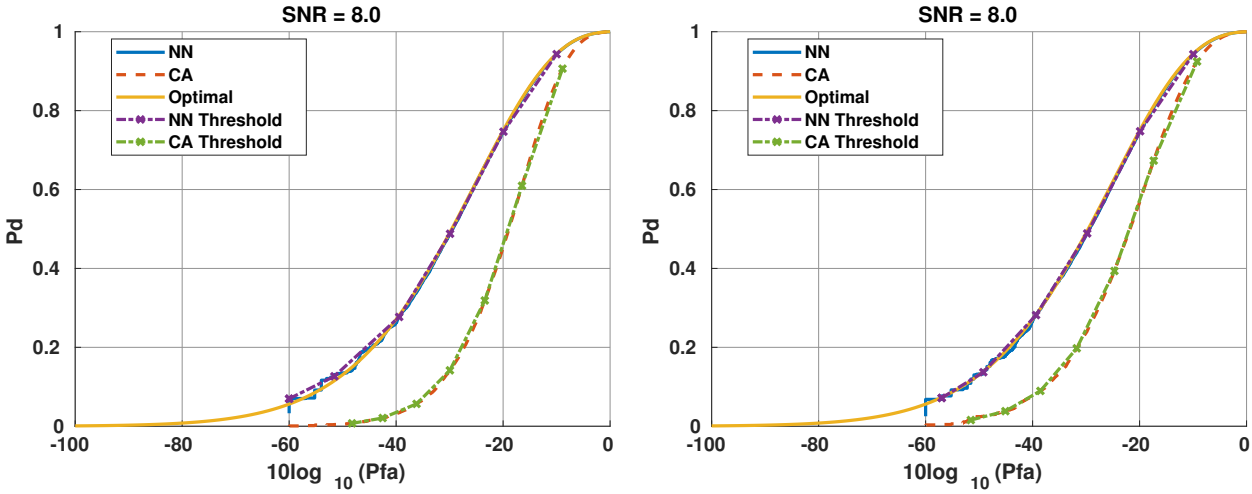
(a) $L = 5$ (b) $L = 10$

Fig. 3—ROC Comparison: SNR = 6 dB



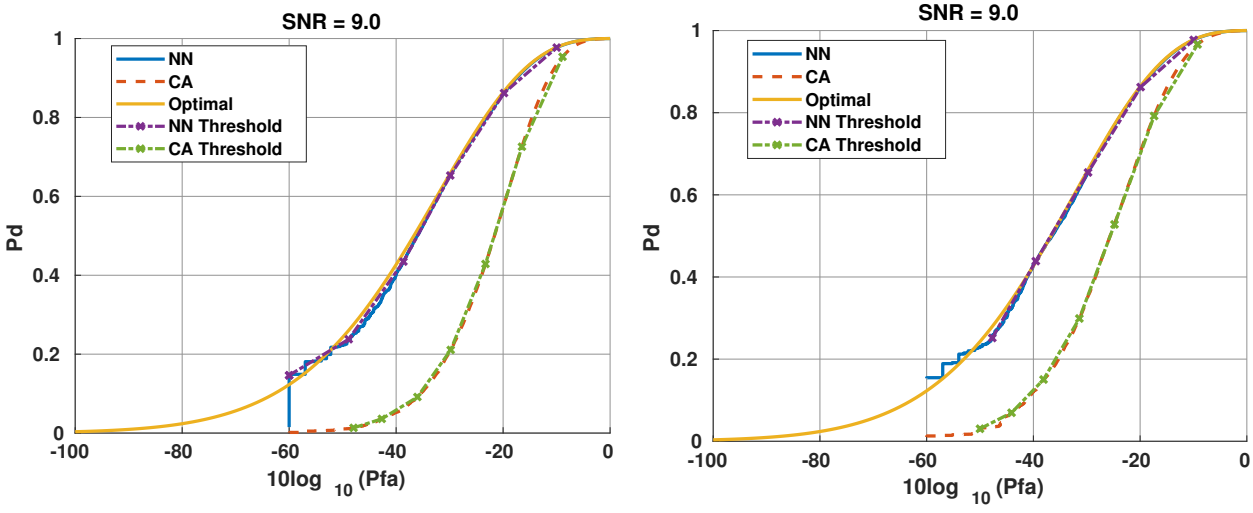
(a) $L = 5$ (b) $L = 10$

Fig. 4—ROC Comparison: SNR = 7 dB



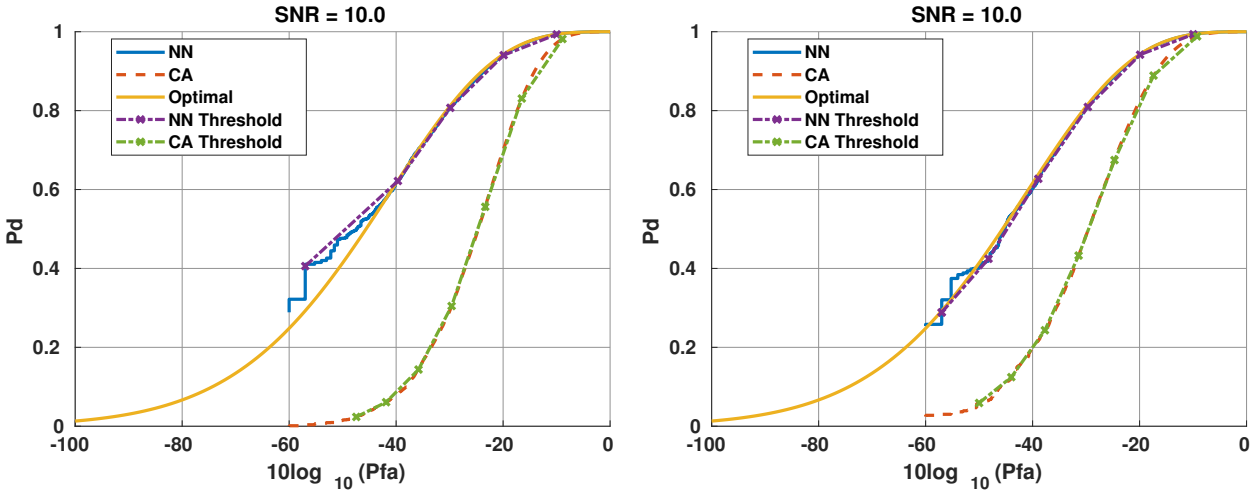
(a) $L = 5$ (b) $L = 10$

Fig. 5—ROC Comparison: SNR = 8 dB



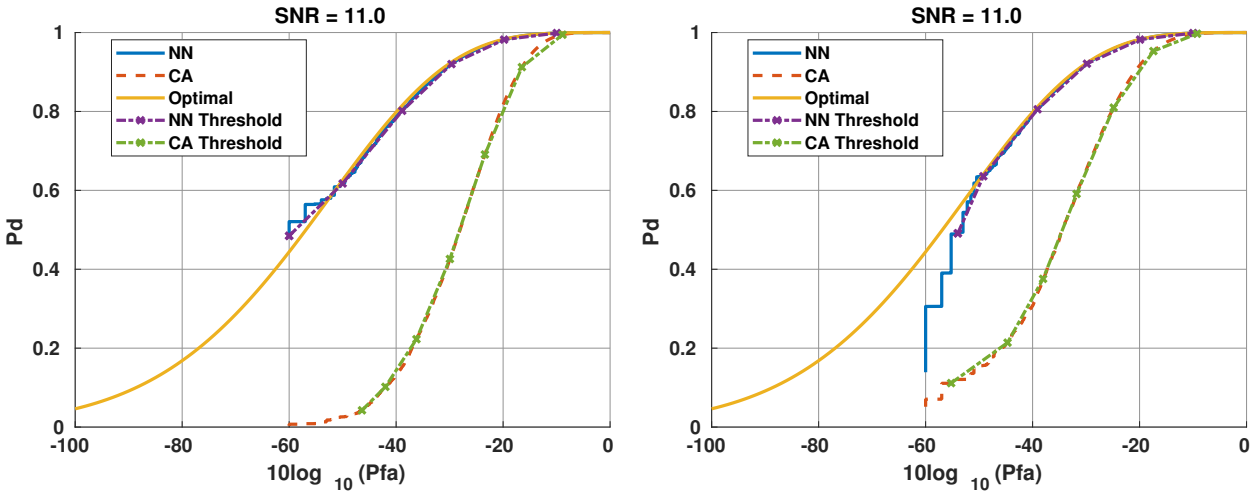
(a) $L = 5$ (b) $L = 10$

Fig. 6—ROC Comparison: SNR = 9 dB



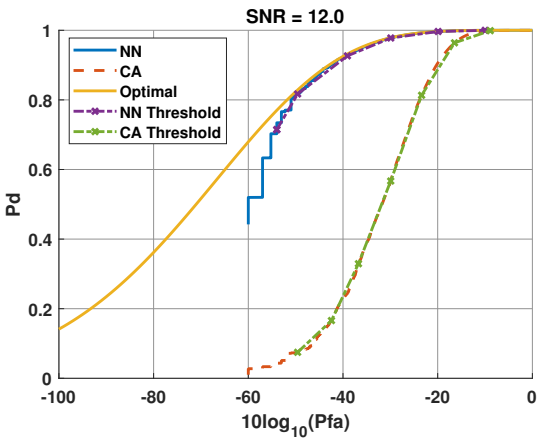
(a) $L = 5$ (b) $L = 10$

Fig. 7—ROC Comparison: SNR = 10 dB

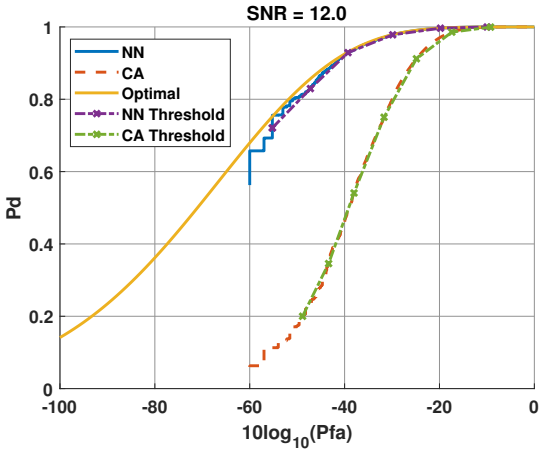


(a) $L = 5$ (b) $L = 10$

Fig. 8—ROC Comparison: SNR = 11 dB

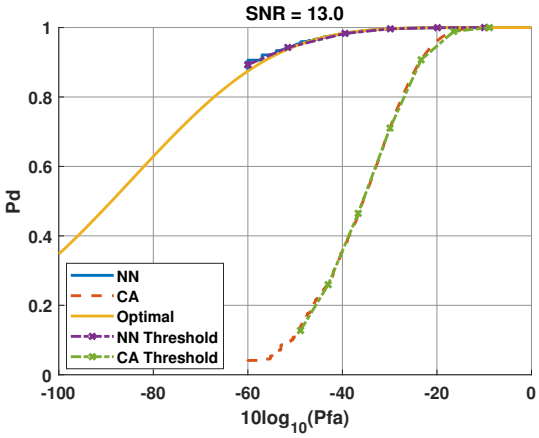


(a) $L = 5$

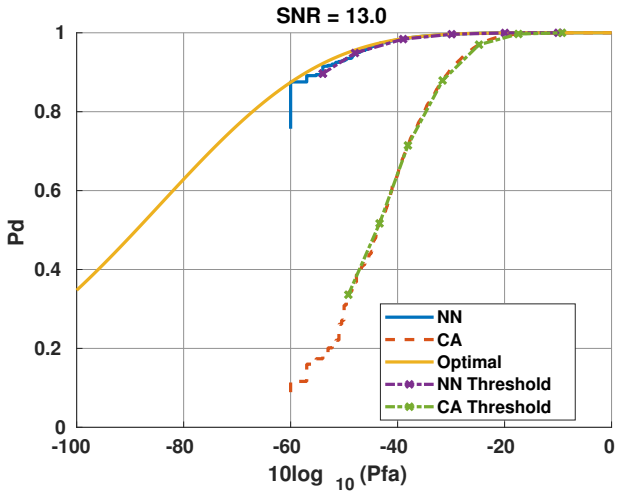


(b) $L = 10$

Fig. 9—ROC Comparison: SNR = 12 dB

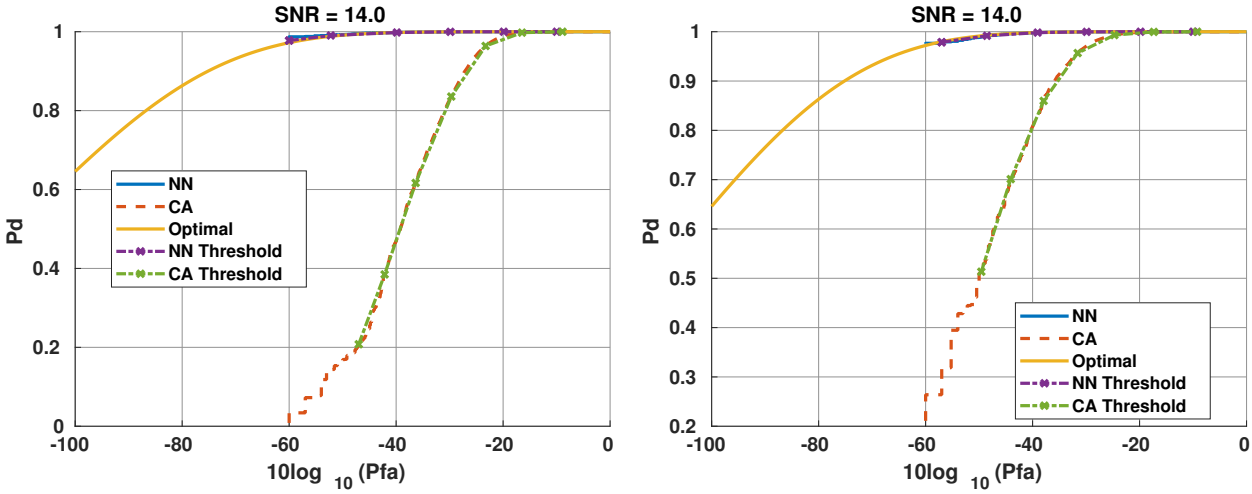


(a) $L = 5$



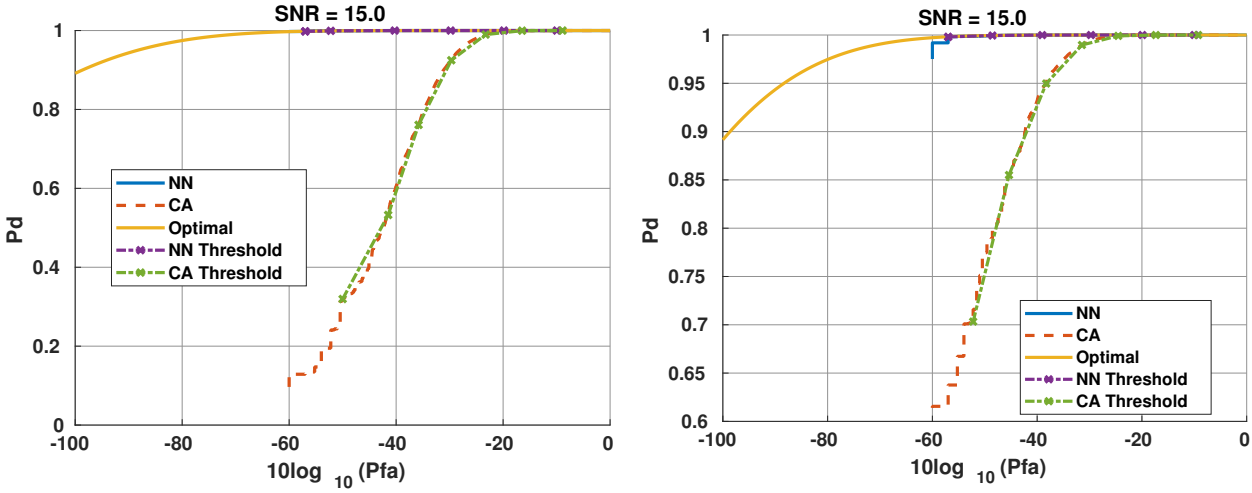
(b) $L = 10$

Fig. 10—ROC Comparison: SNR = 13 dB



(a) $L = 5$ (b) $L = 10$

Fig. 11—ROC Comparison: SNR = 14 dB



(a) $L = 5$ (b) $L = 10$

Fig. 12—ROC Comparison: SNR = 15 dB

is in the training cells would also be an area of future research. Additionally, performing hyper parameter optimization and the impacts of reducing the number of training cells, L is also an area of future research.

ACKNOWLEDGMENTS

The author thanks the Office of Naval Research for funding this work via the NRL base program.

REFERENCES

1. M. Richards, J. Scheer, and W. Holm, *Principles of Modern Radar: Basic principles, Volume 1*, Electromagnetics and Radar (Institution of Engineering and Technology, 2010), ISBN 9781891121524. URL https://books.google.com/books?id=TrTQxwEACAAJ.

2. P. Gandhi and S. Kassam, "Analysis of CFAR processors in nonhomogeneous background," *IEEE Transactions on Aerospace and Electronic Systems* **24**(4), 427–445 (1988), doi:10.1109/7.7185.

3. M. A. Richards, *Fundamentals of Radar Signal Processing* (McGraw-Hill Professional, <country>US</country>, 2005), ISBN 0071444742, doi:10.1036/0071444742. URL https://mhebooklibrary.com/doi/book/10.1036/0071444742.

4. P. Swerling, "Probability of detection for fluctuating targets," *IRE Transactions on Information Theory* **6**(2), 269–308 (1960), doi:10.1109/TIT.1960.1057561.

5. K. P. Murphy, *Machine learning : a probabilistic perspective* (MIT Press, Cambridge, Mass. [u.a.], 2013), ISBN 9780262018029 0262018020. URL https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2.

6. C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1 ed. (Springer, 2007), ISBN 0387310738. URL http://www.amazon.com/Pattern-Recognition-Learning-Information-Statistics/dp/0387310738%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0387310738.

7. J. Davis and M. Goadrich, "The Relationship between Precision-Recall and ROC Curves," Proceedings of the Proceedings of the 23rd International Conference on Machine Learning, ICML '06, New York, NY, USA (Association for Computing Machinery), 2006, p. 233–240. ISBN 1595933832, doi:10.1145/1143844.1143874. URL https://doi.org/10.1145/1143844.1143874.

8. H. Finn and S. Johnson, "Adaptive detection mode with threshold control as a function of spatially sampled clutter level estimates," *RCA Review* **29**, 414–464 (1968).

9. F. Dekking, *A Modern Introduction to Probability and Statistics: Understanding Why and How*, Springer Texts in Statistics (Springer, 2005), ISBN 9781852338961. URL https://books.google.com/books?id=XLUMIlombgQC.