Adversarial ML Radar Literature

Dr. Uttam K. Majumder

Email: Uttam.Majumder@ieee.org

Approved for Public Release, NGA-U-2024-09458
Nov. 05, 2024.

Outline

PART 1: Understanding AI/ML from Sufficient Statistics

- Closed form solution
- Convex function
- Maximum likelihood estimation (MLE)
- Maximum likelihood classification (MLC)
- Maximum a priori (MAP) estimation
- Near closed form solution

PART 2: Adversarial AI/ML Experimentations and Some Results

- Relevant literature
- Adversarial attacks literature
- Adversarial noise generation literature
- AI/ML models
- Adversarial training
- SAR datasets
- MSTAR data standard operating condition
- CVDome standard operating condition
- Robustness: adversarial noise, extending operating condition, phase errors

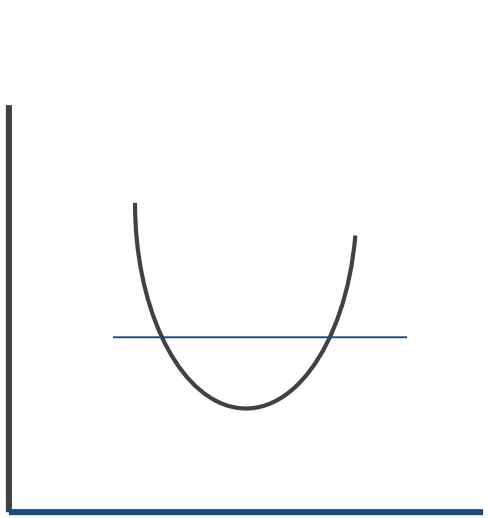
Closed Form Solution

- A closed form solution is an expression for an exact solution with only a finite amount of data.
- Example:
  - Sum of 1, 2, 3, ….n has the closed form solution n(n+1)/2

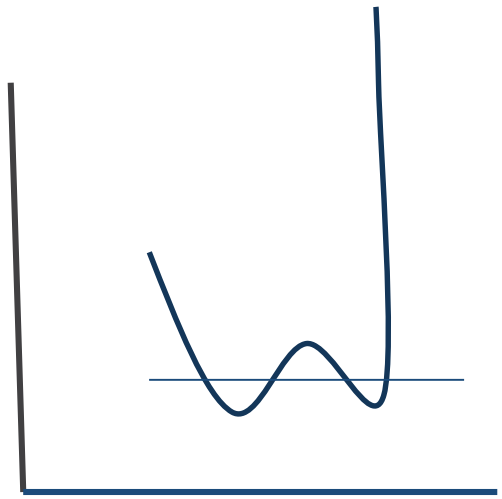
- One Key Question is:
  - What the “big deal” is whether we have closed form solutions or we solve a given problem numerically.
  - ► For many problems, if there is a closed form solution, the time it takes to get the result might be 1/10th (or even 1/100th) of numerical solution

PR# NGA-U-2023-01021

Convex Function



Convex



Non-Convex

This Figures are unclassified; drawn by Author (Uttam Majumder)

- Neural network loss functions are in general non-convex in parameter space.
- No global optimum solution!

PR# NGA-U-2023-01021

Deep Neural Network (DNN) and Non-closed Form Solution

- DNN based classification solution will never be the exact solutions

- However, many of the non-convex functions provide approximate solutions
- **Key Technical Challenge:**
    - **How to obtain “Near Closed Form Solution” or “Consistent Classification Accuracy”**
- Does adding “More Samples” achieves better accuracy?

PR# NGA-U-2023-01021

Maximum Likelihood Classification

- Maximum Likelihood Classification (MLC) is derived from MLE concept

Consider $S = \{C_1, C_2, C_3, \ldots . C_N\}$ be N samples of training class data

The **maximum likelihood Classification (MLC)** of $Class\ C_1$ ... C_N is the value of $a\ particular\ class$ that maximizes the likelihood of that class with the optimization criteria (reduces error function)

$$\widehat{C_i} = \arg \max_{\theta = salient\ features} \{p(Ci|S)\},$$

that is decide class C_i that maximize $p(Ci|S)$

Maximum Likelihood Classification

To accomplish, **correct classification** (most of the time i.e., **robust and reliable**), we must have **sufficient amount** of Labeled Training data in **observation space** to captures the **salient features of a target… NO EXCUSE**

➔ **This is the reason we get good vs. bad results**

→ **We must be confident that we provided enough labeled samples**

→ **That Captures all the "salient features" of a target**

→ **Diverse geometric angles, Noise and clutter situations**

Otherwise, we may not trust the ML-based classification results

→ **The Main Take away: AI/ML is not a "magic-box"; it provides answer to our input data**

PR# NGA-U-2023-01021

Maximum a Posteriori (MAP) Estimator:

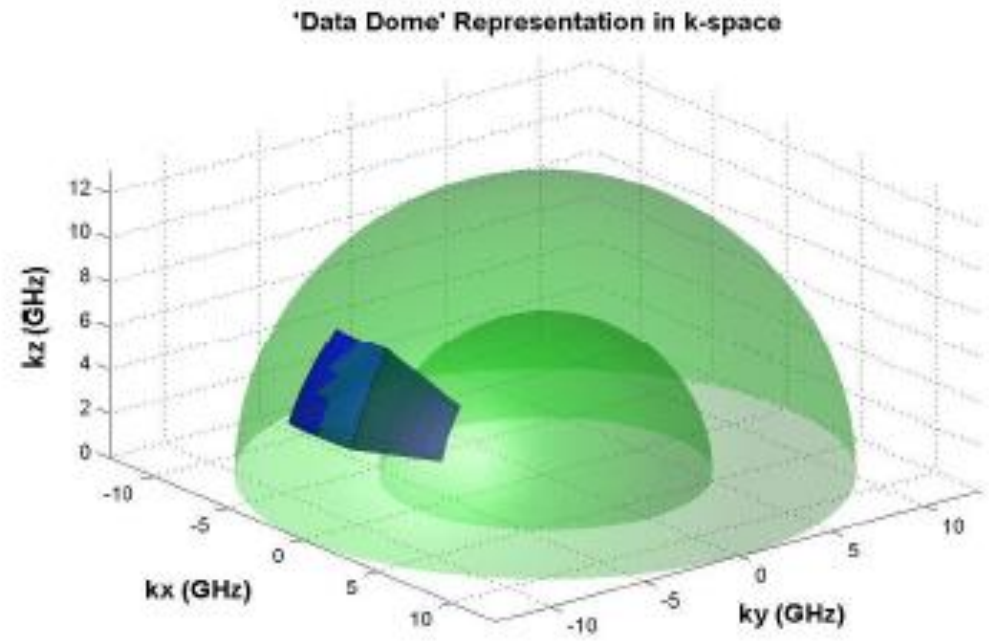
- MAP is MLE or MLC and prior information (Bayesian sense)
- Reduces False Alarm by using context information of the imaging scene.
- Foundation/Scene Geometry Aided ATR

$$\widehat{C_i} = \arg \max_{\theta = salient\ features} \{p(Ci|S)\},$$

+ p(x), Prior information of the imaging scene

PR# NGA-U-2023-01021

Sufficient Data for Radar Object Classification



The image is unclassified; public released by Air Force Research Lab.

LeRoy Gorham, Kiranmai D. Naidu, Uttam Majumder, Michael A. Minardi, "Backhoe 3D "gold standard" image," Proc. SPIE 5808, Algorithms for Synthetic Aperture Radar Imagery XII, (19 May 2005); doi:10.1117/12.609907

PR# NGA-U-2023-01021

PART 1: Conclusion

- Deep Learning Based “Classification” relies on Robust and Reliable Model Development
  - Sufficient amount of samples that capture/represent Salient Features of the Object

- We should not accept simplistic statement such as “more samples” is always better for AI/ML training / model development

PART 2: SAR AML Literature

Papers/Literature/Book

- N. Inkawhich, E. Davis, U. Majumder, C. Capraro and Y. Chen, "Advanced Techniques for Robust SAR ATR: Mitigating Noise and Phase Errors," *2020 IEEE International Radar Conference (RADAR)*, 2020, pp. 844-849, doi: 10.1109/RADAR42522.2020.9114784

- Majumder, Blasch, Garren. “Deep Learning for Radar and Communications ATR”. Artech House, July 2020

- Benjamin Lewis, Kelly Cai, Courtland Bullard, "Adversarial training on SAR images," Proc. SPIE 11394, Automatic Target Recognition XXX, 113940K (28 April 2020); doi: 10.1117/12.2558362

Paper 1 (IEEE RadarConf 2020)

- Very comprehensive research on adversarial ML
- Summary:
  - Two datasets have been used
    - MSTAR: Insert/perturb adv. noise to the image
    - CVDome: Insert/Perturb adv. noise to the phase history and then form images
  - Mitigation techniques
    - Adversarial training
    - Robust DNN models (ResNet18)

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Adversarial Attacks:

A mathematical formulation

- **Fast Gradient Sign Method (FGSM) [1]:**
  - FGSM is a simple, one-step attack to input data point, x
  - We can compute an FGSM adversarial example as:

$$x = x + \varepsilon \, sgn\left(\nabla_x L(x, y;\, \theta)\right)$$

- **Projected Gradient Descent (PGD)[1]:**
  - PGD is more powerful, multi-step attack to input data point, x
  - We can compute a PGD adversarial example as:

$$x^{t+1} = \prod_{x+s} \left(x^t + \varepsilon \, sgn\left(\nabla_x L(x, y; \theta)\right)\right)$$

- **Others (RF/Radar Specific):**
  - Random Noise Per Pixel, Signal Phase Errors, Interference/Jamming, Data Collection Geometry Mismatch in Training and Testing (Elevation, Azimuth, etc.)

[1] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," ArXiv, vol. abs/1706.06083, 2017.

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Adv. Noise Generation: FGSM

Fast gradient sign method (FGSM):

$$adv\_x = x + \epsilon * \text{sign}(\nabla_x J(\theta, x, y))$$

where

- adv_x : Adversarial image.
- x : Original input image.
- y : Original input label.
- ϵ : Multiplier to ensure the perturbations are small.
- θ : Model parameters.
- J : Loss.

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Adv. Noise Generation: PGD

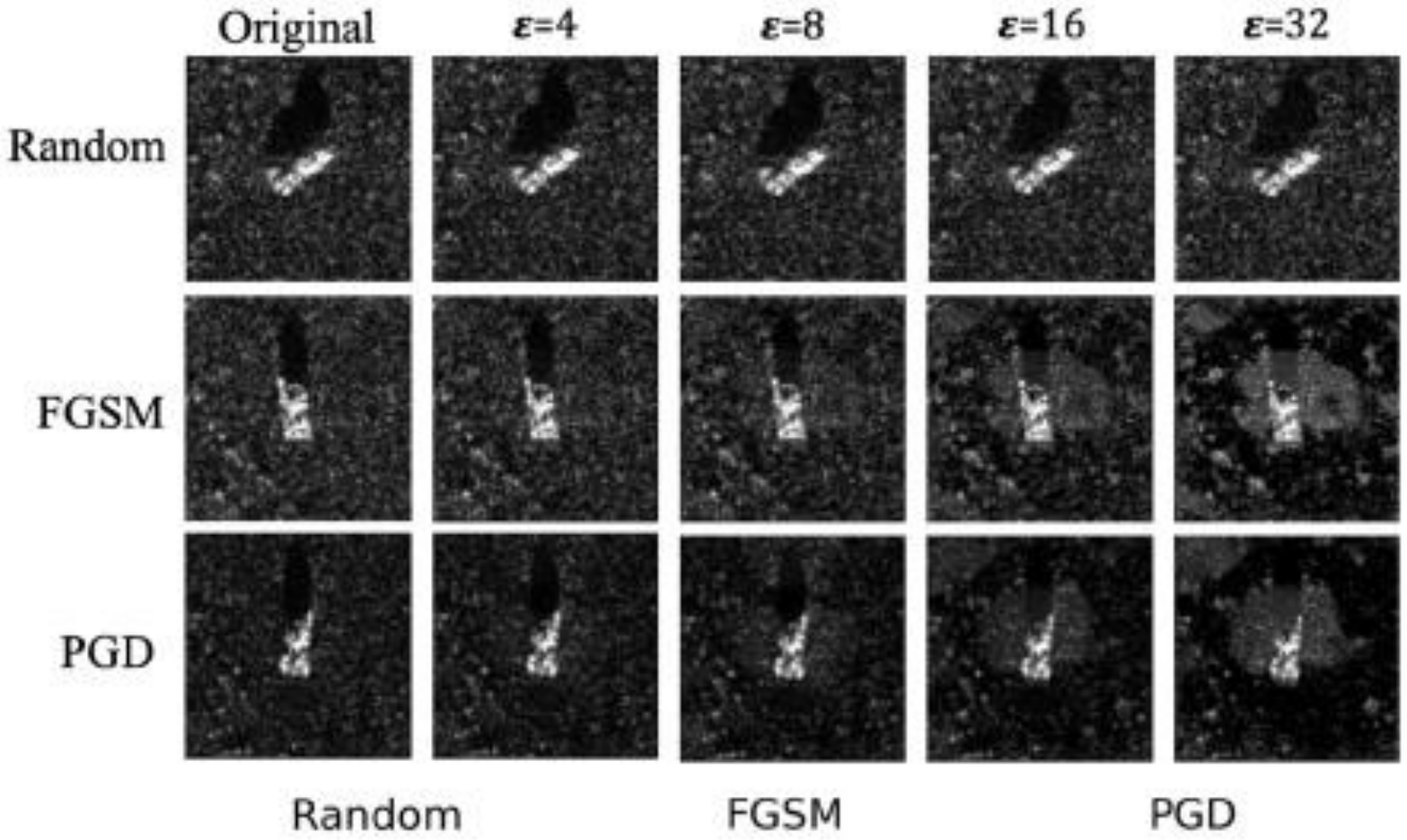
Projected gradient descent (PGD):

One can interpret this attack as a simple one-step scheme for maximizing the inner part of the saddle point formulation. A more powerful adversary is the multi-step variant, which is essentially projected gradient descent (PGD) on the negative loss function

$$x^{t+1} = \Pi_{x+\mathcal{S}} \left(x^t + \alpha \operatorname{sgn}(\nabla_x L(\theta, x, y)) \right) .$$

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Adversarial Noise Generation



Images are unclassified; generated by Majumder and Inkawhich using AFRL public released MSTAR data

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

ATR Models

- SAR ATR Community Models – A-ConvNet and ConvNetB
- Computer Vision Models – ResNet18, VGG11bn, ShuffleNetv2
- L_∞-norm – limit pixel-wise perturbation amount
- ε=0 → "standard" trained model

TABLE I
DNN MODEL INFORMATION

model	lr	# params	MSTAR Acc	CVDome Acc
aconv	0.001	373,898	98.39	91.91
convb	0.001	9,512,970	98.54	90.96
rn18	0.1	111,753,370	97.57	95.67
vgg11	0.01	598,698	98.94	-
shuf	0.1	115,863	96.89	-

The Table is unclassified; generated by Majumder and Inkawhich using AFRL public released CVDome data

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Adversarial Training

- Train the network parameters to minimize an “adversarial loss”
- Decision boundaries respect L_∞-norm balls around the training data
- Notable tradeoffs between clean data accuracy, model capacity, and dataset size have to be made

Intuition with L_∞ norm ball

Standard Training Objective

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[L(x, y; \theta) \right]$$

Adversarial Training Objective

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in S} L(x + \delta, y; \theta) \right]$$

PGD Adversarial Attack

$$x^{t+1} = \Pi_{x+S}(x^t + \alpha sign(\nabla_{x^t} L(x^t, y; \theta)))$$

Images are unclassified; generated by Inkawhich

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

SAR Datasets

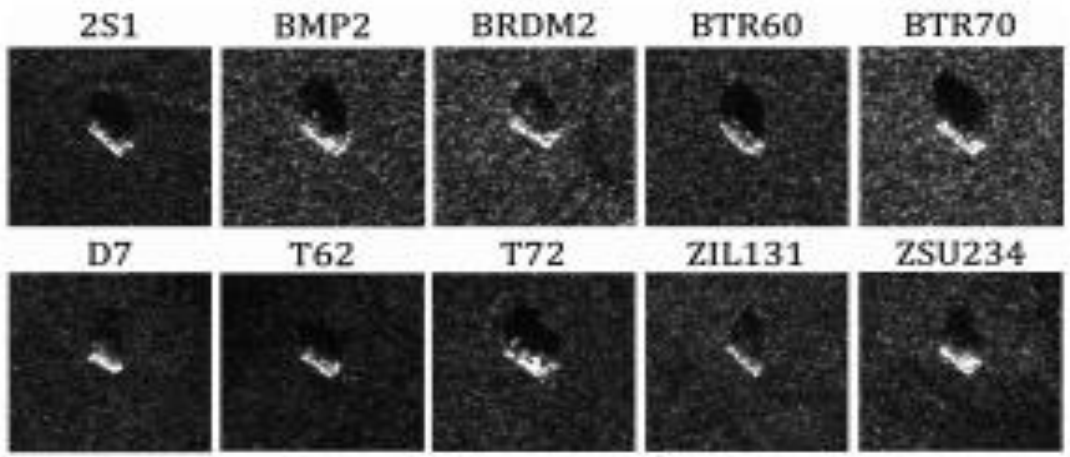


Fig. 1. MSTAR samples.

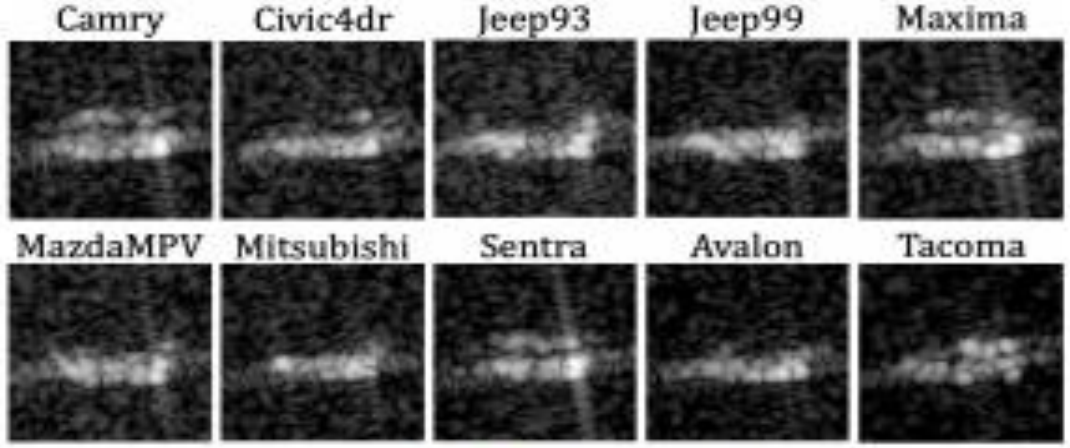


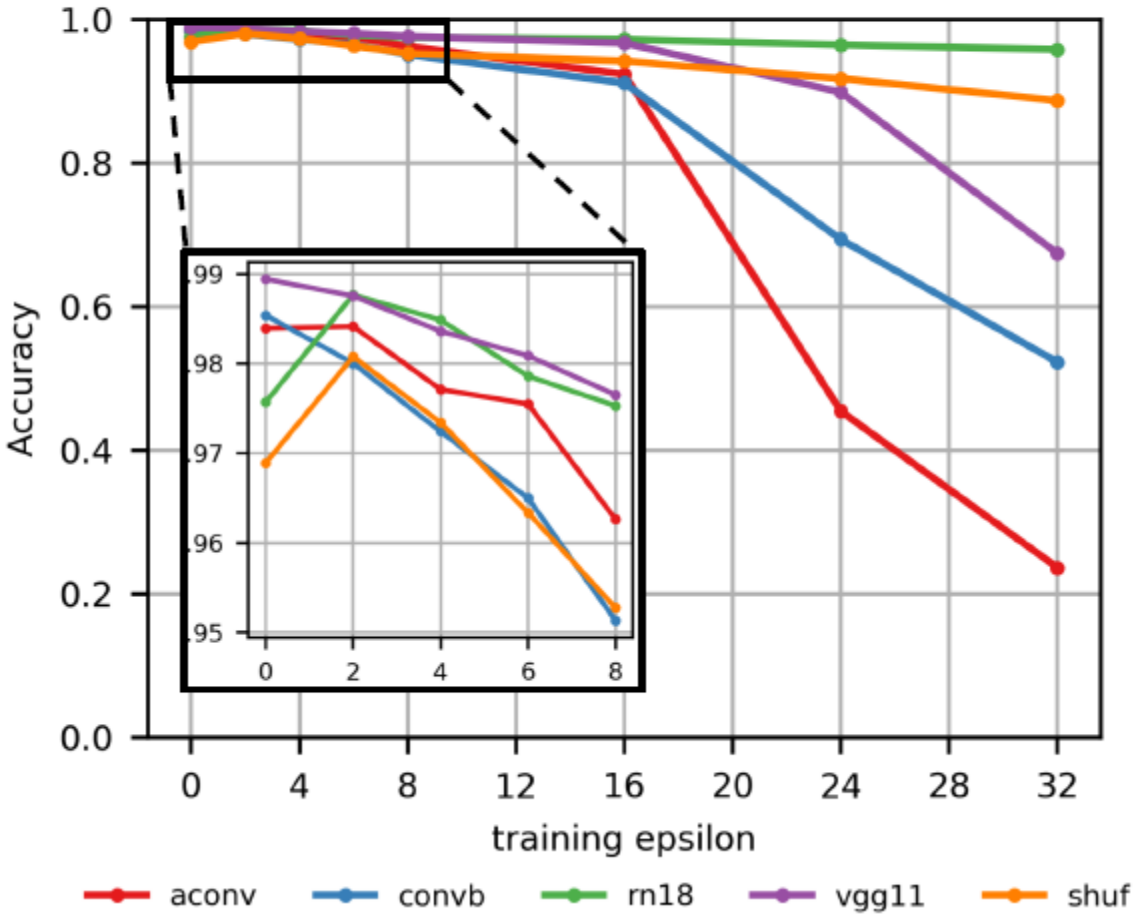
Fig. 2. CVDome samples.

Images are unclassified; generated by Majumder and Inkawhich using AFRL public released CVDome and MSTAR data

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

MSTAR Standard Operating Conditions

- Train: 17°
- Test: 15°
- Accuracy degrades at large training ε
  - Architecture dependent
- Some models have improved performance at small training ε
- Takeaway: small training ε's do not significantly harm MSTAR SOC performance

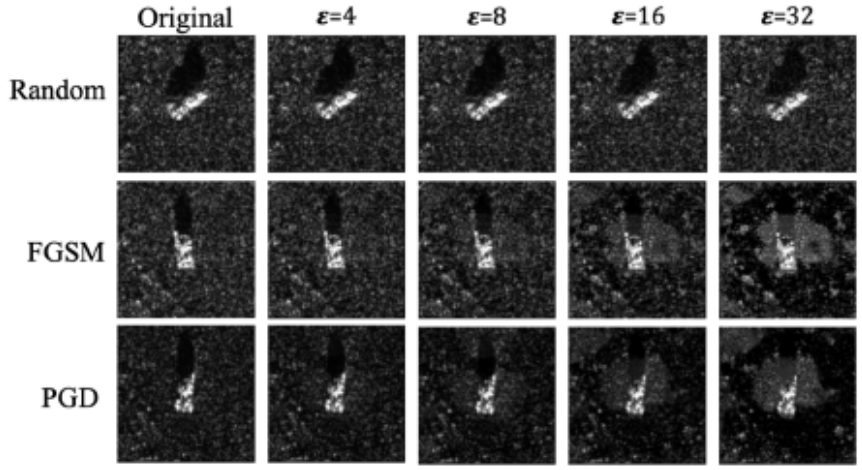


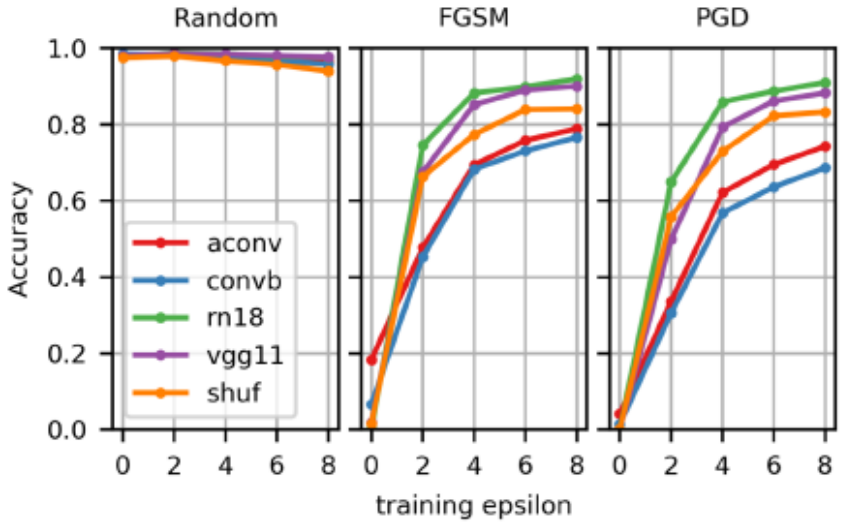
Images are unclassified; generated by Majumder and Inkawhich using AFRL public released MSTAR data

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Robustness: Adversarial Noise

- Worst-case noise, requires access to digital representation of data
- L_∞ noise becomes visible at ε>8
- Attacks completely degrade standard trained models
- AT recovers most accuracy loss
  - *rn18* highest performer
  - *aconv* and *convb* lowest performers





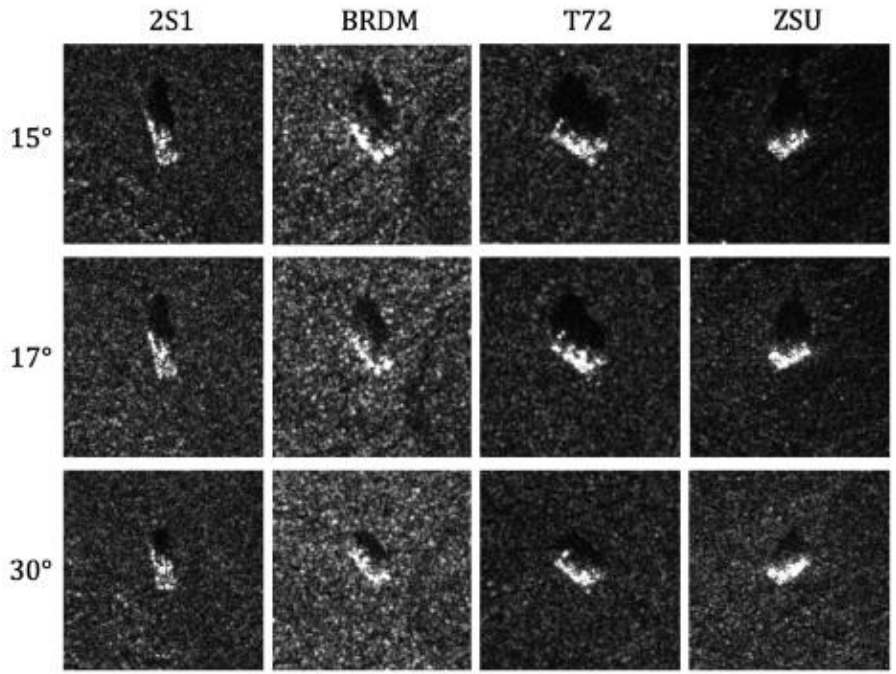
All test attacks generated at ε=8

Images are unclassified; generated by Majumder and Inkawhich using AFRL public released MSTAR data

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Robustness: Extended Operating Conditions

- Train: 17°
- Test: 30°
- The 13° elevation shift causes changes in the shadow regions and target signatures
- Small training ε increases EOC performance
  - Rn18 ε=2 increases accuracy by 11%



ACCURACY IN EXTENDED OPERATING CONDITIONS (MSTAR)

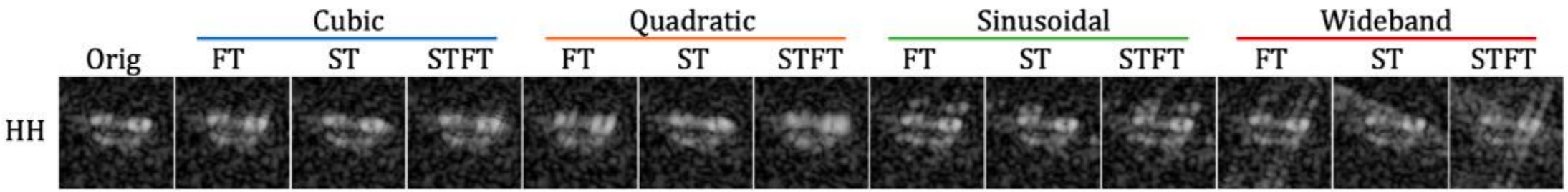
	training epsilon				
	0	2	4	6	8
aconv	82.2	85.1	85.2	84.7	84.6
convb	86.3	87.0	85.7	85.2	83.5
rn18	72.6	83.5	82.7	82.7	82.6
shuf	75.8	81.6	81.7	81.3	80.2
vgg11	69.3	84.4	82.9	82.9	81.8

Images are unclassified; generated by Majumder and Inkawhich using AFRL public released MSTAR data

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Robustness: Phase Errors

- Add cubic, quadratic, sinusoidal, and wideband noise to the CVDome phase history data
- Noise added in the range (fast time = FT) and/or azimuth (slow time = ST) dimensions



Images are unclassified; generated by Majumder and Inkawhich using AFRL public released CVDome data

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Robustness: Phase Errors

- Sinusoidal and wideband noise are the most challenging
- Rn18 is highest performing architecture
- AT increases robustness, in many cases by over 20%

ROBUSTNESS TO PHASE ERRORS (CVDOME)

model	eps	noError	cubicFT	cubicST	cubicSTFT	quadraticFT	quadraticST	quadraticSTFT	sinusoidalFT	sinusoidalST	sinusoidalSTFT	widebandFT	widebandST	widebandSTFT
aconv (HH)	0	91.91	87.84	89.7	83.79	81.97	87.98	73.93	57.35	87.25	44.2	53.79	78.64	33.47
	2	92.02	89.29	90.56	85.62	83.25	88.78	73.66	60.46	88.66	50.74	56.83	81.51	40.41
	4	89.98	88.23	88.05	84.19	80.21	87.03	72.47	59.05	87.23	52.24	58.5	80.38	44.76
	6	86.67	85.53	85.46	81.67	77.26	84.61	69.69	57.52	84.62	52.33	56.52	78.46	46.5
	8	74.97	73.01	73.05	67.71	63.92	71.38	56.02	51.35	73.34	49.32	52.16	67.96	46.67
convb (HH)	0	90.96	87.34	88.79	83.25	82.41	87.05	71.53	59.87	86.44	46.8	50.92	78.78	29.32
	2	93.42	90.87	91.6	86.53	84.84	89.35	73.76	62.39	89.98	52.51	56.64	82.66	36.69
	4	92.56	90.41	90.21	85.6	84.12	89.14	73.91	62.38	89.55	54.2	57.32	82.07	40.17
	6	90.26	88.1	88.55	84.01	82.66	87.02	72.89	60.37	87.46	52.2	56.62	79.03	42.15
	8	84.01	81.44	82.03	76.82	75.06	79.74	64.52	55.79	80.02	50.51	54.24	71.01	45.06
rn18 (HH)	0	95.67	89.89	92.65	82.11	85.56	90.75	71.83	62.7	89.62	48.56	51.26	77.24	27.61
	2	97.84	95.79	96.34	92.28	90.97	93.66	79.34	63.69	93.62	55.08	64.71	86.83	44.97
	4	97.56	95.17	96.16	92.41	90.01	92.71	78.69	62.35	94.33	54.84	66.98	87.7	53.91
	6	96.78	93.57	95.07	90.73	87.98	91.08	75.56	61.15	93.91	53.87	67.67	87.15	55.39
	8	95.51	92.41	94.11	89.42	86.47	89.93	74.39	59.25	92.98	54.55	66.64	86.79	58.14

The Table is unclassified; generated by Majumder and Inkawhich using AFRL public released CVDome data

Public Released: AFRL 88ABW-2019-5326, 88ABW-2020-3313 and 88ABW-2020-0519

Conclusions

- Robustness and interpretability improve with AT
- AT models do not rely on shadow regions

- The impact of AT greatly depends on architecture
  - Small and fast models not good for AT
  - Rn18 works the best here but is the largest

SUMMARY:

- AI/ML based RF object recognition is not a closed form solution
- Adversarial robustness can be enhanced by relevant datasets