Network Intrusion Detection

Adversarial Robustness Evaluation with AdvGAN

CIC-IDS 2017 Dataset

PhD Progress Presentation — February 26, 2026

Mohammed Amir MESSIOUD · LS2N - NII

Outline

Motivation

The Challenge

- Network intrusion detection systems (IDS) rely on **supervised ML** trained on clean traffic
- Adversaries can craft **adversarial flows** , attacks that look benign to the detector
- Standard ML models are **brittle** against even small perturbations

Research Gap

Most IDS research evaluates models on clean data only , **adversarial robustness is rarely studied** in network security.

Our Approach

1. Train a **high-accuracy baseline** IDS (Random Forest)
2. Use **AdvGAN** to generate realistic adversarial attack flows
3. Measure **Evasion Success Rate (ESR)**
4. **Retrain** on adversarial samples to restore robustness

CIC-IDS 2017 Dataset

Dataset Overview

- **Source:** Canadian Institute for Cybersecurity
- **8 CSV files**, one per working day
- **2,830,743** network flow records
- **79 features** extracted from PCAP files
- **15 traffic labels**

Class Distribution

Label	Count	%
BENIGN	2,273,097	80.30
DoS Hulk	231,073	8.16
PortScan	158,930	5.61
DDoS	128,027	4.52
DoS GoldenEye	10,293	0.36
...9 more classes	*<1% each*	

Heavily imbalanced: 80% benign traffic , critical for evaluation strategy.

Preprocessing Pipeline

3.1 Infinity & Negative Values

Replace $\pm$Inf $\rightarrow$ NaN, drop rows

Filter 6 physically invalid cols

3.2 Correlation Filter

Drop features with $|r| > 0.95$

79 $\rightarrow$ 54 features

3.3 RobustScaler

Median + IQR scaling

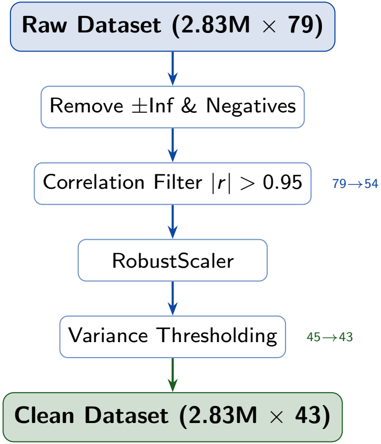
Resilient to DDoS outliers

3.4 Variance Thresholding

Remove constant cols (8 dropped)

Remove quasi-constant ($p = 0.995$)

54 $\rightarrow$ 45 $\rightarrow$ **43** features



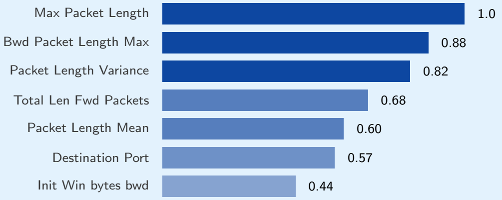
Result: 150 rows removed · 36 features eliminated · 2,827,726 samples retained

Baseline Classifier , Random Forest

Configuration

- **Task:** Binary , BENIGN (0) vs. ATTACK (1)
- **Split:** Stratified 80/20
- **Train:** 2,262,180 samples
- **Validation:** 565,546 samples
- `n_estimators=50`, `max_depth=10`

Top Features by Importance



Classification Report

	P	R	F1	Support
BENIGN	1.00	1.00	1.00	454,235
ATTACK	1.00	0.98	**0.99**	111,311
Accuracy			**1.00**	

AdvGAN , Architecture

Generator G

$$x_{\text{adv}} = x + G(x) \cdot \text{mask} \cdot \varepsilon$$

- Linear(43→128) → BN → ReLU
- Linear(128→256) → BN → ReLU
- Linear(256→43) → **Tanh**
- $\varepsilon = 0.05$ controls perturbation strength
- `mask`: controls which features can change

Discriminator D

- Linear(43→256) → LeakyReLU(0.2)
- Linear(256→128) → LeakyReLU(0.2)
- Linear(128→1) *raw score , no sigmoid*

Dual Loss Objective

WGAN Critic loss:

$$\mathcal{L}_D = -\mathbb{E}[D(x)] + \mathbb{E}[D(G(x))]$$

Generator loss:

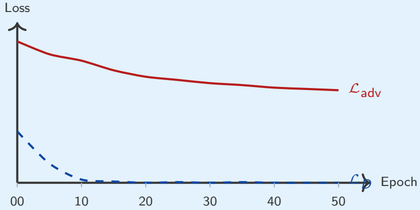
$$\mathcal{L}_G = \underbrace{-\mathbb{E}[D(G(x))]}_{\text{realism}} + \alpha \underbrace{\mathbb{E}[1 - P_{\text{RF}}(\text{BENIGN})]}_{\text{evasion}}$$

- $\alpha = 10$ balances realism vs. evasion
- P_{RF} queried in **black-box** mode
- Trained for **30 epochs**, batch size 64

Training Setup

- GAN trained on **attack-only** samples from training set
- Optimizer: Adam, lr $= 10^{-4}$, $\beta = (0.5, 0.9)$
- All 43 features mutable (mask $= \mathbf{1}$)

Convergence



Evasion Results , Baseline Model

Attack samples tested	111,311
Successful evasions	50,153
ESR	**45.06%**

Nearly half of all adversarial attack flows evaded the baseline detector.

Key Insight

A high-accuracy model (F1 $= 0.99$) is **not robust** against adversarial perturbations , even with $\varepsilon = 0.05$.

Perturbation Analysis

Most Manipulated Features

Rank	Feature	Mean $\lvert\Delta\rvert$
1	Idle Std	0.0157
2	Max Packet Length	0.0113
3	Active Std	0.0102
4	Bwd Packet Length Max	0.0102
5	Bwd Packets/s	0.0098

Perturbations are **small** ($<$ 2% of feature range) yet highly effective.

Correlation Preservation

The adversarial samples maintain **feature correlation structure** similar to real attacks , making them statistically realistic.

- **Real attacks**: strong packet-length correlations
- **Adversarial**: same structure preserved
- Confirms GAN learned **realistic** perturbations, not random noise

Adversarial flows are *statistically indistinguishable* from real attacks yet evade the IDS.

Adversarial Retraining , Restoring Robustness

Augmentation Strategy

- Inject all **111,311 adversarial samples** into training set
- Label them as ATTACK (class 1)
- **Augmented training size:** 2,373,491 samples
- Retrain with **same RF architecture**

Rationale

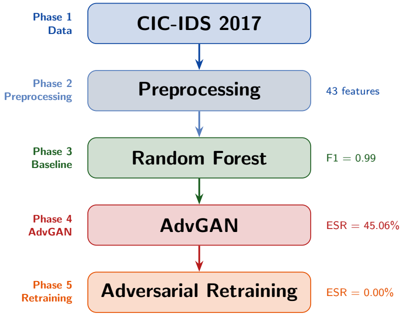
By exposing the classifier to adversarial examples during training, it **learns the perturbation manifold** and becomes robust without architectural changes.

✓Circle Results Comparison

	Baseline	**Robust**
F1 (clean)	0.99	0.99
ESR (adversarial)	**45.06%**	**0.00%**
Evasions	50,153	**1**

Conclusion: Adversarial retraining **completely eliminates** evasion while preserving detection accuracy on clean traffic.

Full Pipeline Summary



Key Contributions

- ✓ Full preprocessing pipeline for CIC-IDS 2017
  79 $\rightarrow$ 43 features, 2.83M samples retained
- ✓ High-accuracy baseline detector
  Random Forest, F1 = 0.99
- ✓ Realistic adversarial attack generation
  AdvGAN, ESR = 45.06% against baseline
- ✓ Full robustness restored via retraining
  ESR drops to 0.00% with no accuracy loss
- ✓ Black-box attack – no internal model access required

Next Steps & Open Questions

Immediate Work

1. **Multi-class classification** , distinguish specific attack types rather than binary detection
2. **Feature mask tuning** , restrict perturbations to only immutable flow features (e.g. packet size, not protocol)
3. **Hyperparameter search** , tune ε, α, GAN depth

Longer-Term Directions

1. **Federated learning integration** , distribute the IDS training across clients (DRL-based client selection)
2. **Transfer attack** , test adversarial flows against other model families (XGBoost, LSTM)
3. **Adaptive adversary** , iterative attack-retrain loop
4. **Evaluation on newer datasets** , CICIDS 2018, UNSW-NB15

Thank You

Questions & Discussion

mohammed-amir.messioud@etu.univ-nantes.fr

Code available at: https://github.com/SYK3S999/Network-Intrusion-Detection