NOVEMBER 12, 2025

CNET – PROJECT

REPORT

HARRIS HASSAN, ABDULLAH NADEEM & M. MUTTAYAB
22I - 1947, 22I - 1925, 22I - 1949
DS(D)

Enhancing Adversarial Robustness in Network Intrusion Detection

Abstract

This report details a replication attempt of the paper "Enhancing Adversarial Robustness in Network Intrusion Detection: A Novel Adversarially Trained Neural Network Approach. The objective was to validate the paper's findings by reproducing its results for four models: a standard Neural Network (NN), a Transformer-NIDS, a Random Forest (RF), and the paper's proposed ADV_NN model.

Our findings are twofold:

1. Replication Success: The core contribution of the paper—the ADV_NN model—was successfully and accurately replicated. Our results for the ADV_NN's clean and robust accuracy are a near-perfect match for the paper's claims, validating their proposed defense strategy.
2. Baseline Contradiction: The paper's baseline results for the Transformer and Random Forest models were not reproducible. The Transformer model failed to train, and the Random Forest model proved highly robust, directly contradicting the paper's claim that it "suffers significant degradation".

Introduction

The objective of this experiment was to replicate the results presented in the paper. The paper evaluates the adversarial robustness of an RF, a standard NN, and a Transformer-NIDS, and then proposes a novel model, ADV_NN, which uses Curriculum Adversarial Training (CAT) and a composite loss function to improve resilience. The replication was performed using a "stricter" Python script (henceforth "the script") designed to follow the paper's methodology as closely as possible.

Implementation Details and Procedures

- Dataset and Preprocessing:
    - Dataset: The UNSW-NB15 dataset was used.
    - Data Split: The script strictly used the provided UNSW_NB15_training-set.csv for training and UNSW_NB15_testing-set.csv for testing. This aligns with the "Mark 2" replication strategy and avoids data leakage from re-splitting.
    - Preprocessing: All preprocessing steps described in the paper (Section 3.2) were followed, including Label Encoding for categorical features, imputation of missing values, and Z-score standardization.
- Model Architectures:
    - NN and ADV_NN: A feedforward neural network with an input layer of 42 features, two hidden layers (128 and 64 neurons), and a 2-neuron output layer was used, matching the paper's architecture.
    - Transformer-NIDS: The architecture from the paper (Section 3.4) was implemented, including 3 encoder layers, 4 attention heads, and a feedforward dimension of 256.

- Random Forest: A standard sklearn.ensemble.RandomForestClassifier was used with default parameters, as none were specified in the paper.
- Substitute Model: For black-box attacks, a simple NN with one 64-unit hidden layer was used as the substitute, per Section 6.3.3.

- Training and Adversarial Defense:

  - NN Training: The baseline NN was trained for 20 epochs using standard Cross-Entropy loss.
  - ADV_NN Training (CAT): The core defense was implemented exactly as described.

    - Curriculum: The PGD attack strength (ϵ) was linearly increased from 0 to 0.20 over 20 epochs
    - Composite Loss: The script implemented the full composite loss function from Section 3.7. This includes the standard cross-entropy loss, the Gradient Alignment Loss, and the Feature Smoothing Loss.
    - Hyperparameters: The paper's specified weights of $\lambda_{align}=1.0$ and $\lambda_{smooth}=0.5$ were used.

- Attack Generation:

  - PGD: Attacks were generated using 10 iterations and a step size (α) of 0.01, as specified in Section 4.1.
  - Black-Box: The script followed the paper's black-box methodology: querying the trained ADV_NN to label 10,000 noisy samples training the substitute model on this new dataset, and then using PGD on the substitute to attack the other models.

Results and Comparative Analysis

Metric	Model	Paper's Claimed Result	Achieved Replication Result
Clean Accuracy	NN (Baseline)	78.56%	89.14%
Clean Accuracy	Transformer	93.02%	32.56%
Clean Accuracy	Random Forest	95.18%	90.07%
Clean Accuracy	ADV_NN	86.04%	86.91%
PGD Acc. (e=0.15)	NN (Baseline)	46.50%	61.89%
PGD Acc. (e=0.15)	Transformer	61.97%	32.56%
PGD Acc. (e=0.15)	Random Forest	49.80%	90.10%
PGD Acc. (e=0.15)	ADV_NN	83.70%	82.63%

Experimental Observations and Challenges

Challenge 1: Replicating the ADV_NN (Success)

- Observation: The ADV_NN model was replicated with near-perfect accuracy. Our clean accuracy (86.91%) was within +0.87% of the paper's (86.04%), and our PGD-robust accuracy (82.63%) was within -1.07% of the paper's (83.70%).
- Conclusion: This is a total success. It proves that the paper's core methodology—combining Curriculum Adversarial Training with the composite L_{align} and L_{smooth} losses—is effective and reproducible as described.

Challenge 2: The Transformer Model Failure

- Observation: The Transformer model's accuracy (32.56%) indicates a complete failure to train. The metrics Precision: 1.0000 and Recall: 0.0092 show it collapsed into only predicting the majority "benign" class.
- Conclusion: The paper's claim of 93.02% accuracy [44]is not reproducible using the architecture and hyperparameters they provided. The script's attempt to implement "class balancing" (as mentioned in Sec 3.4 of the paper) also failed, confirming that critical information for training this model is missing from the paper.

Challenge 3: The Random Forest Contradiction

- Observation: This is the most significant contradiction. The paper claims the RF model "suffers significant degradation" and its accuracy collapses to 49.80% under a PGD attack.
- Our results decisively prove this to be false. The RF's accuracy was almost unchanged, going from 90.07% (Clean) to 90.10% (PGD @ 0.15).
- Conclusion: The paper's methodology for attacking the RF is flawed. The process of generating PGD attacks using an NN surrogate [48] and "transferring" them to an RF is not an effective attack. This replication shows that PGD attacks, in this context, do not transfer well (if at all) from neural networks to tree-based models.

Final Conclusion

This replication was a success that uncovered significant findings including few gaps such as degraded performance of transformers.