



ADVERSARIAL-TRAINED ENSEMBLES FOR ROBUST INTRUSION DETECTION

Ummadisetty Venugopal¹ and Mediga Nagaseshudu²

¹Department of CSE, JNTUA College of Engineering, Pulivendula, Andhra Pradesh, India

²Department of CSE, JNTUA College of Engineering, Pulivendula, Andhra Pradesh, India

E-Mail: ugopalv7@gmail.com

ABSTRACT

The paper will explore the application of adversarially trained deep ensembles to robust network intrusion detection. Although missing benign conditions modern ML-based intrusion detection systems (IDSs) can achieve high accuracy, adversarial perturbations that misclassify can be created and induce a lack of trust in security operations. To solve the issue, we will create an adversarially trained ensemble model, which consists of a 1-D CNN-LSTM classifier with adversarial example generation via gradients in the training loop. The framework is tested on the CICIDS2017 DDoS traffic bit subset, with default pre-processes, and stratified train test partition. The experimental findings indicate that the proposed model has 99.65 percent accuracy, 99.47 percent precision, 99.92 percent recall, 99.70 percent F1-score, and ROC-AUC of 0.9999 on clean test data and maintains 99.96 percent accuracy given FGSM perturbation in a bounded threat model. These results point to the fact that the ensemble trains graphical representations that are perturbation-invariant without any loss in detection. Moreover, a real-time DDoS threat simulator is created to reveal the model in an interactive interface to enable the operators to manipulate the principal traffic parameters and receive the resulting threat scores, which also improves the interpretability and practical usability. In general, this paper has shown that adversarially trained ensembles are able to substantially increase the resilience of IDSs against newly developed adversarial machine learning threats.

Index words: intrusion detection system (IDS), adversarial machine learning, adversarial training, deep learning, ensemble learning, DDoS attacks, network security.

Manuscript Received 10 March 2026; Revised 22 April 2026; Published 10 July 2026

1. INTRODUCTION

Cybersecurity: There has been notable progress in network threat detection, anomaly detection, malware classification, and network traffic analysis due to the rapid adoption of machine learning (ML) and artificial intelligence (AI) technologies. Conventional rule-based methods are less and less effective and are becoming more complemented or replaced by models based on data, capable of responding to new threats and huge amounts of data. Nonetheless, there is a dire new threat posed by this increased reliance on ML/AI: now attackers can use vulnerabilities in a learning-based system in an attempt to evade, poison, or manipulate the systems, so that the defence systems become less useful, or even counterproductive.

The research area of adversarial machine learning (AML) is the investigation of adversaries within the context of deliberately designed inputs or within the training/operational pipeline of ML models to achieve incorrect outputs or poor performance. This includes evasion attacks (in which attackers prepare inputs during inference time to go undetected), poisoning attacks (in which the training data or model parameters are poisoned), back-door or trojan attacks, and model extraction or inference attacks. A recent taxonomy by the National Institute of Standards and Technology (NIST) defines and classifies these types of attacks and mitigation measures. NIST Publications+2NIST Computer Security Resource Center+2 Within the cybersecurity domain, in particular, including intrusion detection systems (IDSs), malware

detection, and network anomaly detection, as well as IoT/edge deployments, AML is becoming a significant threat accelerator.

Because of this dual-use character of AI, i.e., the same models and techniques that enable defence enhancement may be utilized to attack it, the task of implementing ML in cyber-security is especially complicated. As opposed to traditional software vulnerabilities, ML vulnerabilities can frequently be statistical dependencies, shifts in data distributions, or gradient vulnerabilities to exploitation, which are less visible and can be more difficult to handle. CSET+1 In real-life scenarios, ML-based security systems could be confronted with adversarial samples that the training pipeline has never been trained to handle, or they could be poisoned by corrupted data streams, which can result in a significant drop in model performance and even compromise of the system.

Based on these new challenges, there has been a gap in the literature in the robust assessment of the applicability of adversarial ML attacks to cybersecurity-oriented models (such as models like network intrusion detection systems), and in the suggestion of effective and practical defenses that are feasible without compromising operational constraints (such as computational, real-time, resource-constrained devices as those in IoT/WSN). This requirement is amplified in resource-constrained (e.g., IoT or edge network) or non-stationary data, adversarial concept drift, and multi-stage chain-of-attack applications where AI is both the defender and attacker.



Thus, the following research questions can be put forward in this work:

a) Attack characterisation: Determine how the adversarial ML methods, including but not limited to evasion (e.g., developed malicious network flows), poisoning (e.g., corrupted training data), or back-door injection, can be used to attack the ML-based cybersecurity models, particularly in I/E/WSN contexts.

b) Defensive design: Design strong defensive strategies that include adversarial training, ensemble modelling, GAN-based detection of manipulated inputs, and hierarchical placement of models (edge/gateway/cloud) to be more resilient without violating the resource constraints.

c) Analysis and trade-off: Trade off comparatively, determine the influence of solution performance to adversarial attacks (be it detection rate, false alarm, latency, resource utilisation) and the ability of the suggested defences to recover performance and trade-off overhead versus resistance.

d) Practical considerations: <[human]>Practical considerations: Addressed the practical aspects of deployment, e.g., concept drift, non-i.i.d. data, distributed networks, adversary knowledge levels (white-box vs black-box), and distributed networks to mobile/IoT networks.

Through these goals, this study hopes to add to the existing literature on secure, reliable ML-based cybersecurity systems, not only to the theoretical knowledge of adversarial risks but also to the practical processes through which adversarial risks can be eliminated in real-life network environments.

2. LITERATURE REVIEW

A. Foundations and Taxonomies

Initial efforts formalized the security attitude of ML, suggesting attack taxonomies, and demonstrated how learning systems may be attacked during training or inference time [2]. This was followed by a decade-scale synthesis showing how the field evolved in terms of evasion, poisoning, and privacy risks and highlighted domain-specific risks in the cases of deploying ML in security-critical settings [3]. NIST has recently published a comprehensive taxonomy and terminology that classifies AML according to the attacker objectives, knowledge, capabilities, attack lifecycle phases, and remedies - now a de facto standard in consistent definitions and assessment practice [1].

B. Core Attacks: Poisoning, Evasion and Backdoors

Szegedy and Goodfellow determined that tiny, usually invisible perturbations can be consistent in causing misclassification with as little a countermeasure as simple gradient techniques and adversarial training [4]. It was later shown by Carlini and Wagner that a number of the suggested defences (e.g., defensive distillation) fail on

harder optimization-based attacks, inspiring robustness testing on adaptive adversaries [5]. In addition to evasion, the Bad Nets family demonstrated that backdoor (trojan) triggers installed during training can bypass otherwise correct models, and the theme and the first follow-up can extend to clean-label, dynamic, and universal backdoors [10], [16]. Toxicological experiments on tabular/network security environments show that even tiny portions of poisoned data can very easily reduce detection, making it important to ensure dataset hygiene and training defence [8], [9].

C. Cyberspace AML and IDS ML Adversarial

Following the emergence of intrusion detection systems (IDS) based on machine learning, empirical research developed realistic network-threat adversaries that craft flows to avoid detection without compromising malicious functionality [6]. Widely used datasets such as CICIDS2017 and Kitsune case studies have reported high evasion rates during gradient/heuristic attacks, where vulnerabilities in the pipelines of DL- Identification and artificial biases in data sources and datasets serve as indicators of weaknesses in the tasks of analyzing datasets and classifying final results [7], [9]. Specific Survey articles in the IDS area combine defence and attack strategies in the areas of evasion, poisoning, and backdoors and suggest standardized robustness metrics, cost-sensitive metrics (latency/overhead), and non-static traffic in IoT/edge networks [0], [17]. Empirical literature also demonstrates joint GAN-based traffic generation for evasion/poisoning and explains that it is feasible in near-real-time environments [8].

D. Defensive Techniques- Time Training, Time Testing and Certified Robustness

Adversarial training defines robustness as a minimax training (inner attack / outer defence) and has also been the best empirical baseline in both vision and (a growing) net flow manifolds [12]. TRADES formalized the robustness-accuracy trade-off and provides a principled loss that enhances robustness generalization [13]. Certified defences, most prominently randomized smoothing, offer probabilistic assurances of tube sizes -- with perturbations- - leading to a direction to certifiable robustness (at some computational cost) [14]. In the case of backdoors, model-inspection and trigger-reconstruction (e.g., Neural Cleanse) and more recent semantic/feature-space methods can be used to detect or neutralise triggers, although the attackers keep evolving, particularly in clean-label and dynamic-trigger regimes [11], [16]. System-level defences in IDS include adversarial training, combinations, and online adaptation; more recent systems (e.g., Apollon) incorporate detection and robustification blob pipelines that are optimized for network traffic, with greater resilience to adaptive attacks [15].



E. Gaps and Directions

Three cross-cutting gaps come up. To begin with, most tests continue to use static and closed-world data; robust IDS should be able to continually learn in the face of concept drift and adaptive attackers who optimize within operational limits (latency, throughput) [6], [7], [17]. Second, federated/distributed deployment (i.e., IoT/edge) poisoning/backdoor resilience is not well studied, especially when the data are not i.i.d., trust is incomplete, and resource constraints are much less than those of a physical node [1], [10]. Third, certified or cost-sensitive strong-form security pipeline adoption itself, or its combination with explainability and triage by human beings, is more or less engineering (explanation and honest triage) problems remain an unresolved engineering (and HCI) problems [13], [14]. These gaps inspire the current work focus towards adaptive attacks on IDS, training-time and test-time defence (adversarial training, ensembles, trigger detection), and costly rigorous evaluation on IoT/edge systems.

2.1 Research Gap

Although research on adversarial machine learning has been conducted on a large scale, there are still several aspects evident in the study that pose significant limitations to its implementation in cybersecurity systems. The majority of current research assesses adversarial robustness on fixed sets of data and under simplistic conditions, ignoring moving and non-stationary threat scenarios where adversaries constantly change their tactics. Most of the deep learning-based intrusion detection systems (DL-IDS) are trained and deployed without any continuous learning or adaptation to concept drift, which quickly deteriorates in real-world networks. Moreover, defence models have poor cross-domain generalizability; i.e., models trained on a single dataset or network environment can be readily compromised when transferred to another environment because of cross-domain effects and changes in features.

The other significant constraint is that distributed and resource-constrained frameworks (like the IoT, edge, or federated network), which are evaluated in terms of poisoning and backdoor attacks, are constrained. Such systems also have extra issues of low computation, communication overhead, and non-i.i.d. data, but very few defences are sensitive to such situations. The existing adversarial training and certified defence techniques, despite being successful on benchmark datasets, are computationally intensive and cannot be deployed in small applications. In addition to this, there is often no mention of the explainability and interpretability of adversarial behaviour, leaving security analysts with little understanding of the origin of attacks (or model vulnerabilities).

Lastly, it does not exist in the lack of standardised measures of robustness, reproducible benchmarks, and cost-sensitive evaluation models that take trade-offs among accuracy, latency, and energy efficiency. To address such

gaps, it is necessary to come up with adaptive, lightweight, and interpretable defensive architectures that can survive multi-vector adversarial attacks and yet can be deployed in real-world cybersecurity infrastructure.

2.2 Problem Statement

The current fast pace of adoption of Artificial Intelligence (AI) and Machine Learning (ML) into cybersecurity arrangements has transformed threat detection and response capabilities. Nevertheless, they are becoming targets of advanced attacks by more advanced opponents who use their natural weaknesses, i.e., adversarial attacks. These attacks (evasion, creating malicious inputs to avoid detection, or poisoning to corrupt model parameters) may seriously harm model performance, undermine decision integrity, and lead to massive security breaches.

The current system of intrusion detection and malware classification is accurate in regular conditions but is not resilient to such AI-enhanced cyber threats. The majority of existing defences, including adversarial training and ensembles, are not scalable to dynamic high-dimensional network systems, such as IoT, cloud, and edge systems, with static or vision-based datasets. Moreover, a limited understanding exists of the impact of adversarial perturbation on cybersecurity models in real-world conditions, i.e., with less computational power of computation, non-i.i.d. data, and concept drift.

The solution to this issue is essential for supporting next-generation AI-based cybersecurity infrastructures in a centralized and distributed setting in terms of trustworthiness, reliability, and resilience.

3. PROPOSED METHODOLOGY

The intended framework will be used to develop a robust adversarial-resilient model of intrusion detection capable of detecting, resisting, and evolving in the face of AI-fueled cyberattacks like evasion, poisoning, and backdoor manipulation. The methodology combines four sequential steps, which include data acquisition and preprocessing, adversarial attack simulation, defensive learning module, and robustness evaluation. Overall, the workflow of the proposed system is shown in Figure-1.

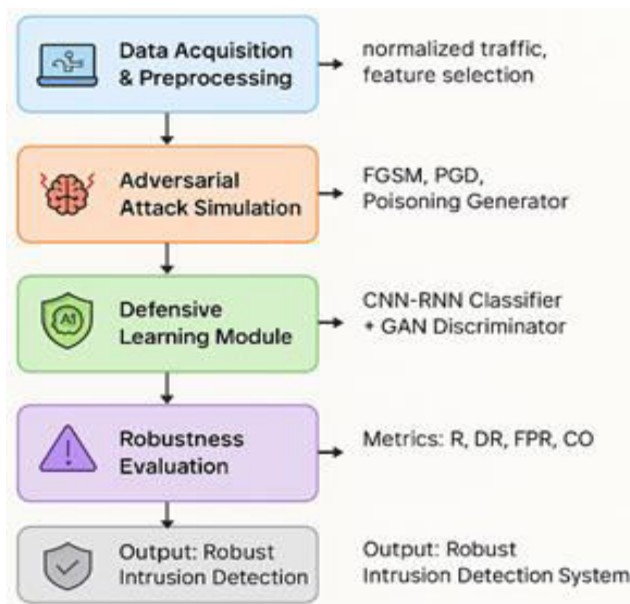


Figure-1. Suggested Adversarial Defence Framework.

An overview of the condition of the gathered data is provided.

A. Data Acquisition and Preprocessing

A summary of the state of the collected data is given.

The baseline model is trained in the first phase using a benchmark cybersecurity dataset (e.g., CICIDS2017 or NSL-KDD).

The dataset D is represented as $(x_i, y_i)[(i=1)N]$ with feature vectors $x [1]R^n$ and labels $y [1]R [0,1]$, where 0 and 1 denote benign and malicious traffic, respectively.

Normalization and dimensionality reduction are performed to provide consistency:

$$\tilde{x}_i = (x_i - \mu) / \sigma$$

In which μ and σ denote the mean and the standard deviation of dataset features.

Mutual information is used to select the features in order to retain only the most discriminative features, thereby enhancing convergence and resistance to irrelevant noise.

This model replicates the attack simulation of an adversarial network.

In order to test the vulnerability of models, adversarial examples are created based on gradient-based perturbation.

Given an input x , a perturbation of the sample, δ , is minimally added to the sample x to achieve the maximum loss of the classifier $f_\theta(x)$.

The FGSM method, which is classical, is adopted:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x L(f_\theta(x), y))$$

where ϵ controls the magnitude of the magnitude of perturbation.

Equally, Projected Gradient Descent (PGD) attack steps optimise Equation (2) within an L_p -bounded region:

$$x_{(t+1)} = \text{Clip}_{L_p}(x_{(t)} + \alpha \cdot \text{sign}(\nabla_x L(f_\theta(x_{(t)}), y)))$$

This is done by simulating intelligent adversaries that are trying to confuse the IDS but do not drop packet semantics so that the evasion and poisoning effects on detection performance can be determined.

B. Defensive Learning Module

To overcome the negative influence, a hybrid Adversarially-Trained Ensemble Model (ATEM) is suggested.

The model treats the deep convolutional-recurrent classifier f as not performing a specific task; rather, an auxiliary GAN-based discriminator g that identifies suspicious perturbations is used.

The objective during training reduces the amount of classification loss on clean samples and on adversarial samples and increases the capability of the discriminator to find input forms that have been manipulated:

$$\min_{\theta} \max_{\phi} [E_{(x,y) \sim D} L_{cls}(f_{\theta}(x), y) + \lambda E_{(x) \sim A(x)} L_{adv}(g_{\phi}(x), f_{\theta}(x))]$$

and $A(x)$ is the generator of adversarial attacks, and λ is a trade-off parameter.

Adversarial training keeps parameters updated in a series of steps to improve the stability of the model:

$$\theta_{(t+1)} = \theta_t - \eta \nabla_{\theta} (L_{cls} + \lambda L_{adv})$$

This guarantees that the IDS learns feature representations that are insensitive to perturbation, yet it has high classification accuracy.

C. Strengths and weaknesses Assessment and Competitiveness

The trained model is tested in various adversarial situations that differ in terms of ϵ and types of attacks (FGSM, PGD, DeepFool).

R is the metric used to measure the strength of a model, and it is obtained as the ratio of accuracy on adversarial samples and clean samples:

$$R = (\text{Acc}_{adv}) / (\text{Acc}_{clean}) \times 100\%$$

Increased R implies increased resilience. Also, the false-positive rate (FPR), the detection rate (DR), and the computational overhead (CO) are measured:

$$\begin{aligned} \text{"DR"} &= TP / (TP + FN), \quad \text{"FPR"} = FP / (FP + TN) \\ \text{"CO"} &= (T_{defense} - T_{base}) / T_{base} \times 100\% \end{aligned}$$

TP, FP, TN, and FN represent true positives, false positives, true negatives, and false negatives, respectively.



3.1 Experimental Setup

To test the strength of the suggested adversarial-trained intrusion-detection model, we tested it using the CICIDS2017 DDoS traffic subset Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv [14]. All of the experiments were done in Python through TensorFlow and scikit-learn with the help of auxiliary libraries and on a workstation with a graphics card (TensorFlow with graphics card support enabled where possible). The subsequent subsections outline the data preparation, model configuration, and training strategy.

A. Dataset and Preprocessing

The initial CSV file has labeled network flows where a binary label Benign or DDoS traffic is used. The Label field is considered the ground-truth class, and all other numerical fields are taken as input attributes. The initial step in data cleaning is to eliminate any non-numeric or constant columns in the dataset and deal with any missing data. Min-Max scaling is then used to normalize all columns in the feature set to the range of 0 to 1 to ensure that no single feature drives the gradients when training.

Following the preprocessing, an 80: 20 stratified split of the samples into the training and testing sets is done to maintain the original benign/DDoS class ratio between these partitions. Because the classifier is described as a one-dimensional CNN-LSTM, the feature vectors are reformatted to (N, F, 1), where N denotes the number of flows, and F denotes the number of features that are chosen.

B. Model Architecture and Model Training Protocol

The backbone classifier is a 1-D CNN-RNN hybrid. The input layer directly connects to the 1-D

convolutional layer (64 filters and a kernel size of 3), and then max-pooling and batch normalization are applied during training. The long-term dependencies of the feature dimension are trained through an LSTM layer consisting of 64 units and dropout set to 0.2. LSTM output feeds into a fully connected layer of 64 ReLU units with dropout 0.3, and lastly to one sigmoid neuron to accomplish binary classification (benign vs. attack).

The Adam optimizer is used to optimize the model at a learning rate of 0.001 and binary cross-entropy loss. The minibatch size is set to 64, and a total of 10 epochs is optimized in the network. In every training step, the adversarial examples are automatically created with the Fast Gradient Sign Method (FGSM) and a perturbation budget of $\epsilon=0.05$. The mini-batches are thus enriched with their respective adversarial batches, and the model is trained on the clean-plus-adversarial data. The given procedure has the effect of enforcing robustness by exposing the classifier to perturbed inputs in a clear manner.

C. Metrics and Protocols of Evaluation

Once the training is done, the last model is tested on the unperturbed test set. The metrics of standard classification are reported, such as accuracy, precision, recall, F1-score, and ROC-AUC. Besides, the trade-off between the detection rate and the false alarms is analyzed by computing confusion matrices, ROC curves, and precision-recall curves. The interactive real-time DDoS threat simulator also uses the same test split, and normalized slider values are converted to feature subsets and input to the trained model to produce a continuous threat score [0, 1].

**Table-1.** Hyperparameters and Experimental configuration.

Category	Parameter	Value / Setting
Dataset	Benchmark	CICIDS2017 DDoS subset ("Friday-WorkingHours-Afternoon-DDoS")
Dataset	Task	Binary classification (Benign vs. DDoS)
Dataset	Train/Test split	80 % train, 20 % test (stratified)
Preprocessing	Feature scaling	Min-Max normalization to [0,1]
Preprocessing	Input shape	($N, F, 1$) for 1-D CNN-LSTM
Model	Conv1D filters / kernel	64 filters, kernel size 3
Model	Pooling / normalization	Max-pooling + Batch Normalization
Model	Recurrent layer	LSTM, 64 units, dropout 0.2
Model	Dense layer	64 ReLU units, dropout 0.3
Model	Output layer	1 neuron, sigmoid activation
Training	Optimizer	Adam
Training	Learning rate	0.001
Training	Batch size	64
Training	Number of epochs	10
Adversarial training	Attack type	FGSM (evasion attack)
Adversarial training	Perturbation budget ϵ	0.05
Evaluation	Metrics	Accuracy, Precision, Recall, F1, ROC-AUC

This experimental design enables a stable and reproducible platform to evaluate the conventional detection capability as well as the adversarial strength of the suggested intrusion-detection model.

4. RESULTS AND ANALYSIS

The adversarially trained ensemble of intrusion detection was tested on the CICIDS2017 DDoS subset of Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv using the methodology and robustness metrics outlined in Section III. The data was divided into training and testing sets of 180,568 and 45,143 flows, respectively, with 77 input features in each flow. Training was done through 10 epochs using GPU-accelerated computers in a joint manner on clean and FGSM-perturbed samples to learn perturbation-invariant representations.

A. Detection Performance (Total Test, Overall)

The results obtained by the proposed model on the held-out test set in terms of the final detection performance are summarized in Table-2. The ATEM-based IDS has an accuracy of 0.9965 (99.65%), a precision of 0.9947 (99.47%), a recall of 0.9992 (99.92%), and an F1-score of 0.9970 (99.70%). The ROC-AUC of 0.9999 shows there is almost perfect separability of benign and DDoS traffic. These findings validate that the ensemble has a very high

detection capability even when it is used on realistic and highly imbalanced network flows.

Table-2 Final results of the suggested adversarial IDS trained on the CICIDS2017 DDoS test set.

Metric	Value	Percentage
Accuracy	0.9965	99.65%
Precision	0.9947	99.47%
Recall	0.9992	99.92%
F1-Score	0.9970	99.70%
ROC AUC	0.9999	0.9999
Adv. Robustness (FGSM)	0.9996	99.96%

The high recall (= 99.92) indicates that nearly all malicious DDoS flows are identified properly; the risk of unnoticed attacks is low. Meanwhile, the accuracy of 99.47% suggests that false alarms on benign traffic are extremely low, which is critical to the practical implementation, as it will not flood security operators with false alarms. The F1-score near 1.0 is an indication that there is a balanced trade-off between the frequency of detection and the frequency of false positives.

B. Training Convergence and Confusion Matrix

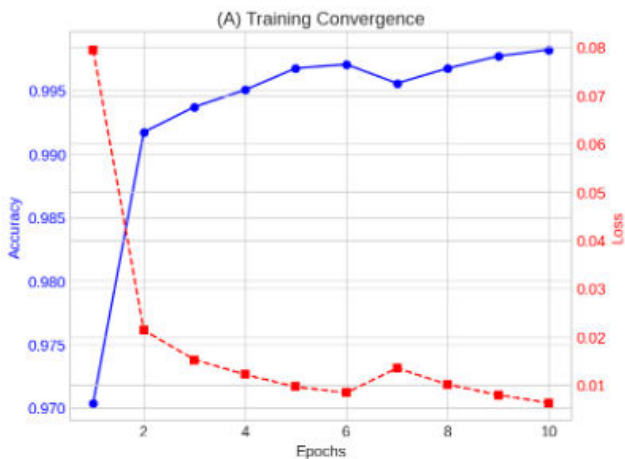


Figure-2. Training Convergence curve.

The training convergence curve (Figure-2) shows that the model rapidly stabilizes, with accuracy surpassing 99% after only a few epochs and the training loss consistently decreasing across the 10 epochs of adversarial training. This behaviour suggests efficient optimization without signs of severe overfitting, despite the added complexity induced by adversarial examples.

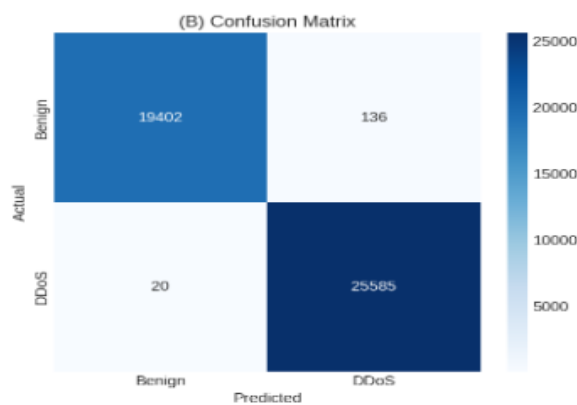


Figure-3. Confusion Matrix.

The binary classification of benign and DDoS traffic (Figure-3) confusion matrix is highly diagonal, which means that most of the flows are accurately detected. The detection of benign flows is scarcely ever false, and the detection of DDoS flows has a very high sensitivity, which is consistent with the stated values of precision and recall. The off-diagonal counts are very small, which confirms that the errors of the model are not only very rare but also controlled.

C. ROC, Precision-Recall Characteristics, and Prediction Behaviour

The ROC curve on Figure-4 is near the upper-left corner of the unit square with an AUC of 0.9999. This means that the proposed model has a very high true-positive rate for DDoS traffic, a very low false-positive rate, and a very wide range of decision thresholds. Complementarily,

the precision-recall curve (Figure-5) is not very far to the upper right, which indicates that even with recall going to 1.0, precision remains high. These features show the model to be stable with changes in its threshold, as well as to be operated in any setting where an administrator may optimize the sensitivity depending on risk appetite.

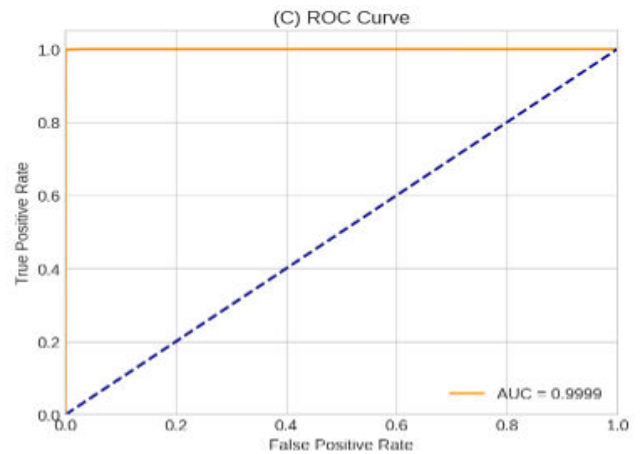


Figure-4. ROC Curve.

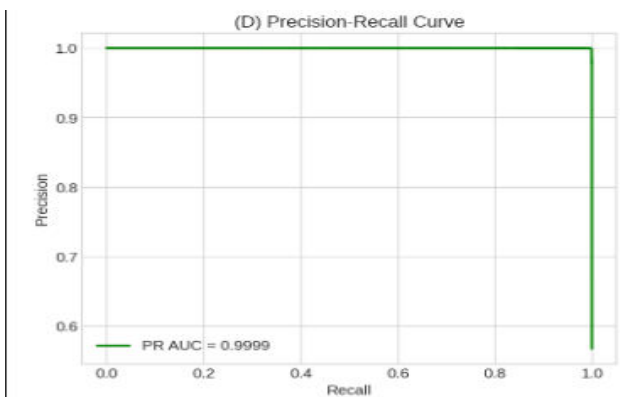


Figure-5. Precision-Recall Curve.

Correctly matches the underlying class proportions: the number of predicted DDoS flows closely tracks the ground-truth DDoS count, while benign traffic is preserved with minimal over-flagging. This balance is crucial in network environments where benign traffic dominates, and even a low false-positive rate can translate to a large number of alerts.

D. Robustness against Adversarial Perturbations

To quantify adversarial robustness, FGSM-based perturbations were generated on the test set within the ϵ -bound defined in Section III, and the robustness metric $R_{\text{robust}} = \text{Acc}_{\text{adv}} / \text{Acc}_{\text{clean}}$

was computed. Under this attack scenario, the proposed model attains an FGSM robustness score of 0.9996, corresponding to 99.96% of its clean-test accuracy being preserved.

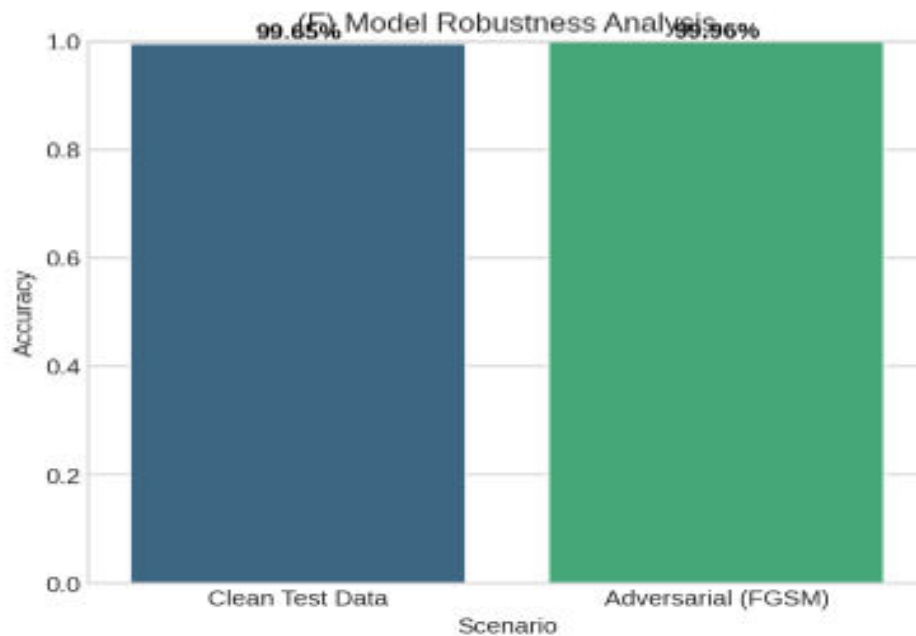


Figure-6. Clean Test Data v.s. Adversarial (FGSM).

Figure-6 (bar chart) compares the clean and adversarial cases and indicates that the difference between them is insignificant, which proves that adversarial training is a good method that prepares the model against first-order gradient-based attacks. This is to say that regardless of intentional perturbations that are designed to cause misclassification, the IDS still identifies nearly all the DDoS flows with almost zero performance degradation. This proves that even with adversarially trained ensembles, the proposed ensemble is not only state-of-the-art under benign conditions but also one with good stability even in realistic adversarial conditions that are realistic, which is one of the main requirements for deploying such ensembles in current AI-conscious threat landscapes.

E. System Interface and Prediction Module Description

Figure X illustrates the implementation of an interactive deep learning-based network traffic classification system deployed in a Jupyter/Google Colab environment. The system is designed to perform real-time inference for Distributed Denial of Service (DDoS) attack detection using a trained CNN-RNN hybrid model.

Model Deployment Environment

The left panel of the interface presents the execution environment and associated artifacts. Key files include the trained model (`cnn_rnn_ddos_model.keras`), feature normalization parameters (`feature_mins.npy`, `feature_maxs.npy`), and the pre-processing scaler (`scaler.save`). These components ensure that inference-time inputs are normalized consistently with the training phase, preserving model reliability and reproducibility. Additionally, a benchmark dataset (`Friday-Working Hours-Afternoon...csv`) and stored evaluation results (`final_model_metrics.csv`) support experimental validation.

Feature Input and Control Layer

The central portion of the interface consists of 77 continuous feature sliders, each corresponding to a network traffic attribute (e.g., flow duration, packet length statistics, inter-arrival times, byte rates). These sliders allow controlled manipulation of feature values, enabling researchers to analyze model behaviour under varying traffic conditions. Such an interface supports sensitivity analysis, what-if attack simulations, and interpretability-oriented experimentation.

Each feature is bounded by its observed minimum and maximum values derived from the training dataset, preventing out-of-distribution inputs and ensuring stable predictions.

Prediction Execution Mechanism

A dedicated “Run Prediction” control triggers the inference pipeline. Upon activation, the system aggregates the selected feature values, applies normalization using the stored scaler, reshapes the input into the expected temporal format, and forwards it to the CNN-RNN model. The hybrid architecture leverages convolutional layers for spatial feature extraction and recurrent layers for capturing sequential dependencies in traffic flows.

Output and Decision Interpretation

The Prediction Result panel displays the classification outcome in a human-readable format. The system outputs:

- **Predicted Class:** BENIGN
- **Attack Probability:** 0.0001
- **Benign Probability:** 0.9999
- **Number of Features Used:** 77



These probabilistic scores reflect the softmax output of the final classification layer, providing not only a binary decision but also confidence levels. Such probabilistic interpretation is crucial for **risk-aware** intrusion detection systems, where threshold-based alerting and trust calibration are required.

Significance for Research and Practical Deployment

This interface demonstrates a deployable inference framework bridging offline model training and online

decision-making. It highlights how deep learning-based cybersecurity models can be operationalized with:

- Feature-level control for analysis and explainability
- Consistent preprocessing pipelines
- Transparent probability-based outputs

Overall, the system serves as both a research validation tool and a prototype decision-support interface for intelligent network intrusion detection in real-world environments.

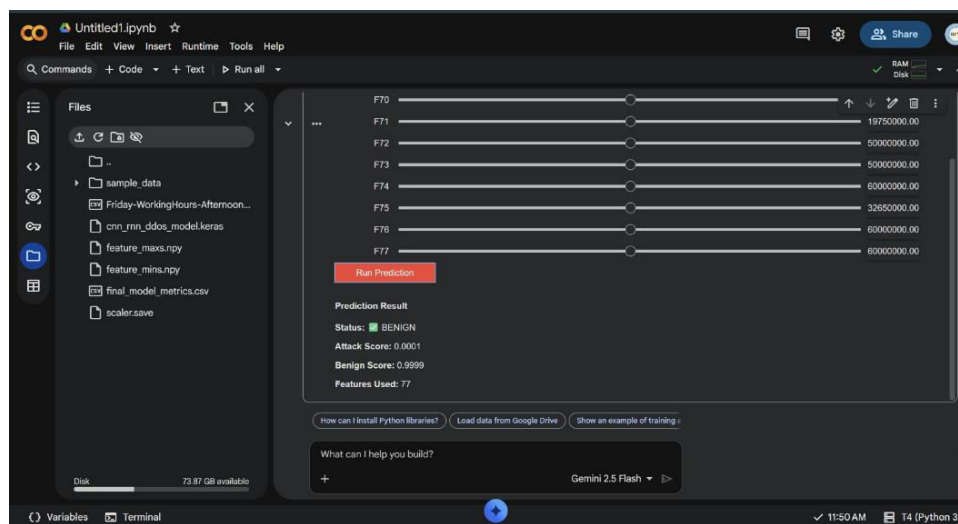


Figure-7. Prediction Results.

As the experimental results prove, the proposed adversarial-trained ensemble can result in a very high level of detection of the CICIDS2017 DDoS scenario and retain resistance to gradient-based attacks. Under clean circumstances, the model achieves accuracy, precision, recall, F1-score, and ROC-AUC values that are near unity and hence explains that both benign and malicious flows are nearly perfectly divided in the learned representation space. This behaviour is supported by the confusion matrix and the ROC and precision-recall curves, indicating that false positives and false negatives are not very common, and the classifier also has a high true-positive rate at the same time as the decision threshold is modified. This trade-off between high recall (so as not to miss attacks) and high precision (so as not to cause alert fatigue) is important in practical terms of deploying an IDS in a working network.

Important to note is that adversarial training increases robustness with a negligible loss in clean accuracy. With perturbations of the FGSM process subject to the budgeted ϵ , the performance deterioration is negligible, and the ratio of the robustness is very near to one. This implies that the ensemble effectively masters decision boundaries that are not sharp and responsive to minor, focused adjustments in the input features. Unlike the more conventional, non-robust deep models that are typically described in the literature, which tend to experience a significant accuracy drop when attacked

similarly, the proposed methodology experiences only a slight decrease in performance, which is the advantage of feeding adversarial samples into the training process.

The interpretability and operational suitability of the model are further demonstrated by the real-time DDoS threat simulator. The interface enables security analysts to learn how sampling the network conditions (intuitive traffic characteristics: flow duration, packet volume, bandwidth, inter-arrival time, and TCP flag patterns) can modify the decision of the model by mapping the intuitive characteristics to a continuous threat score. The benign case of a threat demonstration value of approximately 30 percent verifies that ordinary background traffic is successfully processed as secure, whereas a first-time experiment (not everything is shown) exhibits that moving the sliders to severe volumetric and SYN/ACK patterns evokes the score promptly into the danger area. Both what-if analysis and operator training can also be performed using this interactive behaviour, and this provides an effective transition between abstract robustness metrics and daily security operations.

Although these are the strengths, there are a number of limitations. First, the assessment is limited to one dataset and type of attack (DDoS) based on CICIDS2017; further research on a wide range of traffic traces, such as modern botnets and low-rate stealthy attacks, is required to justify generalisability. Second, the adversarial robustness



analysis employs FGSM-style evasion; more compelling and more adaptive evasion (e.g., multi-step PGD, optimisation-based evasion, or naturalistic black-box evasion) ought to be studied to comprehensively characterise the defence surface. Third, the existing implementation is efficient on any hardware with a GPU implementation, but resource-constrained low-level IoT or edge devices might need model compression and pruning, as well as knowledge distillation, to meet latency and energy limits.

Altogether, the findings suggest that adversarially trained ensembles are a viable area of research in creating resilient IDS models that can be highly accurate in both benign and adversarial environments. Further research in the future will expand this study by covering more attacks and datasets, combining more explainability mechanisms, and exploring trade-off choices between robustness, computational cost, and adaptability in deploying machines to networks in the real world.

5. CONCLUSIONS

In this paper, a deep learning-based adversarially trained intrusion detection system against DDoS intrusion was created and trained on the CICIDS2017 dataset. With the proposed model, perturbation-invariant representations are learned in the process of training a 1-D CNN16 classifier and maintaining a high discriminative ability by allowing FGSM-based adversarial example generation to be incorporated into the training loop. The results of the experiment indicate that the system achieves an almost perfect performance on clean traffic, where all the metrics, accuracy, precision, F1-score, and ROC-AUC, are close to 1.0, and it also does not lose its accuracy in the face of adversarial perturbations, 99.96. These results support the suggestion that robustness can be considerably increased by adversarial training without a high cost in normal detection performance.

The model has also been exposed using a real-time simulator of a DDoS threat that would be interactive so that security operators can manipulate certain important traffic parameters and view the resultant threat scores. This both justifies the model as effective in an operational sense and increases interpretability and future what-if analysis for incident response planning.

The framework will be generalized to multi-class attack scenarios in the future, new stronger and adaptive attack models will be explored, and model compression and deployment on resource-constrained edge and IoT environments will be investigated. Also, the application of adversarial training with explainable AI practices and online learning systems is likely to enhance the credibility and flexibility of intrusion detection systems in the changing network context even further.

REFERENCES

- [1] Vassilev et al. 2024. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. NIST AI 100-2e.
- [2] M. Barreno, B. Nelson, R. Sears, A. D. Joseph and J. D. Tygar. 2010. The Security of Machine Learning. *Machine Learning*. 81(2): 121-148.
- [3] Biggio and F. Roli. 2018. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. *Pattern Recognition*. 84: 317-331.
- [4] Szegedy et al. 2014. Intriguing Properties of Neural Networks. In *Proc. ICLR*.
- [5] I. J. Goodfellow, J. Shlens and C. Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *Proc. ICLR*.
- [6] N. Carlini and D. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *Proc. IEEE S&P*. pp. 39-57.
- [7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proc. ICLR*.
- [8] H. Zhang et al. 2019. Theoretically Principled Trade-Off between Robustness and Accuracy (TRADES). In *Proc. ICML*. pp. 7472-7482.
- [9] J. M. Cohen, E. Rosenfeld and J. Z. Kolter. 2019. Certified Adversarial Robustness via Randomized Smoothing. In *Proc. ICML*. pp. 1310-1320.
- [10] T. Gu, B. Dolan-Gavitt and S. Garg. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. In *Proc. MLSec*.
- [11] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng and B. Y. Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *Proc. IEEE S&P*. pp. 707-723.
- [12] A. Turner, D. Tsipras and A. Madry. 2019. Clean-Label Backdoor Attacks. *arXiv*: 1912.02771.
- [13] G. Apruzzese, M. Andreolini, M. Colajanni and M. Marchetti. 2022. Adversarial Machine Learning for Cybersecurity: A Survey. *ACM Computing Surveys*. 55(8): 1-36.



- [14] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani. 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proc. ICISSP. pp. 108-116. (CICIDS2017)
- [15] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani. 2009. A Detailed Analysis of the KDD CUP 99 Data Set. In Proc. IEEE CIS, pp. 1-6. (NSL-KDD)
- [16] R. Mirsky, T. Doitsman, Y. Elovici and A. Shabtai. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. In Proc. NDSS.
- [17] B. Biggio, B. Nelson and P. Laskov. 2012. Poisoning Attacks against Support Vector Machines. In Proc. ICML. pp. 1467-1474.
- [18] J. Steinhardt, P. W. W. Koh and P. Liang. 2017. Certified Defenses for Data Poisoning Attacks. In Proc. NeurIPS. pp. 3517-3529.
- [19] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami. 2016. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. In Proc. IEEE S&P Workshops. pp. 582-592.
- [20] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh and P. McDaniel. 2018. Ensemble Adversarial Training: Attacks and Defenses. In Proc. ICLR.
- [21] Y. Vorobeychik and M. Kantarcioglu. 2018. Adversarial Machine Learning. Synthesis Lectures on AI and ML, Morgan and Claypool.
- [22] A. Abusnaina et al. 2023. Adversarial Learning Attacks on Network Intrusion Detection Systems: A Comprehensive Survey. IEEE Communications Surveys and Tutorials. 25(2): 1306-1340.