I.J. Information Technology and Computer Science, **2026, 4, 170-198**
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2026.04.11

Adversarial Transferability in AI-based Network Intrusion Detection: A Comparative Study of ANN and CNN Models

Aasim Zafar
Department of Computer Science, Aligarh Muslim University, Aligarh, Uttar Pradesh, 202002, India
E-mail: azafar.cs@amu.ac.in
ORCID iD: https://orcid.org/0000-0003-1331-014X

Shazra Wali
Department of Computer Science, Aligarh Muslim University, Aligarh, Uttar Pradesh, 202002, India
E-mail: walishazra@gmail.com
ORCID iD: https://orcid.org/0009-0009-7812-5595

Sheikh Burhan ul Haque*
Department of Computer Applications, Cluster University of Srinagar, Jammu and Kashmir, 190008, India
E-mail: sheikh.burhan@cusrinagar.edu.in
ORCID iD: https://orcid.org/0000-0003-3306-5671
*Corresponding author

Received: February 25, 2026; Revised: April 16, 2026; Accepted: June 15, 2026; Published: August 8, 2026

Abstract: Network Intrusion Detection Systems (NIDS) play a vital role in modern cybersecurity by leveraging artificial intelligence (AI), particularly deep learning (DL) and machine learning (ML), to detect and mitigate malicious activities. However, these AI-driven systems are highly vulnerable to adversarial attacks, where small, imperceptible perturbations in input data can deceive models and significantly reduce detection accuracy. This raises critical concerns about the security and reliability of intrusion detection, especially in real-world scenarios where attackers exploit adversarial transferability to bypass defenses. This research investigates the threat posed by black-box adversarial attacks via surrogate models, focusing on the ability of adversarial examples to transfer across different architectures. This study simulates real-world adversarial threats, demonstrating how attacks crafted on one model can effectively deceive another, compromising NIDS security. A comparative study is conducted on two widely used AI models: an Artificial Neural Network (ANN) and a Convolutional Neural Network (CNN), both trained on the CICIDS 2019 dataset. The study evaluates the robustness of these models against two gradient-based adversarial attack methods, Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), to determine their susceptibility under black-box adversarial conditions. Experimental results indicate that CNN-based NIDS are more vulnerable to adversarial attacks than ANN-based models, with adversarial examples successfully transferring across architectures. These findings highlight the critical risks associated with adversarial transferability, underscoring the need for enhanced security measures to strengthen AI-driven intrusion detection systems against evolving cyber threats.

Index Terms: Adversarial Attacks, Network Security, Intrusion Detection, Deep Learning, Machine Learning.

1. Introduction

With the rapid expansion of digital networks, cybersecurity threats have evolved in complexity, making traditional security mechanisms inadequate for safeguarding sensitive systems. One of the most critical components in modern cybersecurity is NIDS, which monitors and analyzes network traffic to identify malicious activities, unauthorized access, and potential cyberattacks. NIDS can be broadly categorized into signature-based and anomaly-based detection approaches. Signature-based NIDS, such as Snort and Suricata, rely on predefined attack signatures to detect known threats, making them highly effective against previously documented attacks. Fig. 1 illustrates the various types of NIDS approaches. However, this method fails against zero-day exploits and polymorphic malware, where attack patterns

continuously evolve. On the other hand, anomaly-based NIDS use statistical analysis and machine learning techniques to identify deviations from normal network behavior, providing a more adaptive approach [1]. However, these models often suffer from high false-positive rates, limiting their practical usability. DL approaches, including ANNs and CNNs, offer superior feature extraction capabilities by automatically learning hierarchical patterns from raw network traffic data. DL models have demonstrated strong performance in intrusion detection by recognizing spatial correlations in network traffic features [2]. These models continuously evolve and adapt, making them highly effective against sophisticated and previously unseen threats. Unlike traditional NIDS, which rely on static signatures, AI-driven models analyze network traffic dynamically, making them more effective against evolving threats. When a network traffic packet enters a system, ML models assess its behavior in real-time, identifying unusual deviations even if the attack method is slightly altered. For example, if an attacker disguises malicious traffic by modifying packet headers or payload structure, DL models can still detect hidden patterns by analyzing the packet's byte sequences and flow characteristics. This ability to recognize behavioral anomalies, rather than relying solely on predefined rules, allows AI-powered NIDS to adapt and respond proactively to new and sophisticated cyber threats.

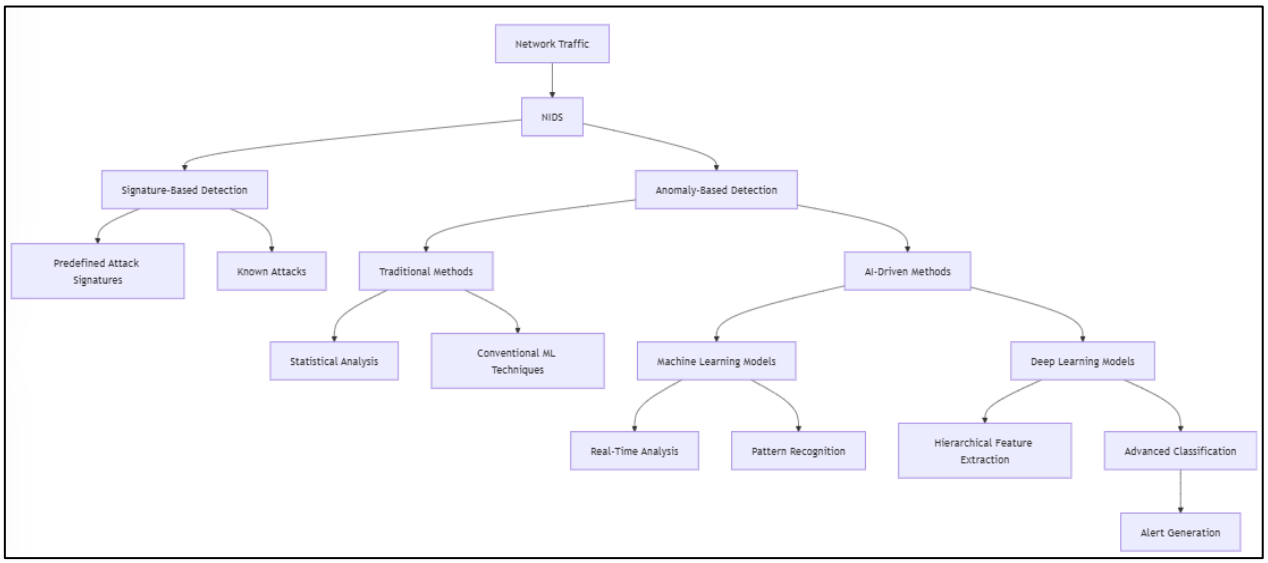


Fig.1. Various Network Intrusion Detection System (NIDS) approaches

ML and DL have significantly improved NIDS by enhancing their ability to identify and mitigate cyber threats [3-4]. However, ML models often struggle to adapt to evolving attack patterns, while DL models, despite their superior performance, are computationally intensive and highly susceptible to adversarial attacks [5-6]. Adversarial examples—small, carefully designed modifications to input data—can deceive these models, leading to misclassification and security breaches. One of the most critical concerns is adversarial transferability, where an attack crafted on one model can effectively deceive another [7]. This is particularly problematic in black-box attack scenarios, where attackers do not have direct access to the target model but can still manipulate it using surrogate models. These vulnerabilities highlight the urgent need for more robust AI-driven NIDS, as cyber threats continue to evolve and exploit weaknesses in AI-based security systems. Understanding how different model architectures react to adversarial manipulations is crucial for developing effective countermeasures and improving the reliability of intrusion detection.

This study addresses the research gap in evaluating the adversarial robustness of AI-driven NIDS by comparing the vulnerability of two widely used architectures: ANNs and CNNs. Unlike previous research that primarily focused on model accuracy, this study investigates adversarial transferability, assessing how attacks generated on one model can deceive another. ANNs and CNNs are selected due to their distinct learning approaches: ANNs process data sequentially, making them effective for general classification, while CNNs utilize convolutional filters to extract spatial dependencies, which are crucial for analyzing network traffic structures. To simulate real-world black-box attacks, a surrogate model approach is used, where an attacker trains a substitute model to approximate the target model's behavior. Adversarial examples are generated using FGSM [8-9] and PGD [10-11], with FGSM providing quick perturbations and PGD generating stronger attacks. These perturbations are crafted using real-world traffic samples from the CICIDS 2019 dataset [12-14], a benchmark dataset containing a diverse mix of benign and malicious network traffic. By evaluating the impact of these attacks on ANN and CNN models, this study aims to provide insights into their security vulnerabilities and the effectiveness of adversarial transferability in compromising AI-based intrusion detection systems.

The key motivation of this study is to address the challenges faced by CNN and ANN-based NIDS. With ever-emerging technology in AI, the vulnerabilities posed by adversarial attacks are severe. Although DL models such as ANN and CNN have significantly enhanced threat detection capabilities, their vulnerability to adversarial perturbations—especially in black-box scenarios—raises serious security concerns. This research seeks to bridge a critical gap by comparing the adversarial robustness of ANN and CNN, two models that embody distinct learning paradigms. By

employing FGSM and PGD attacks, which represent varying intensities and methodologies of adversarial manipulation, this study aims to elucidate how adversarial examples crafted on surrogate models can transfer across architectures, thereby compromising detection accuracy. Understanding these dynamics is paramount for developing more resilient NIDS frameworks capable of countering the evolving tactics of cyber adversaries.

The significant contributions of the research study are as follows: Addressing the gap by systematically evaluating the susceptibility of ANN and CNN models to adversarial attacks.

- Crafting adversarial examples on real-world network traffic (CICIDS2019 dataset) to assess attack effectiveness in practical scenarios.
- Investigating the impact of different attack intensities (FGSM vs. PGD) to determine how varying perturbation strengths affect model performance.
- Analyzing adversarial transferability, highlighting the security risks posed by black-box attacks.
- Providing insights into AI-based NIDS vulnerabilities, emphasizing the need for more secure AI-driven intrusion detection systems. Providing a benchmark study for future adversarial research in NIDS, aiding researchers in developing more resilient intrusion detection frameworks.

2. Related Works

Several studies have explored the vulnerability of ML and DL-based NIDS to adversarial attacks. Haroon and Ali [15] investigate the susceptibility of machine learning-based intrusion detection systems (IDS) to adversarial attacks. They create adversarial samples on typical datasets such as NSL-KDD, UNSW-NB15, and CICIDS17 using methods such as FGSM and JSMA, and then evaluate the robustness improvements achieved through adversarial training. Their findings show that, while adversarial training can improve the resilience of IDS models, it does not fully alleviate the negative impacts of such assaults.

This review paper by Shilin Qiu et al. [16] provides a comprehensive overview of adversarial attacks and defense mechanisms in DL. It categorizes adversarial attacks based on the stage at which they occur, during training or testing, and discusses various defense strategies, such as modifying data, adjusting model architectures, and employing auxiliary tools. The paper highlights that although numerous attack methods have been proposed across domains like computer vision and cybersecurity, achieving robust defenses remains challenging. It also identifies key research gaps, underlining their critical importance, and offers future directions for improving the resilience of AI systems against adversarial threats.

Similarly, Šircelj and Škocaj [17] introduce the "accuracy-perturbation curve" as a novel metric for evaluating classifier robustness. This framework visualizes how a model's accuracy degrades with increasing adversarial perturbation, providing a more nuanced and hyperparameter-independent method to compare different attack techniques and the effectiveness of adversarial training.

Another study by Alatwi et al. [18] highlighted the role of feature selection in cybersecurity systems by demonstrating how different feature sets affect the transferability of adversarial attacks. Their findings indicated that certain feature selection methods could significantly reduce the success of attacks.

The paper by Ibitoye et al. [19] investigates the impact of adversarial attack techniques—such as FGSM, PGD, and JSMA—on DL-based Network Intrusion Detection Systems (NIDS). Through extensive experiments on standard benchmark datasets, the authors demonstrate that even minor perturbations can lead to significant misclassifications in both shallow and deep models. The study underscores the vulnerability of current IDS implementations to adversarial manipulation and discusses the trade-offs between attack strength and detection performance. The results underscore the critical need for more robust defense mechanisms in NIDS, highlighting the importance of this study's conclusions.

Shahad Alahmed et al. [20] propose a generative adversarial network (GAN)-based approach to defend ML models for NIDS against adversarial attacks. They used a black-box attack, namely, the Zeroth-Order Optimization (ZOO) attack, to evaluate the resilience of the designed system. The experiment was carried out using a realistic and recent public network dataset: CSE-CICIDS2018. The simulation results showed that GAN-based adversarial training enhanced the resilience of the IDS model and mitigated the severity of the black-box attack.

Samar M. Nour et al. [21] proposed a performance evaluation model for enhancing NIDS using DL. It explores CNNs, RNNs, Attention Mechanisms, and DNNs while addressing challenges like scalability, interpretability, and adversarial attacks. Experimental results on real-world datasets show CNNs outperform other models in detecting cyber threats.

In study [22], authors explore the use of adversarial machine learning in network security. It begins with a discussion of machine learning and adversarial attacks. They proposed a new method for classifying adversarial attacks in network security. They also introduce the concept of space and feature space dimensional classification of adversarial attacks in network security. Finally, they introduce the concept of adversarial risk in computer and network security. They provide a comprehensive risk mapping for evaluating the risk of adversarial attacks in network security, based on the discriminative or directive autonomy of the machine learning tasks and techniques, respectively, to reassure the audience of the thoroughness of the research.

Reference [23] investigates the vulnerability of DL-based face mask detection systems to black-box adversarial attacks. The authors demonstrate that by using a substitute model, they can generate adversarial examples that

significantly reduce the accuracy of a target face mask detection model. This research highlights the security risks associated with adversarial transferability in surveillance systems, especially in applications like pandemic monitoring. The study provides empirical evidence by showing a reduction in the model's accuracy from 97.18% to 46.52% using the black-box attack method.

In [24], the authors study the rigorous evaluation process by training a DL model to detect network intrusions. They meticulously assessed the model's performance in the presence of adversarial examples, using two comprehensive datasets for Network Intrusion Detection. Their white box attack on the trained deep-learning model was a testament to their thoroughness. They evaluated how different feature sets affected adversarial attack success, revealing that the DL model was vulnerable, and that feature choice played a critical role in attack effectiveness.

Dingcheng Yang [25] proposed a method for training a surrogate model with dark knowledge to boost the transferability of the adversarial examples generated by the surrogate model. The proposed method consists of two key components: a teacher model extracting dark knowledge and a mixing augmentation technique that enhances the dark knowledge of training data.

Md. Ahsan Ayub et al. [26] focus on the challenge of adversarial attacks against machine learning-based IDS. The authors developed a machine learning approach for intrusion detection using a Multilayer Perceptron (MLP) network. They demonstrated the effectiveness of their model on two network-based IDS datasets, achieving high accuracy in detecting malicious traffic.

The paper [27] addresses the vulnerability of machine learning-based NIDS to adversarial examples, which are malicious data samples designed to evade detection. Vivek Kumar, Kamal Kumar and Maheep Singh proposed a novel approach using a variational autoencoder (VAE) to generate adversarial examples that not only deceive NIDS but also comply with domain constraints, ensuring their practicality. By perturbing the latent space representation of data samples, the method crafts adversarial examples under limited perturbation, achieving an attack success rate of 64.8% against ML/DL-based NIDS.

Studies have shown their impact on NIDS. For instance, Sharma et al. (2024) [28] analyzed their effects on machine learning models. Similarly, elShehaby et al. (2024) [29] examined the practicality of these attacks in real-world scenarios and proposed strategies to enhance model resilience against them. These perturbations are crafted using real-world traffic samples from the CICIDS2019 dataset to evaluate their impact on NIDS. This dataset, which includes a diverse mix of benign and malicious network traffic, serves as a benchmark for assessing how adversarial examples deceive ANN and CNN models. Abood et al. [30] (2025) demonstrated the practical application of CICIDS 2019 for classifying various DDoS attack types using machine learning techniques, highlighting its effectiveness and relevance in cybersecurity research.

While these studies provide valuable insights into adversarial attacks on AI-driven NIDS, they have certain limitations. Most existing research focuses on improving detection accuracy or applying adversarial training, but few studies have systematically analyzed adversarial transferability across different architectures. Additionally, prior work has primarily evaluated individual models rather than comparing their robustness against black-box adversarial threats. Table 1 summarizes the techniques, models, datasets and key highlights of existing studies in adversarial attacks against NIDS. The novelty of this study lies in its comparative analysis of ANN and CNN architectures, examining how adversarial examples generated using FGSM and PGD transfer between models. By utilizing a surrogate model approach on the CICIDS2019 dataset, this research provides new insights into the vulnerability of AI-based NIDS and highlights the security risks posed by adversarial manipulation. The findings will contribute to the development of more resilient intrusion detection frameworks capable of countering evolving cyber threats.

Table 1. Existing work in adversarial attacks and NIDS

Study	ML and DL Algorithms	Adversarial Attack Technique	Adversarial Defense Technique	Dataset	Metrics	Key Highlights
Haroon et al. [15] (2022)	ML based DT, RF, SVM, KNN, LR, NB	JSMA, FGSM	Adversarial Training	NSL-KDD, UNSW-NB15, CICIDS-17	Accuracy, F1-Score, AUC	Models trained with adversarial samples showed improved robustness compared to those without adversarial training. K-NN performed better in NSL-KDD and UNSW-NB15 datasets, while Random Forest was superior in CICIDS-17. Overall, Naïve Bayes was the worst performer.
Qiu et al. [16] (2019)	DNN, SVM	FGSM, JSMA, C&W, DeepFool, HOUDINI, One Pixel Attack, ZOO, NES, ANGRI, ANTs, Momentum	Adversarial Training, Gradient Hiding, Blocking Transferability, Data Compression, Data Randomization, Regularization, Defensive Distillation, Feature	MNIST	Misclassification Rate, Success Rate	Comprehensive review of adversarial attack and defense techniques. Categorizes attacks by training vs. testing stage. Explores defense methods like adversarial training and gradient hiding. Emphasizes challenges in defense and future research directions.

			Squeezing, Deep Contractive Network, Mask Defense, Parseval Networks, Defense-GAN, MagNet, High-Level Representation Guided Denoiser (HGD)			
Šircelj et al. [17] (2021)	CNNs, MLP, RBFNs, LR	FGSM, BIM, Deep Fool, AutoPGD	Adversarial Training	MNIST, CIFAR-10	Accuracy, Image-Perturbation	Introduced "accuracy-perturbation curves" for evaluating robustness. Compared FGSM, BIM, DeepFool, AutoPGD on various classifiers. Found RBFNs more resilient to adversarial attacks. Showed adversarial training enhances robustness and reduces attack transferability.
Alatwi et al. [18] (2021)	ML based SVM,NB, MLP, LR, DT,RF,KNN, AE, DAGMM,AnoGAN, ALAD,DSVDD,OC-SVM,IF, GMM, SAE,RBM, GB DL based CNN,DNN,GAN,LSTM	Evasion, Posioning, Oracle	AT , EL	NSL-KDD, CICIDS2017, KDDCup99, UNB-CICT or ADFA-LD, DREBIN, UNSW-NB15, MAWILab, Kyoto2006+ WSN-DS, CICIDS2018 , DARPA SYN flood Kitsune CTU-13	TPR, FPR, FNR, TNR, FAR, F1Score, Accuracy, Recall, Precision, AUC, Detection Rate	NIDSs are vulnerable to adversarial attacks. Black-box attacks work without model knowledge. Small perturbations can severely degrade NIDS performance. More research is needed for robust defenses.
Ibitoye et al. [19] 2023	decision trees, SVM, NN, clustering, CNN, RNN	FGSM, PGD, JSMA.	Adversarial training, input validation, network hardening.	NSL-KDD, CICIDS2017, MNIST, CIFAR10 Image net	Accuracy, precision, recall, F1-score, detection rate, False positive rate.	ML/DL effectiveness, model vulnerability, defense effectiveness.
Shahad Alahmed et al [20] (2022)	RF, GANs	ZOO	GAN-based Adversarial Training	CSE-CIC-IDS2018	Accuracy, Precision, Recall, F1-Score, MSE, AUC-ROC	GAN-based adversarial training enhances IDS resilience PCA-based feature selection improves performance ZOO black-box attack significantly reduces IDS accuracy Compared with previous studies, the proposed model showed superior accuracy and robustness
Samar M.Nour [21](2024)	ANN,CNN,	Perturbation-Based Adversarial Attacks	Adversarial training, Feature Engineering	CICIDS	Accuracy, Precision, Recall, F1-score, AUC-ROC,	Deep learning model for intrusion detection, performance comparison with ML models, adversarial robustness evaluation, realistic dataset usage
MARIA MA et al.[22] (2021)	RNN,CNN, KNN,RF	FGSM,JSMA,DeepFool,GAN	Adversarial training, Input transformation techniques	NSL-KDD	accuracy, true positive rate, and misclassification rates	RNN-IDS performed better than traditional ML models but is still vulnerable to adversarial attacks.

Hang et al. [23] (2020)	CNN	Black-box attacks (FGSM, PGD) with SCES and SPES ensemble strategies	Adversarial training and ensemble adversarial training	MNIST, GTSRB, USPS	Success rate, transfer rate	More diverse substitute models improve attack success; adversarial training is not fully effective
Hesamodin et al. [24] (2022)	DNN	White-box based FGSM attack	NA	CIC-IDS2017, CIC-DDoS2019	Attack success rate	NIDS models are vulnerable to adversarial attacks, with Forward and Flow-based features being most effective for attacks.
Yang et al. [25] (2023)	ResNet18, DenseNet121, MobileNetV2	-Transfer-based black-box attack. - DSM	NA	CIFAR-10, ImageNet	Success rate	Dark Knowledge improves adversarial transferability significantly, outperforming standard surrogate models.
Ayub et al.[26] (2020)	MLP	JSMA	Adversarial training, gradient masking, and decision boundary	CICIDS2017, TRAbID 2017	Accuracy	The JSMA attack significantly reduces IDS accuracy, showing its vulnerability to adversarial samples. Countermeasures are suggested but not implemented.
Vivek et al.[27] (2024)	VAE	Grey-box attack using latent space perturbation	No explicit defense applied, but study suggests integrating trained models into IDS for improved robustness.	CICIDS2017	success rate	Adversarial examples follow domain constraints, making them more realistic. Traditional attacks may generate invalid network data samples.

3. Dataset Description

The CICIDS 2019 dataset is a well-established benchmark for NIDS, offering realistic network traffic with labeled benign and attack instances. Developed by the Canadian Institute for Cybersecurity (CIC), the dataset replicates real-world network conditions by simulating both normal user behavior and various cyberattacks, making it highly relevant for intrusion detection research. It includes a wide range of attack types such as DDoS, brute force attacks, botnets, web-based exploits, and infiltration attempts. The dataset is structured chronologically, with each day's traffic representing specific attack scenarios: Monday contains only benign traffic, serving as a baseline. Tuesday captures brute force attacks targeting FTP and SSH services. Wednesday includes Denial of Service (DoS) attacks and the Heartbleed vulnerability. Thursday features web-based threats like SQL injection, cross-site scripting (XSS), and an infiltration attack. Friday, the focus of this study, contains a mix of benign traffic and Distributed Denial of Service (DDoS) attacks, making it particularly valuable for analyzing large-scale network disruptions.

The dataset is available in PCAP and CSV formats, allowing for diverse cybersecurity applications, including machine learning-based intrusion detection and adversarial attack research. Compared to traditional datasets such as KDDCUP-99, NSL-KDD, and CICIDS-2017, CICIDS 2019 offers a more modern and comprehensive representation of network threats. It provides higher feature richness, updated threat models, and realistic traffic patterns, addressing the limitations of older datasets that lack modern attack vectors and diverse network behaviors.

This study specifically focuses on a DDoS attack subset extracted from the Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv file, which includes both normal and malicious traffic. The dataset contains 41 statistical network traffic features, covering flow-based metrics, packet characteristics, protocol-specific flags, and behavioral indicators. Due to its realistic attack diversity, this file is ideal for evaluating how adversarial attacks impact intrusion detection models in distinguishing between benign and malicious activities.

3.1. Data Preprocessing

The dataset underwent several preprocessing steps to ensure data consistency, improve model performance, and facilitate adversarial robustness evaluation. The first step involved data cleaning and handling missing values. The dataset was checked for null and infinity values, and any missing values were either removed or replaced using statistical imputation techniques to maintain data integrity. Next, feature selection and scaling were performed. A predefined subset of features was manually selected using the column's list, which included flow metrics, flag counts, and statistical measures. This ensured that only the most relevant network traffic characteristics were used for training. To standardize the dataset, scikit-learn's StandardScaler function was applied, transforming each feature by subtracting the mean (μ) and dividing by the standard deviation (σ), as shown in Equation (1). The distribution of the dataset is shown in Table 2.

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

This normalization process ensured that all numerical features were on a similar scale, preventing any single feature from dominating the model's learning process. Following feature selection, data splitting was carried out to divide the dataset into training and testing sets. The dataset was split into 60% training and 40% testing using the train_test_split function from scikit-learn, with a random state parameter of 42 to ensure reproducibility. Finally, adversarial robustness preprocessing was integrated to assess the impact of adversarial attacks on intrusion detection models. The Adversarial

Robustness Toolbox (ART) was used to generate adversarial examples, and KerasClassifier was applied to train the model while incorporating adversarial training techniques, ensuring robustness against attacks. Fig. 2 visualizes the preprocessing steps. These preprocessing steps ensured that the dataset was well-structured, enabling effective training, evaluation, and adversarial robustness analysis in network intrusion detection systems.

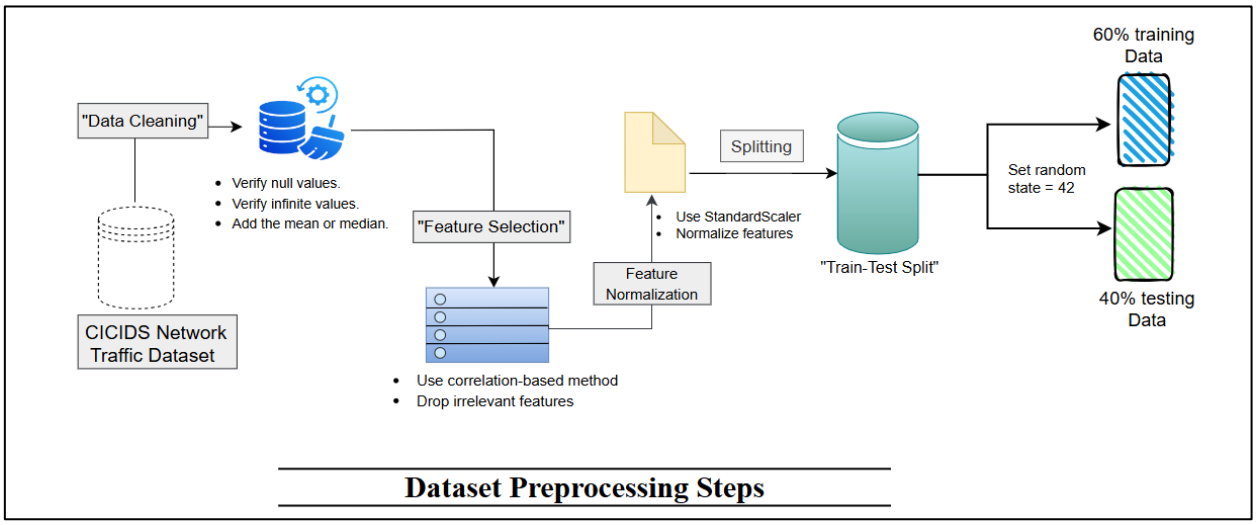


Fig.2. Data preprocessing

Table 2. Dataset composition (Number of samples for each class)

Details	Counts
Total dataset samples	197,718
Training samples	118,630
Testing samples	79,088
Training - Testing Split	60% - 40%
Random state	42
Total benign and attack samples	('BENIGN', 'DDoS') – (97,718, 100,000)
Training benign and attack samples	('BENIGN', 'DDoS') – (58,631, 60,000)
Testing benign and attack samples	('BENIGN', 'DDoS') – (39,087, 40,000)

4. Methodology

This study primarily focuses on analyzing adversarial transferability within the context of NIDS. It examines how adversarial examples created using one model (the surrogate) can be effectively transferred to compromise another model (the target) in a black-box attack scenario. The study involves the development and training of two DL models, ANN and CNN, as primary intrusion detection models. These models are designed to learn and classify network traffic as benign or malicious, using the CICIDS 2019 dataset. The dataset is preprocessed by normalizing features and converting categorical data into numerical form before training. The ANN model is constructed by stacking multiple fully connected layers, while the CNN model leverages convolutional layers to extract spatial patterns from the input data. Both models are trained using an optimization algorithm that minimizes classification loss, and they are evaluated on a separate test set to ensure generalization. However, a critical aspect of model evaluation is robust testing. DL models are vulnerable to adversarial attacks, so their robustness is rigorously tested by generating adversarial examples and evaluating their effectiveness in a black-box setting**.** Fig. 3 illustrates the basic architecture of proposed attack on DL model via surrogate model.

In a black-box attack, the attacker does not have direct access to the target model's parameters or training data. Instead, the attacker's key strategy is to train a separate surrogate model. This surrogate model plays a crucial role in approximating the decision boundary of the target model. It is trained on a subset of the CICIDS 2019 dataset and is designed to replicate how the primary ANN and CNN models classify network traffic. The surrogate ANN model is a smaller feedforward neural network with fewer layers and neurons than the target ANN model, while the surrogate CNN model has a reduced number of convolutional and fully connected layers but still mimics the decision boundary of the target CNN model.

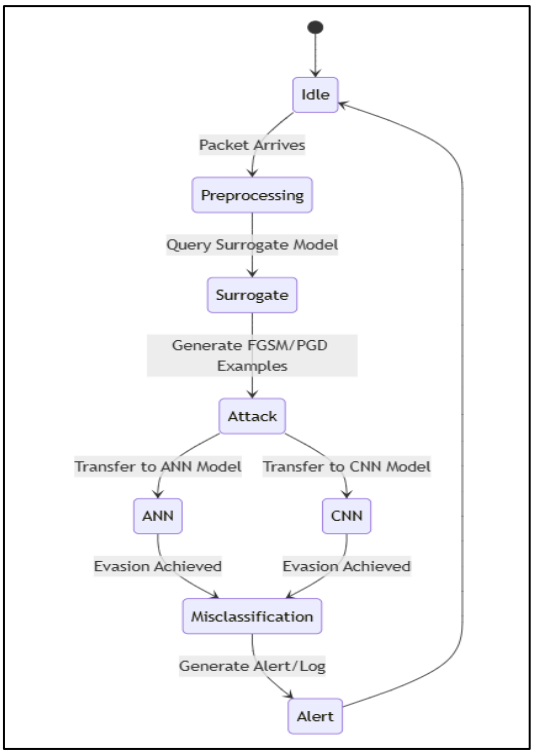


Fig.3. Basic overview of proposed attack on DL models

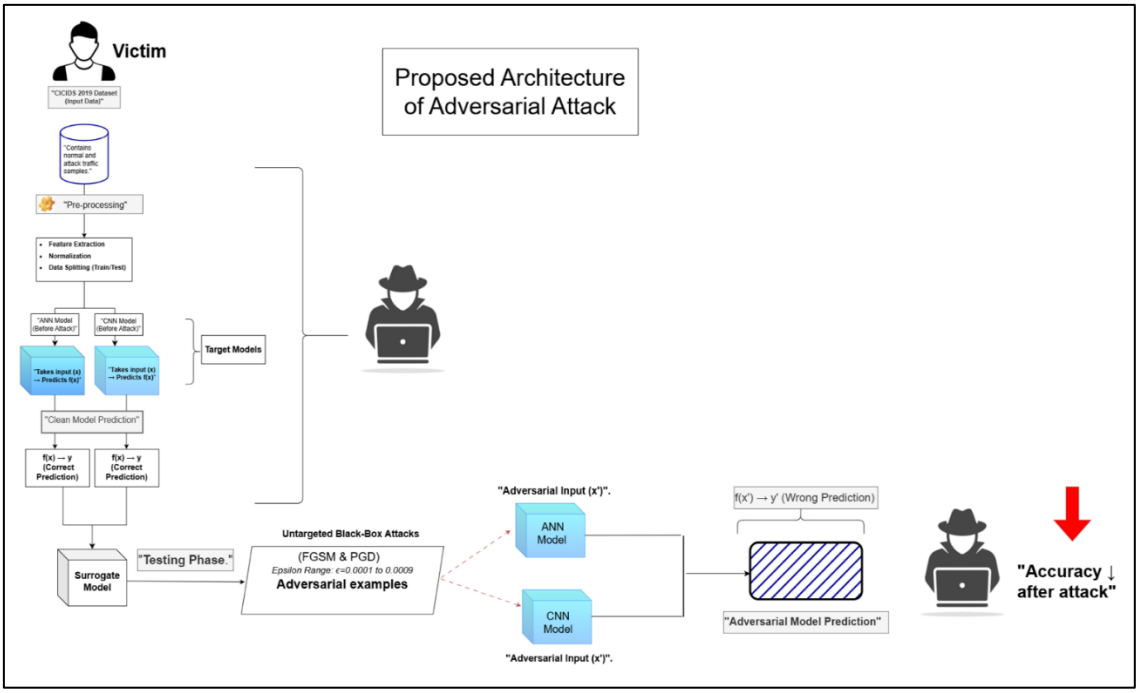


Fig.4. Proposed architecture of adversarial attack in DL NIDS

These primary models, which are the ultimate targets of the attack, are then subject to the adversarial examples generated from the surrogate models. After training, gradient-based attacks, FGSM and PGD are applied to generate these adversarial examples. If the adversarial examples successfully mislead the primary models, it demonstrates the transferability of adversarial attacks across different architectures. Fig. 4 illustrates the workflow of the proposed black-box adversarial attack on target model (ANN and CNN) via surrogate model.

4.1. Architecture of basic ANN and CNN Models

ANNs are computational models inspired by the human brain, designed to process numerical inputs and extract patterns for decision-making. The structure of an ANN consists of an input layer, one or more hidden layers, and an output layer, where each layer comprises neurons that apply mathematical transformations. Fig. 5 illustrates the basic ANN DL

Model. During training, ANNs adjust weights and biases through activation functions and updates via backpropagation and gradient descent. Table 3 defines the details of hyperparameters of ANN. They can adapt and improve themselves. This makes them useful for tasks like classification, regression, and function approximation. They work well in many areas, such as finance, healthcare, and cybersecurity.

Table 3. Hyperparameters and their details for ANNs

Hyperparameters	Details
Learning Rate (α)	Controls the step size of weight updates during training. A smaller value leads to slow convergence, while a larger value may cause instability.
Number of Hidden Layers	Determines the depth of the network, affecting its ability to capture complex patterns.
Number of Neurons per Layer	Specifies how many neurons exist in each hidden layer, influencing the model's capacity and computational cost.
Activation Function	Defines the transformation applied to the input signal in each neuron, introducing non-linearity (e.g., ReLU, Sigmoid, Tanh).
Batch Size	The number of training samples processed before the model updates its parameters. Smaller batches improve generalization but increase noise in updates.
Epochs	The number of times the entire dataset is passed through the network during training.
Optimization Algorithm	Defines how the model updates weights (e.g., SGD, Adam, RMSprop) to minimize the loss function.
Loss Function	Measures how well the model's predictions match the actual output (e.g., Cross-Entropy for classification, MSE for regression).
Dropout Rate	The fraction of neurons randomly dropped during training to prevent overfitting.
Weight Initialization	The method is used to set initial weights (e.g., Xavier, He initialization) to improve convergence.

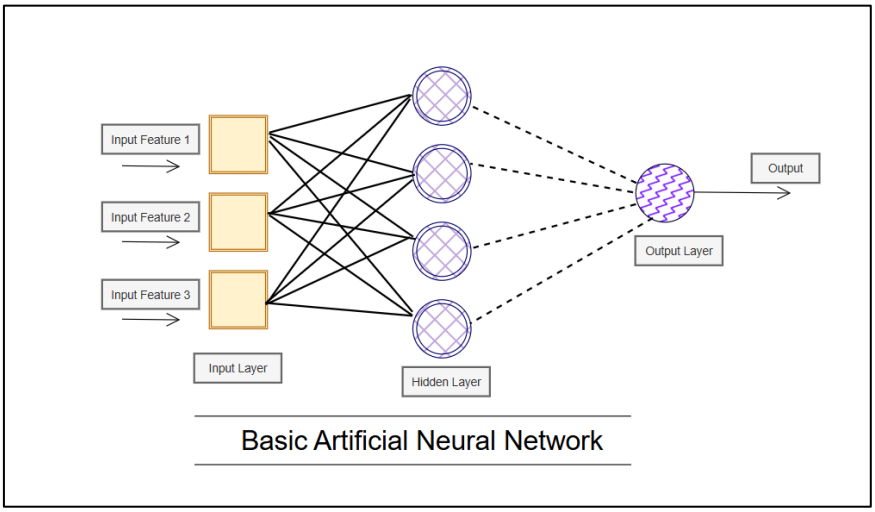


Fig.5. Basic ANN model

CNNs represent specialized DL architecture designed for the analysis of image and spatial data. In contrast to traditional ANNs, CNNs incorporate convolutional layers that facilitate the automatic extraction of spatial features from the input data. This advancement substantially diminishes the requirement for manual feature engineering, enhancing the efficiency of the model development process. The architecture consists of an input layer, convolutional layers that apply filters to detect patterns, pooling layers that down sample the feature maps, fully connected layers that interpret the extracted features, and an output layer for classification or regression. Fig. 6 illustrates the basic architecture of CNN DL model. CNNs, with their widespread use in various applications such as image recognition, object detection, and medical image analysis, have proven to be a reliable and versatile technology. Table 4 defines the details of hyperparameters of CNN.

Table 4. Hyperparameters and their details for CNNs

Hyperparameters	Details
Kernel Size	The size of the filter used in the convolutional layer to extract features from the input image. Common sizes are (3x3) or (5x5).
Stride	Defines the steps size at which the filter moves across the input image. Larger strides reduce feature map size but may lose information.
Padding	Determines whether extra pixels (zero-padding) are added around the input image to maintain spatial dimensions after convolution
Number of Filters (Kernels)	Specifies the number of feature detectors applied in a convolutional layer, affecting feature extraction complexity.
Pooling Size	Defines the window size for pooling operations (e.g., 2×2 for max or average pooling), which helps reduce dimensionality and retain important features.

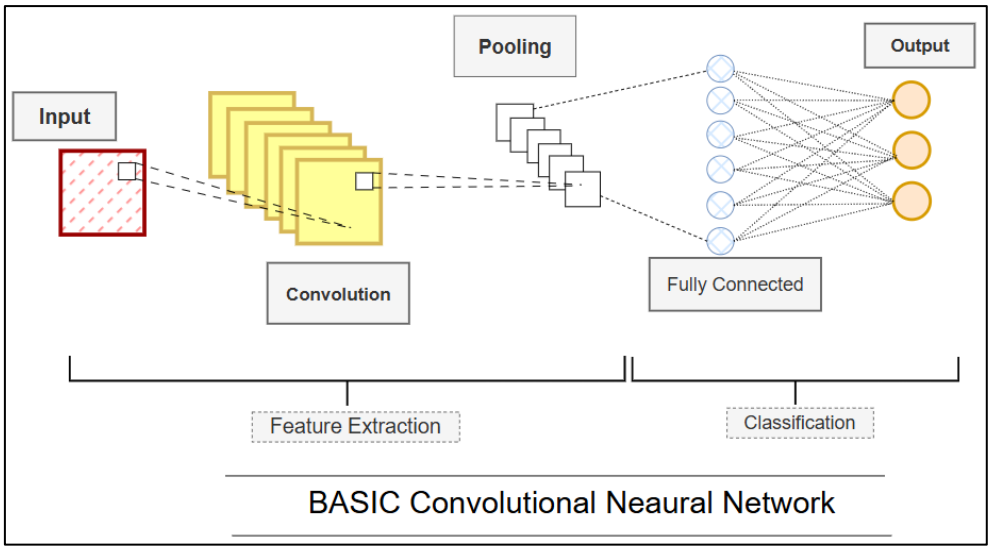


Fig.6. Basic CNN DL model

4.2. Architecture of Proposed Target ANN Model

The target Artificial Neural Network (ANN) model is a carefully crafted deep neural network designed to achieve precise classification of network traffic patterns. Its architecture comprises an input layer, three hidden layers, and a final output layer. The hidden layers increase the model's capacity to capture complexity, incorporating nonlinearity through the Rectified Linear Unit (ReLU) activation function. This nonlinear transformation is pivotal for detecting intricate patterns within the data. The output layer employs the softmax activation function to convert raw logits, the unnormalized scores generated by the model, into probability distributions. This step is critical for binary classification, as it establishes a clear decision boundary to distinguish between benign and malicious classes.

To improve generalization and mitigate overfitting, L2 regularization is applied to the first hidden layer, while dropout layers are strategically inserted between the hidden layers to reduce the model's dependence on individual neurons. The proposed ANN model is trained using the Adam optimizer, which adaptively adjusts the learning rate, paired with the binary crossentropy loss function, an optimal choice for binary classification tasks. Network weights are refined through backpropagation, enabling the model to iteratively minimize classification errors across multiple training epochs. The hyperparameters and configuration details of the proposed ANN model are summarized in Table 5.

Table 5. Hyperparameters and configuration

Parameter	Values
Input Features	Same as dataset features
Hidden Layers	3
Neurons per Layer	50,30,20
Activation Function	ReLU (hidden layers), Softmax(output)
Regularization	L2 Regularization (first layer)
Loss Function	Binary Crossentropy
Optimizer	Adam
Learning Rate	0.001

The classifier for this target ANN model is implemented using the TensorFlowV2Classifier from the Adversarial Robustness Toolbox (ART), tailored for compatibility with TensorFlow 2.x models. This classifier facilitates integration with adversarial attack and defense mechanisms. In this setup, the classifier takes the trained ANN model as input and employs categorical cross-entropy as the loss function for optimization. It is used to evaluate the model's predictions and is subsequently incorporated into adversarial attack experiments to assess the model's robustness. The classifier is configured for binary classification, predicting whether network traffic is benign or malicious. The softmax activation in the output layer generates probabilities for the two classes, enabling effective classification. The classifier's configuration is detailed in Table 6.

Table 6. Classifier hyperparameters

Parameter	Details
clip_values	Ensures input values stay within the defined range (min/max of training data)
nb_classes	Specifies the number of output classes (2 for binary classification)
input_shape	Matches the shape of the input features from the dataset
Loss Function	Categorical cross-entropy with logits disabled (from_logits=False)

4.3. Surrogate ANN Model

For the execution of black-box adversarial attacks, we constructed a surrogate Artificial Neural Network (ANN) model. This model features an input layer, four hidden layers, and an output layer, engineered to mimic the decision boundary of the target ANN. Although its structure resembles that of the target model, the surrogate includes minor architectural deviations. These variations stem from the premise that an attacker lacks precise knowledge of the target model's specifications and are intended to eliminate any bias. The output layer employs the sigmoid activation function to yield probability scores for binary classification. The proposed hyperparameters of the surrogate ANN are outlined in Table 7. The primary function of the surrogate model is to produce adversarial examples through gradient-based techniques, such as FGSM and PGD. These adversarial samples are then applied to the target model to evaluate the success of black-box attack strategies.

Table 7. Hyperparameters and configuration of surrogate model

Parameter	Values
Input Features	Same as dataset features
Hidden Layers	4
Neurons per Layer	60,50,30,20
Activation Function	ReLU (hidden layers), Sigmoid(output)
Regularization	L2 Regularization (first layer)
Loss Function	Binary Crossentropy
Optimizer	Adam
Learning Rate	0.0001

The surrogate model also functions as a binary classifier, with the key difference being using a sigmoid activation function in the final layer instead of softmax. This allows the model to output classification probability scores between 0 and 1. Like the target model, it is implemented using the TensorFlowV2Classifier from the ART, ensuring seamless integration with adversarial attacks, thereby facilitating robustness evaluation.

4.4. Architecture of Target CNN Model

The target CNN model was designed to efficiently learn hierarchical spatial features from network traffic data. The architecture consists of 3 convolutional layers, each followed by activation functions and pooling layers to reduce dimensionality while retaining critical features. The extracted features are then passed through fully connected (dense) layers for final classification. Dropout layers were incorporated to prevent overfitting, and batch normalization was applied to stabilize learning. The model was trained using the Adam optimizer and a binary cross-entropy loss function, optimizing for accurate classification of benign and malicious traffic. Table 8 presents the proposed hyperparameters for the CNN model.

The target CNN model was wrapped using the TensorFlowV2Classifier from the ART, like the target ANN model. This classifier enables adversarial robustness evaluations and facilitates integration with attack strategies. The setup remains the same as in the ANN model, ensuring consistency in adversarial evaluation.

Table 8. Hyperparameters and configuration

Parameter	Values
Input Features	Same as dataset features
Number of Convolutional Layers	3
Filters per Layer	64, 128, 256
Kernel Size	3x3
Activation Function	ReLU (hidden layers), Softmax (output)
Pooling Type	Max Pooling
Dropout Rate	0.5
Normalization	Yes

Optimizer	Adam
Loss Function	Binary Crossentropy
Learning Rate	0.001

4.5. Surrogate CNN model

To perform black-box adversarial attacks, we trained a surrogate CNN model, which approximates the decision boundary of the target CNN model using a subset of the dataset. It follows a similar architecture to the target CNN but differs in filter sizes and dropout rates. It also uses a sigmoid activation function instead of softmax for probability score outputs. The model was trained using binary cross-entropy loss and the Adam optimizer. Table 9 presents the proposed hyperparameters for the surrogate CNN model. It was then used to generate adversarial examples via FGSM and PGD attacks, which were later transferred to the target CNN model for robustness evaluation. The classifier setup remains the same as in the target CNN model, ensuring consistency in adversarial evaluation.

Table 9. Hyperparameters and configuration of surrogate model

Parameter	**Values**
Input Features	Same as dataset features
Number of Convolutional Layers	3
Filters per Layer	32,64, 128
Kernel Size	3x3
Activation Function	ReLU (hidden layers), Softmax (output)
Pooling Type	Max Pooling
Dropout Rate	0.4
Normalization	Yes
Optimizer	Adam
Loss Function	Binary Crossentropy
Learning Rate	0.0001

4.6. Adversarial Attack Analysis

DL-based Network Intrusion Detection Systems (NIDS) excel in identifying malicious network traffic but are inherently vulnerable to adversarial attacks. These attacks introduce subtle, intentional perturbations to input data, exploiting the sensitivity of DL models to minor changes and causing misclassifications. Such vulnerabilities pose a significant security risk, as adversaries could craft deceptive traffic to bypass detection, undermining the reliability of NIDS. This section investigates the impact of adversarial attacks on DL-based NIDS, focusing on two untargeted black-box attack methods: the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD). These attacks aim to degrade model performance by generating adversarial examples that lead to incorrect predictions without targeting a specific misclassification. By analyzing their effects, this study seeks to quantify the susceptibility of DL-based NIDS and highlight the need for robust countermeasures.

In black-box attack scenarios, the attacker lacks access to the target model's internal parameters, such as weights or gradients. Instead, adversarial examples are crafted using a surrogate model trained on a similar data distribution, relying on the transferability property—where perturbations effective on one model generalize to others. The success of these attacks hinges on how well the surrogate approximates the target's decision boundary and the transferability of the generated perturbations across different architectures. This research evaluates the robustness of two DL models—an Artificial Neural Network (ANN) and a Convolutional Neural Network (CNN)—against FGSM and PGD attacks, using the CICIDS 2019 dataset as the foundation for training and testing.

A. Implementation of Untargeted Black-Box Attacks

To establish a baseline, both the ANN and CNN models were trained and evaluated on the CICIDS 2019 dataset, a comprehensive collection of benign and malicious network traffic. The ANN achieved an accuracy of 99%, while the CNN reached 98%, demonstrating their effectiveness in distinguishing normal from anomalous traffic under standard conditions. However, their resilience against adversarial perturbations was subsequently tested using black-box attacks facilitated by a surrogate model.

The surrogate model, an ANN with four hidden layers and a sigmoid output layer, was developed to approximate the target model's behavior while maintaining architectural independence, reflecting the black-box assumption. Trained on the same CICIDS 2019 dataset, the surrogate generates adversarial examples using FGSM and PGD, which are then transferred to the target ANN and CNN models. The attack effectiveness is measured by the reduction in accuracy and the increase in misclassification rates post-attack. FGSM employs a single-step perturbation, while PGD uses an iterative approach to craft stronger, more refined adversarial samples. These samples are stored and tested on the target models to assess their robustness, simulating a real-world black-box scenario where the target has no prior knowledge of the

perturbations.

The analysis revealed a marked decline in model performance under adversarial conditions. For the ANN, accuracy dropped to 40%–50% after FGSM attacks and further to 30%–40% after PGD attacks. The CNN exhibited slightly greater resilience, with accuracy falling to 50%–60% for FGSM and 40%–50% for PGD. These results underscore PGD's superior attack strength due to its iterative refinement, as well as the varying susceptibility of ANN and CNN architectures. Notably, PGD perturbations showed higher transferability, effectively deceiving both models despite being crafted on the surrogate. This confirms the black-box attack's viability, as adversarial examples successfully compromised the target models without requiring access to their internal structures.

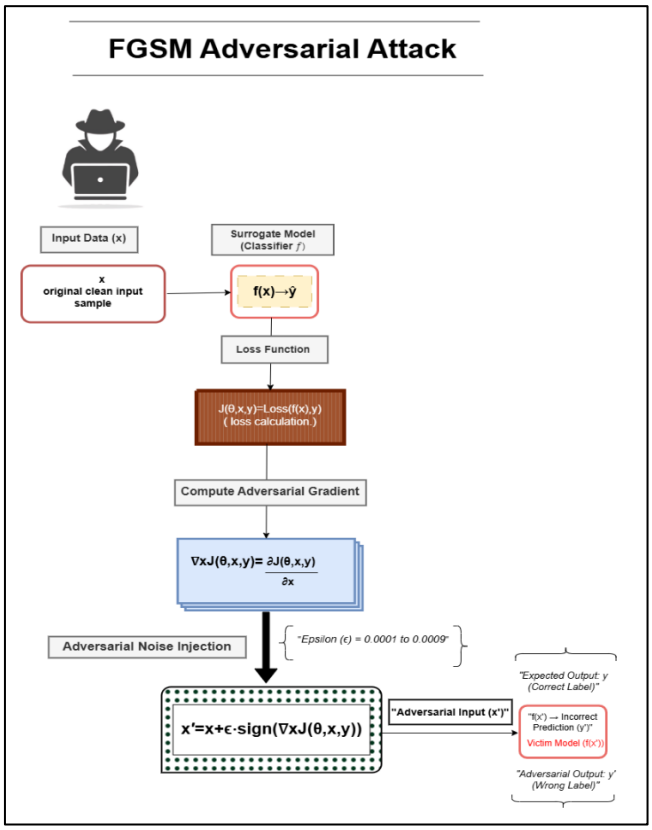


Fig.7. FGSM adversarial attack

Fast Gradient Sign Method (FGSM) - Untargeted Black-Box attack

FGSM is a simple yet powerful adversarial attack that perturbs an input sample in the direction of the gradient of the loss function. It generates adversarial examples by making small changes to the input, ensuring the perturbation is minimal but effective in causing misclassification. The FGSM Equation (1) works by perturbing the input X in the direction of the gradient of the loss function J (θ, X, y) with respect to the input. The sign function ensures the perturbation is applied consistently across all input dimensions. Table 10 defines the general equations for the parameters and hyperparameters of FGSM. The magnitude of the perturbation is controlled by ϵ, which determines how much the input is altered. The method generates a list of adversarial perturbations for different epsilon values ranging from 0.0001 to 0.0009, in this study. A higher ϵ value increases the attack strength and makes the perturbations more detectable. Fig. 7 illustrates the proposed FGSM adversarial attack. In a black-box attack scenario, adversarial examples are crafted on a surrogate model using this equation and then transferred to the target model to assess its vulnerability. Table 11 lists the specific features and parameters used for the FGSM attack in this study. Algorithm 1 outlines the steps of proposed surrogate model attack via FGSM.

$$Xadv = X + \epsilon.sign\big(\nabla xJ(\theta, X, y)\big) \tag{2}$$

Table 10. Hyperparameters and parameters in FGSM

Parameters	Details
X	Original input sample
X_{adv}	Adversarial input
ϵ	Perturbation magnitude
∇xJ	Gradient of loss w.r.t. input
J (θ, X, y)	Loss function
Sign(.)	Sign function to keep perturbation minimal

Algorithm 1: FGSM-Based Adversarial Attack

Input:

- Surrogate model M (Pre-trained classifier)
- Test dataset X_{test}
- True labels Y_{test} corresponding to X_{test}
- List of perturbation magnitudes $\mathcal{E} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_n\}$

Output:

- List of adversarial samples $X_{adv-list}$
- Attack performance results (accuracy, precision, recall, F1-score)

Algorithm Steps:

1.Initialize an empty list for attack performance metrics:

$$\text{attack\_results} \leftarrow []$$

2. For each perturbation magnitude $\epsilon \in \mathcal{E}$:

a) Create FGSM attack instance:

$$\text{fgsm\_attack} \leftarrow \text{FastGradientMethod}\ (\ \text{classifier} = M, \text{eps} = \epsilon, \text{targeted} = \text{False}, \text{batch\_size} = 128)$$

b) Generate adversarial samples using FGSM:

$$X_{adv} \leftarrow \text{fgsm\_attack.generate}\ (X_{test}\ )$$

c) Evaluate the surrogate model on adversarial samples:

$$Y_{pred\_adv} \leftarrow M.\,\text{predict}(X_{adv})$$

d) Convert predictions into class labels:

$$Y_{pred\_labels} \leftarrow \text{argmax}\ (Y_{pred\_adv}\ , \text{axis} = 1)$$

e) Compute the evaluation metrics:

$$\text{Accuracy} \leftarrow \text{accuracy\_score}\ (Y_{test}\ , Y_{pred\_labels}\ )$$

$$\text{Precision} \leftarrow \text{precision\_score}\ (Y_{test}\ , Y_{pred\_labels}\ , \text{average} = '\ '\ \text{weighted}\ ', \text{zero\_division} = 1)$$

$$\text{Recall} \leftarrow \text{recall\_score}\ (Y_{test}\ , Y_{pred\_labels}\ , \text{average} = '\ \text{weighted}\ ', \text{zero\_division} = 1)$$

$$\text{F1-score} \leftarrow \text{f1\_score}\ (Y_{test}\ , Y_{pred\_labels}\ , \text{average} = '\ '\ \text{weighted}\ ', \text{zero\_division} = 1)$$

f) Store the metrics into the results list:

attack_results.append({epsilon : ϵ, accuracy : Accuracy, precision : Precision, recall : Recall, F1-score : F1-score })

g) Print the evaluation metrics for the current perturbation magnitude ϵ.

3. (Optional) Plot the attack impact graphically:

- Extract epsilon values and corresponding accuracy and F1-score from attack_results.
- Plot accuracy vs. epsilon and F1-score vs. epsilon using matplotlib.pyplot.

Table 11. Feature and parameter for FGSM

Parameters	Details
Attack Type	Single-step untargeted black-box attack
Computational Cost	Low
Perturbation Refinement	No
Transferability	Moderate
Impact on ANN	High
Impact on CNN	Moderate

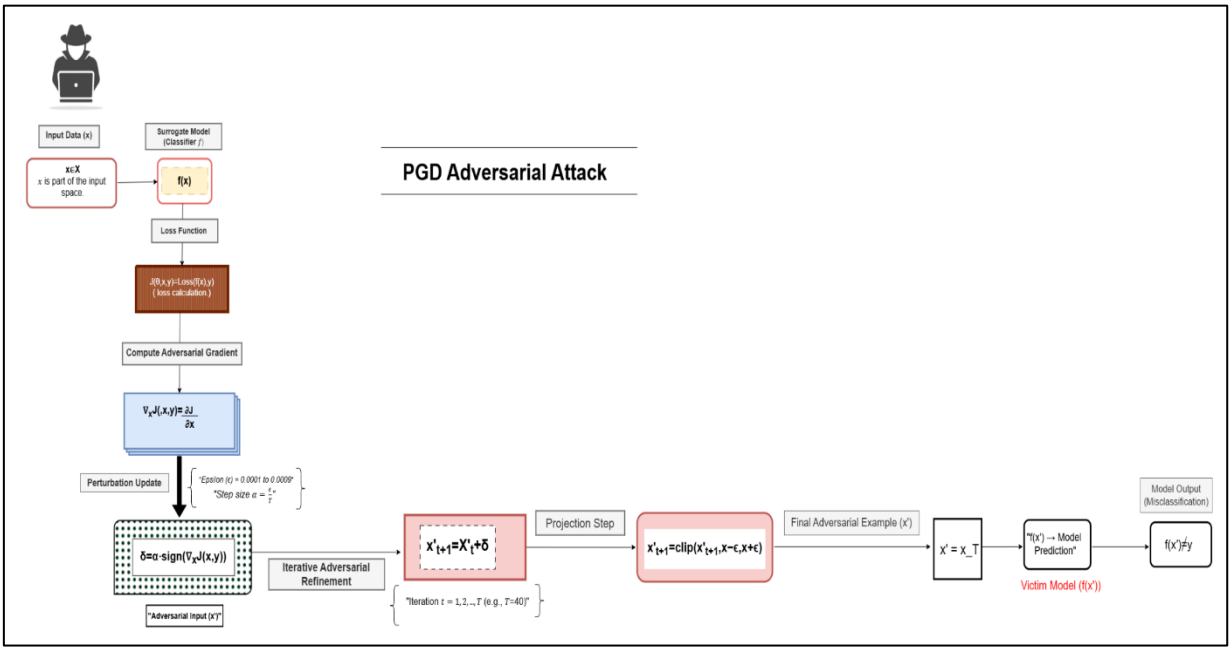


Fig.8. PGD adversarial attack

B. Projected Gradient Descent (PGD)-Untargeted Black-Box Attack

PGD is an iterative variant of FGSM and is considered one of the strongest first-order adversarial attacks. It generates adversarial examples by applying multiple small perturbation steps while ensuring that the perturbed sample remains within a bounded region around the original input. Unlike FGSM, which is a single-step attack, PGD refines the perturbation over multiple iterations, making it significantly more effective in fooling deep-learning models. Fig. 8 illustrates the proposed PGD adversarial attack. The PGD attack works by iteratively modifying the input in the direction of the gradient of the loss function while keeping the perturbation constrained within a defined epsilon (ϵ) bound. Table 12 defines the hyperparameters and parameters for the PGD attack. At each step, a small change is made using a step size (α), and the adversarial sample is clipped to ensure it does not exceed the perturbation limit. This makes PGD a more robust and stronger attack compared to FGSM. Algorithm 2 outlines the steps of the PGD-based black-box attack on the target model. The derivation of the PGD attack equation is presented below:

Step 1: Starting from FGSM

The FGSM computes a single-step perturbation: $\mathrm{Xadv} = \mathrm{X} + \epsilon \cdot \mathrm{sign}\,(\nabla \mathrm{XJ}\,(\theta, \mathrm{X}, \mathrm{y}))$

Step 2: Iterative Gradient Updates

Instead of applying the full perturbation in one step, PGD applies multiple updates with a step size α:

$$\mathrm{X}(\mathrm{t}+1) = \mathrm{X}(\mathrm{t}) + \alpha \cdot \mathrm{sign}\,(\nabla \mathrm{XJ}\,(\theta, \mathrm{X}(\mathrm{t}), \mathrm{y}))$$

At each iteration t, we update X in the direction of the gradient. The update step size α is smaller than ϵ, ensuring finer perturbations.

Step 3: Projection to Keep Perturbation in Bound

Since PGD performs multiple updates, the adversarial input may exceed the allowed perturbation range ϵ. To prevent this, we clip the perturbation so that the final adversarial example stays within the epsilon-ball around the original input:

$$X(t+1) = \text{clip}X, \epsilon\left(X(t) + \alpha \cdot \text{sign}\left(\nabla X J(\theta, X, y)\right)\right)$$

- Clipping ensures that $\|X^{(t+1)} - X\|_\infty \leq \epsilon$
- This prevents the adversarial sample from going outside the valid perturbation space.

Table 12. Hyperparameters and parameters in PGD

Parameters	Details
$X^{(0)}$	Original input sample
$X^{(t+1)}$	Adversarial input at step t+1
ϵ	Maximum perturbation magnitude
J (θ, X, y)	Loss function
α	Step size per iteration
Iterations	Number of update steps performed
Clip function	Ensures perturbation does not exceed ϵ

Algorithm 2: Adversarial perturbation generation with PGD and surrogate model

Input:

- M_s : Trained surrogate model used for generating adversarial samples
- X_{test} : Original test dataset
- E : List of perturbation magnitudes (ϵ values)

Output:

- X_{adv_list} : List of adversarially perturbed inputs for different ϵ values

1. Initialize the surrogate model as an ART classifier:

C ← KerasClassifier (model = M_s, clip_values = $\left(\min(X\_test), \max(X\_test)\right)$, use_logits = False)

2. Define the PGD attack function: Function pgd_attack(C, X_test, ε) :

```
PGD ← ProjectedGradientDescent (
estimator = C,
        eps = ε,              // Maximum perturbation magnitude
        eps_step = ε / 5,     // Step size per iteration
        max_iter = 40,        // Number of iterations
        targeted = False,     // Untargeted attack
        batch_size = 128      // Batch size for efficiency
      )
  X_adv ← PGD.generate(X_test)
  Return X_adv
End Function
```

3. Generate adversarial samples for multiple epsilon values:

```
X_adv_list ← []
For each ε ∈ E do:
  X_adv ← pgd_attack(C, X_test, ε)
  X_adv_list.append(X_adv)
```

4. Evaluate the attack on the surrogate model:

```
For each X_adv ∈ X_adv_list do:
  Y_pred_adv ← M_s.predict(X_adv)
  Y_pred_adv_labels ← argmax(Y_pred_adv, axis = 1)
  Compute performance metrics:
    acc ← accuracy_score(Y_test, Y_pred_adv_labels)
    prec ← precision_score(Y_test, Y_pred_adv_labels, average = 'weighted', zero_division = 1)
    rec ← recall_score(Y_test, Y_pred_adv_labels, average = 'weighted', zero_division = 1)
    f1 ← f1_score(Y_test, Y_pred_adv_labels, average = 'weighted', zero_division = 1)
  Store metrics for ε: {ε: ε, accuracy: acc, precision: prec, recall: rec, f1_score: f1}
```

5. Optional: Visualize attack impact:
Extract ε values and corresponding acc, f1 from stored metrics
Plot acc vs. ε and f1 vs. ε using matplotlib.pyplot

6. Return X_adv_list

5. Experimental Setup and Description

This section delineates the technical environment utilized for conducting experiments, encompassing the hardware, software, and libraries essential for building, training, and evaluating the deep learning (DL) models and adversarial attack methodologies presented in this study. The experimental setup was designed to ensure efficient computation, robust model training, and reliable assessment of adversarial robustness, leveraging both local hardware and cloud-based resources.

Hardware Configuration: The experiments were primarily executed on an MSI laptop equipped with an 11th Generation Intel Core i5-11400H processor, operating at a base frequency of 2.69 GHz, and paired with 16 GB of RAM (15.7 GB usable). This hardware configuration provided sufficient computational capacity to handle the training and evaluation of DL models, as well as the execution of adversarial attack algorithms, on moderately large datasets such as CICIDS 2019. The 64-bit operating system with an x64-based processor ensured compatibility with modern software frameworks and efficient memory management, facilitating seamless performance during experimentation.

To augment computational power, particularly for resource-intensive tasks such as model training and large-scale adversarial sample generation, Google Colab was employed. This cloud-based platform provided access to free GPU and TPU resources, significantly accelerating the training and evaluation processes. By leveraging Colab's high-performance computing capabilities, the study mitigated the limitations of local hardware, enabling rapid experimentation without requiring dedicated high-end infrastructure.

Software Environment: The software environment was carefully structured to support the development and analysis of DL-based Network Intrusion Detection Systems (NIDS) and adversarial attacks. Python served as the primary programming language, executed within Google Colab's Linux-based runtime, which offers native compatibility with a suite of machine learning and scientific computing libraries. Key frameworks and tools included TensorFlow and Keras for constructing and training DL models, the Adversarial Robustness Toolbox (ART) for generating adversarial examples, and additional libraries such as NumPy, Matplotlib, Seaborn, and Scikit-learn for numerical operations, visualization, and performance evaluation. This combination of tools ensured an integrated and efficient workflow, from data preprocessing to adversarial robustness assessment.

Libraries and Tools: The specific libraries employed in this research are detailed in table 13, alongside their purposes:

Table 13. Software tools and libraries

Library/Tool	Purpose
TensorFlow/Keras	Framework for building, training, and evaluating deep neural networks.
Adversarial Robustness Toolbox (ART)	Generation of adversarial examples (e.g., FGSM, PGD) and robustness analysis.
NumPy	Numerical computations, including array and matrix operations for data handling.
Matplotlib/Seaborn	Visualization of model performance and adversarial attack impacts.
Scikit-learn	Data preprocessing, model evaluation, and computation of performance metrics (e.g., accuracy, precision, recall).

These libraries were selected for their robustness, efficiency, and widespread adoption in the machine learning community, ensuring reliable implementation of the proposed models and attacks.

Experimental Setup Summary: The complete experimental setup, integrating hardware and software components, is summarized in Table 14.

Table 14. Experimental setup configuration

Component	Description
Device Name	MSI laptop
Processor	11th Gen Intel® Core™ i5-11400H @ 2.69 GHz
Installed RAM	16.0 GB (15.7 GB usable)
System Type	64-bit operating system, x64-based processor
Google Colab Environment	Cloud-based platform providing free GPU/TPU resources for accelerated computation

Primary Libraries	- TensorFlow/Keras: DL model training and evaluation
	- ART: Adversarial attack generation
	- NumPy: Numerical operations
	- Matplotlib/Seaborn: Data visualization
	- Scikit-learn: Model evaluation and preprocessing
Reason for Setup	Leveraged Colab's GPU/TPU resources for efficient training and attack generation; selected libraries optimized for DL and adversarial robustness analysis.

The combination of an MSI laptop and Google Colab was chosen to balance accessibility with computational power. The local hardware supported initial development and smaller-scale testing, while Colab's GPU/TPU resources enabled efficient training of DL models on the CICIDS 2019 dataset and rapid execution of adversarial attack algorithms. The selected software libraries provided a cohesive ecosystem for implementing neural network architectures, crafting adversarial examples (e.g., FGSM and PGD), and evaluating model performance under attack. This setup ensured reproducibility, scalability, and precision in assessing the robustness of DL-based NIDS against black-box adversarial threats, aligning with the study's objectives.

Performance metrics: To evaluate the effectiveness of adversarial attacks and model robustness, I employed several performance metrics, which provide insights into the accuracy and reliability of the models under normal and adversarial conditions. Below are the commonly used performance metrics in this experiment:

- **Accuracy**: Measures the percentage of correct predictions. A decrease indicates the model's vulnerability to adversarial attacks.

$$\text{Accuracy} = \frac{\text{Total Predictions}}{\text{Correct Predictions}}$$

- **Precision:** Indicates the proportion of positive predictions that are correct. Higher precision means fewer false positives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

- **Recall:** Measures the model's ability to correctly identify positive instances. High recall means fewer positive instances are missed.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

- **F1-Score**: The harmonic mean of precision and recall, balancing both metrics to evaluate the model's overall performance.

$$\text{F1 Score} = 2\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision+Recall}}$$

- **Confusion Matrix**: Visualizes the model's performance, showing true positives, false positives, true negatives, and false negatives. Fig. 9 demonstrates the Confusion Matrix.

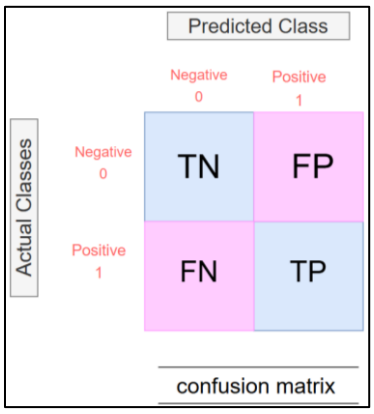


Fig.9. Confusion matrix

5.1. Experimental Parameters

The effectiveness of adversarial attacks largely depends on the choice of parameters, which influence the strength and subtlety of the perturbations applied to input data. In this study, the FGSM and PGD were used as adversarial attack techniques, and their parameter selection was carefully chosen to ensure a balanced evaluation of adversarial robustness.

For FGSM, the primary parameter is Epsilon (ε), which controls the magnitude of perturbations added to the input. Different values of ε were selected—small (e.g., 0.0001), medium (e.g., 0.01), and significant (e.g., 0.2)—to analyze how varying attack strengths impact model performance. Using various values allows for a comparative study of minimal, moderate, and vigorous adversarial perturbations while maintaining realistic attack scenarios.

For PGD, the attack is iterative, requiring additional parameters beyond ε. The step size (α) determines how much input is modified in each iteration, and it was chosen as $\alpha = \varepsilon / k$ to ensure controlled perturbation growth without overshooting the optimal attack direction. The number of iterations (k) varied across 10, 20, and 40 iterations, allowing for an analysis of how attack strength changes with increasing steps. Lower iteration values provide faster attack execution but may result in weaker adversarial examples, while higher iterations create more substantial perturbations at the cost of increased computational resources.

These parameters were selected to ensure a systematic evaluation of adversarial effectiveness. Using the same ε values for FGSM and PGD enables a direct comparison between the attack methods. The step size and iteration variations in PGD help us understand how iterative refinement affects attack strength. These choices provide insights into adversarial transferability and model vulnerability across different attack intensities.

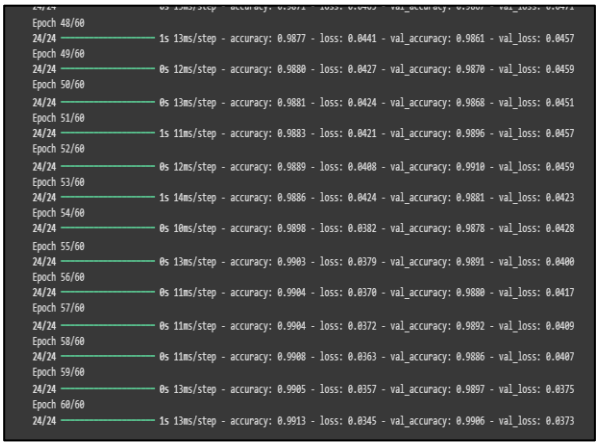
5.2. Results and Performance Evaluation

This section presents the results obtained from the adversarial attacks on different models and evaluates their performance under varying attack strengths. The impact of FGSM and PGD attacks on ANN and CNN models is analyzed using key performance metrics such as accuracy, precision, recall, F1-score, and adversarial transferability. A comparative analysis examines how different models respond to various levels of disturbances and investigates the possibility of creating adversarial examples that can transfer between models. This evaluation not only uncovers the vulnerabilities of the models but also assesses the effectiveness of the attacks and the overall resilience of the architectures against adversarial threats.

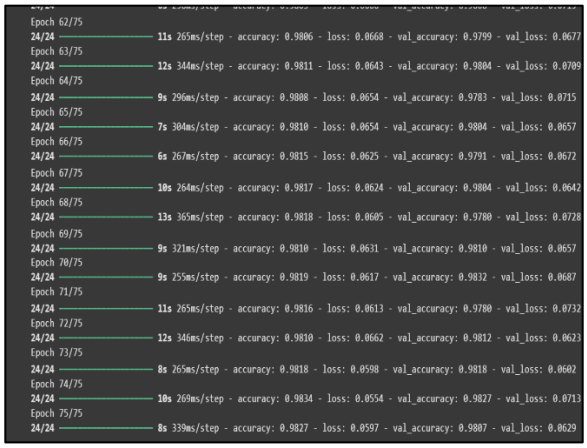
5.3. Performance of Target ANN/CNN on Clean Data

Prior to assessing the impact of adversarial attacks, it is imperative to establish the baseline performance of both ANN and CNN on clean, unperturbed datasets. Fig. 10 depicts the training process for both a) the ANN and b) the CNN target models prior to the application of attacks. The evaluation of the models was performed utilizing a variety of key performance metrics. These included accuracy, precision, recall, F1-score, and a detailed analysis of the confusion matrix, all aimed at ensuring a thorough and comprehensive assessment of the model's effectiveness.

The results show that both models achieve high accuracy on clean data, demonstrating their effectiveness in classifying network traffic without adversarial perturbations. The ANN achieved an accuracy of 99%, while the CNN achieved 98%, indicating that both models can correctly classify most instances in the dataset. Fig. 11 and Fig. 12 illustrate the accuracy and loss curves of the target ANN and CNN models before the attack, respectively. These high accuracy and low loss values affirm that before adversarial attacks, the models perform well and generalize effectively on normal data.



a. ANN target model performance



b. CNN target model performance

Fig.10. Training of target a) ANN b) CNN model before attack

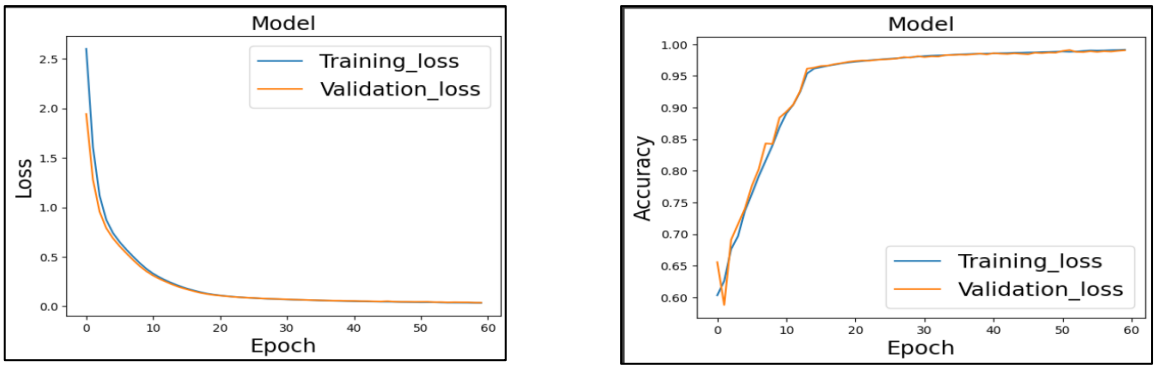


Fig.11. Accuracy and loss of target ANN model before attack

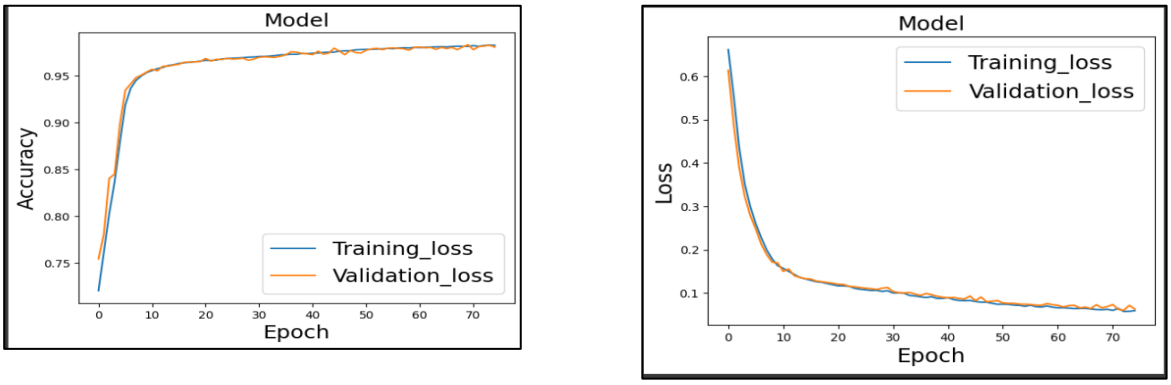
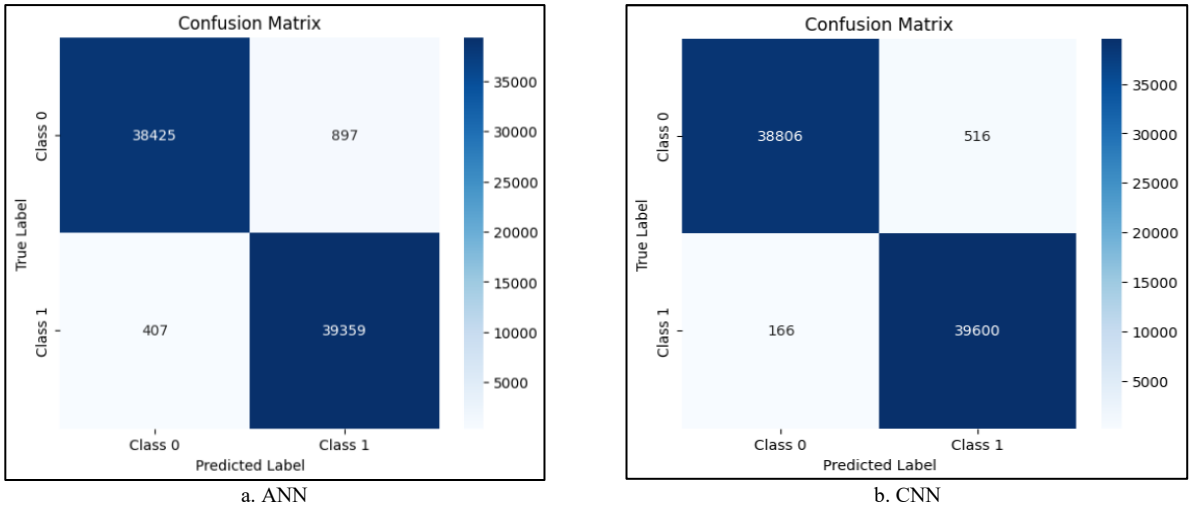


Fig.12. Accuracy and loss of target CNN model before attack

To further analyze the classification performance, a confusion matrix was generated for both models. Fig. 13 illustrates the confusion matrix of the target model. The results indicate that while both ANN and CNN target models classify most instances correctly, ANN slightly outperforms CNN in this setting.



a. ANN b. CNN

Fig.13. Confusion Matrix of target models before Attack a. ANN b. CNN

However, precision, recall, and F1-score provide additional insights into model effectiveness. Table 15 contains performance metrics of the target models. Fig. 14 and Fig. 15 illustrate the classification graphs and ROC curves of the target ANN and CNN models before the attack, respectively, highlighting how well they distinguish between normal and attack instances.

Table 15. Performance measure

Target Models	Accuracy	Precision	Recall	F1 Score	Roc Curve
ANN	0.9914	0.9871	0.9958	0.9915	0.9993
CNN	0.9835	0.9777	0.9898	0.9837	0.9979

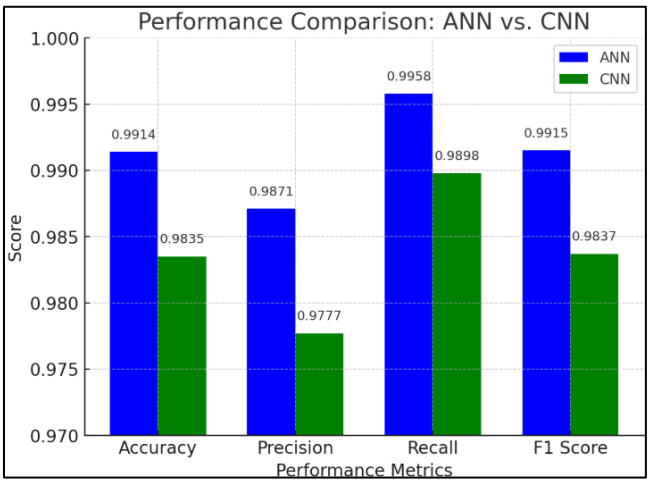


Fig.14. Classification graph of ANN/CNN before attack

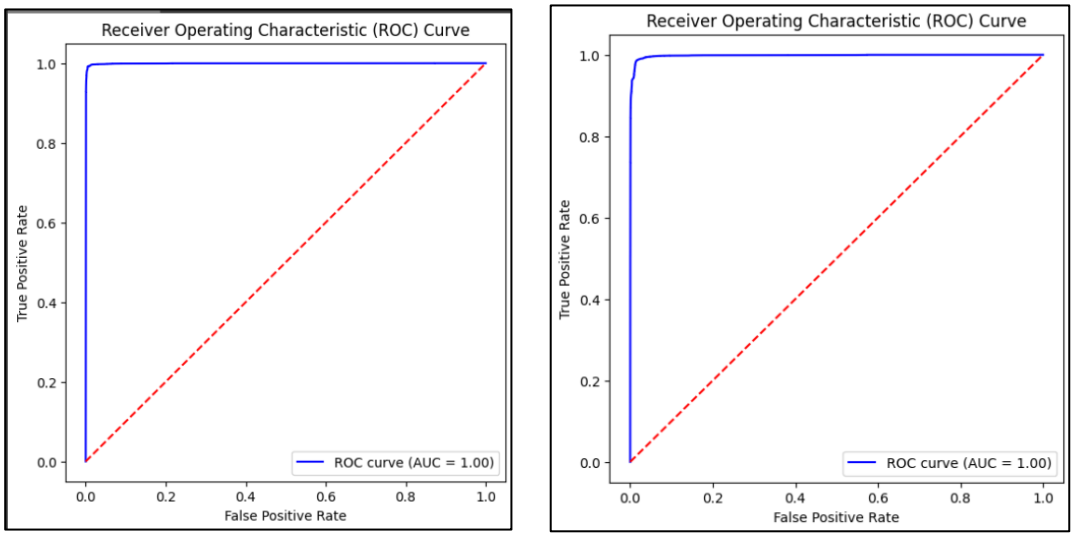
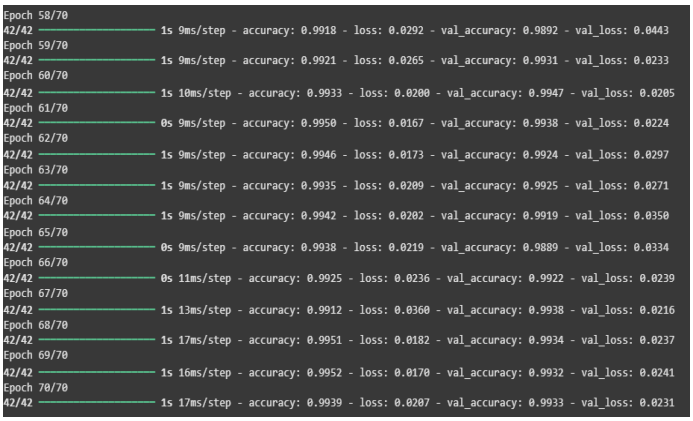


Fig.15. ROC Curve graph of a) ANN and b) CNN before Attack

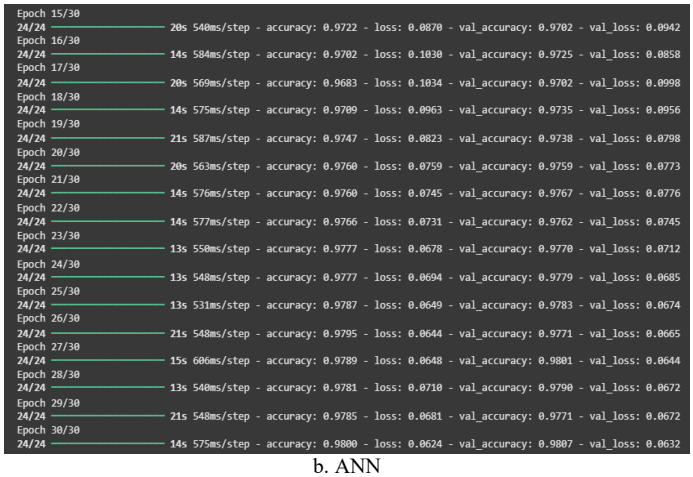
A. Performance of Surrogate ANN/CNN on Clean Data

In this study, a surrogate model was developed to replicate the behavior of a target model, offering a strategic advantage in black-box attack scenarios where the attacker has no direct access to the target model's parameters. Under these conditions, adversarial examples are crafted using the surrogate model and then transferred to the target model for evaluation. This approach enables the assessment of adversarial robustness without requiring knowledge of the target's internal structure. Specifically, either the ANN or CNN served as the surrogate to generate adversarial examples, which were subsequently tested on the alternate model to investigate adversarial transferability. The successful deception of a CNN using adversarial examples generated by an ANN surrogate, or vice versa, indicates that these perturbations are not model-specific but can generalize across distinct architectures. This finding is pivotal for understanding the vulnerabilities and resilience of DL models against adversarial attacks, highlighting a broader susceptibility that transcends architectural differences. The ANN surrogate model, during training on clean data, achieved an accuracy of 99%, reflecting its robust classification performance. Complementing this, a loss value of 0.02 indicates a high degree of alignment between the model's predictions and the true labels. Fig. 16 illustrates the training process of the ANN and CNN surrogate models. The low loss suggests that the ANN effectively captured meaningful patterns in the dataset, rendering it a reliable source for generating transferable adversarial examples.

Likewise, the CNN surrogate model attained an accuracy of 98%, slightly lower than the ANN due to its deeper and more complex architecture. Its loss value of 0.06 further underscores its learning efficacy. Given CNNs' ability to exploit spatial hierarchies in data, the adversarial perturbations they produce may exhibit unique transferability characteristics compared to those from the ANN. These performance metrics establish both models as credible surrogates, facilitating a deeper exploration of how architectural differences influence the effectiveness and generalizability of adversarial attacks.



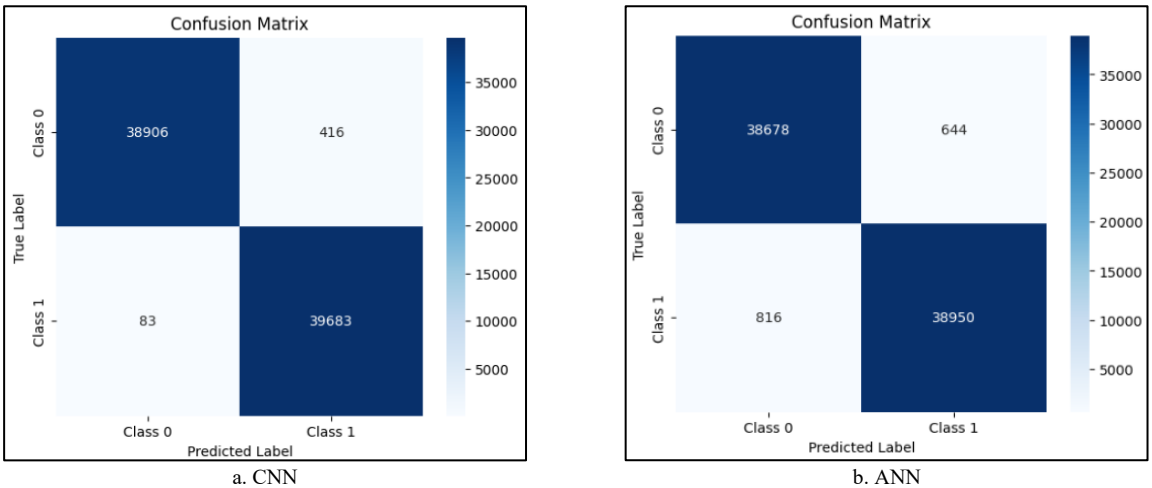
a. CNN



b. ANN

Fig.16. Training of the surrogate a) ANN and b) CNN models

To gain deeper insight into the classification behavior of the models, the confusion matrices for both the ANN and CNN surrogate models were analyzed. For the ANN, the confusion matrix [[38906, 416], [83, 39683]] reveals that the model accurately classified the vast majority of instances, with only a limited number of false positives (416) and false negatives (83). Similarly, the CNN's confusion matrix [[38678, 644], [816, 38950]] exhibits a comparable trend, indicating strong and reliable performance, though with slightly higher misclassification rates—644 false positives and 816 false negatives. Fig. 17 illustrates the confusion matrices for both the ANN and CNN surrogate models. These subtle differences in misclassified instances suggest potential variations in how adversarial perturbations transfer between the models, influencing the overall success of adversarial transferability.



a. CNN b. ANN

Fig.17. Confusion matrices of the a) ANN and b) CNN surrogate models

The surrogate model's influence on performance evaluation is pivotal for understanding adversarial transferability. The classification efficacy of the surrogate model directly affects how well adversarial examples perform when applied to the target model. High accuracy, coupled with well-balanced precision, recall, and F1-score metrics, suggests that the surrogate model closely mirrors the decision boundaries of the target model, enhancing the transferability of adversarial attacks. Table 16 presents the classification evaluation metrics for the surrogate models, providing a detailed breakdown of their performance. Conversely, disparities in classification performance between the surrogate and target models can diminish the success rate of attacks, as the adversarial examples may fail to generalize effectively. Figs. 18 and 19 depict the classification performance and ROC curves of the surrogate ANN and CNN models, respectively, offering visual insights into their performance characteristics. Thus, the surrogate model's capacity to produce effective adversarial perturbations is a critical factor in assessing the robustness and vulnerability of the target model to adversarial attacks.

Table 16. Classification evaluation metrics

Surrogate Models	Accuracy	Precision	Recall	F1 Score	Roc Curve
ANN	0.9937	0.9896	0.9979	0.9938	0.9996
CNN	0.9815	0.9837	0.9795	0.9816	0.9967

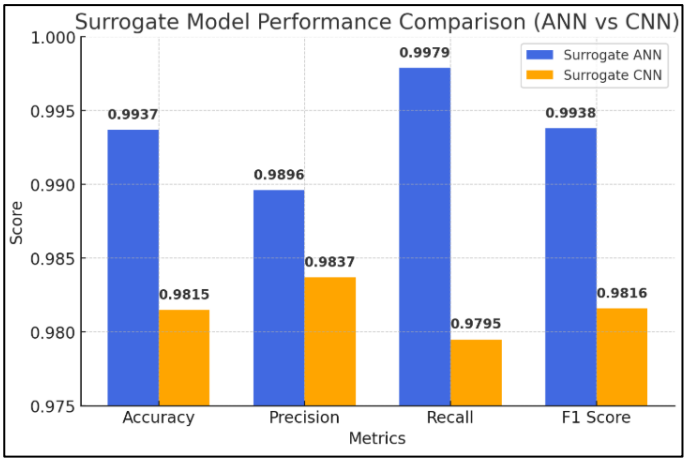


Fig.18. Classification performance of surrogate ANN and CNN models

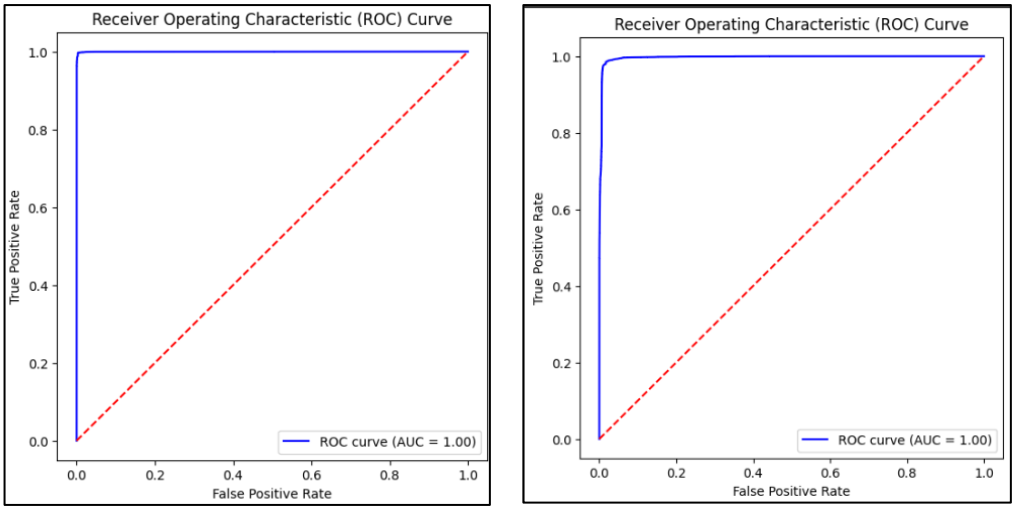


Fig.19. ROC curves of surrogate (a) ANN and (b) CNN models

B. Performance of Target ANN/CNN on Adversarial Data

This study utilized black-box untargeted adversarial attacks, specifically the FGSM and PGD, to evaluate the robustness of ANN and CNN models. The primary objective was to examine how adversarial examples influence model accuracy and classification performance, while also exploring the transferability of these attacks across distinct architectures. Adversarial examples were generated using a surrogate model in a black-box setting and subsequently applied to the target models, simulating a realistic scenario where attackers lack direct access to the target's internal parameters.

The introduction of adversarial perturbations via FGSM and PGD led to a marked decline in the accuracy of both ANN and CNN models relative to their baseline performance on unperturbed data. These perturbations caused misclassifications, with the degree of accuracy reduction tied to the attack's strength—determined by the perturbation magnitude (ϵ) and the intrinsic resilience of each model. Larger ϵ values intensified the perturbations, increasing the likelihood of successful deception. For instance, under FGSM, the ANN's accuracy dropped from 99% to 44%, while the CNN's fell from 98% to 29%. Similarly, PGD attacks reduced ANN accuracy to 42% and CNN accuracy to 41%, as depicted in Figs. 20 and 21, respectively. These findings illustrate that even subtle alterations to input data can severely compromise the reliability of DL models.

A key observation from this study is the transferability of adversarial examples across architectures. Adversarial samples crafted using the ANN surrogate model effectively misled the CNN target model, and those generated with a CNN surrogate similarly deceived the ANN target. This cross-architecture transferability highlights shared vulnerabilities between ANN and CNN models, despite their differing structural designs. Rather than targeting specific misclassifications, these untargeted attacks broadly disrupted prediction accuracy, amplifying their potential impact. The ability of adversarial perturbations to generalize across models underscores a significant security concern for DL-based NIDS, revealing that such systems are universally susceptible to similar adversarial threats. These results emphasize the urgent need for enhanced defensive strategies to bolster the robustness of DL models against evolving adversarial tactics.

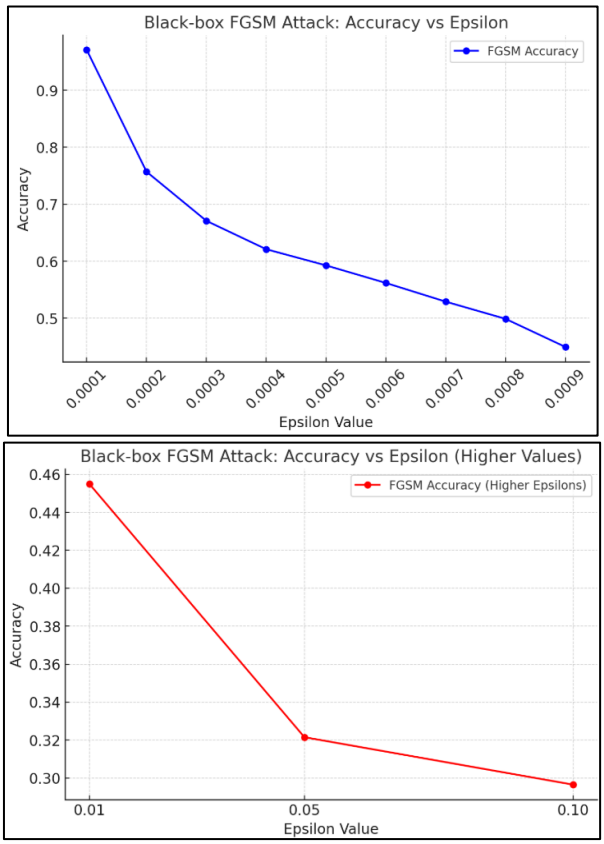


Fig.20. Accuracy of ANN and CNN models under FGSM attack

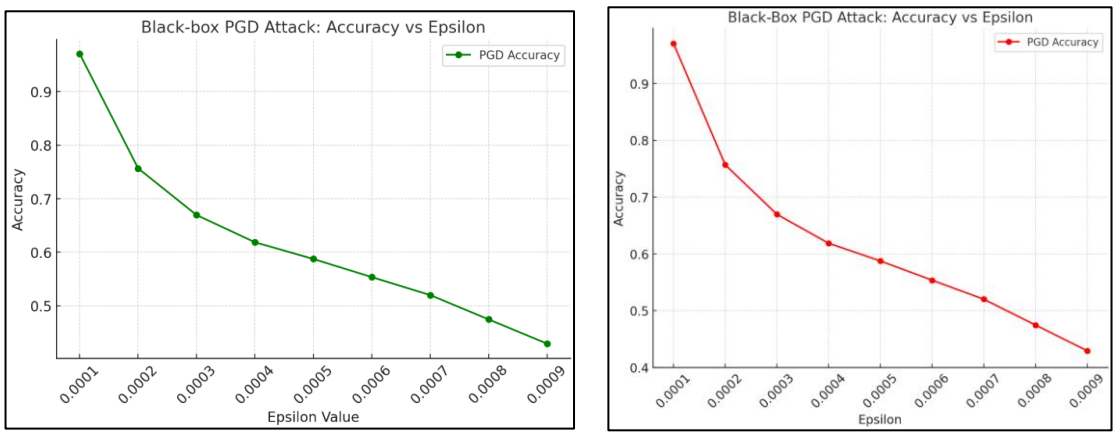


Fig.21. Accuracy of ANN and CNN models under PGD attack

The efficacy of FGSM and PGD attacks is contingent upon specific parameters that dictate the intensity of the perturbations applied. In the case of FGSM, the principal parameter, ε, regulates the magnitude of the perturbation introduced to the input data. An increased epsilon value leads to more substantial perturbations, thereby resulting in a greater degradation of accuracy. However, since FGSM is a single-step attack, lower epsilon values may be insufficient to mislead more robust models. In contrast, PGD, being an iterative attack, refines the adversarial perturbations over multiple steps. The effectiveness of PGD is influenced by several parameters. The parameter ε determines the magnitude of the perturbation applied, while the step size α regulates the extent of modification made during each iteration. The number of iterations (k) plays a crucial role in enhancing the perturbation across multiple steps, which increases the effectiveness of the attack. Additionally, incorporating random restarts in the PGD method makes it more challenging for defensive measures to succeed, thereby improving its efficacy compared to the FGSM. The comparative results demonstrate that PGD generally causes a greater accuracy drop than FGSM, confirming its superiority as an adversarial attack.

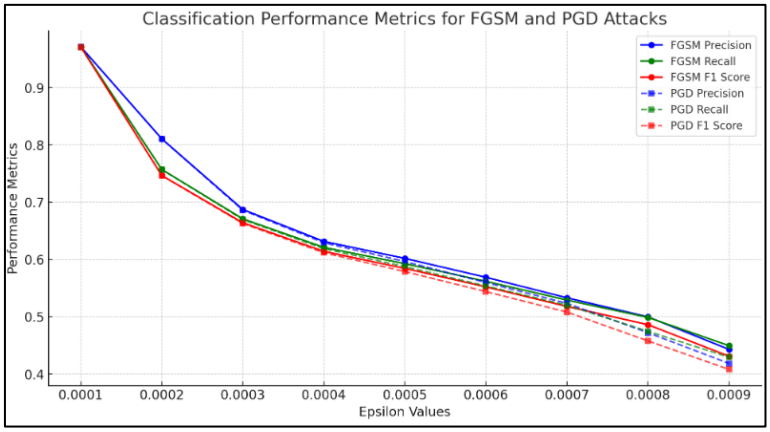


Fig.22. Classification performance metrics of ANN and CNN models

The final comparative study results confirm that adversarial black-box attacks significantly degrade the performance of both ANN and CNN models. Fig. 22 illustrates the classification performance metrics of the ANN and CNN models. The key findings indicate that ANN and CNN both experienced substantial accuracy reductions, proving their vulnerability to adversarial perturbations. Furthermore, PGD exhibited a stronger attack impact compared to FGSM, demonstrating its superior ability to deceive the models. The transferability of adversarial examples between ANN and CNN further highlights that different architectures are susceptible to similar attack strategies. In addition to accuracy degradation, the deterioration in Precision, Recall, and F1 Score validates that adversarial attacks not only reduce accuracy but also impact overall classification reliability. Ultimately, this study underscores the importance of developing robust defense mechanisms, as both ANN and CNN are vulnerable to adversarial perturbations, emphasizing the need for models that are robust against adversarial attacks.

6. Limitations and Future Work

This study provides significant insights into the adversarial transferability of black-box attacks, specifically the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), and their impact on Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) models within Network Intrusion Detection Systems (NIDS). However, certain limitations constrain the scope of the findings, and opportunities for future research could further advance the understanding and mitigation of adversarial threats.

The current investigation focuses exclusively on attack strategies without incorporating defensive measures such as adversarial training, defensive distillation, or input preprocessing. This omission limits the study's ability to assess how these models might perform with countermeasures in place, a critical aspect for real-world deployment. Additionally, the analysis is restricted to two untargeted black-box attack methods—FGSM and PGD—excluding more sophisticated techniques like the Carlini & Wagner (CW) attack or AutoAttack, which could reveal further vulnerabilities. The reliance on a single dataset, CICIDS 2019, while comprehensive, may not fully represent the diversity of network traffic and attack scenarios encountered across different environments, potentially limiting the generalizability of the results. Finally, the examination of transferability is confined to ANN and CNN architectures, leaving unexplored how adversarial examples might behave across other deep learning models, such as transformers or ensemble architectures, which are increasingly relevant in cybersecurity applications.

Future research could address these limitations by integrating defense strategies to evaluate their efficacy in mitigating the impact of adversarial attacks on ANN and CNN-based NIDS. Techniques such as adversarial training, model ensembling, or input transformation could be explored to enhance model resilience, providing a more holistic view of security in adversarial settings. Expanding the range of attack variants—incorporating white-box, adaptive, or more

advanced black-box methods like CW or AutoAttack—would offer a broader understanding of the threat landscape and test model robustness under diverse conditions. Extending the analysis to additional datasets, such as NSL-KDD, UNSW-NB15, or CSE-CIC-IDS2018, could validate the findings across varied network contexts and attack types, improving generalizability. Investigating cross-model transferability beyond ANN and CNN to include other architectures, such as Recurrent Neural Networks (RNNs), transformers, or hybrid models, would provide deeper insights into the universal vulnerabilities of deep learning systems in NIDS. Finally, translating these findings into real-world deployable systems by testing adversarial robustness in live network environments could bridge the gap between theoretical analysis and practical application, ensuring that AI-driven NIDS can withstand evolving cyber threats effectively.

7. Conclusions

This study investigates the adversarial transferability of black-box attacks (FGSM and PGD) on AI-driven NIDS using the CICIDS 2019 dataset, revealing the vulnerability of ANN and CNN models. Initially, both models achieved high accuracies of 99.14% (ANN) and 98.35% (CNN) on clean data. However, the introduction of adversarial examples significantly reduced accuracies to as low as 29% for CNN under FGSM and 41% for both models under PGD, exposing significant vulnerabilities. Notably, these perturbations, generated on surrogate models, transferred effectively between ANN and CNN architectures, indicating shared vulnerabilities despite structural differences. PGD proved more effective than FGSM due to its iterative nature, while CNNs were found to be more susceptible, highlighting the impact of architecture on robustness. These findings highlight the need for robust defense mechanisms to protect AI-based NIDS against evolving adversarial threats.

All the Declarations and Statements

Author Contributions Statement

Aasim Zafar - Conceptualization, Methodology, Supervision. Proposed the core research ideas, constructed the overall framework for evaluating adversarial transferability in ANN and CNN-based NIDS, and supervised the entire project execution, including experimental design and manuscript finalization.
Shazra Wali - Data Curation and Software Implementation. Handled data acquisition from the CICIDS2019 dataset, performed dataset preprocessing (cleaning, feature selection, normalization), and implemented the software pipeline, including model architectures and adversarial attack generation using the Adversarial Robustness Toolbox (ART).
Sheikh Burhan ul Haque - Model Training, Validation, and Performance Evaluation. Led the training and validation of the ANN and CNN models, implemented surrogate and target model setups, conducted benchmarking against baseline performance, generated adversarial examples using FGSM and PGD, and evaluated results under black-box transferability conditions.
All authors contributed to the interpretation of results, reviewed drafts, and approved the final version of the manuscript.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare that they have no conflicts of interest.

Funding Declaration

This research received no specific external grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement

This study analyzed a publicly available dataset. The CICIDS2019 (specifically the DDoS subset from the Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv file) is available from the Canadian Institute for Cybersecurity at: https://www.unb.ca/cic/datasets/ddos-2019.html (accessed on [Month Year, e.g., February 2025]). The code generated and/or analyzed during the current study is available from the corresponding author upon reasonable request.

Ethical Declarations

This research does not involve human subjects, animals, or any procedures requiring ethical approval from an institutional review board. All experiments were conducted using publicly available, anonymized network traffic data with no personal or sensitive information.

Acknowledgments

The authors sincerely thank the Canadian Institute for Cybersecurity for providing the CICIDS2019 dataset, which served as a realistic benchmark for this study. We also gratefully acknowledge the high-performance computing resources and support provided by the Department of Computer Science, Aligarh Muslim University, Aligarh, India and Department of data science and AIML, Cluster University of Srinagar. Valuable discussions with colleagues and access to open-source

libraries (TensorFlow/Keras, Adversarial Robustness Toolbox, scikit-learn) greatly facilitated the implementation and evaluation phases.

Declaration of Generative AI in Scholarly Writing
No large language models (LLMs) or other generative AI tools were used in the conceptualization, coding, data analysis, figure generation, or drafting of this manuscript. All content, including text, interpretations, and conclusions, was produced by the authors. Minor language polishing for clarity (e.g., grammar checks via standard word processors) was performed manually. No generative AI tools were used for content creation.

Abbreviations
The following abbreviations are used in this manuscript:

AI - Artificial Intelligence
ANN - Artificial Neural Network
ART - Adversarial Robustness Toolbox
CIC - Canadian Institute for Cybersecurity
CNN - Convolutional Neural Network
DDoS - Distributed Denial of Service
DL - Deep Learning
FGSM - Fast Gradient Sign Method
IDS - Intrusion Detection System
ML - Machine Learning
NIDS - Network Intrusion Detection System
PGD - Projected Gradient Descent

References

[1] R.N. Anaedevha, A.G. Trofimov, "Improved Robust Adversarial Model against Evasion Attacks on Intrusion Detection Systems", Optical Memory and Neural Networks, Vol.33, No.4, pp.S414–S423, 2024. DOI:10.3103/s1060992x24700681
[2] Khushnaseeb Roshan, Aasim Zafar, "A Novel DL-based Model to Defend Network Intrusion Detection System against Adversarial Attacks", 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp.2043-2048, 2023. DOI:10.1109/ICAC3N56670.2022.10112520
[3] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, X. Yin, "Evaluating and Improving Adversarial Robustness of Machine Learning-Based Network Intrusion Detectors", IEEE Journal on Selected Areas in Communications, Vol.39, No.8, pp.2632–2647, 2021. DOI:10.1109/jsac.2021.3087242
[4] G. Kocher, G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges", Soft Computing, Vol.25, No.13, pp.9731–9763, 2021. DOI:10.1007/s00500-021-05893-0
[5] C. Zhang, X. Costa-Perez, P. Patras, "Adversarial Attacks against Deep Learning-Based Network Intrusion Detection Systems and Defense Mechanisms", IEEE/ACM Transactions on Networking, Vol.30, No.3, pp.1294–1311, 2022. DOI:10.1109/tnet.2021.3137084
[6] Hesamodin Mohammadian, "An adversarial attack framework for DL-based NIDS", Master's Thesis, University of New Brunswick, pp.1-115, 2022.
[7] Aasim Zafar, Tarab Malik, and Sheikh Burhan ul Haque, "Comparative Analysis of Black-Box Targeted Adversarial Attacks on DL-Based NIDS: A Study of C&W and JSMA using CICDDoS2019," International Journal of Computer Networks & Communications (IJCNC), vol. 18, no. 1, pp. 101-119, January 2026. DOI: 10.5121/ijcnc.2026.18107.
[8] W. Villegas-Ch, A. Jaramillo-Alcázar, S. Luján-Mora, "Evaluating the Robustness of DL Models against Adversarial Attacks: An Analysis with FGSM, PGD and CW", Big Data and Cognitive Computing, Vol.8, No.1, pp.8, 2024. DOI:10.3390/bdcc8010008
[9] K. Roshan, A. Zafar, S.B.U. Haque, "Untargeted white-box adversarial attack with heuristic defense methods in real-time DL-based network intrusion detection system", Computer Communications, Vol.218, pp.97–113, 2023. DOI:10.1016/j.comcom.2023.09.030
[10] A. Alotaibi, M.A. Rassam, "Enhancing the Sustainability of Deep-Learning-Based Network Intrusion Detection Classifiers against Adversarial Attacks", Sustainability, Vol.15, No.12, pp.9801, 2023. DOI:10.3390/su15129801
[11] Sheikh Burhan ul Haque, "A Fuzzy-based Frame Transformation to Mitigate the Impact of Adversarial Attacks in Deep Learning-based Real-time Video Surveillance Systems," Applied Soft Computing, Vol. 167, Art. No. 112440, 2024. DOI: 10.1016/j.asoc.2024.112440
[12] I. Sharafaldin, A.H. Lashkari, S. Hakak, A.A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy", 2019 International Carnahan Conference on Security Technology (ICCST), pp.1–8, 2019. DOI:10.1109/ccst.2019.8888419
[13] Z.K. Maseer, R. Yusof, N. Bahaman, S.A. Mostafa, C.F.M. Foozy, "Benchmarking of Machine Learning for Anomaly-Based Intrusion Detection Systems in the CICIDS2017 Dataset", IEEE Access, Vol.9, pp.22351–22370, 2021. DOI:10.1109/access.2021.3056614
[14] Canadian Institute for Cybersecurity, "CICDDoS2019 Dataset", University of New Brunswick, [Online], 2019.
[15] Sheikh Burhan ul Haque, Aasim Zafar, "Robust Medical Diagnosis: A Novel Two-Phase Deep Learning Framework for

Adversarial Proof Disease Detection in Radiology Images", Journal of Imaging Informatics in Medicine, Vol. 37, No. 1, pp. 308-338, 2024. DOI: 10.1007/s10278-023-00916-8
[16] S. Qiu, Q. Liu, S. Zhou, C. Wu, "Review of Artificial Intelligence Adversarial Attack and Defense Technologies", Applied Sciences, Vol.9, No.5, pp.909, 2019. DOI:10.3390/app9050909
[17] J. Šircelj, D. Skočaj, "Accuracy-Perturbation Curves for Evaluation of Adversarial Attack and Defense Methods", 2020 25th International Conference on Pattern Recognition (ICPR), pp.6290-6297, 2021.
[18] Huda Ali Alatwi, Amjad Aldweesh, "Adversarial Black-Box Attacks against Network Intrusion Detection Systems: A Survey", 2021 IEEE World AI IoT Congress (AIIoT), pp.34-40, 2021. DOI:10.1109/AIIoT52608.2021.9454214
[19] O. Ibitoye, R. Abou-Khamis, M.E. Shehaby, A. Matrawy, M.O. Shafiq, "The threat of adversarial attacks on machine learning in Network Security: A Survey", arXiv:1911.02621, pp.1-15, 2019.
[20] S. Alahmed, Q. Alasad, M.M. Hammood, J.-S. Yuan, M. Alawad, "Mitigation of Black-Box Attacks on Intrusion Detection System-Based ML", Computers, Vol.11, No.7, pp.115, 2022. DOI:10.3390/computers11070115
[21] P. Kumar, S. Rathore, "DL Performance Evaluation Model for Enhancing Network Intrusion Detection Systems", 2023 International Conference on Computer Communication and Informatics (ICCCI), pp.1-7, 2024. DOI:10.1109/ICAC3N60023.2023.10753184
[22] Elie Alhajjar, Paul Maxwell, Nathaniel Bastian, "Adversarial machine learning in Network Intrusion Detection Systems", Expert Systems with Applications, Vol.186, pp.115782, 2021. DOI:10.1016/j.eswa.2021.115782
[23] Sheikh Burhan ul Haque, "Mitigating Adversarial Threats in Deep CT Image Diagnosis Models via a Dual-Stage Inference-Time Defense", Applied Soft Computing, Vol. 163, Art. No. 111909, 2024. DOI:10.1016/j.asoc.2024.111909.
[24] H. Mohammadian, A.H. Lashkari, A.A. Ghorbani, "Evaluating Deep Learning-Based NIDS in Adversarial Settings", Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP), pp.435-444, 2022. DOI:10.5220/0010867900003120
[25] Sheikh Burhan ul Haque, Aasim Zafar, Sheikh Moeen ul Haque, Sheikh Riyaz ul Haq, Mohassin Ahmad, "Securing AI in Healthcare: A Three-Layer Defense to Mitigate Adversarial Noise Impact in Radiology Imaging", Biomedical Signal Processing and Control, Vol. 109, Art. No. 107969, 2025. DOI:10.1016/j.bspc.2025.107969
[26] Md. Ahsan Ayub, William A. Johnson, Douglas A. Talbert, Ambareen Siraj, "Model Evasion Attack on Intrusion Detection Systems using Adversarial Machine Learning", 2020 54th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, pp.1-6, 2020, doi: 10.1109/CISS48834.2020.1570617116.
[27] V. Kumar, K. Kumar, M. Singh, "Generating practical adversarial examples against learning-based network intrusion detection systems", Annals of Telecommunications, Vol.79, pp.1-14, 2024. DOI:10.1007/s12243-024-01021-9
[28] S. Sharma, Z. Chen, "A systematic study of adversarial attacks against network intrusion detection systems", Electronics, Vol.13, No.24, pp.5030, 2024. DOI:10.3390/electronics13245030
[29] M. elShehaby, A. Matrawy, "Adversarial Evasion Attacks Practicality in Networks: Testing the Impact of Dynamic Learning", arXiv:2306.05494, pp.1-12, 2023.
[30] Mohammad Jawad Kadhim Abood and Ghassan Hameed Abdul-Majeed, "Enhancing Multi-Class DDoS Attack Classification using Machine Learning Techniques," Journal of Advanced Research in Applied Sciences and Engineering Technology, vol. 43, no. 2, pp. 75–92, 2024, doi: 10.37934/araset.43.2.7592.

Authors' Profiles

Prof. Aasim Zafar is a Professor in the Department of Computer Science at Aligarh Muslim University (AMU), Aligarh, India. He holds a Master's degree in Computer Science and Applications and a Ph.D. in Computer Science from Aligarh Muslim University. With over 32 years of teaching, research, and administrative experience at national and international levels, Prof. Zafar has published over 100 research papers in reputed international journals and conferences and has guided 11 PhD scholars. His research interests include Mobile Ad hoc and Sensor Networks, Image Processing and Video Analytics, Information Retrieval, E-Systems, Cybersecurity, Virtual Learning Environments, Neuro-Fuzzy and Soft Computing, and Software Engineering. He served as Chairperson/Head of the Department of Computer Science from January 2021 to January 2024 and is currently the Registrar of Aligarh Muslim University. He has been actively involved in ICT infrastructure development and serves as UGC SWAYAM Coordinator, Coordinator of the IGNOU Study Centre at AMU, and Convener of the University Website Committee.

Shazra Wali is a Research Scholar in the Department of Computer Science at Aligarh Muslim University, Aligarh, India. She specializes in network security, machine learning, and deep learning applications in cybersecurity. Her research interests focus on adversarial robustness of AI-driven intrusion detection systems, machine learning-based security mechanisms, and the development of defensive measures against adversarial attacks. She has contributed to data curation, preprocessing, and software implementation of critical research projects involving network traffic analysis and adversarial attack generation using advanced machine learning frameworks. Her work emphasizes understanding vulnerabilities in AI-based network security systems and developing robust solutions to enhance the reliability of intrusion detection systems against evolving cyber threats.

Dr. Sheikh Burhan ul Haque is an Assistant Professor in the Department of Computer Application at Cluster University of Srinagar, Jammu and Kashmir, India. Currently, he is the HOD of the Department of Data Science at Cluster University of Srinagar. He holds advanced degrees in Computer Applications and has specialized expertise in deep learning, machine learning, and adversarial machine learning. His research interests encompass vulnerabilities in AI models, adversarial attacks and defenses, Generative Adversarial Networks (GANs), deep learning-based video surveillance systems, and network intrusion detection systems (NIDS). Dr. Burhan has published research on adversarial robustness, AI security, and applications of deep learning in cybersecurity. He has published more than 18 journal articles out of which 11 are indexed in highly impacted SCI/ SCIE journals with most of them in Q1 quartile. He is actively involved in developing defense mechanisms and mitigation strategies against adversarial threats in AI-driven security systems. He has also published 2 patents so far. He has led model training, validation, and performance evaluation initiatives in adversarial robustness research and serves as a corresponding author on multiple international publications.

How to cite this paper

Aasim Zafar, Shazra Wali, Sheikh Burhan ul Haque, "Adversarial Transferability in AI-based Network Intrusion Detection: A Comparative Study of ANN and CNN Models", International Journal of Information Technology and Computer Science(IJITCS), Vol.18, No.4, pp.170-198, 2026. DOI:10.5815/ijitcs.2026.04.11