Washington DC, USA

15-18 December 2024

AAG: Adversarial Attack Generator for evaluating the robustness of Machine Learning Models against Adversarial Attacks

Dimitrios Christos Asimopoulos, Panagiotis Radoglou-Grammatikis, Thomas Lagkas, Vasileios Argyriou, Ioannis Moscholios, Jorgen Cani, Georgious Th. Papadopoulos, Evangelos K. Markakis, and Panagiotis Sarigiannidis

MetaMind Innovations P.C, Greece

Authors & Contributors

Funded by the European Union

AI4CYBER – Artificial Intelligence for next generation CYBERsecurity

Minds — Dimitrios-Christos Asimopoulos

International Hellenic University — Dimitrios-Christos Asimopoulos

University of Western Macedonia — Panagiotis Radoglou Grammatikis, Panagiotis Sarigiannidis

K3Y R&D and Cyber Security — Panagiotis Radoglou Grammatikis

Harokopio University of Athens — Jorgen Cani, Georgios Th. Papadopoulos

Democritus University of Thrace — Thomas Lagkas

Kingston University London — Vasileios Argyriou

University of Peloponnese — Ioannis Moscholios

Hellenic Mediterranean University — Evangelos K. Markakis

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101070450 (AI4CYBER) .

Presentation Structure

Introduction

1

- Introduction
- Related Work
- Contributions

Main Part

2

- Adversarial Attack Generator Architectural Design
- White Box Adversarial Attacks
- Black Box Adversarial Attacks
- AAG Evaluation & Results

Conclusions

3

- Sum up important key points
- Future Work
- Q/A

Introduction, Relevant Work & Contributions

Introduction

Artificial Intelligence (AI) has significantly improved applications in image recognition, natural language processing, and autonomous systems, but it also introduce vulnerabilities, particularly to adversarial attacks.

Adversarial attacks involve intentionally crafted perturbations that mislead AI model predictions, exposing critical security risks in various fields

Critical infrastructures, like the smart electrical grid, are vulnerable to multi-step adversarial attacks and Advanced Persistent Threats (APTs), which can lead to widespread service outages, financial losses, and even potential fatalities.

AI-driven defense systems can detect unknown anomalies and zero-day cyberattacks, yet they remain vulnerable to adversarial manipulations that aim to bypass detection or create false alarms, highlighting the need for robust models.

Related Work

2020

Martins et al.

- This survey examines adversarial threats against intrusion and malware detection systems, analysing attack techniques, as well as defences.

2021

Bai et al

- This work reviews advancements in adversarial training methods to improve the robustness of deep learning models. It presents a taxonomy of techniques and identifies unresolved challenges in making models resilient to adversarial attacks.

2022

Ghaffari Laleh et al.

- This study investigates the vulnerability of convolutional neural networks (CNNs) in oncology diagnostics, highlighting the susceptibility of AI models to adversarial attacks and exploring mitigation strategies

2023

Asimopoulos et al.

- This research explores vulnerabilities in AI-based intrusion detection systems used in industrial applications, particularly within the energy sector, and evaluates the resilience of various models like Decision Trees, Random Forests, and MLPs against attacks like FGSM and CTGAN.

Contributions

Adversarial Attack Generator (AAG) against OCPP dataset:

- An Adversarial Attack Generator is provided to train the models and test the impact of various attacks. For this purpose, two ML/DL models are used and compared.

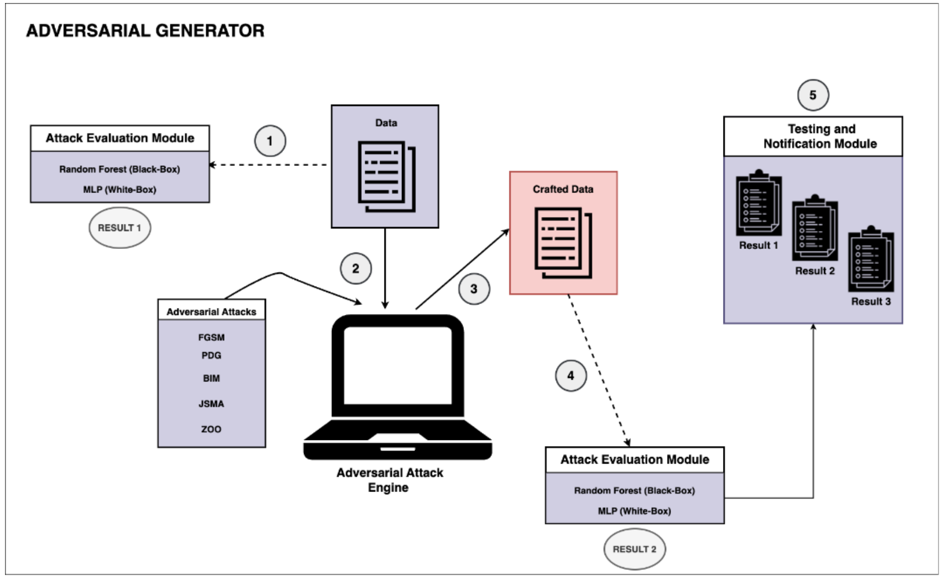
Evaluation of various adversarial attacks (FGSM, BIM, PGD, C&W, JSMA, ZOO)

- We investigate how various adversarial attacks affect the detection performance of the ML/DL models.

Adversarial Attack Generator Architectural Design

Adversarial Attack Generator (AAG)

- The architecture of the AAG is based on the methodology of the C4 model
- This methodology includes four key diagram types.



- **System Context Diagram:** Shows users and external entities interacting with the system.
- **Container Diagram:** This represents the system as a set of independent services that interact with each other.
- **Component Diagram:** Breaks down each container to detail components as function blocks performing specific tasks.
- **Code Diagram:** Describes the implementation of each component, utilizing UML diagrams and entity relationship diagrams.

AAG System Context Diagram

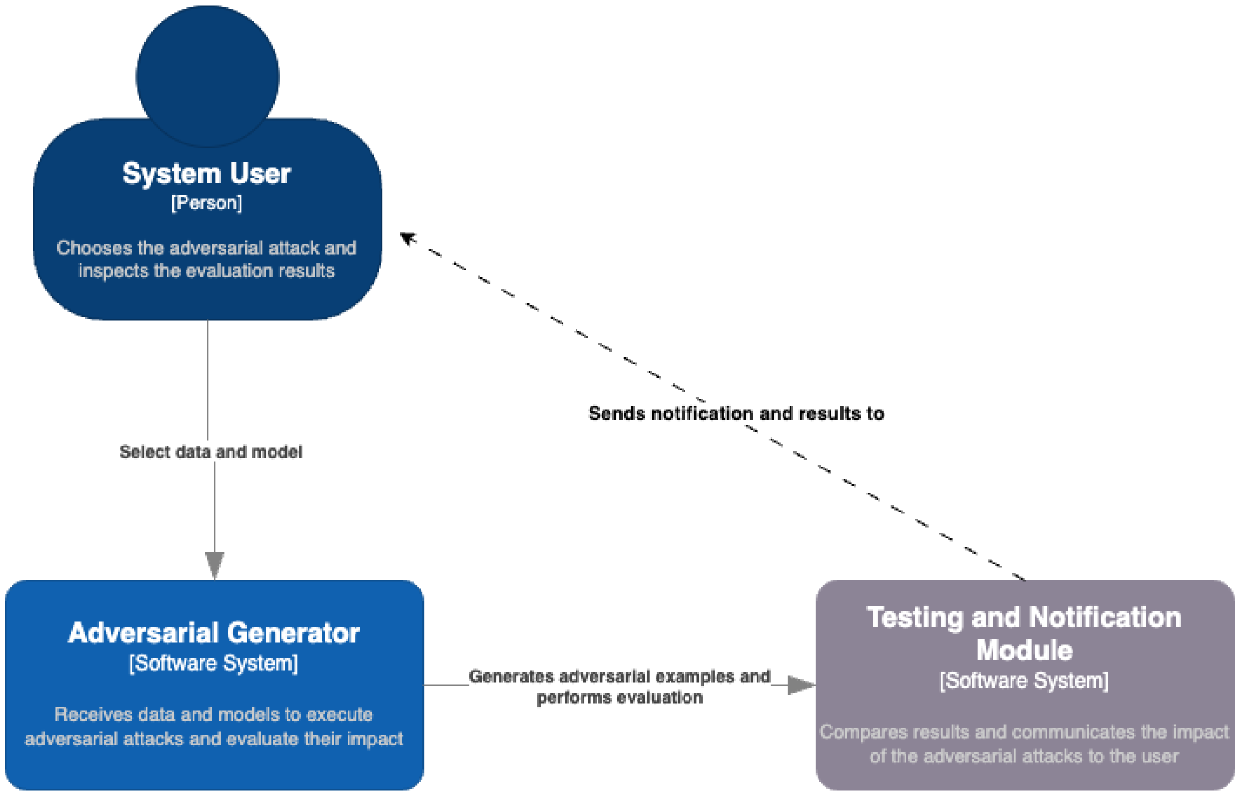
System User: Selects the type of adversarial attack, provides data (and model, if needed), and initiates the attack generation process.

Adversarial Generator: Core component that:

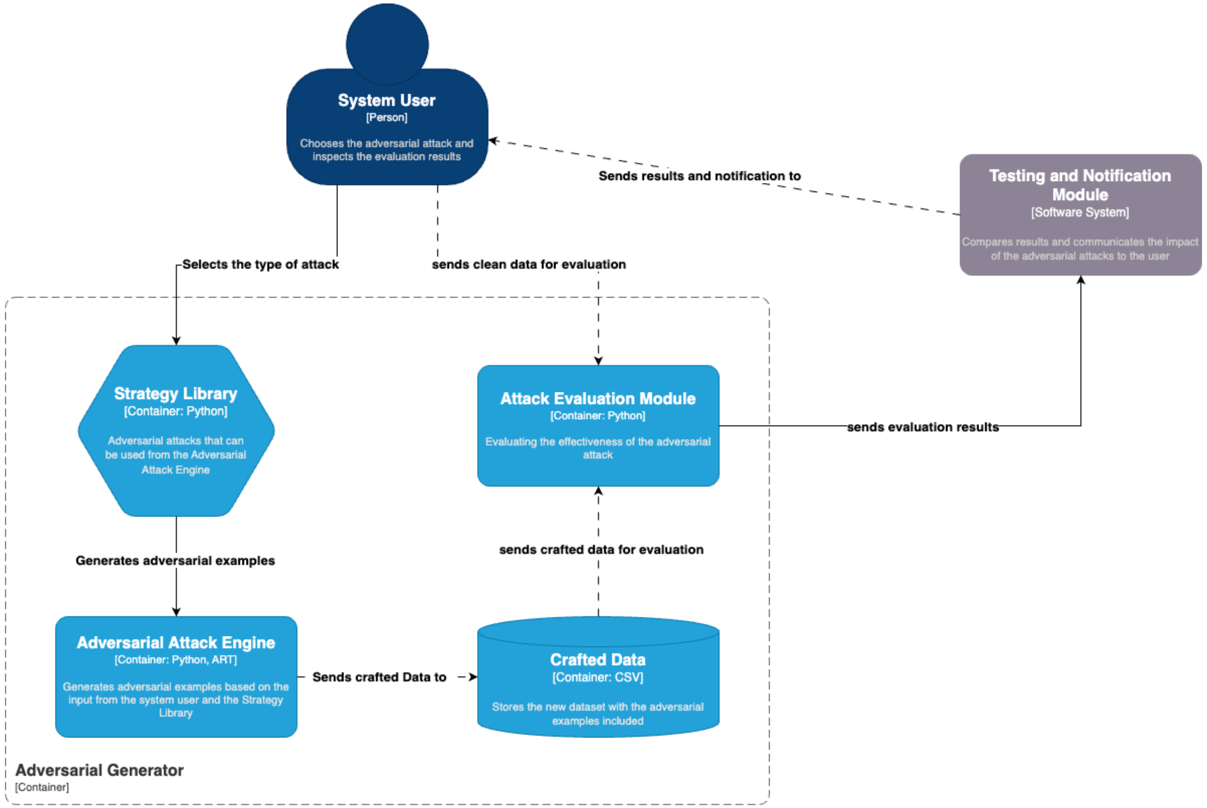
- Receives input data and model.
- Generates adversarial examples.
- Evaluates model robustness
- Sends results to the Testing and Notification Module

Testing and Notification Module:

- Processes and compares evaluation outcomes.
- Sends final performance results, including model behavior under adversarial conditions, back to the System User for review and analysis.



AAG Container Context Diagram



Strategy Library:
• Contains adversarial attack algorithms like FGSM, PGD, JSMA, BIM, C&W, and ZOO.
• Provides the attack methods accessed by the Adversarial Attack Engine for adversarial generation.

Adversarial Attack Engine:
• Core container that generates adversarial examples from the clean dataset.
• Uses attacks from the Strategy Library and supports both white-box and black-box scenarios.

Crafted Data:
• Stores the adversarially perturbed datasets in CSV format.
• Ensures compatibility with downstream evaluation processes.

Attack Evaluation Module:
• Assesses the robustness of machine learning models against adversarial examples.
• Evaluate model performance under both white-box and black-box conditions.

White Box Adversarial Attacks

Fast Gradient Sign Method (FGSM)

FGSM adds targeted noise to the input data to exploit model vulnerabilities, making it an essential technique for adversarial robustness testing due to its simplicity and computational efficiency.

- FGSM generates adversarial examples by adding a small, crafted perturbation to the input.
- Perturbation is calculated by taking a step in the direction of the gradient sign of the loss function, maximizing prediction error.
- Efficient and widely used for evaluating the robustness of machine learning models.

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

adv_x $\rightarrow$ Adversarial data

$x \rightarrow$ Original data

$y \rightarrow$ Original input label

$\epsilon \rightarrow$ Multiplier to ensure the perturbations are small

$\theta \rightarrow$ Model parameters

$J \rightarrow$ Loss function. In our case, the CrossEntropy function

Jacobian-Based Saliency Map Attack (JSMA)

JSMA is a targeted adversarial attack that identifies and manipulates the most impactful features (e.g., pixels) to achieve misclassification while keeping perturbations minimal and harder to detect.

JSMA steps:

Compute Jacobian Matric: Measures the sensitivity of each model output with respect to each input feuture. Provides insights into what features impact the model's prediction

$$J_F(X) = \left[\frac{\partial F_j(X)}{\partial X_i}\right]_{i,j}$$

Calculate the Saliency Map: Identifies the features that maximize the probability of the target class θ without increasing the probabilities on not-targeted classes.

$$S_{\mathrm{map}}(i,\theta) = \begin{cases} 0 & \text{if } \frac{\partial F_\theta(X)}{\partial X_i} < 0 \text{ or } \sum_{j\neq\theta} \frac{\partial F_j(X)}{\partial X_i} > 0, \\ \left(\frac{\partial F_\theta(X)}{\partial X_i}\right)^2 - \left(\sum_{j\neq\theta} \frac{\partial F_j(X)}{\partial X_i}\right)^2 & \text{otherwise.} \end{cases}$$

Select and Perturb Features: Features with the highest saliency scores are chosen for perturbation.

Project Gradient Descent (PGD) – Basic Iterative Method (BIM)

PGD and BIM are both iterative, white-box attack methods that apply controlled, small perturbations to input data, aiming to mislead deep learning models into misclassification.

- Iteratively applies perturbations in the gradient direction to maximize the model's prediction error.
- PGD executes FGSM in small steps, repeatedly projecting perturbations to keep them undetectable.

$$\delta_{\text{new}} = \text{Clip}\epsilon\left(\delta + \alpha \cdot \text{sign}(\nabla_x J(\theta, x, y))\right)$$

Where $\delta_{\_new}$ keeps perturbations within bounded constraints

- An enhancement of FGSM that applies iterative, precise perturbations.
- Steps are calculated to maximize model misclassification, with a clipping function ensuring changes remain within a specified epsilon neighbourhood.

$$X^{(n+1)} = \text{Clip}X, \varepsilon\left\{X^{(n)} + \alpha \cdot \text{sign}\left(\nabla_X J(\theta, X^{(n)}, Y\text{true})\right)\right\}$$

Where $X^{(n+1)}$ is the adversarial example for the next iteration

Carlini & Wagner (C&W)

The C&W attack creates adversarial examples with minimal visible changes, aiming to deceive neural network classifiers while keeping the perturbation nearly imperceptible.

Optimization-Based: Finds the smallest perturbation required for misclassification. PGD executes FGSM in small steps, repeatedly projecting perturbations to keep them undetectable.

Norm Variants: Supports multiple norms for flexibility in perturbation size and visibility:

- L_0 : Alters the fewest components.
- L_2 : Minimizes the Euclidean distance (overall similarity).
- $L\infty$: Limits maximum change to any component.

Gradient Descent: Used to solve the optimization problem, balancing between achieving misclassification and maintaining imperceptibility.

$$\text{minimize } \|x - x_0\|_2^2 + c \cdot l(x),$$

$$l_9(x) = \begin{cases} 0, & \text{if } \max_{j \neq t}\{g_j(x)\} - g_t(x) \leq 0, \\ +\infty, & \text{otherwise.} \end{cases}$$

where x_0 is the original input, x is the adversarial example, g represents model logits, and t is the target class.

Black Box Adversarial Attacks

Zero Order Optimization (ZOO)

ZOO attack as a black Box attack doesn't require access to the model's gradients, unlike white box attacks. ZOO uses function evaluations (not gradients) to estimate the directions of perturbations.

Gradient Estimation: Employs finite differences by slightly perturbing input data and observing output changes.

Iterative Optimizations: Adjusts input incrementally to maximize the loss function, thereby crafting an adversarial example

Short Description	This experiment aims to generate adversarial examples using a real-world dataset. The goal is to craft perturbations that deceive the model's detection mechanism, causing it to misclassify the adversarial inputs. The crafted dataset will be used to test the *model's* behavior after the adversarial attack.
Attack Name	Zero Order Optimization (ZOO)
Attack Type	*Black-box attack*
Target Model Architecture	*Random Forest*
Dataset	*The Dataset used for the execution of the scenario is the OCPPFlowMeter (CSV)*
Target Task	*The type of task that the scenario executes is a classification task.*
Perturbation Method	*The method used to generate adversarial examples is a gradient-based attack (ZOO)*
Perturbation	*confidence*=0.1, *targeted*=False, *learning_rate*=0.01, *max_iter*=10, *binary_search_steps*=10, *initial_const*=1.0, *abort_early*=False, *use_resize*=False, *use_importance*=False, *nb_parallel*=1, *batch_size*=1, *variable_h*=0.1

AAG Evaluation & Results

Dataset Overview

OCPP CICFlow Meter

Source: The dataset was parsed using CICFlow Meter to extract network flow statistics.

Format: Data recorded in PCAP CSV format, providing insights into network traffic.

Includes both normal and benign traffic and multiple types of cyberattacks such as FDI Charging Profile, DOC ID Tag, DOS Flooding Heartbeat, and DOS Flooding EVCS Rejected attacks.

Evaluation Metrics

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

True Positive Rate

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate

$$FPR = \frac{FP}{FP + FN}$$

F1 Score

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$TP \rightarrow$ True Positives
$TN \rightarrow$ True Negatives
$FP \rightarrow$ False Positives
$FN \rightarrow$ False Negatives

Model Evaluation Results

	Random Forest	MLP
Accuracy	0.9909	0.9890
TPR	0.9909	0.9890
FPR	0.0130	0.0212
F1 score	0.9909	0.9890

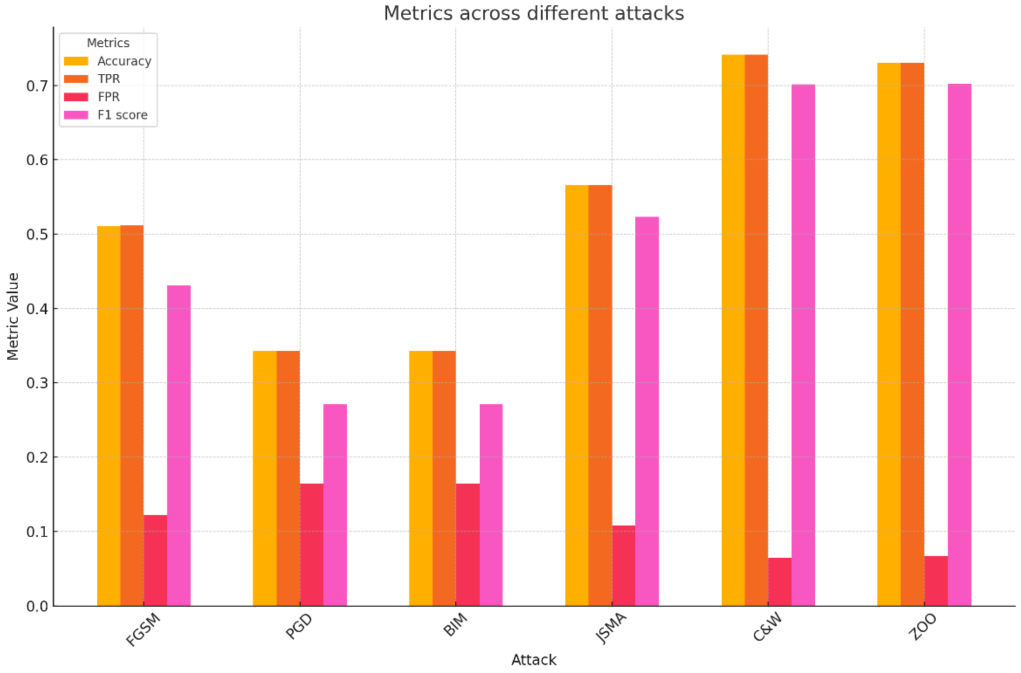
1st step of the AAG is to evaluate the performance of the model using clean data.

The results show that the model performs outstandingly having an accuracy, TPR, FPR, and F1 score of 0.99

AAG Evaluation results (1)

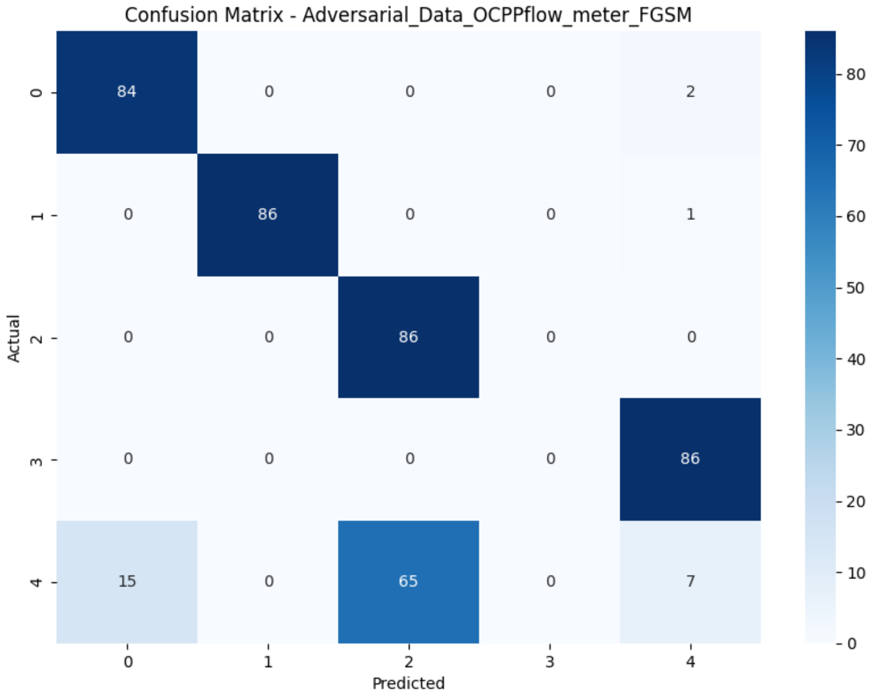
	White Box					Black Box
	FGSM	PGD	BIM	JSMA	C&W	ZOO
Accuracy	0.5109	0.3434	0.3434	0.5659	0.7417	0.7307
TPR	0.5116	0.3433	0.3433	0.5660	0.7413	0.7304
FPR	0.1219	0.1641	0.1641	0.1082	0.0646	0.0672
F1 score	0.4310	0.2712	0.2712	0.5232	0.7013	0.7024

2nd step is to apply the adversarial attacks and evaluate the performance of the model. The results show that the model's prediction has been decreased in comparison with the evaluation results in the 1st step. The most effective adversarial attack based on the results is with PGD and BIM.

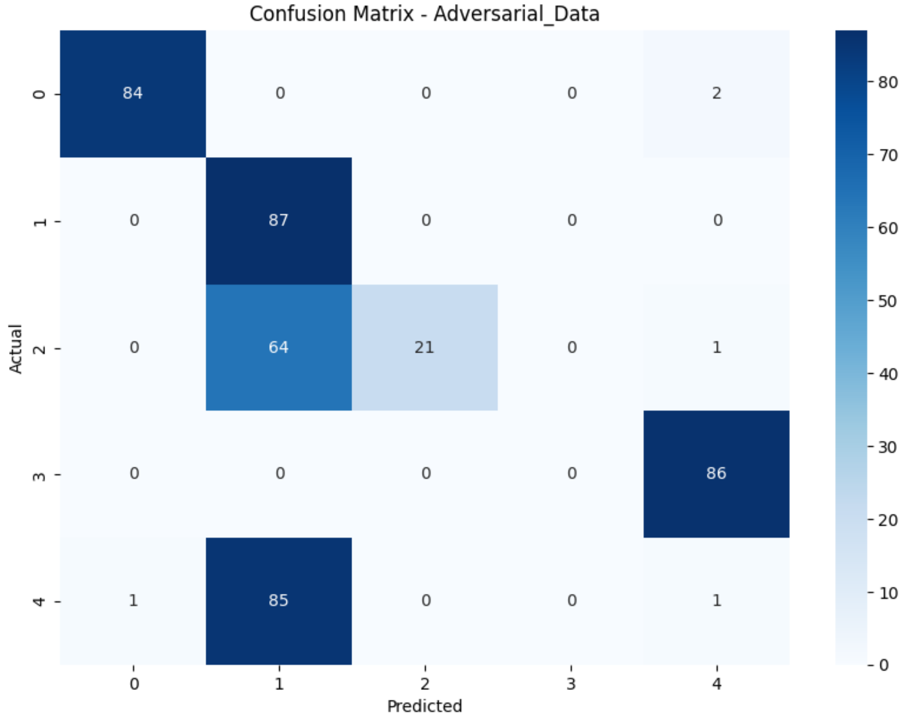


AAG Evaluation results (2)

FGSM



ZOO



Conclusions

Evaluated the impact of various adversarial attacks on ML models for intrusion detection using the OCPP dataset.

The white box and black box attacks were used in order to evaluate the resilience of the models.

While ML models effectively detect standard anomalies, adversarial attacks pose significant risks, emphasizing the need for robust defences in AI-driven intrusion detection systems.

Future work

Investigate and implement **defensive techniques** against adversarial attacks, such as: Adversarial training, Defensive distillation, Gradient masking

Extend the research to focus on **industrial control systems** and other high-stakes environments, evaluating defence strategies in real-world scenarios.

Aim to increase the resilience of AI security **models** against adversarial attacks to safeguard critical systems.

Thank you for your attention!