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[††], Georgios Th. Papadopoulos[‡‡] Evangelos K. Markakis[x], Panagiotis Sarigiannidis[‡]

***Abstract*—With the ongoing integration of machine learning models into critical infrastructure, the resilience of these systems against adversarial attacks is important for all domains. This paper introduces an adversarial attack generator framework against a network dataset that is part of OCPP Dataset using CICFlowMeter parser . We conduct a comprehensive evaluation of various prominent adversarial attacks, including FGSMA, JSMA, PGD, C&W, and more to assess their efficacy on the OCCP dataset. The Adversarial Generator is meticulously evaluated, demonstrating a significant impact in the models performance to detect potential perturbations. The results showcased the impact of the different type of adversarial attacks, contributing to a critical advancement in future defense strategies that need to be utilised in order to protect industrial control systems.**

***Index Terms*—Adversarial attacks, white-box, Black-box, evasion**

I. Introduction

In recent years, the introduction and evolution of Artificial Intelligence (AI) brought huge advances across many applications, such as image recognition, natural language processing, and autonomous systems. However, every advantage comes with a throwback. These kind of systems are threatened by multiple types of AI attacks, with one of them being adversarial attacks. Adversarial attacks, insert intentionally crafted perturbations to mislead model predictions and underscore a critical vulnerability in machine learning algorithms. This vulnerability compromises the reliability of AI systems and more specifically creates security risks in sensitive applications such as cybersecurity, healthcare, and autonomous systems in many domains for example energy sector [1]. In particular, multi-step attack scenarios and Advanced Persistent Threats (APTs) against critical infrastructures (such as the smart electrical grid) can result in various cascading effects with widespread service outages, financial losses, or even fatal accidents. AI has the potential to significantly improve defense systems by enabling the detection of unknown anomalies and zero-day cyberattacks [2]. However, AI-powered detection systems are vulnerable to hostile attempts that try to compromise their security and are prone to false alarms. This phenomenon has created the need to defend against this kind of attacks, by using more robust models.

In order to defend against cyberattacks that target the network systems protocol, first its important to have a better understanding of the adversarial attacks by creating an Adversarial Attack Generator (AG). Despite being widely used in Industrial IoT (IIoT) applications, particularly in the energy sector, network systems are characterized by serious security issues because it lacks authentication and access control mechanisms, making them possible for potential cyberattacker(s) to carry out unauthorized and Man-In-the-Middle (MITM) activities [9]. In terms of Machine Learning (ML) and Deep Learning (DL) models, different AI techniques are utilized for this goal, Random Forest for black-box attack, and Multi-Layer Perceptron (MLP) for white-box attacks. More specifically, we use multiple adversarial attacks to test our system. Therefore, based on the aforementioned remarks, the contributions of this paper are summarized as follows.

- **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 with each other.

[*] This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101070450.

[*] Dimitrios-Christos Asimopoulos is with MetaMind Innovations, Kila, 50100 Kozani, Greece - E-Mail: dasimopoulos@metamind.gr

[†] Dimitrios-Christos Asimopoulos is also with the Department of Information and Electronic Engineering, International Hellenic University, Sindos Campus 57400, Thessaloniki, Greece - E-Mail: dimiasim3@ihu.gr

[‡] P. Radoglou-Grammatikis and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Campus ZEP Kozani, 50100 Kozani, Greece - E-Mail: {pradoglou,psarigiannidis}@uowm.gr

[§] P. Radoglou-Grammatikis is also with K3Y, Sofia, Bulgaria - E-Mail: pradoglou@k3y.gr

[¶] T. Lagkas is with the Department of Computer Science, Democritus University of Thrace, Kavala Campus, 65404, Kavala, Greece - E-Mail: tlagkas@cs.duth.gr

[‖] V. Argyriou is with Kingston University London, Surrey, UK - E-Mail: vasileios.argyriou@kingston.ac.uk

[**] I. Moscholios is with the Department of Informatics & Telecommunications, University of Peloponnese, Tripolis, 22131, Greece - E-Mail: idm@uop.gr

[††] Jorgen Cani and Georgios Th. Papadopoulos are with the Department of Informatics and Telematics, Harokopio University of Athens, Thiseos 70, Athens, GR 17676, Attiki, Greece - E-Mail: {cani,g.th.papadopoulos}@hua.gr

[x] E. K. Markakis is with the Hellenic Mediterranean University, 71004 Heraklion, Greece - E-Mail: emarkakis@hmu.gr

- **Investigation of various adversarial attacks (FGSM, BIM, PGD, C& W, JSMA, ZOO):** We investigate how various adversarial attacks affect the detection performance of the previous ML/DL models.

The rest of the paper is organized as follows. Section II presents a background and similar works in this field. In section III, our Adversarial Generator against CICFlow Meter parser is described. Next section IV provide an overview of the different type of adversarial attacks used in this study. Finally, section VI focuses on the evaluation analysis and experimental results, while VII concludes this paper.

II. Related Work

Adversarial attacks target machine learning (ML) and deep learning (DL) models by subtly altering input data to induce errors in predictions. These attacks are categorized based on the attacker's knowledge into white-box (full system knowledge), black-box (access only to input/output), and grey-box (limited model knowledge) approaches. White-box attacks leverage model gradients for optimized perturbations, while black-box attacks explore input-output relationships through queries. Grey-box attacks utilize partial knowledge, often exploiting transferability from other models. Adversarial tactics also include poisoning (corrupting the training data to impair learning) and evasion (altering inputs to cause misclassification without detection by humans), both aiming to exploit model vulnerabilities for manipulation. Understanding these strategies is crucial for developing robust defenses against such malicious interventions. In [3], Yihua Zhang et al, introduce an adversarial training method utilizing bi-level optimization to enhance deep neural network robustness against adversarial attacks. This approach sidesteps the limitations of conventional methods by avoiding gradient sign-based generation, leading to significant robustness improvements without explicit robust regularization. This method demonstrates superior performance across various models and datasets. In [4], Narmin Ghaffari Laleh et al, investigate the vulnerability of AI models, specifically convolutional neural networks (CNNs), to adversarial attacks in oncology diagnostic workflows. They highlight the susceptibility of CNNs to both white- and black-box attacks in weakly-supervised classification tasks. Exploring mitigation strategies like adversarially robust training and dual batch normalization, they find their effectiveness limited without precise attack knowledge, and demonstrate that vision transformers (ViTs) offer superior robustness to these attacks, attributed to more resilient latent representations of clinical categories. Their recommendation aligns with theoretical insights, advocating for ViTs over CNNs for enhancing model security in clinical applications. In [[5]] Jiacheng Huang and Long Chen introduce a defense against word-level adversarial attacks in natural language processing by leveraging a semantic associative field for textual embedding. Recognizing the necessity for a relation between original and perturbed words. This approach enhances word embeddings through related word vectors and weighted sampling, simulating semantic interconnections. Extensive experiments show that this method outperforms traditional defenses, offering universality and maintaining training efficiency without depending on model architecture. In [6] Tao Bai et al explore the advancements in adversarial training to enhance the robustness of deep learning models against adversarial attacks. With a taxonomy, they review recent progress, address generalization issues from multiple perspectives, and highlight unresolved challenges. This comprehensive analysis identifies potential future directions for research in making models inherently resistant to adversarial threats.

In their survey, N. Martins et al. examine adversarial threats against intrusion and malware detection outlined in [7]. The study assesses attack techniques like L-BFGS, FGSM, JSMA, DeepFool, C&W, GAN-based methods, and ZOO, then reviews defenses including adversarial training, gradient masking, defensive distillation, feature squeezing, and universal perturbation defenses. Additionally, it explores how adversarial attacks are utilized in intrusion and malware detection systems, highlighting the need for future research in this area. In [8] Aleksander Madry et al introduce enhancing the adversarial robustness of neural networks via robust optimization, offering a comprehensive perspective on existing efforts to combat adversarial attacks. This principled approach facilitates the development of reliable, universally applicable methods for training and defending against adversaries, providing concrete security guarantees. By focusing on resistance against first-order adversaries, they pave the way for deep learning models that are inherently more secure and robust. In [9] Afnan Alotaibi and Murad A. Rassam survey the intersection of adversarial machine learning and intrusion detection systems (IDS), highlighting the dual challenge of detecting malicious activities while mitigating the risk of misclassification due to novel attacks. They explore the potential of machine learning to enhance IDS accuracy, acknowledging the vulnerability of these systems to adversarial perturbations that can disrupt threat detection. By examining various adversarial attacks and defense mechanisms, they provide insights into reducing their impact on IDS. The survey also identifies existing research gaps and proposes directions for future investigation, emphasizing the need for robust defense strategies in the evolving cybersecurity landscape.

III. Adversarial Generator

As depicted in Fig 1, the architecture of the proposed Adversarial Attack Generator consists of four three modules: a) Adversarial Attack Engine, b) Attack Evaluation Module, and c) Testing and Notification Module. The first module generates adversarial examples to add perturbations to the dataset using adversarial attacks. These attacks are listed in a library named Strategy Library. It contains various adversarial attack algorithms (FGSM, PGD, JSMA, BIM, ZOO, C& W) to test against the model.

The second module is responsible for assessing the effectiveness of the adversarial attacks, generated by the adversarial attack engine, by using various ML/DL models from an evaluation model library. More specifically, there are two different

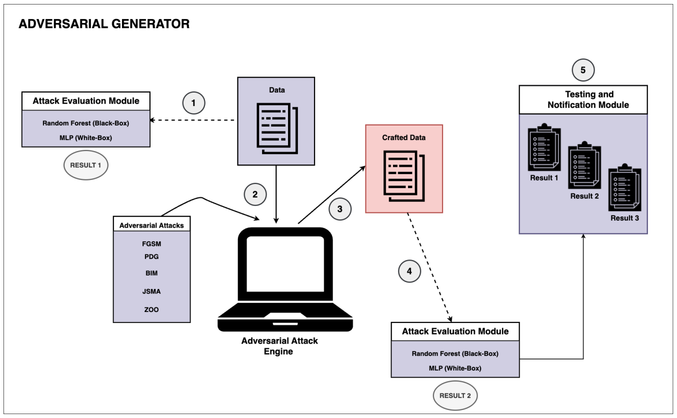

Fig. 1: Adversarial Generator

scenarios tested: a) White-Box attacks where the attacker has knowledge of the model used and b) Black-Box attack where the only knowledge the attacker has is the dataset. The final module is responsible for testing the models after their training in the Adversarial Attack Engine by comparing the results and communicating the results of the attacks.

IV. Overview of the Adversarial Attacks

This section explores different adversarial attacks, for recognizing vulnerabilities in deep learning models. The attacks that were investigated are the Gast Gradient Sign Method (FGSM); Basic Iterative Method (BIM), which progressively refines adversarial perturbations; the Projected Gradient Descent (PGD), esteemed for its efficiency and labeled as the ”universal adversary”; the Jacobian-based Saliency Map Attack (JSMA), focusing on exploiting model sensitivity to input features; the Zeroth Order Optimization (ZOO) attack, facilitating black-box attacks without gradient information; and the Carlini & Wagner (C& W) attack, noted for its complexity and capability to evade defensive measures. These attacks are analyzed and investigated as to how they mislead deep learning models, and finally, the need for defense mechanisms is investigated.

A. Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is a widely recognized adversarial attack technique used to generate adversarial examples by exploiting the vulnerabilities of machine learning models. Developed by Goodfellow et al., FGSM works by adding a perturbation to the input data that maximizes the model's prediction error. This perturbation is crafted by computing the gradient of the loss function with respect to the input data and then applying a small, scaled step in the direction of the gradient sign. The process is mathematically represented as

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla$$

x J$(\theta, \mathbf{x}, \mathbf{y}))$, where $\mathbf{x}_{\text{adv}}$ is the adversarial example, $\mathbf{x}$ is the original input, ϵ is a small constant determining the perturbation size, J is the loss function, ` represents the model parameters, and $\mathbf{y}$ is the true label. FGSM is effective due to its simplicity and efficiency, making it a fundamental method for evaluating the robustness of machine learning models against adversarial attacks.

B. Jacobian-based Saliency Map Attack (JSMA)

The Jacobian-based Saliency Map Attack (JSMA) focuses on the manipulation of inputs to deceive deep learning models [[10]]. JSMA is a greedy algorithm that utilizes the saliency map concept, a method that aims to identify and modify pixels that have the most significant input on the output. In this way, misclassification is achieved and the perturbations are few and precise making the attack difficult in detection. The saliency map in JSMA is calculated using the gradient of the model's output with respect to its input, aimed at identifying the most impact changes to the input that would result in misclassification. The calculation of the saliency map is detailed through the following steps.

The first step involves computing the Jacobian matrix of the model's output with respect to its input. For a given input vector X and a model F that outputs a probability distribution over classes, the Jacobian matrix *Jf(X)* is defined as:

$$J_F(X) = \left[\frac{\partial F_j(X)}{\partial X_i}\right]_{i,j} \tag{1}$$

where i indexes the input features and j indexes the output classes. This matrix captures how changes in each input feature influence the predictions for each class.

Second step is to decide which features to modify. In order to do that JSMA calculates a saliency map based on the Jacobian matrix. The saliency score $S_{\text{map}}(i, \theta)$ for modifying feature i towards changing the class to a target class θ is defined as:

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

This formula ensures that feature i is only considered for perturbation if its modification increases the probability of the target class θ (positive gradient) and does not increase the sum of probabilities for other classes (negative sum of gradients for non-target classes).

Based on the saliency map, features with the highest saliency scores are selected for perturbation. The adversarial example X is then crafted by applying a perturbation δ to the selected features of the original input X, aiming to mislead the model into classifying X as the target class θ. By iteratively applying these steps, JSMA generates an adversarial example that is visually similar to the original input but is classified differently by the target model, demonstrating the effectiveness of this attack in exploiting model vulnerabilities.

C. Project Gradient Descent (PGD)

The Projected Gradient Descent (PGD) is a white-box adversarial attack, wherein its main operation is to apply a small perturbation to the input data at every iteration.

The perturbation is generated by multiplying the sign of the gradient of the loss function with respect to the input data and the direction it faces, guiding the input towards the direction that maximizes the loss. These steps are repeated for a fixed number of iterations or until the input is misclassified. [[11]]

PGD can be considered as executing the Fast Gradient Sign Method (FGSM) multiple times with small steps, while projecting the adversarial samples back onto the ℓ_∞ ball containing perturbations after each step. This ensures that the perturbations do not become overly large and remain undetectable.

The algorithm is initialized by setting the perturbation δ to small random values. At each step of the algorithm, the gradient of the loss function $\nabla_x J(\theta, x, y)$ with respect to the input x is computed. The perturbation δ is then updated in the direction of this gradient, scaled by a small factor α, and the result is clipped to ensure it remains within the specified ϵ-bounded ℓ_∞ ball:

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

where Clip_ϵ ensures that $||\delta_{\text{new}}||_\infty \leq \epsilon$, thereby keeping the adversarial perturbations within the allowable range. This iterative process is repeated until the adversarial example x' is misclassified or the maximum number of iterations is reached.

D. Basic Interative Method (BIM)

The Basic Iterative Method (BIM), an evolution of the Fast Gradient Sign Method (FGSM), stands as a sophisticated technique designed to test the robustness of deep learning models [[12]]. BIM iteratively applies small but targeted perturbations to the input data, with each step calculated to maximally increase the loss function with respect to the model's current prediction, thus steering the model towards misclassification. The iterative nature of BIM allows for more precise control over the perturbation process compared to FGSM, enabling the generation of adversarial examples that are both effective in deceiving models and subtle to human observers. The update formula for an adversarial example at iteration n is given by:

$$X^{(n+1)} = \text{Clip}_{X,\varepsilon}\left\{X^{(n)} + \alpha \cdot \text{sign}\left(\nabla_X J(\theta, X^{(n)}, Y_{\text{true}})\right)\right\} \quad (3)$$

where $X^{(n+1)}$ is the adversarial example at iteration $n+1$, $X^{(n)}$ is the adversarial example from the previous iteration, α is the step size, $\text{Clip}_{X,\varepsilon}$ is a clipping function that ensures the perturbed image does not go beyond an ε-neighbourhood of the original image, ∇_X denotes the gradient with respect to X, J is the loss function, θ represents the model parameters, and Y_{true} is the true label. This methodical adjustment of the input exemplifies the calculated exploitation of model vulnerabilities, highlighting the critical importance of incorporating adversarial robustness in the development and evaluation of machine learning models. By crafting inputs that lead to consistent misclassification, BIM not only exposes potential weaknesses in model architectures but also serves as a benchmark for enhancing their defensive capabilities against adversarial threats.

E. Carlini and Wagner (C&W)

Developed by Nicholas Carlini and David Wagner, the C&W attack is a sophisticated method designed to generate adversarial examples with minimal perturbation, aiming to fool neural network classifiers. Unlike earlier attacks, the C&W method focuses on crafting adversarial samples that are almost indistinguishable from original samples, highlighting vulnerabilities in deep learning models. The core of the C&W attack is an optimization problem designed to find the smallest change to the input data that results in a misclassification. It can be formulated in terms of different norms (L_0, L_2, and L_∞), each representing a different measure of perturbation size:

- L_0 norm focuses on altering the least number of components in the input vector.
- L_2 norm minimizes the Euclidean distance between the original and the adversarial example.
- L_∞ norm limits the maximum change to any component of the input vector.

The attack uses gradient descent to minimize a loss function that combines the misclassification objective with a term controlling the size of the perturbation, effectively balancing between imperceptibility and misclassification rate. The main idea of classifiers is the optimization:

$$\text{minimize } \|x - x_0\|_2^2 + c \cdot l(x), \quad (4)$$

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

C&W's objective is to minimize the first equation, where $x - x_0$ ensures the adversarial example x is close to the original example x_0, making this way the perturbation more undetectable, and $l_2(x)$ ensures that the adversarial example is misclassified into a specific class. By solving this optimization problem, we find an adversarial example that is both close to the original image and misclassified as desired, fulfilling the objectives of the C&W adversarial attack. The result is a powerful method to test and potentially exploit the vulnerabilities of machine learning classifiers.

F. Zeroth Order Optimization (ZOO)

Zeroth Order Optimization is an adversarial attack on machine learning models which unlike the Carlini & Wagner (C&W) attack uses gradient descent and requires access to the model's gradients. The ZOO attack is a black-box attack method that does not require such access. Instead, it estimates the gradients using only function evaluations, hence the term ”zeroth order,” which refers to using zeroth order (or direct) optimization techniques. The ZOO attack aims to create adversarial examples without needing to know the internal workings of the model. The attacker only has access to the output of the model, such as the final classification scores. It uses a technique known as finite differences to estimate the gradient of the model's loss function concerning the input. This is done by slightly perturbing the input and observing the change in output. The process begins with the attacker perturbing the input slightly and observing the corresponding changes in the output. By employing different methods, the gradient of the model's loss function concerning the input is estimated. These estimations are then used to adjust the input incrementally, to maximize the loss function. Through iterative optimization, the ZOO attack carefully crafts an adversarial example that misleads the model into a false classification. This gradient estimation and optimization approach allows the ZOO attack to sidestep the need for internal model details, making it a potent tool for assessing the robustness of machine learning classifiers in situations where an adversary is limited to only query access to the model.

V. Experiment Setup

The experimental results were carried out with Windows 11 Pro, Intel Core i9-10980XE CPU @ 3.00 GHz, Nvidia, 64GB Random Access Memory (RAM) and 1TB Solid Disk Drive (SSD). Notably, to handle the intensive computations inherent in ML/DL tasks, a GeForce 3080 Ti GPU was employed. The preferred deep learning framework for this experiment was TensorFlow, chosen for its adaptability and seamless integration with the chosen hardware setup.

A. Dataset

The dataset utilized in this study is part of the OCPP (Open Charge Point Protocol) Dataset, which was parsed using CICFlowMeter to extract network flow statistics. This dataset comprises various network flow features recorded in PCAP CSV format, providing detailed insights into network behavior. Key input features include flow duration, total forward and backward packets, total length of forward and backward packets, packet length statistics, and various inter-arrival times, among others. These features are crucial for detecting network anomalies and potential cyberattacks. The dataset includes both normal/benign traffic and multiple types of cyberattacks such as FDI Charging Profile, DOC ID Tag, DOS Flooding Heartbeat, and DOS Flooding EVCS Rejected attacks. Preprocessing steps involved feature engineering to drop non-predictive features, identifying and handling null values, label encoding of target values, and standard scaling of the features to ensure they are on a common scale. This comprehensive dataset was then used to create an adversarial dataset using as mentioned above the Adversarial Attack Generator in order to compare the results between different type of attacks.

B. Evaluation Metrics

The metric of accuracy (Equation 5) measures the proportion of correct classifications in relation to the total instances. This evaluation metric is considered appropriate when the training dataset is balanced, meaning it contains an equal number of instances for all classes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where:

$$\begin{aligned} TP &\rightarrow \text{True Positives} \\ TN &\rightarrow \text{True Negatives} \\ FP &\rightarrow \text{False Positives} \\ FN &\rightarrow \text{False Negatives} \end{aligned}$$

TPR (Equation 6) represents the fraction of actual intrusion instances that were correctly identified as intrusions.

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

FPR (Equation 7) indicates the proportion of normal instances that were incorrectly classified as cyberattacks, reflecting the balance between the accurate identification of normal instances and the occurrence of false alarms.

$$FPR = \frac{FP}{FP + FN} \tag{7}$$

The F1 score (Equation 8) is a metric that captures the balance between true positive rate (TPR) and precision. Precision is defined as the ratio of true positives to the sum of true positives and false positives.

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{8}$$

VI. Experimental Results

The experimental results were conducted in two steps. First step, as shown in the architecture design of Adversarial Attack Generator, is to evaluate the models using the clean test dataset before using the adversarial attacks to craft the original dataset. For the evaluation two models where used: Random Forest and a custom MLP. The MLP is a Sequential neural network model implemented using Keras to perform multi-class classification. The architecture of the model consists of seven fully connected (Dense) layers. The first layer includes 32 neurons with ReLU activation, followed by a second layer with 64 neurons and Tanh activation. The subsequent layers alternate between 32 and 64 neurons, all utilizing ReLU activation, to effectively capture non-linear relationships within the data. The final layer comprises 5 neurons with a softmax activation function, designed to output a probability distribution across the 5 target classes. This model, with a total of 14,309 trainable parameters, was selected for its capacity to learn complex patterns and provide accurate classifications. The combination of varying neuron counts and activation functions across layers ensures a robust representation of the input features, facilitating improved model performance on the classification task. The results as shown in Table I are good since the Random Forest model achieved $Accuracy = 0.9909$, $TPR = 0.9909$, $FPR = 0.0130$ and $F1 = 0.9909$ the MLP model achieved

TABLE I: Evaluation Metrics in original dataset

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

TABLE II: Evaluation Metrics in adversarial dataset

	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

$Accuracy = 0.9890$, $TPR = 0.9890$, $FPR = 0.0212$ and $F1 = 0.0.9890$.

The second step of the evaluation process is to use MLP to compare the impact of the different white-box attacks used to the original dataset and Random Forest to compare the results of the black-box attack. The results highlight the varying effectiveness of different adversarial attack techniques. Among the white-box attacks, PGD and BIM stand out as the most effective, achieving the lowest accuracy and TPR with the highest FPR, making them the most potent methods for degrading the model's classification performance. FGSM and JSMA have a moderate impact, while the C&W attack is the least effective, maintaining the highest accuracy and TPR with the lowest FPR.

In the black-box scenario, the ZOO attack proves to be less effective, nearly matching the performance of the least impactful white-box attack (C&W). This underscores the resilience of models against black-box attacks compared to white-box attacks. Overall, these metrics provide critical insights into the strengths and weaknesses of each attack method, guiding the development of more robust defenses against adversarial attacks in machine learning models.

VII. Conclusion & Future Work

In conclusion, this study explored the efficacy of various adversarial attacks on machine learning models used for intrusion detection using the OCPP dataset. On the one hand, the results showcased that machine learning models can detect multiple anomalies in different sectors, and more specifically in the network section as mentioned and studied. However, in the digital era of AI there are many techniques used to bypass intrusion detector leading to devastating consequences. In particular, first the paper studied the detection accuracy of ML/DL models such as a custom MLP and Random Forest against the original OCPP dataset. The results showed that both models performed excellent in detecting anomalies. Next, adversarial attacks were implemented in order to test there impact against ML/DL models. The attacks utilised were devided into white-box and black-box attacks. White-box attacks used the custom MLP model and the FGSM, PGD, BIM, JSMA, and C&W attacks, while black-box attacks used ZOO attack and Random Forest as a classification model. The results proved that adversarial attacks are capable to evade and craft the datasets in order to mislead the classification models. Based on the results our future plan aim to investigate the different type of defences against adversarial attacks in order to improve the resilience of the AI security models and contribute to the protection of industrial control systems.

Acknowledgment

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

References

[1] P. R. Grammatikis, P. Sarigiannidis, A. Sarigiannidis, D. Margounakis, A. Tsiakalos, and G. Efstathopoulos, "An anomaly detection mechanism for iec 60870-5-104," in *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. Bremen, Germany: IEEE, 2020, pp. 1–4.

[2] V. Kumar and D. Sinha, "A robust intelligent zero-day cyber-attack detection technique," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2211–2234, 2021.

[3] Y. Zhang, G. Zhang, P. Khanduri, M. Hong, S. Chang, and S. Liu, "Revisiting and advancing fast adversarial training through the lens of bi-level optimization," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 693–26 712.

[4] N. Ghaffari Laleh, D. Truhn, G. P. Veldhuizen, T. Han, M. van Treeck, R. D. Buelow, R. Langer, B. Dislich, P. Boor, V. Schulz *et al.*, "Adversarial attacks and adversarial robustness in computational pathology," *Nature communications*, vol. 13, no. 1, p. 5711, 2022.

[5] J. Huang and L. Chen, "Defense against adversarial attacks via textual embeddings based on semantic associative field," *Neural Computing and Applications*, vol. 36, no. 1, pp. 289–301, 2024.

[6] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," *arXiv preprint arXiv:2102.01356*, 2021.

[7] N. Martins, J. M. Cruz, T. Cruz, and P. H. Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: a systematic review," *IEEE Access*, vol. 8, pp. 35 403–35 419, 2020.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[9] A. Alotaibi and M. A. Rassam, "Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense," *Future Internet*, vol. 15, no. 2, p. 62, 2023.

[10] W. Zhang, X. Zhang, K. Hao, J. Wang, and S. Zhang, "Optimized jacobian-based saliency maps attacks," *International Journal of Network Security*, vol. 24, no. 6, pp. 1020–1030, 2022.

[11] W. Villegas-Ch, A. Jaramillo-Alcázar, and S. Luján-Mora, "Evaluating the robustness of deep learning models against adversarial attacks: An analysis with fgsm, pgd and cw," *Big Data and Cognitive Computing*, vol. 8, no. 1, p. 8, 2024.

[12] B. H. Mohammed, H. Sallehudin, S. A. Mohamed, N. S. M. Satar, and A. H. B. Hussain, "Internet of things-building information modeling integration: Attacks, challenges, and countermeasures," *IEEE Access*, vol. 10, pp. 74 508–74 522, 2022.