Realistic Adversarial Machine Learning to improve Network Intrusion Detection

Research Group on Intelligent Engineering and Computing
for Advanced Innovation and Development

2022 / 2023

João Pedro Machado Vitorino

1180851

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Realistic Adversarial Machine Learning to improve Network Intrusion Detection

Research Group on Intelligent Engineering and Computing
for Advanced Innovation and Development

João Pedro Machado Vitorino

1180851

Thesis submitted for:

Master's Degree in Artificial Intelligence Engineering

Supervised by:

Dr. Isabel Cecília Correia da Silva Praça Gomes Pereira

Coordinator Professor, School of Engineering, Polytechnic of Porto

Dr. Eva Catarina Gomes Maia

Senior Research Fellow, School of Engineering, Polytechnic of Porto

Thesis Jury

President:

Dr. Maria Goreti Carvalho Marreiros
Coordinator Professor with Aggregation, School of Engineering, Polytechnic of Porto

Vocals:

Dr. Pedro Ricardo Morais Inácio
Associate Professor, University of Beira Interior

Dr. Isabel Cecília Correia da Silva Praça Gomes Pereira
Coordinator Professor, School of Engineering, Polytechnic of Porto

Porto, July 2023

« The quieter you become,
the more you are able to hear. »
Rumi, the poet

Acknowledgments

The work described in this thesis was partially supported by the European Union's Horizon 2020 research and innovation program, under project SeCoIIA (grant agreement no. 871967) and project VALU3S (grant agreement no. 876852).

I would like to thank GECAD for giving me this challenge, as well as all those who supported me throughout this journey. In particular:

To my supervisors, Isabel and Eva, for their invaluable advice and feedback.

To my teammates, especially Shaq and Tiago, for all the moments we shared.

To my family, for their unconditional support. To my parents, Margarida and Mário, to whom I owe who I am today. To my brother André, who is always there for me.

To Joana, for the constant encouragement.

Abstract

Modern organizations can significantly benefit from the use of Artificial Intelligence (AI), and more specifically Machine Learning (ML), to tackle the growing number and increasing sophistication of cyber-attacks targeting their business processes. However, there are several technological and ethical challenges that undermine the trustworthiness of AI.

One of the main challenges is the lack of robustness, which is an essential property to ensure that ML is used in a secure way. Improving robustness is no easy task because ML is inherently susceptible to adversarial examples: data samples with subtle perturbations that cause unexpected behaviors in ML models. ML engineers and security practitioners still lack the knowledge and tools to prevent such disruptions, so adversarial examples pose a major threat to ML and to the intelligent Network Intrusion Detection (NID) systems that rely on it.

This thesis presents a methodology for a trustworthy adversarial robustness analysis of multiple ML models, and an intelligent method for the generation of realistic adversarial examples in complex tabular data domains like the NID domain: Adaptative Perturbation Pattern Method (A2PM). It is demonstrated that a successful adversarial attack is not guaranteed to be a successful cyber-attack, and that adversarial data perturbations can only be realistic if they are simultaneously valid and coherent, complying with the domain constraints of a real communication network and the class-specific constraints of a certain cyber-attack class.

A2PM can be used for adversarial attacks, to iteratively cause misclassifications, and adversarial training, to perform data augmentation with slightly perturbed data samples. Two case studies were conducted to evaluate its suitability for the NID domain. The first verified that the generated perturbations preserved both validity and coherence in Enterprise and Internet-of-Things (IoT) network scenarios, achieving realism. The second verified that adversarial training with simple perturbations enables the models to retain a good generalization to regular IoT network traffic flows, in addition to being more robust to adversarial examples.

The key takeaway of this thesis is: ML models can be incredibly valuable to improve a cybersecurity system, but their own vulnerabilities must not be disregarded. It is essential to continue the research efforts to improve the security and trustworthiness of ML and of the intelligent systems that rely on it.

Keywords: Realistic adversarial examples, Adversarial robustness, Tabular data,
Machine learning, Cybersecurity

Resumo

Organizações modernas podem beneficiar significativamente do uso de Inteligência Artificial (AI), e mais especificamente Aprendizagem Automática (ML), para enfrentar a crescente quantidade e sofisticação de ciberataques direcionados aos seus processos de negócio. No entanto, há vários desafios tecnológicos e éticos que comprometem a confiabilidade da AI.

Um dos maiores desafios é a falta de robustez, que é uma propriedade essencial para garantir que se usa ML de forma segura. Melhorar a robustez não é uma tarefa fácil porque ML é inerentemente suscetível a exemplos adversos: amostras de dados com perturbações subtis que causam comportamentos inesperados em modelos ML. Engenheiros de ML e profissionais de segurança ainda não têm o conhecimento nem as ferramentas necessárias para prevenir tais disrupções, por isso os exemplos adversos representam uma grande ameaça a ML e aos sistemas de Deteção de Intrusões de Rede (NID) que dependem de ML.

Esta tese apresenta uma metodologia para uma análise da robustez de múltiplos modelos ML, e um método inteligente para a geração de exemplos adversos realistas em domínios de dados tabulares complexos como o domínio NID: Método de Perturbação com Padrões Adaptativos (A2PM). É demonstrado que um ataque adverso bem-sucedido não é garantidamente um ciberataque bem-sucedido, e que as perturbações adversas só são realistas se forem simultaneamente válidas e coerentes, cumprindo as restrições de domínio de uma rede de computadores real e as restrições específicas de uma certa classe de ciberataque.

A2PM pode ser usado para ataques adversos, para iterativamente causar erros de classificação, e para treino adverso, para realizar aumento de dados com amostras ligeiramente perturbadas. Foram efetuados dois casos de estudo para avaliar a sua adequação ao domínio NID. O primeiro verificou que as perturbações preservaram tanto a validade como a coerência em cenários de redes Empresariais e Internet-das-Coisas (IoT), alcançando o realismo. O segundo verificou que o treino adverso com perturbações simples permitiu aos modelos reter uma boa generalização a fluxos de tráfego de rede IoT, para além de serem mais robustos contra exemplos adversos.

A principal conclusão desta tese é: os modelos ML podem ser incrivelmente valiosos para melhorar um sistema de cibersegurança, mas as suas próprias vulnerabilidades não devem ser negligenciadas. É essencial continuar os esforços de investigação para melhorar a segurança e a confiabilidade de ML e dos sistemas inteligentes que dependem de ML.

Palavras-chave: Exemplos adversos realistas, Robustez adversa, Dados tabulares, Aprendizagem automática, Cibersegurança

Contents

List of Figures

List of Tables

List of Acronyms

A2PM	Adaptative Perturbation Pattern Method
A&M	Access and Misuse
ACM	Association for Computing Machinery
AI	Artificial Intelligence
ANN	Artificial Neural Network
CGAN	Conditional Generative Adversarial Network
CVAE	Conditional Variational Autoencoder
DDoS	Distributed Denial-of-Service
DoS	Denial-of-Service
GAN	Generative Adversarial Network
GECAD	Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development
IAT	Inter-Arrival Time
IDT	In-Distribution
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet-of-Things
JSMA	Jacobian-based Saliency Map Attack
LGBM	Light Gradient Boosting Machine
MDPI	Multidisciplinary Digital Publishing Institute
ML	Machine Learning
MLP	Multilayer Perceptron
NID	Network Intrusion Detection
OOD	Out-of-Distribution
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RCN	Reconnaissance

RF	Random Forest
SeCoIIA	Secure Collaborative Intelligent Industrial Assets
SVM	Support Vector Machine
VAE	Variational Autoencoder
VALU3S	Verification and Validation of Automated Systems' Safety and Security
WGAN	Wasserstein Generative Adversarial Network
XGB	Extreme Gradient Boosting

1 Introduction

This chapter contextualizes the work described in this thesis and presents the problem statement, the established objectives, and the formulated research questions. The main scientific contributions and the structure of this document are also described.

1.1 Context and Motivation

The digital transformation is a paradigm change for both the private and public sectors. Throughout the 21st century, and especially since the COVID-19 pandemic, there has been an accelerated adoption of digital systems to meet the ever-increasing demand for interconnected services, electronic commerce, remote working, and more resilient supply chains. The European Commission considers this as one of the top priorities for the coming years, as presented in its targets for 2030 in the "A Path to the Digital Decade" policy program [1].

Modern organizations can benefit from the technological advances associated with this transformation to re-engineer their business processes, integrating their control and information systems and automating their decision-making procedures. Nonetheless, as organizations become more and more dependent on digital systems, the threat posed by cyber-attacks skyrockets [2]. Every novel technology adds new vulnerabilities that can be exploited in multiple attack vectors to disrupt the normal operation of a system. This is particularly concerning for organizations that deal with confidential information and sensitive personal data, or manage critical infrastructure, such as the healthcare and energy sectors [3].

The disruptions caused by a successful cyber-attack can be extremely costly for an organization. In 2022, the average cost of a data breach was reported to be 4.35 million United States dollars, an increase of 12.7% since 2020 [4]. This continued growth of both the number of successful cyber-attacks and their associated costs in various sectors and industries denotes that modern organizations face tremendous security challenges. Furthermore, since attack mitigation is not a trivial process and small enterprises commonly fall short of security best practices, most go out of business within 6 months of suffering a breach [5].

With financial security and business continuity on the line, it is essential for enterprises of all sizes to adequately monitor their systems, detect suspicious activity, and mitigate possible threats. This is where Artificial Intelligence (AI), and more specifically Machine Learning (ML), can be incredibly valuable [6]. ML models can originate from numerous algorithms, including tree-based algorithms and deep learning algorithms based on Artificial Neural Networks (ANNs), and can be trained to automate several tasks, ranging from the recognition of patterns and anomalies in benign network traffic to the classification of complex cyber-attack classes.

The adoption of intelligent cybersecurity solutions can shorten the time required to detect an intrusion by up to 76 days, leading to cost savings of up to 3 million United States dollars [4]. However, despite the benefits of ML to tackle the growing number and increasing sophistication of cyber-attacks, it is not flawless. There are several technological and ethical challenges that undermine its trustworthiness and hinder a large-scale adoption of intelligent systems, including the lack of explainability, interpretability, fairness, and robustness [7].

Robustness is a highly desirable property because it is essential to ensure that ML is used in a secure way. Nonetheless, improving robustness is no easy task because ML is inherently susceptible to adversarial examples: data samples with subtle perturbations that cause unexpected behaviors in ML models [8]. For instance, a sample that originates from a faulty data recording of benign network traffic may cause a misclassification in an intelligent NID system that relies on ML, leading to false alarms. Furthermore, even though the malicious purpose of a cyber-attack causes it to have distinct characteristics that could be recognized in a thorough analysis by security practitioners, an attacker may craft an adversarial cyber-attack example with specialized inputs capable of evading detection.

ML engineers and security practitioners still lack the knowledge and tools to prevent such disruptions, so adversarial examples pose a major threat to ML and to the intelligent systems that rely on it [9], [10]. To improve the security of ML, reliable defense strategies must be adopted during the training and deployment of a model [11]. Nonetheless, efforts to increase a model's robustness against all perturbed data samples that might occur must not disregard the importance of the model's generalization to regular data samples that it will certainly encounter when it is deployed. Therefore, the lack of robustness of current ML models is a pertinent challenge that requires further research efforts [7].

This thesis details the research and development work performed to address ML robustness, with a focus on the NID domain. It was done in the scope of the ongoing work of the intelligent systems laboratory of the Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD) [12]. GECAD is a research unit within the School of Engineering, Polytechnic of Porto, with the mission of developing scientific research and innovation for the incorporation of intelligence in engineering and decision sciences.

The developed work was aligned with the participation of GECAD in two international projects of the European Union's Horizon 2020 research and innovation program: Secure Collaborative Intelligent Industrial Assets (SeCoIIA) [13] and Verification and Validation of Automated

Systems' Safety and Security (VALU3S) [14]. The former aimed at securing the digital transition of the manufacturing industry, addressing industrial Internet-of-Things (IoT) communication networks in aerospace, automotive, and naval construction use cases. The latter aimed at improving the verification and validation processes of automated systems to ensure their safety and robustness in various complex domains.

1.2 Problem Statement

In the Network Intrusion Detection (NID) domain, cyber-attacks can be identified by analyzing the characteristics of network traffic flows, which are represented in a tabular data format. The features of a flow may be required to follow specific data distributions, according to the specificities of a communication network and of the utilized protocols. Furthermore, due to their distinct malicious purposes, different cyber-attacks may exhibit entirely different feature correlations. Since a data sample must represent a real network flow, either benign activity or a cyber-attack class, it must fulfill all the constraints of this complex tabular data domain.

An ML model trained for a classification task in a domain like NID would learn to distinguish between two or more classes based on the characteristics of its training data. Considering that the provided training set correctly represented the target domain, the model would be able to generalize well to new data that is In-Distribution (IDT) within that specific domain, so it would correctly classify previously unseen samples that fulfill all the required constraints. Therefore, in a proper deployment, a model must only be provided IDT data and it must be safeguarded from Out-Of-Distribution (OOD) data because it does not represent real network traffic.

However, adversarial attack methods commonly generate random data perturbations [15], which can lead to OOD adversarial examples in tabular data domains. Throughout the current scientific literature, various studies apply adversarial attacks to complex domains like NID and provide the examples as direct input to a model without questioning if they are viable for a real deployment scenario [16]. This may result in misleading robustness evaluations where a model seems to be robust because it was tested against unrealistic attacks with OOD examples that it will not encounter in a real scenario in the target domain.

Moreover, augmenting a model’s training set with the examples created by adversarial attack methods, a defense designated as adversarial training [17], may not be as beneficial as it seems. Even though it is meant to improve robustness, training with OOD samples will make a model learn distorted characteristics that will not be exhibited by IDT samples [18]. This raises a major security concern because including unrealistic data in a training set can not only be detrimental to a model’s generalization, but also lead to accidental data poisoning and to the introduction of hidden backdoors that leave a model even more vulnerable [19].

In short, the lack of constrained data generation approaches is a major obstacle to ML robustness because realistic adversarial examples can only be crafted if the constraints of the NID domain are fulfilled during the generation of adversarial data perturbations.

1.3 Objectives and Research Questions

The main goal of this thesis was developing a solution to improve the security of ML in complex tabular data domains, with a focus on the NID domain, tackling the lack of robustness through an adversarial training approach with realistic adversarial examples. With that broader goal in mind, four more specific objectives were established:

- **OB1**: Investigate the state-of-the-art adversarial ML methods, constrained data generation approaches, and their applications.
- **OB2**: Formulate an approach to perform a trustworthy analysis of a model's adversarial robustness in a realistic NID scenario.
- **OB3**: Develop a method capable of crafting realistic adversarial examples for adversarial attacks and training in the NID domain.
- **OB4**: Validate and test the developed method in NID case studies adequate for the SeCoIIA and VALU3S projects.

To guide the research performed in the scope of this thesis and successfully accomplish the established objectives, the main research question to be investigated was carefully formulated: *"How can specific constraints be fulfilled during data generation to create realistic adversarial examples for NID?"*. The main question was divided into three narrower sub-questions:

- **RQ1**: "What are the state-of-the-art methods used to create adversarial examples for adversarial attacks and defense strategies?".
- **RQ2**: "What are the current approaches to generate data for adversarial ML according to specific constraints?".
- **RQ3**: "What are the most reliable NID datasets for the creation of realistic adversarial examples of network traffic flows?".

1.4 Scientific Contributions

Throughout the development of this thesis, significant research was performed, various concepts were introduced, and several experimental evaluations were conducted at GECAD. The main scientific contributions of the performed research and development work can be summarized in four key points:

- A literature review of the recent advances in adversarial ML methods and strategies, constrained data generation approaches, and public NID datasets.

- A methodology for an adversarial robustness analysis, with a realistic evasion attack vector and a comparison of multiple ML models.

- An intelligent method for the generation of realistic adversarial examples for adversarial attacks and training in complex tabular data domains.

- Two case studies of the realism of the examples crafted by the developed method and of the generalization of adversarially trained ML models in NID scenarios.

At the time of submission of this document, the developed work has resulted in a total of five peer-reviewed scientific publications. The publications included parts of the performed literature review, descriptions of the introduced concepts and methods, and experiments with several datasets and models. Nonetheless, two additional articles have been submitted and are currently under revision, and an additional manuscript is being prepared with further experiments for posterior submission.

Regarding scientific journals, two open access articles were published in impactful Q1 and Q2 journals of internationally recognized publishers, in the first and second quartiles, and another two articles are currently in a final revision stage before publication:

- **João Vitorino**, Nuno Oliveira, and Isabel Praça, "Adaptative Perturbation Patterns: Realistic Adversarial Learning for Robust Intrusion Detection", *Future Internet*, volume 14, issue 4, 2022, doi: 10.3390/fi14040108 [20].

- **João Vitorino**, Isabel Praça, and Eva Maia, "Towards Adversarial Realism and Robust Learning for IoT Intrusion Detection and Classification", *Annals of Telecomunications*, 2023, doi: 10.1007/s12243-023-00953-y [21].

- **João Vitorino**, Tiago Dias, Tiago Fonseca, Isabel Praça, and Eva Maia, "Constrained Adversarial Learning and its applicability to Automated Software Testing: a systematic review", to appear in *Information and Software Technology* (under revision) [22].

- **João Vitorino**, Isabel Praça, and Eva Maia, "SoK: Realistic Adversarial Attacks and Defenses for Intelligent Network Intrusion Detection", to appear in *Computers & Security* (under revision) [23].

Regarding scientific conferences, three papers were presented and published in the proceedings of several well-established conferences and symposiums, with experiments with several datasets and models in different domains:

- **João Vitorino**, Rui Andrade, Isabel Praça, Orlando Sousa, and Eva Maia, "A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection", presented in *14th International Symposium on Foundations and Practice of Security (FPS)*, 2022, pages 191–207, doi: 10.1007/978-3-031-08147-7_13 [24].

- Rui Andrade, **João Vitorino**, Sinan Wannous, Eva Maia, and Isabel Praça, "LEMMAS: a secured and trusted Local Energy Market simulation system", presented in *18th International Conference on the European Energy Market (EEM)*, 2022, pages 1–5, doi: 10.1109/EEM54602.2022.9921159 [25].

- **João Vitorino**, Lourenço Rodrigues, Eva Maia, Isabel Praça, and André Lourenço, "Adversarial Robustness and Feature Impact Analysis for Driver Drowsiness Detection", presented in *21st International Conference on Artificial Intelligence in Medicine (AIME)*, 2023, pages 108–113, doi: 10.1007/978-3-031-34344-5_13 [26].

In addition to the publications and to the direct contributions to the SeCoIIA and VALU3S projects, this thesis also contributed to the formulation of a new research and development project submission to the Horizon Europe funding program. The proposed project aims to demonstrate tools and techniques that enable enhanced collaboration, reduced response cost, and improved resilience of strategic European supply chains. It will address anomaly and attack detection, response optimization, and collective recovery capabilities.

1.5 Document Structure

This document is divided into multiple chapters that were organized to facilitate the reading of the thesis as a whole or of each chapter separately.

The current chapter, Chapter 1, contextualized the challenge addressed by this thesis and presented the objectives, the research questions, and the main scientific contributions.

Chapter 2 presents the performed literature review. It is divided into four sections: one for each research question, where the adopted research methodology is described and the findings are presented and discussed, and an additional one for the concluding remarks of the review.

Chapter 3 describes the types of constraints required for an adversarial cyber-attack example to be realistic, defines a methodology for a trustworthy adversarial robustness analysis, and details the design and implementation of the developed adversarial method.

Chapters 4 and 5 present the two case studies, divided into three sections each: the study configuration, the results and discussion, and the concluding remarks. The first study analyzed the realism of the examples created by the developed method. The second analyzed the generalization of adversarially trained ML models for binary and multi-class classification.

Finally, Chapter 6 provides the main conclusions of this thesis, highlighting the completion rate of each objective. The key benefits and limitations of the proposed solution are described, indicating possible improvements and research topics to be explored in the future.

2 State-of-the-art

This chapter presents the literature review that was performed to thoroughly investigate the formulated research questions. In the following sections, the research methodology adopted for each question is described, and the findings are presented and discussed.

2.1 Adversarial Machine Learning

The concept of an adversarial example was formalized in 2014. Szegedy et al. [8] demonstrated a phenomenon where subtle data perturbations in the input of an ML model caused a misclassification, even though the differences between the original and perturbed data samples were almost imperceptible to humans. This discovery that ML models had previously unsuspected vulnerabilities sparked a wave of research that led to the creation of the adversarial ML area of research within the broader ML field [11].

Over the years, researchers have developed a wide range of adversarial attacks and have worked on various defenses to safeguard ML models from these attacks. Therefore, this section intends to answer RQ1: “What are the state-of-the-art methods used to create adversarial examples for adversarial attacks and defense strategies?”.

2.1.1 Research Methodology

The research performed to investigate RQ1 was based on the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [27], which is a standard reporting guideline that aims to improve the transparency of literature reviews. Search terms were used in reputable bibliographic databases, and several inclusion and exclusion criteria were defined to screen the found publications. Nonetheless, since screening the titles and abstracts of the publications was sufficient to assess their eligibility, their full texts were directly reviewed without systematic exclusion rounds being necessary.

After a careful initial analysis of the literature, several search terms were chosen. The *adversarial* keyword was combined with other suitable terms like *example* and *perturbation* to ensure a comprehensive coverage of relevant publications. The search query also included concepts related to NID, such as *anomaly detection* and *cyber-attack classification*, as well as *wireless* and *IoT* communication networks. Table 1 provides an overview of the utilized terms for each scope. The different scopes were combined in a search query with AND operators.

Table 1. Search terms for RQ1.

Scope	Terms
Adversarial	*adversarial*
Learning	(*learning* OR *example* OR *perturbation* OR *attack* OR *defense*)
Network	(*network* OR *wireless* OR *IoT*)
Intrusion	(*intrusion* OR *anomaly* OR *cyber-attack*)
Detection	(*detection* OR *classification*)

The primary search source was Science Direct [28], which is a large bibliographic database of scientific journals and conference proceedings provided by the internationally recognized publisher Elsevier. Due to their acknowledged relevance for scientific literature of ML, computing, software engineering, and information technology, the search also included the digital libraries of the Association for Computing Machinery (ACM) [29], the Institute of Electrical and Electronics Engineers (IEEE) [30], and the Multidisciplinary Digital Publishing Institute (MDPI) [31]. It is important to note that the PRISMA backward snowballing process of checking the references of the findings also led to additional publications that were not directly obtained from querying these databases.

Since adversarial ML is an active area of research, the search was limited to peer-reviewed publications from 2017 onwards, in the English language. It included surveys and reviews that addressed key developments, as well as more recent works introducing innovative methods. The publications that did not have a full-text available or were duplicated in multiple databases were excluded. Table 2 provides an overview of the defined inclusion and exclusion criteria that were applied to screen the found publications.

Table 2. Inclusion and exclusion criteria for RQ1.

Inclusion Criteria	Exclusion Criteria
IC1: Peer-reviewed journal article or conference paper IC2: Available in the English language IC3: Published from 2017 onwards IC4: Addressed key developments or innovative methods for adversarial machine learning	EC1: Duplicated publication EC2: Full text not available

2.1.2 Findings and Discussion

The information present in the obtained RQ1 findings was consolidated and combined into three subsections: the main adversarial data perturbation crafting processes, the most relevant attack methods, and the most effective defense strategies.

2.1.2.1 Data Perturbations

ML has been increasingly used to make digital systems more intelligent, but it is not flawless. For instance, if an ML model is trained with non-representative data that has missing or biased information, it may become underfit, performing poorly on both its training data and new data, or even overfit, performing very well on its training data but still poorly on previously unseen testing data [32]. These generalization errors can be quickly noticed during the development of an intelligent system, and better results can be achieved by improving data quality and fine-tuning the utilized models [33]. However, even if a model generalizes well to the testing data, it is not guaranteed to always have a stable performance. During the inference phase, when it is deployed to make predictions on live data, it may sometimes behave unexpectedly with seemingly ordinary data samples [34], [35].

In a set of very similar IDT samples of the same class, a model may correctly classify all but one. That specific sample may be assigned to a completely different class with a high confidence score because the model wrongly considers that it is different from the others. Ultimately, this unexpected behavior is caused by unnoticed generalization errors during a model's training phase [18]. Since a training set does not cover all the IDT samples that a model will encounter in its inference phase when deployed in a real system, the model will inevitably learn some simplifications that lead to incorrections in its internal reasoning [36], [37]. These incorrections can be hard to notice because the intricate mechanics of ML models cause the misclassifications to only occur in very specific samples, which are designated as **adversarial examples** [8].

An adversarial example may have very subtle perturbations that are almost imperceptible to humans but make it significantly different from regular samples to an ML model. Such perturbations can occur naturally in faulty data recordings with incorrect readings, but they can also be specifically crafted with specialized inputs to exploit the generalization errors [17], [38]. Even though all ML models are inherently susceptible to adversarial examples, different models will learn distinct simplifications of the target domain and create distinct decision boundaries. Hence, some models may be more vulnerable to perturbations in a certain feature than others, presenting model-specific edge cases that are hard to detect and address [39], [40].

Due to the advances in computer vision technologies and their increasing use in various sectors and industries, the major developments in adversarial ML have been focused on the image classification domain [41], [42]. In adversarial images, the perturbed features are pixels with a value freely assigned from 0 to 255, but it is pertinent to understand how these research efforts can be applied in cybersecurity solutions and if the concepts are transferable to a NID system in a real communication network. In the current scientific literature, the perturbations that turn

a regular sample into an adversarial example can be crafted using two main concepts: an **adversarial patch** that heavily modifies a few features, and an **adversarial mask** that slightly modifies many or all features [43].

Adversarial patches are the most straightforward way to disrupt a cyber-physical system. Since live data from a physical environment is not easily controllable, there is a greater risk for perturbed samples to affect an ML model [44]. For instance, for a model trained to classify street signs, a perturbed sample of a stop sign with small black and white patches can be misclassified as a completely unrelated sign, such as a speed limit sign (Figure 1) [45]. These patches are devised to cause the model to make a mistake when it encounters the sign at a certain angle, although a human would still easily recognize a stop sign [46].

Figure 1. Adversarial perturbation via a patch, based on [45].

Despite being harder to apply adversarial masks in physical environments, they are very well-suited for digital systems. For instance, for a model that performs handwritten digit recognition, a picture of a digit with a subtle change to several pixels can be misclassified as another digit (Figure 2) [47]. Such model can have a wide range of applications, from certified documents and bank check processing to authentication via a picture of an identification document. If a person applies a filter that has a built-in adversarial mask before submitting the requested picture, the automated verification systems that rely on this model can be deceived [48]. Furthermore, there are even some adversarial masks that exploit the intrinsic vulnerabilities of ML and turn every image of well-established datasets into an adversarial image, which denotes that adversarial examples may not be as difficult to create as previously thought [49].

Figure 2. Adversarial perturbation via a mask, based on [47].

Even though most developments in the adversarial ML area of research have addressed image classificstion, the susceptibility of ML models to these examples has also been noticed in other domains with different data types, such as audio, text, tabular data, and time series [15], [39]. For the NID domain, adversarial perturbations must follow a tabular data format, where each feature is a categorical or numerical variable representing a characteristic of a network traffic flow [50], [51]. The tabular format requires more complex perturbations, but they can also be based on the concepts utilized for images. For a tabular classification model, a patch-like perturbation could fully replace the values of categorical variables, which may include the communication protocol or the endpoint port number, and a mask-like perturbation could slightly increase or decrease the values of numerical variables, such as the amount of sent packets or the download-to-upload ratio (Figure 3) [52], [53]. Nonetheless, not all perturbations are suitable for NID because there are specific constraints that must be complied with.

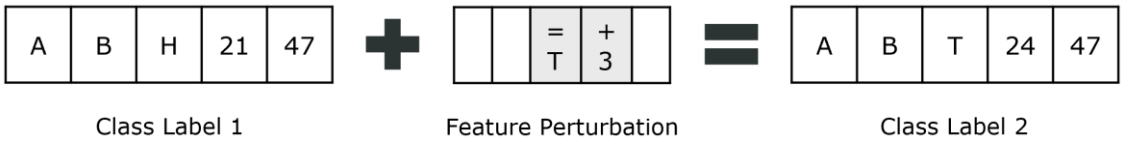

Figure 3. Adversarial perturbation on tabular data, based on [52].

In contrast with the pixels of an image, each tabular feature may have a different range of possible values, according to the characteristic it represents. Furthermore, a feature may also be highly correlated to several other features, being required to exhibit specific values depending on the other characteristics of a sample [52]. Therefore, to ensure that an example represents a real network traffic flow that can be transmitted through a real communication network, the constraints of the utilized protocols and the malicious purpose and functionality of a cyber-attack must be taken into account when generating the perturbations [54], [55]. Despite the current difficulty in creating realistic adversarial cyber-attack examples, the growing popularity of adversarial ML is leading to the development of novel methods to attack various types of algorithms, which is very concerning for the security of ML and of the intelligent cybersecurity solutions that rely on it [10], [16], [50].

2.1.2.2 Attack Methods

The susceptibility of ML to adversarial examples can be exploited for diverse malicious purposes with methods that automatically generate the intended adversarial perturbations. An attacker may use multiple methods to perform a wide range of attacks, which can be divided into two primary categories: **poisoning attacks** during a model's training phase, and **evasion attacks** during the inference phase [56].

Poisoning attacks inject adversarial examples in a model's training data to compromise its internal reasoning. These attacks can perform **model corruption** that make it completely unusable, or even introduce **hidden backdoors** that make it exhibit a biased behavior in specific samples, which is difficult to detect and explain because the model only deviates from its expected behavior when triggered by very specific data perturbations [19], [57]. This is a serious

security risk for organizations that rely on third-party datasets or outsource their intelligent cybersecurity solutions, such as the development of facial recognition models for biometric authentication systems [58], [59]. Nonetheless, since NID systems are commonly developed in secure environments with thoroughly verified network traffic data, an external attacker does not usually have access to a model to compromise it during its training phase [55], [60].

On the other hand, evasion attacks use adversarial examples to deceive a vulnerable model after it has been deployed. The misclassifications caused by these attacks can be directly used to **evade detection** from an intelligent system, or for more complex goals, such as **membership inference** and **attribute inference** to check if a model was trained with a certain sample and certain features, **model inversion** to reconstruct a training set, and **model extraction** to steal its functionality and replicate it in a substitute model [61]–[63]. If confidential or proprietary information is used to train a model, an attacker can cause significant damage to an organization by gathering that information during the inference phase [64], [65]. Even though a model must be queried many times to obtain the information, advances in wireless communications and IoT technologies are making NID systems process larger and larger amounts of network traffic, which substantially increases query opportunities and therefore the feasibility of evasion attacks [66]–[68].

In recent years, numerous methods have been developed to automate the misclassification attempts for evasion attacks. A method may require access to a model in one of three settings: **black-box**, **gray-box**, and **white-box**. The first is model-agnostic and solely queries a model's predictions, whereas the second may also require knowledge of its architecture or the utilized features, and the third needs full access to its internal parameters [44], [69]. Additionally, a black-box or gray-box method may solely use class predictions, a **decision-based** approach, or require a model to output the confidence scores of the predictions, a **score-based** approach [60], [70]. These characteristics affect the choice of an adversarial method because it must be able to attack the targeted model and system, while also being useful to the fulfillment of the end goals of the attacker.

Since the focus of adversarial ML has been image classification, the common attack approach is to freely exploit the internal gradients of an ANN in a white-box setting [15], [56]. Consequently, most state-of-the-art methods do not support other settings nor other models, which severely limits their applicability to other domains. Considering that a deployed NID system is securely isolated, having full access to a model and its feedback is highly unlikely, and an attacker will only know if a certain example evades detection if the entire cyber-attack is successfully completed. This will be a decision-based interaction in black-box or gray-box settings, depending on the available system information about the model and feature set [48], [55]. Furthermore, various other types of ML models can be used for classification tasks with tabular data. For instance, tree-based algorithms and ensembles like Random Forest (RF) are remarkably well-established for NID, but are also susceptible to adversarial attacks [32], [71], [72]. Therefore, an attacker will need to resort to methods that support these models and all the specificities of a communication network.

Various adversarial evasion attack methods have been made open-source software and have started being used to target the ML models of intelligent NID systems. Table 3 summarizes the characteristics of the most relevant methods of the current literature that have been used in NID, noting if they could potentially fulfill the constraints of complex communication networks. Even though some methods were introduced as black-box, they require knowledge of the utilized features to determine how which feature will be perturbed, so they were categorized as gray-box. The Scores keyword corresponds to models that can output confidence scores for a score-based approach. In turn, the Gradients keyword corresponds to models that provide full access to their internal loss gradients, which includes ANNs.

Table 3. Characteristics of relevant adversarial evasion attack methods.

Method	Attack Setting	Supported Models	Could Fulfill Constraints	Reference
BIM	White-box	Gradients	✖	[38]
C&W	White-box	Gradients	✖	[73]
DeepFool	White-box	Gradients	✖	[74]
FGSM	White-box	Gradients	✖	[17]
Hierarchical	White-box	Gradients	✖	[75]
Houdini	White-box	Gradients	✖	[76]
JSMA	White-box	Gradients	✓	[77]
PGD	White-box	Gradients	✖	[78]
Structured	White-box	Gradients	✖	[79]
DoSBoundary	Gray-box	Scores	✓	[53]
GSA-GAN	Gray-box	Scores	✖	[80]
IDS-GAN	Gray-box	Scores	✖	[81]
Polymorphic	Gray-box	Scores	✓	[82]
BFAM	Black-box	Scores	✖	[83]
BMI-FGSM	Black-box	Scores	✖	[84]
OnePixel	Black-box	Scores	✓	[85]
RL-S2V	Black-box	Scores	✖	[86]
WGAN	Black-box	Scores	✖	[87]
ZOO	Black-box	Scores	✖	[88]
Boundary	Black-box	Any	✖	[89]
CGAN	Black-box	Any	✖	[90]
CVAE	Black-box	Any	✖	[91]
GADGET	Black-box	Any	✖	[92]
HopSkipJump	Black-box	Any	✖	[93]
Optimization	Black-box	Any	✖	[94]

Several methods initially developed for the generation of adversarial images have been adapted to generate adversarial examples of network traffic flows. However, most do not account for the constraints of the utilized communication protocols nor the functionalities of the cyber-attacks, so only a few could potentially generate realistic examples [60], [72]. The potentially suitable methods are described below.

The Polymorphic attack [82] was developed for NID, so it attempts to address the preservation of original class characteristics to create examples compatible with a cyber-attack's purpose. A feature selection algorithm is applied in a gray-box setting to obtain the most relevant features for the distinction between benign and cyber-attack classes. Then, the remaining features, which are considered non-relevant for the functionality of a cyber-attack, are perturbed by a version of a Generative Adversarial Network (GAN) [95]: a Wasserstein GAN (WGAN) [87]. By not modifying the most important features of each class, the characteristics required for a successful cyber-attack can be preserved. Nonetheless, the perturbations generated by WGAN in the remaining features disregard the constraints on the structure of a network traffic flow, which mostly leads to OOD examples with incompatible values for a real scenario.

The distinction between benign activity and cyber-attack classes was further explored in the DoSBoundary attack [53], which iteratively optimizes the perturbations that are performed on each feature of a Denial-of-Service (DoS) flow according to the specified constraints of a communication network. However, it requires expert knowledge to manually configure the specific values of each feature and all the possible perturbations, which corresponds to a gray-box setting where a security practitioner must thoroughly analyze the characteristics of each class. Even though the method could potentially preserve the correlations between the features of a network flow and generate IDT examples, the required expert analysis does not scale well to scenarios with multiple cyber-attack classes in complex communication networks.

Despite being originally developed to attack image classification models, the perturbation crafting processes of the Jacobian-based Saliency Map Attack (JSMA) [77] and the OnePixel attack [85] could potentially preserve the structure of a network flow. The former attempts to minimize the number of modified pixels in an adversarial image, requiring full access to the internal gradients of an ANN in a white-box setting, whereas the latter only modifies a single pixel, based on the confidence scores of a model in a black-box setting. These methods could be suitable for NID because they only perturb the most appropriate features without affecting the remaining ones, although their lack of constraints could lead to incompatible values, resulting in a mix of OOD examples and IDT examples created by chance.

Due to the different characteristics of existing methods and diverse goals of attackers, efforts are being made to systematize the possible adversarial attack vectors in the Adversarial Threat Landscape for Artificial-Intelligence Systems [96] knowledge base, and to complement it with case studies and demonstrations based on real-world observations. As novel adversarial methods continue to be developed, it is becoming essential to raise awareness of the diverse strategies that attackers can use to exploit the vulnerabilities of ML models and the security risks they pose to modern organizations.

2.1.2.3 Defense Strategies

The growing ML attack surface led to a never-ending arms race where attackers continuously exploit newly discovered vulnerabilities and defenders develop countermeasures against each novel threat. However, the defenders are always a step behind because it can take a long time until the effects of an attack are detected, and then it is difficult to retrace it and develop a countermeasure for it [9]. To get ahead of attackers, organizations should follow a security-by-design development approach and proactively search for vulnerabilities themselves. By simulating adversarial attacks in realistic scenarios and analyzing entire attack vectors, ML engineers and security practitioners can anticipate possible threats and use that knowledge to preemptively revise and improve their defense strategy (Figure 4) [97].

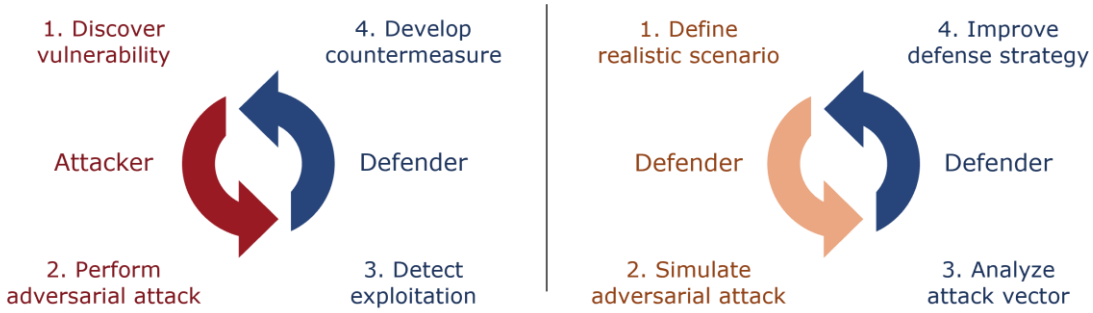

Figure 4. Adversarial arms race (left) and security-by-design (right), based on [97].

A defense strategy can combine multiple techniques to address different security concerns. Due to their proven value against several adversarial attacks, some defenses have been standardized across the literature, divided into two primary categories: **proactive defenses** during the training phase, and **reactive defenses** during the inference phase [15].

Regarding reactive defenses, they attempt to mitigate the effects of corrupted data on a model's predictions by safely processing its input and output. These defenses can rely on several preprocessing techniques, such as **data denoising** and **feature squeezing** to reduce the search space for an attack, and postprocessing techniques, such as mechanisms that deal with **model uncertainty** and require predictions with **high confidence** scores [63], [69], [98]. Even though reactive defenses can be valuable against both erroneous data and adversarial attacks purposely exploiting a model, they represent an additional software layer that attempts to encapsulate a vulnerable model. This layer is always needed for a NID system to convert the recorded network activity into the utilized feature set and then convert the predictions of a model into relevant alerts, but it does not fully protect that model [35], [55].

On the other hand, proactive defenses tackle the susceptibility of ML to adversarial examples, aiming to reduce the vulnerabilities and intrinsically improve a model's robustness against adversarial examples during its training phase. These defenses include several techniques, such as **adversarial training** with perturbed samples in a training set, **regularization** to better calibrate the learning process, and **defensive distillation** to create smaller models less sensitive to data variations [16], [42], [99]. It is not yet clear how to completely resolve this susceptibility

and achieve an adversarially robust generalization in a classification task, but progress is being made in robustness research with regularization and optimization techniques [100]–[102]. These scientific advances are starting to give ML engineers and security practitioners better tools to address ML security during the entire lifecycle of an intelligent system, including its development, testing, deployment, and maintenance phases.

Most proactive defenses are focused on improving the robustness of deep learning algorithms based on ANNs against evasion attacks in the image classification domain [103]–[106], although some also take measures against backdoors [19], [58]. Despite ANN defenses being difficult to apply to other models and domains, the protection of tree-based algorithms has been drawing attention for cybersecurity systems [107], [108]. Some defenses have been developed to improve the robustness of entire tree ensembles at once [109], [110], whereas others address each individual decision tree at a time [111], [112]. Still, proactive defenses often trade-off some performance on regular IDT samples to improve performance on OOD adversarial examples. This trade-off affects the choice of a defense strategy because there is a need to balance the robustness against occasional adversarial examples and the generalization to regular data that a model will commonly encounter.

Defense strategies continue to be enhanced with better techniques, but the most effective and widespread defense is still adversarial training because it anticipates the data variations that an ML model may encounter [113]–[115]. Performing data augmentation with examples created by an adversarial attack method enables a model to learn additional characteristics that the samples of each class can exhibit, so it becomes harder to deceive it. This augmented training data with more data variations can improve a model's robustness not only against attack methods similar to the utilized one, but also against a wide range of different attacks with distinct data perturbations [101], [116], [117]. Nonetheless, to improve a model's robustness to adversarial data without deteriorating its generalization to regular network traffic, it is essential to ensure that adversarial training is performed with realistic examples that are IDT within the NID domain and preserve the malicious purpose of a cyber-attack [18], [48].

2.2 Constrained Data Generation

High-quality data is essential for all organizations that integrate AI solutions in their critical business processes [118], but it is especially needed in adversarial ML approaches to ensure their reliability. Due to the difficulty of obtaining real data to train, validate, and test an intelligent solution, synthetic data may be generated. Likewise, due to the difficulty of obtaining real faulty input data to evaluate the robustness of an ML model, adversarial examples may be generated instead. In the case of the NID domain, all data samples must represent real network traffic flows and actual cyber-attacks, so the generated data must comply with the constraints of a communication network, the utilized communication protocols, and the functionalities and malicious purposes of the cyber-attack classes [55].

Due to the lack of constrained data generation approaches observed in the findings of the previous research question, this research topic was further investigated in a systematic manner. This section intends to answer RQ2: "What are the current approaches to generate data for adversarial ML according to specific constraints?".

2.2.1 Research Methodology

To achieve a transparent, replicable, and complete systematic review, the full PRISMA reporting guideline was followed for RQ2. Therefore, the titles and abstracts of the found publications were assessed in a screening phase, then their full texts were carefully analyzed in an eligibility assessment phase, and only then the remaining publications were included in the review.

Since constrained data generation is a less studied research topic without standardized keywords, the search had to cover broader terms to prevent narrowing it down too much and obtain relevant publications that would otherwise be disregarded. In addition to including the *adversarial* keyword to obtain approaches related to adversarial data perturbations, some word variations were also considered. This widened the search to more common concepts like *number generators* and *conditional generators*, as well as approaches that address *constraints* and *restrictions*. Table 4 provides an overview of the utilized terms for each scope, which were combined in a search query with AND operators.

Table 4. Search terms for RQ2.

Scope	Terms
Constrained	(*constrained* OR *conditional* OR *restricted* OR *constraint* OR *condition* OR *restriction*)
Adversarial	*adversarial*
Data	(*data* OR *sample* OR *example* OR *number*)
Generation	(*generation* OR *generator*)

As in the previous research question, the search was conducted using the Science Direct bibliographic database as the primary source, together with the digital libraries of the ACM, IEEE, and MDPI sources. The review intended to analyze the yearly advances of peer-reviewed publications introducing or applying constrained generation approaches. Considering that there have been relevant technological advances and scientific developments in recent years, the selected time frame was 2017 to 2022, the latest 6 whole years.

The widened search query led to more relevant publications within the intended time frame, but also to many publications that were not aligned with the purpose of the review. To screen them, several exclusion criteria were defined to remove general surveys and reviews that did not specify concrete approaches, publications that mentioned constraints but not during their data generation processes, and publications that did not detail how the utilized constraints were applied nor how they affected the data generation. Additionally, to limit the findings to

publications stored in the selected databases and time frame, snowballing was not performed for RQ2. Table 5 provides an overview of the defined criteria.

Table 5. Inclusion and exclusion criteria for RQ2.

Inclusion Criteria	Exclusion Criteria
IC1: Peer-reviewed journal article or conference paper	EC1: Duplicated publication
IC2: Available in the English language	EC2: Survey or review
IC3: Published from 2017 to 2022	EC3: Constraints not in generation
IC4: Introduced or applied an approach for constrained data generation	EC4: Constraints not detailed
	EC5: Full text not available

2.2.2 Findings and Discussion

A total of 1756 records were initially obtained for RQ2 by applying the query to the contents of the publications stored in the selected databases. After removing duplicates and performing the screening phase, the full texts of 247 records were carefully assessed to check if they contained the required details. Finally, 93 publications were included in the review, to be consolidated and systematized (Figure 5).

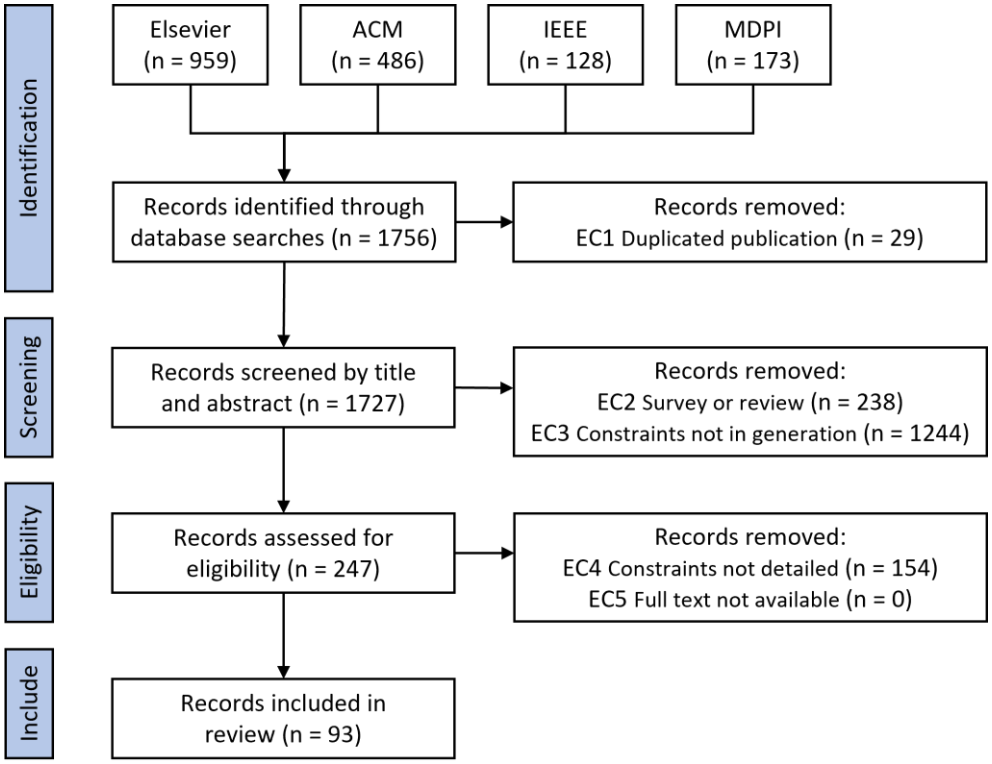

Figure 5. PRISMA search process for RQ2.

The approaches detailed in the included records were systematized to identify the current applications of constrained data generation and the methods they are based on. Since some records addressed multiple tasks, a total of 97 applications were identified across 11 sectors and industries, although a few were more general and not specific to a single industry.

In recent years, constrained data generation has started drawing more attention. The number of applications grew from 2 in 2017 to 33 in 2022, with the cybersecurity and healthcare sectors being the most prominent and having more diverse applications. There were also several approaches developed for the aerospace and energy sectors, possibly due to their growing importance for modern organizations. Furthermore, mechanical applications to improve the resilience of industrial machinery have spiked in 2022, which may be a result of the additional research efforts of the two previous years to tackle the disruptions in manufacturing and supply chains caused by the COVID-19 pandemic (Figure 6).

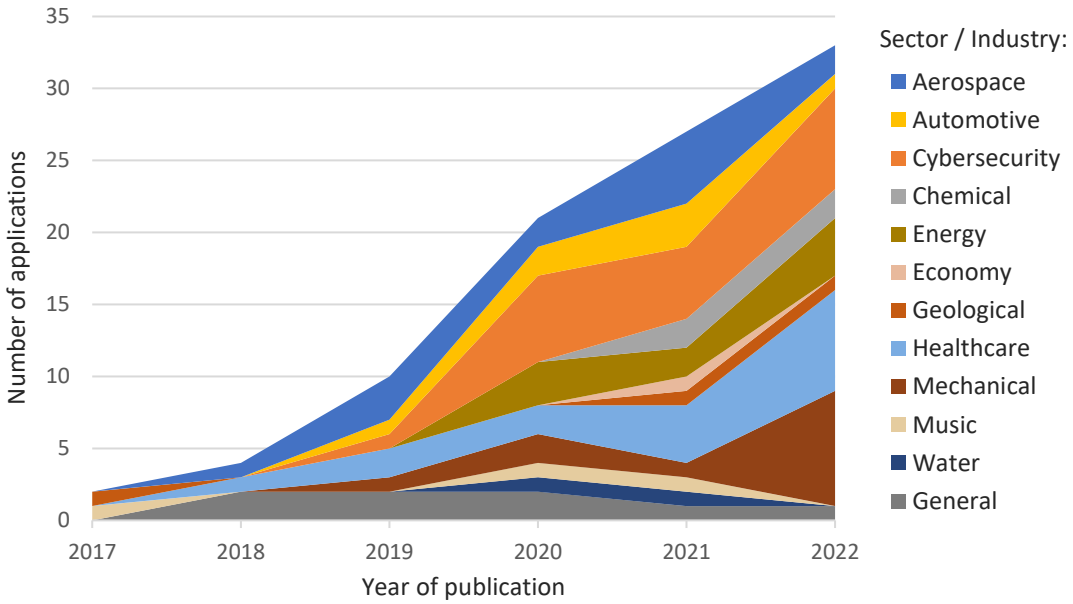

Figure 6. Industry applications of constrained data generation per year.

The increasing use of constraints across various sectors and industries highlights that better data can be generated when the specificities of the task at hand are considered. Nonetheless, since most publications handle image classification datasets, they employ methods with relatively simple constraints to generate synthetic images according to the classes of a dataset. Most are based on conditional versions of GAN [95] and Variational Autoencoder (VAE) [119]: a Conditional GAN (CGAN) [90] and a Conditional VAE (CVAE) [91]. Even though these methods spread across the literature because they learn to generate different data for different classes, the samples of each class are freely generated without addressing any complex constraint.

For more complex tasks in sectors that require other data types like tabular data and time series, some approaches are starting to rely on methods that support more rigorous configurations and more complex constraints. For instance, some researchers use evolutionary computation with swarm intelligence and genetic algorithms [120], and others apply fuzzy logic to deal with

the uncertainty of the information [121]. Table 6 summarizes the reviewed constrained data generation applications and indicates the methods they were based on. The base methods were either explicitly built upon or their concepts were implicitly used in new implementations.

Table 6. Applications of constrained data generation methods.

Sector / Industry	Found Applications	Base Methods
Aerospace	Synthetic-Aperture Radar (SAR) image reconstruction, and hyperspectral classification [122]–[125]	CGAN, CVAE
	Satellite mapping of surface water, road surface area, reflectance, and radiance [126]–[129]	CGAN
	Precipitation estimation, aircraft detection, and remote sensing image segmentation [130]–[132]	CGAN
	Object detection in drone inspection with Unmanned Aerial Vehicles (UAVs) [133], [134]	CGAN
Automotive	Urban traffic trajectory and parking occupancy estimation [135], [136]	CGAN, CVAE
	Inertial Measurement Unit (IMU) and vehicle transmission gear reliability analysis [137], [138]	CGAN
	Ultrasonic signal sensors for autonomous driving systems [139]	CGAN
	Vehicular and transportation network simulation [140], [141]	CGAN
Chemical	Chemical production process fault diagnosis [142], [143]	CGAN, WGAN
	Soft sensors for chemical compound analysis [144]	CGAN, WGAN
	Extraction of environmental features from aerosol optical depth sensors [145]	CGAN
Cybersecurity	Network intrusion and anomaly detection, and cyber-attack classification [120], [146]–[148]	CGAN, evolutionary algorithms
	Illegal webpage and malicious domain name detection [146], [149], [150]	CGAN, evolutionary algorithms
	Software testing and malware detection in code blocks [121], [151], [152]	CGAN, fuzzy algorithms
	Fake user detection and user behavior generation for recommendation systems [153], [154]	CGAN
	Speech recognition and denoising, and voice spoofing detection [155]–[158]	CGAN
	Data masking, obfuscation, anonymization, and compression [159]–[161]	CGAN
Economy	Monte Carlo simulation design for economic and financial studies [162]	CGAN, WGAN, econometric algorithms

Sector / Industry	Found Applications	Base Methods
Energy	Power distribution and economic load dispatch simulation for electrical grids [141], [163], [164]	CGAN, graph algorithms
	Anomalous energy consumption and electricity theft detection [165], [166]	CVAE, WGAN
	Charging behavior and electric load forecasting [167]–[170]	CGAN
Geological	Seismic data interpolation [171]	CGAN
	Virtual landscape and terrain authoring [172], [173]	CGAN
Healthcare	Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) analysis [174]–[178]	CGAN, WGAN
	Electrocardiogram (ECG) and Electroencephalogram (EEG) signal analysis [179]–[181]	CGAN, WGAN
	Electrodermal Activity (EDA) generation for stress detection [182]	CGAN
	Mammographic density segmentation and breast cancer diagnosis [183]–[185]	CGAN
	Lung and skin lesions detection [186]	CGAN
	Microscopic blood smear test [187]	CGAN
	Vocal cord and voice disorder detection [188]	CGAN
	Connectomes segmentation for reconstruction of neural circuits [189]	CGAN
Mechanical	Rotating machinery and rolling bearing fault diagnosis [190]–[195]	CGAN, CVAE, WGAN
	Industrial equipment failure and conveyor belt damage detection [196]–[198]	CGAN, CVAE, WGAN
	Wind turbine gearbox fault diagnosis [199]–[201]	CGAN, CVAE
Music	Variable-length music and audio generation [202], [203]	CGAN, CVAE
	Cross-modal musical performance generation [204]	CGAN
Water	Water flow simulation for water supply and distribution systems [141], [164]	CGAN, graph algorithms
General	Object stretches, compressions, and motion simulation [205], [206]	CGAN, WGAN, skinning algorithms
	Picture generation according to age and facial attributes [207]–[209]	CGAN
	Wireless indoor positioning and tracking [210]	CGAN
	Handwritten text and digit recognition [211], [212]	CGAN

Despite the wide range of applications of constrained data generation, the few approaches that could potentially be transferable to the NID domain were mainly developed for cyber-physical applications in energy and water networks. Due to the low-quality and OOD data that

conventional methods would generate in these domains, some researchers encapsulated the data generation processes in mechanisms that enforce task-specific constraints and attempt to be as realistic as possible. The potentially transferable approaches are summarized below.

In [164], the authors intended to simulate electrical grids and water supply and distribution systems. Each action or event in one part of these cyber-physical systems would affect the entire energy and water networks, so the presence of a given value at a given feature would restrict the values that other features could have. Linear equality constraints were defined to fulfill the physical capacity requirements of each network and multiple simulations were performed. This optimization approach could be useful to model the benign network traffic flows and the expected behavior of devices in a communication network, but the constraints would need to be carefully redesigned for each minor change in the infrastructure.

To provide a structure that could be adapted to changes, in [141], energy and water networks were addressed through graph theory. The graph structure provided a more rigorous representation of the electric and water flow physics between different nodes in a network, enabling the simulation of cyclic trends and acyclic congestions, as well as their effects in other nodes. Nodes and their connections could be added or removed to adapt the structure to different flow networks, although it was tailored to the specific physical constraints of those domains. Additionally, it is also relevant to highlight the energy function approximation introduced in [205] to perform physically accurate simulations of the phases of object stretches and compressions. Despite not being as rigorous as flow graphs, this function could be redesigned to represent the phases of a network connection, enforcing the correct order of network packets for the setup, data transfer, and closing of a connection.

Regarding biomedical images, in [186], the authors addressed the insufficient label granularity of most classification datasets. Since multiple sub-types of a disease can be aggregated into a single broader label, the samples of a class can exhibit entirely different characteristics. Even if a CGAN is used to distinguish between different classes, the generated samples may mix the characteristics of multiple sub-types and therefore may not correspond to an actual disease. Conditioning vectors were used to restrict the modification of relevant disease features according to the values exhibited by other similar samples that were presumed to be of the same sub-type. Despite this mechanism still being based on a CGAN with relatively simple constraints, the underlying concept of configuring different constraints for each class could also benefit tabular data generation methods.

Even though the reviewed approaches present valuable insights for the generation of network traffic data, they are tailored to the intricate requirements of their very specific domains. To support the significantly different constraints of a standard communication network, they would need to be completely redesigned with cybersecurity expert knowledge. Nonetheless, it could be beneficial to develop a way of enforcing specific constraints for each cyber-attack class and ensure that the generated network traffic is still an actual cyber-attack. Improving tabular data quality is essential to enhance the robustness and resilience of intelligent cybersecurity solutions, so further research efforts are required to better address NID constraints.

2.3 Network Intrusion Detection

Intrusion detection has played a crucial role since the early days of computer systems. By analyzing the files and audit data of local machines, a security application could detect evidence of an intrusion and mitigation measures could be applied. As communication networks became larger and more complex, NID systems were developed to analyze the network traffic patterns of multiple machines and perform real-time detection of ongoing cyber-attacks. Even though these systems commonly perform a signature-based detection with a set of rules recognizing known threats, novel approaches are starting to employ ML for anomaly-based detection, identifying changes in behavior that indicate the occurrence of an intrusion.

To securely use ML models for NID, their training sets must have reliable and up-to-date data about the target domain. Otherwise, if they were trained with non-representative data that had biased information or missing values, their predictions would be untrustworthy and the NID system might not be able to detect intrusions when deployed in the intended network. Therefore, this section intends to answer RQ3: "What are the most reliable NID datasets for the creation of realistic adversarial examples of network traffic flows?".

2.3.1 Research Methodology

The adopted research methodology for RQ3 was also based on the PRISMA methodology, but it included a straightforward process to screen dataset descriptions and abstracts, and then assess the eligibility of the utilized features and classes.

To ensure a comprehensive coverage of relevant datasets, the search query was created with the same terms related to NID that were used in RQ1. Table 7 provides an overview of the utilized terms for each scope, which were combined with AND operators.

Table 7. Search terms for RQ3.

Scope	Terms
Network	(*network* OR *wireless* OR *IoT*)
Intrusion	(*intrusion* OR *anomaly* OR *cyber-attack*)
Detection	(*detection* OR *classification*)

Unlike in the previous research questions, the search was conducted using IEEE DataPort [213], Zenodo [214], and Kaggle [215] as sources because they are well-established platforms for the storage and sharing of datasets across the scientific community and industry practitioners. Public datasets of relatively recent network activity were included, but those without any class labels to distinguish between benign traffic and cyber-attacks were disregarded. Table 8 provides an overview of the defined inclusion and exclusion criteria.

Table 8. Inclusion and exclusion criteria for RQ3.

Inclusion Criteria	Exclusion Criteria
IC1: Publicly available dataset IC2: Created from 2017 onwards IC3: Contained communication network activity with both benign traffic and cyber-attacks	EC1: Duplicated dataset EC2: Class labels not available

2.3.2 Findings and Discussion

The obtained RQ3 findings were several publicly available datasets containing labeled recordings of network flows, which represented either benign traffic that was part of the regular network activity or malicious traffic that was part of a cyber-attack. The found datasets were compared considering the common cyber-attack groups.

Cyber-attacks can be grouped considering the confidentiality, integrity, and availability triad, which form the basic principles for network security [216]. Different groups present different functionalities to cause disruptions in distinct security principles, according to the end goals of an attacker. Reconnaissance (RCN) attacks commonly consist of eavesdropping and probing of the machines in a communication network, affecting the confidentiality of the business processes of an organization. After an initial reconnaissance stage, attackers usually attempt to perform Access and Misuse (A&M) or Denial-of-Service (DoS) attacks. The former attempt to gain access to a vulnerable device and use it for malicious purposes, affecting the integrity of the data, whereas the latter attempt to make a machine or a server inaccessible to its intended users, disrupting the availability of a system [217], [218]. Table 9 provides an overview of the main cyber-attack groups, including their common subgroups.

Table 9. Cyber-attack groups and disrupted security principles.

Group	Designation	Subgroup	Disrupted Principle
RCN	Reconnaissance	Eavesdropping Probing	Confidentiality
A&M	Access and Misuse	Brute-Force Injection Spoofing	Integrity
DoS	Denial-of-Service	Amplification Flooding	Availability

The cyber-attacks within each group can be directly performed but can also be used in more complex attacks with multiple steps. A pertinent example of a complex attack is a botnet created from self-replicating malware. Due to the increasing use of IoT devices with unresolved vulnerabilities, botnets can cause significant disruptions in the digital systems of a modern organization [219]. After a malware is injected into a device, it stealthily replicates to other

vulnerable devices, and then starts to control them to perform unintended actions as part of A&M or DoS attacks. The latter are especially dangerous because when a botnet controls many devices, they can be commanded to perform a Distributed DoS (DDoS) [220] and quickly bring down a communication network, making it completely inaccessible.

Each of the found datasets was analyzed to identify the cyber-attack groups they included, considering the labels and descriptions of their network traffic flows. Table 10 summarizes the characteristics of the most relevant datasets, ordered by the publication year, and provides a concise highlight of their differentiating characteristics.

Table 10. Main characteristics of relevant NID datasets.

Dataset	RCN	A&M	DoS	Highlight	Year	Reference
CIC-IDS2017	✓	✓	✓	Enterprise computer networks	2017	[221]
N-BaIoT	✓	✖	✓	Mirai and BASHLITE malwares	2018	[222]
Bot-IoT	✓	✓	✓	Information theft in IoT networks	2019	[223]
MQTT-IoT-IDS2020	✓	✓	✖	MQTT communication protocol	2020	[224]
IoT-23	✓	✓	✓	Mirai, Torii, Gafgyt, Kenjiru, Okiru, and other malwares	2020	[225]
HIKARI-2021	✓	✓	✖	HTTPS encryption protocol	2021	[226]
5G-NIDD	✓	✖	✓	5G wireless networks	2022	[227]

There is a scarcity of publicly available datasets, and the existing ones are mostly imbalanced towards benign activity because of the low rate of recorded cyber-attacks. Nonetheless, some datasets are well-established across the scientific community and are commonly used to benchmark ML models for NID. The most complete datasets are described below.

CIC-IDS2017 [221] was made public by the Canadian Institute for Cybersecurity and consists of 7 labeled captures of common cyber-attacks performed on a standard enterprise computer network. It includes benign activity and several classes of RCN, A&M, and DoS attacks, which were recorded in July 2017 in an heterogenous testbed environment with 12 interacting machines (Figure 7). The complete attack interactions were recorded and converted to a tabular data format with more than 80 features using the CICFlowMeter [228] analysis tool. Despite some flaws having been noticed on a portion of the resulting network flows [229], CIC-IDS2017 continues to be used as a standard benchmark dataset to compare the performance of novel ML models with baseline models from previous studies.

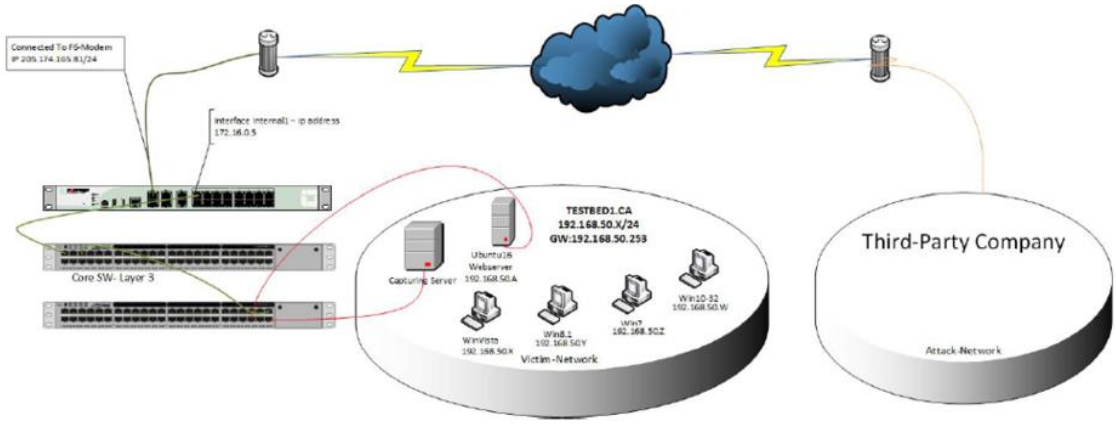

Figure 7. Enterprise network testbed environment [221].

In contrast, IoT-23 [225] and Bot-IoT [223] address the wireless communications between the interconnected devices of the emerging IoT networks. These are public datasets containing labeled captures of benign activity and malicious traffic caused by malware. The former was created by the Stratosphere Research Laboratory and contains 23 captures of the network activity of a real IoT network, using IoT services and devices with unrestrained access to an internet connection. The latter resulted from a realistic IoT testbed environment developed at the University of New South Wales, with simulated IoT devices being controlled by botnets (Figure 8). Both datasets are extremely valuable because the recorded data manifests real IoT network traffic patterns and includes various classes of common malware attacks.

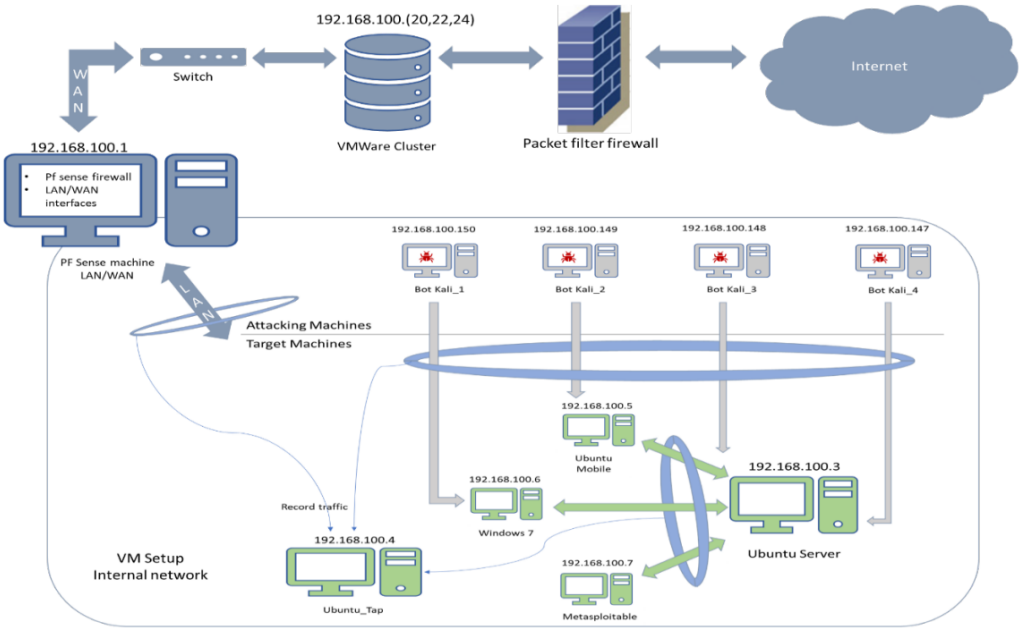

Figure 8. IoT network testbed environment [223].

Despite the valuable information of existing datasets, devices are starting to generate larger and larger amounts of network traffic. An attacker may exploit different device vulnerabilities with novel attack vectors, rendering the ML models that were created with old datasets

ineffective. This is a major obstacle to the adoption of ML for NID systems, but a promising strategy to overcome it is benefiting from adversarial examples to perform training data augmentation. By providing more cyber-attack variations to a model during its training phase, its detection performance may be substantially increased.

2.4 Chapter Remarks

In this chapter, the formulated research questions were answered by investigating the scientific literature in a methodical manner. Considering that adversarial ML is an active area of research, RQ1 was addressed by exploring the findings of the surveys and reviews already present in the current literature, as well as more recent publications introducing innovative methods. Since constrained data generation is a less studied research topic, a systematic review was performed for RQ2 to thoroughly investigate the recent developments. Regarding RQ3, well-established platforms were explored to find and select publicly available NID datasets with labeled network traffic flows that could be used for the intended case studies.

The key developments in adversarial ML were divided into several types of attack methods and defense strategies. A wide range of attack methods have been created, but most were not intended for the specific constraints of a communication network and of the utilized protocols, so they may lead to OOD data perturbations and unrealistic examples in the NID domain. Even though some methods could possibly be used to create IDT samples in tabular data, they either require an entirely manual configuration based on expert knowledge or are only able to generate IDT samples by chance because they do not account for any constraint. Research efforts continue to be made to better protect various types of algorithms with reactive and proactive defenses, and a security-by-design approach throughout the entire lifecycle of an intelligent system is becoming essential to tackle the growing ML attack surface.

Despite the increasing use of constrained data generation approaches in various sectors and industries, most applications only differentiate between classes without enforcing any specific constraint, which can still lead to OOD samples. A few recent publications started to notice this obstacle and attempt to address it, but their approaches are tailored to very specific tasks and cannot be adapted to generate data for other complex domains with different constraints. Overall, the approaches present in the current literature were not designed to support the constraints of a complex tabular data domain like NID, which hinders the development of realistic adversarial attack and defense strategies for the creation of robust ML models. This is the gap in the current literature addressed in this thesis.

3 Proposed Solution

This chapter describes the types of constraints required for an adversarial cyber-attack example to be realistic, defines a methodology for a trustworthy adversarial robustness analysis with an adversarial evasion attack vector, and details the design and implementation of the developed method and its adaptative patterns.

3.1 Robustness Analysis

The NID domain is a complex tabular data domain where adversarial cyber-attack examples must resemble real network traffic flows that could evade detection in a real communication network. These flows should be able to deceive an ML model into classifying them as benign activity while continuing to be a part of a cyber-attack. Otherwise, the examples would not be useful for an attacker and could even be counterproductive for a defense strategy like adversarial training because a model would learn incorrect representations that would compromise its internal reasoning.

For an adversarial robustness analysis to be trustworthy and provide relevant results, multiple ML models should be evaluated and compared in a methodical manner. It must use realistic adversarial examples in a realistic adversarial attack vector that could be applicable to a real scenario in a real communication network. Therefore, it is pertinent to establish a methodology suitable for a robustness analysis in complex domains like the NID domain.

3.1.1 Data Constraints

To generate realistic adversarial examples for a robustness analysis in the NID domain, it is necessary to carefully examine the complex constraints of different communication networks, of the utilized communication protocols, and of the cyber-attacks they face.

The features of a network traffic flow may be required to follow specific data distributions, according to the specificities of a communication network and of the utilized protocols. Furthermore, due to their distinct malicious purposes, different cyber-attacks may exhibit entirely different feature correlations. Since the tabular features of a flow may be correlated, the presence of a given value at a given feature may restrict the values that other features can have. Conceptually, this leads to two main types of constraints:

- **Intra-feature constraints**: Restrict the value of a single feature.
- **Inter-feature constraints**: Restrict the values of one or more features depending on the values present in one of more other features.

Both types of constraints can be observed in the packet Inter-Arrival Time (IAT), which is one of the most pertinent characteristics of network traffic. IAT represents the elapsed time between the arrival of two subsequent packets of a flow and can be represented as two tabular features: the Minimum IAT (MinIAT) and the Maximum IAT (MaxIAT). These features are very valuable for the detection of various DoS attack classes. A low MinIAT can indicate a short DoS attack that quickly overloads a server with many requests, whereas a high MaxIAT can indicate a lengthy DoS attack that overwhelms a server by maintaining several long connections open.

These two features, MinIAT and MaxIAT, present a straightforward infra-feature constraint: both must only have positive values because there is always some elapsed time between two packets. Nonetheless, they also have other more complex intra and inter-feature constraints that vary according to the specificities of a communication network and to the intended functionality of a cyber-attack class.

Some more complex constraints can be observed in a Slowloris attack, which is a lengthy DoS attack that attempts to overwhelm a web server by opening multiple connections and maintaining them as long as possible. A network flow utilized in this cyber-attack must use the Transmission Control Protocol (TCP) and the Push (PSH) flag to keep the connection open on the port number 80, its endpoint [230]. The flow may have a varying IAT between 20 and 30 seconds to appear as arbitrary traffic instead of scheduled packets just to keep the connection open. In this scenario, a longer IAT cannot be utilized because this particular web server is configured with a timeout to close connections after 30 seconds of inactivity, which is a common web application security measure [231].

Due to the lack of constraints of the perturbation crafting processes of the image classification domain, it is very difficult to transfer them to the NID domain. A patch-like crafting process could be performed, changing the flow from a TCP connection to another protocol, or from port number 80 to another port, but these modifications would not be useful for a lengthy DoS. The communication protocol, the connection flag, and the port must remain the same, otherwise the crafted example will no longer be a Slowloris flow. Likewise, a mask-like crafting process may change the values of MinIAT and MaxIAT, increasing or decreasing them, but not all perturbations will be suitable for a real communication network.

From the original values of MinIAT and MaxIAT, 20 and 30 seconds, three different examples may be generated for the considered flow: the first with a MinIAT of 22 and a MaxIAT of 28, the second with 18 and 32, and the third with 26 and 24. Even though all three may deceive an ML model and be misclassified as benign, only the first is a harmful Slowloris flow. The second example is harmless because the considered web server will terminate the connection at the 31st second of inactivity before a packet is received at the 32nd second, preventing the functionality of this cyber-attack. In turn, the third example would not even be possible in a real communication network because a flow with packets at least every 26 seconds cannot also have packets at most every 24 seconds (Figure 9).

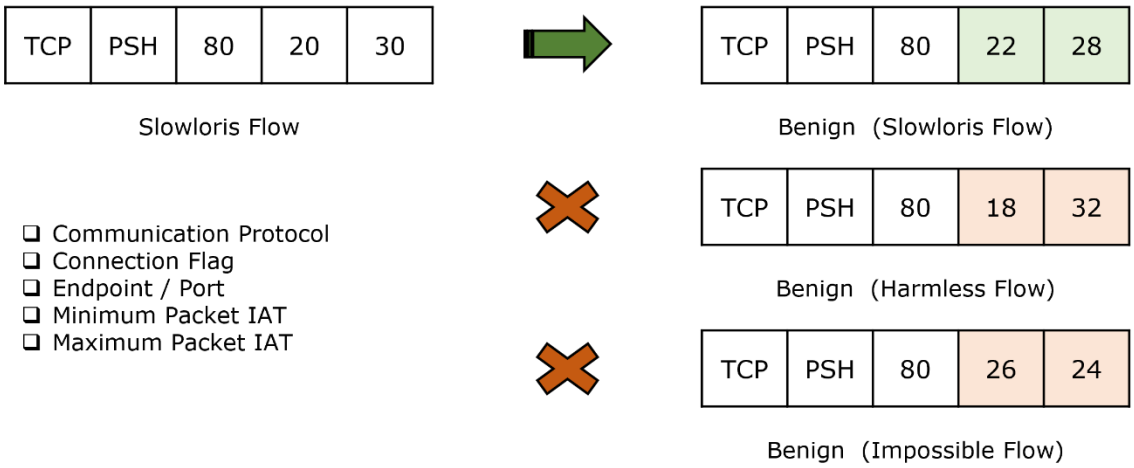

Figure 9. Adversarial perturbation on a Slowloris network traffic flow.

Despite all examples following similar mask-like perturbations of increasing and decreasing some numerical variables, they would lead to very different outcomes in a communication network and only one example could be used against a real NID system. Therefore, a successful adversarial attack is not guaranteed to be a successful cyber-attack.

By inspecting the third example, it can be observed that the reason it is impossible is because it does not comply with the inherent data structure of a network traffic flow. On the other hand, the reason that the second example is harmless is because it does not comply with the intended functionality of the Slowloris class. Hence, it is possible to define two fundamental properties that are required for an adversarial example to resemble a real data sample:

- **Validity**: Compliance with the constraints of a domain, following its data structure.
- **Coherence**: Compliance with the constraints of a specific class, following the characteristics that distinguish it from other classes.

Adversarial realism becomes a unifying concept that combines these two properties. A generated adversarial example can only be fully realistic if it is simultaneously valid within the inherent data structure of its domain and coherent with the characteristics of its class, by fulfilling all domain and class-specific constraints. An exemplification of these properties and of

some of the required constraints is provided below, considering MinIAT, MaxIAT, and another DoS attack class: a short DoS attack.

During a perturbation crafting process, if MinIAT was increased to a value higher than MaxIAT, a network flow could become an adversarial example that a model would misclassify as benign. However, that would be an invalid flow that a model would never encounter in a real deployment scenario because it could not be transmitted through a communication network. Therefore, to preserve validity within the network traffic data structure, an inter-feature domain constraint must be enforced: the perturbed MinIAT must not be higher than the perturbed MaxIAT. These types of constraints, including value ranges and multiple category membership, have started being investigated in [52], [53]. Enforcing constraints improve the feasibility of adversarial attacks for NID, sometimes even mistakenly calling validity as realism.

Nonetheless, validity is not enough for an adversarial attack to be a successful cyber-attack. Even if the previous domain constraint was fulfilled, valid flows with moderately increased MinIATs could be misclassified as benign but not be quick enough to overload a real web server. Consequently, those supposed adversarial examples would not actually be short DoS flows. Instead, they would represent just regular traffic that would not be useful for a cyber-attack, so an ML model would be correct to label them as benign. Therefore, to preserve coherence, it is necessary to also enforce an intra-feature class-specific constraint: the perturbed MinIAT must not be higher than the highest original value of MinIAT for the short DoS class. This is a significantly more complex constraint, but a crafting process must take it into account to ensure that an adversarial attack is feasible against a real NID system.

Even though validity has previously been investigated, it is imperative to address it together with coherence to fully achieve adversarial realism. Hence, an adversarial cyber-attack example must represent valid network traffic capable of being transmitted through a real communication network, as well as a coherent cyber-attack flow capable of fulfilling its intended functionality and the malicious purposes of an attacker.

3.1.2 Analysis Methodology

In addition to using realistic adversarial examples, a robustness analysis must also provide them to an ML model in an adversarial attack vector that could be used in a real scenario against a real NID system. To establish an attack vector, it is important to consider how ML models are created and how NID systems are developed.

A standard practice when creating ML models for NID and cyber-attack classification tasks is to combine the holdout method with cross-validation. A NID dataset can be randomly split into training and holdout sets with 70% and 30% of the samples, with stratification to ensure that the original class proportions are preserved. Then, a 5-fold cross-validation can be performed, creating five stratified subsets, each with 20% of the training set, which corresponds to 14% of the total dataset. In this validation process, five distinct iterations are performed, each training a model with four subsets and validating it with the remaining one. This enables a

hyperparameter optimization to be performed by training and validating a model with 70% of the data, and then an independent performance evaluation with the remaining 30% that the model has not yet seen, to assess its generalization (Figure 10).

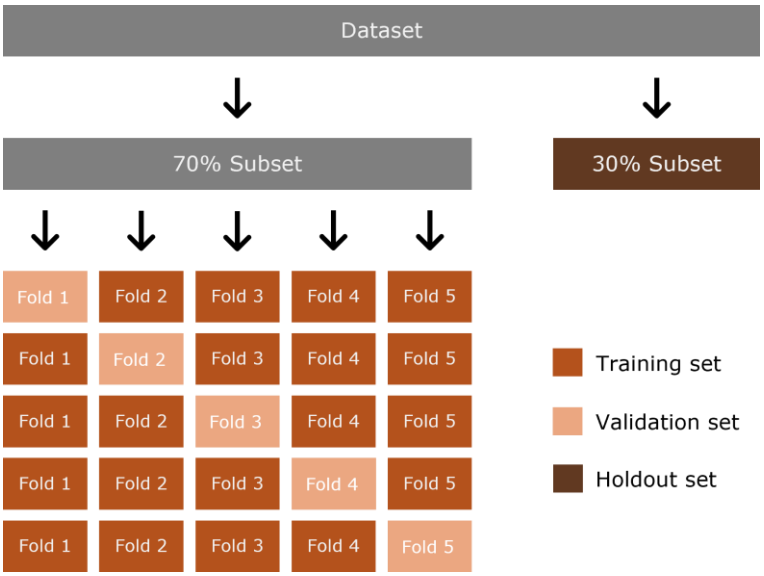

Figure 10. Holdout method combined with 5-fold cross-validation.

To also evaluate a model's robustness, an additional performance evaluation can be done with an adversarial set created by an adversarial attack method. However, the way that an adversarial set is created will heavily influence the results. Considering that NID systems are developed in a secure environment with thoroughly verified data, an attacker will not likely have access to a model's training set, and will have to perform an adversarial evasion attack relying on network traffic recorded during the initial reconnaissance stage of a cyber-attack. Likewise, considering that NID systems are deployed with security measures to encapsulate the utilized models, an attacker will not likely have access to internal parameters nor to confidence scores, and the only feedback will be if the cyber-attack is successful or not.

This evasion attack vector can be simulated by solely giving an adversarial method access to the holdout set and to a model's class predictions, enabling it to create an adversarial holdout set to cause misclassifications in that specific model without introducing any bias in the results. Therefore, a robustness analysis for a single ML model can be performed in 4 steps:

1. Preprocess a dataset, splitting it into training and holdout sets.

2. Train and validate an ML model, using the training set.

3. Perform an adversarial evasion attack to create a model-specific adversarial holdout set, using the regular holdout set and the model's class predictions.

4. Evaluate the model's performance on the regular and adversarial holdout sets, analyzing its generalization to regular data and its robustness to adversarial data.

In addition to a regularly trained model, an adversarial training approach can be included to create a second model and evaluate how training with slightly perturbed samples affects its performance. These two models originate from the same algorithm, so the trade-off of performance on regular data required to improve the performance on adversarial data can be analyzed for that specific algorithm. By creating this pair of models for multiple different algorithms, from tree-based algorithms and ensembles to deep learning algorithms based on ANNs, it is possible to perform a trustworthy comparison of multiple ML models. Therefore, the complete adversarial robustness analysis methodology consists of 5 steps, in which steps 3 and 4 are to be replicated for each algorithm:

1. Preprocess a dataset, splitting it into training and holdout sets.

2. Perform a simple data perturbation in a copy of each sample of the regular training set, creating an augmented adversarial training set with more data variations.

3. Train and validate two ML models, the first using the regular training set and the second using the adversarial training set.

4. Perform two adversarial evasion attacks to create two model-specific adversarial holdout sets, using the regular holdout set and each model's class predictions.

5. Evaluate each model's performance on the regular and adversarial holdout sets, comparing their generalization to regular data and their robustness to adversarial data.

As more regular network traffic flows are misclassified, a model's performance will worsen on the regular holdout set. Likewise, as more adversarial cyber-attack examples are misclassified, a model's performance will worsen on the adversarial holdout set. Therefore, a model should achieve a good balance between the independent evaluation results of the regular and adversarial holdout sets. From the multiple ML models under evaluation, the one that presents the best balance can be considered to have the best adversarially robust generalization. In the NID domain, this model will be the most capable of detecting cyber-attack variations, so it will be the most reliable for an intelligent cybersecurity system.

The proposed methodology aims to enable a security-by-design approach during the development of ML models and during the lifecycle of an intelligent system. From the comparison performed in the last step, the best model can be selected for deployment, but if new data is recorded, the analysis should be repeated to ensure that the model continues to be robust. This methodology is meant to be regularly replicated with up-to-date data recordings to anticipate possible threats to the utilized models and use that knowledge to improve the adversarial defense strategy of the system (Figure 11).

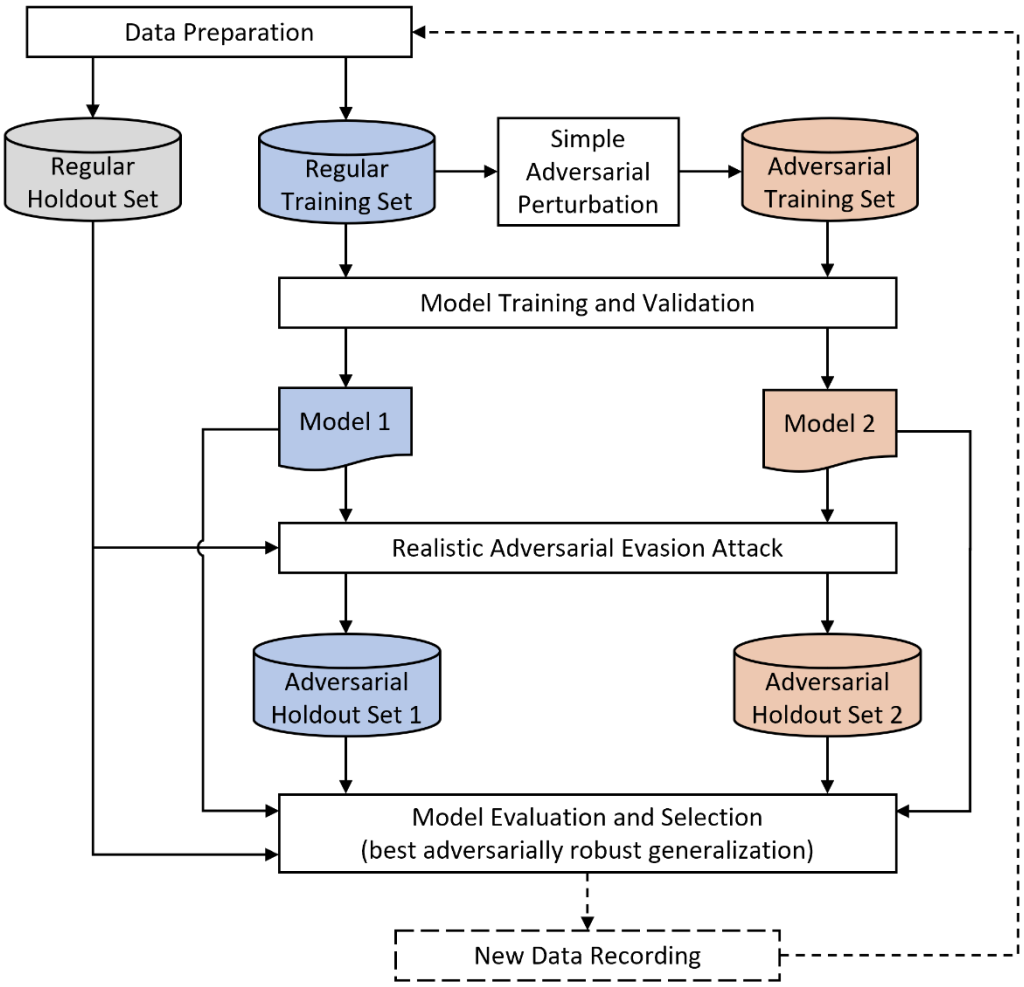

Figure 11. Adversarial robustness analysis methodology.

3.1.3 Model Evaluation

To compare multiple ML models and select the one with the best adversarially robust generalization, all models must be evaluated in a consistent and standardized manner. This evaluation must use metrics that correctly reflect the impact of adversarial examples on a model's performance. The considered evaluation metrics and their interpretation are briefly described below [232]–[234].

The basis for evaluating a model's performance in a classification task in a domain like NID is the confusion matrix. It reports the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) by checking the model's class predictions and comparing them with the true class of each sample. In a multi-class classification task, the values of a confusion matrix can be calculated for each class (Figure 12).

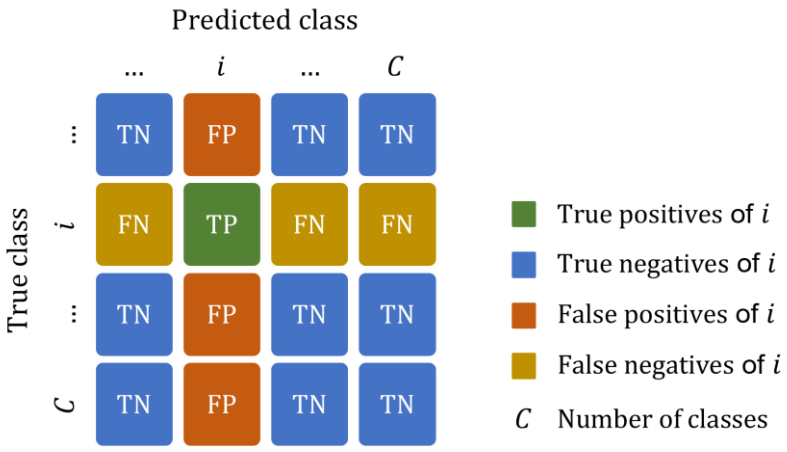

Figure 12. Multi-class confusion matrix.

The standard classification metric across the literature is accuracy, which measures the proportion of correctly classified samples. However, its bias towards the majority classes must not be disregarded when there is class imbalance and the minority classes are particularly relevant, which is the case of the NID domain. Since the adversarial evasion attack of step 4 will generate examples solely for cyber-attack flows, an accuracy as high as the proportion of benign flows could still be achieved even if all examples evaded detection. Therefore, to correctly exhibit the misclassifications caused by an adversarial attack, the accuracy score of a model should be calculated using the samples of all classes except the target class. This metric can be mathematically expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Despite the reliability of accuracy, there are other suitable metrics to measure the impact of an attack across all the different classes. To account for class imbalance during model validation, metrics that can be macro-averaged should be preferred because they give all classes the same relevance, which causes significantly lower scores when there are generalization errors and performance declines in minority classes.

Recall, which corresponds to the true positive rate and is also referred to as sensitivity, is a very valuable evaluation metric. It measures the proportion of samples of class i that were correctly predicted as class i, which reflects a model's ability to identify that class. The macro-averaged recall is defined as:

$$Macro\text{-}averaged\ Recall = \frac{1}{C} * \sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i} \tag{2}$$

where TP_i and FN_i are the TP and FN of class i, and C is the total number of classes.

Precision is also a valuable metric because it measures the proportion of samples predicted as class i that actually belonged to class i, which indicates the relevancy of a model's class predictions. The macro-averaged precision is defined as:

$$Macro\text{-}averaged\ Precision = \frac{1}{C} * \sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i} \quad (3)$$

where TP_i and FP_i are the TP and FP of class i, and C is the total number of classes.

These metrics are consolidated in the F1-Score, which calculates the harmonic mean of precision and recall, considering both FP and FN. The macro-averaged F1-Score is a reliable metric for NID because a high score indicates that the different cyber-attack classes are being correctly identified and there are few false alarms, which is important for security practitioners. Therefore, the macro-averaged F1-Score was the selected metric for the model validation of step 3 of the robustness analysis. It can be expressed as:

$$Macro\text{-}averaged\ F1\text{-}Score = \frac{1}{C} * \sum_{i=1}^{C} \frac{2 * P_i * R_i}{P_i + R_i} \quad (4)$$

where P_i and R_i are the precision and recall of class i, and C is the number of classes.

3.2 Adversarial Method

An adversarial method is only suitable for adversarial attacks and training in the NID domain if it accounts for its complex constraints. Considering the two properties that are required for an adversarial example to be realistic, validity and coherence, it is pertinent to use a perturbation crafting process capable of complying with the domain constraints of a real communication network and the class-specific constraints of a certain cyber-attack class.

The Adaptative Perturbation Pattern Method (A2PM) was specifically developed to address the constraints of complex tabular data domains like the NID domain. By analyzing the characteristics of each class, the method can generate constrained data perturbations that result in realistic adversarial examples (Figure 13). It was implemented in the Python 3 programming language and the *numpy* library was employed to optimize several mathematical operations, using vectorized data structures to consume less memory and provide faster execution speed. The source code repository is available at *GitHub*[1] and the documentation is available at *Read the Docs*[2], with the details of each function and a simple how-to guide.

[1] A2PM source code: https://github.com/vitorinojoao/a2pm

[2] A2PM documentation: https://a2pm.readthedocs.io/en/latest

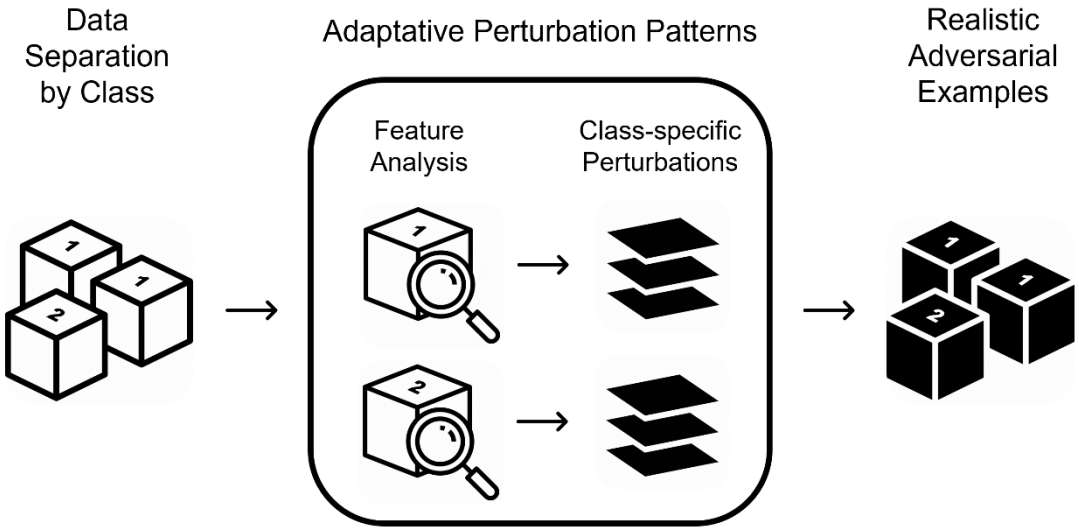

Figure 13. Adaptative perturbation pattern method.

3.2.1 Method Workflow

A2PM was designed with a modular architecture where an independent sequence of adaptative perturbation patterns is assigned to each class. These sequences analyze specific feature subsets to learn the characteristics of original data samples and generate valid and coherent data perturbations. To adjust it to a specific domain, A2PM only requires a simple base configuration for the creation of a pattern sequence. Afterwards, simple data perturbations can be added to original data samples to perform adversarial training, augmenting a training set with more data variations, or realistic adversarial examples can be iteratively created in an adversarial evasion attack against a classification model (Figure 14).

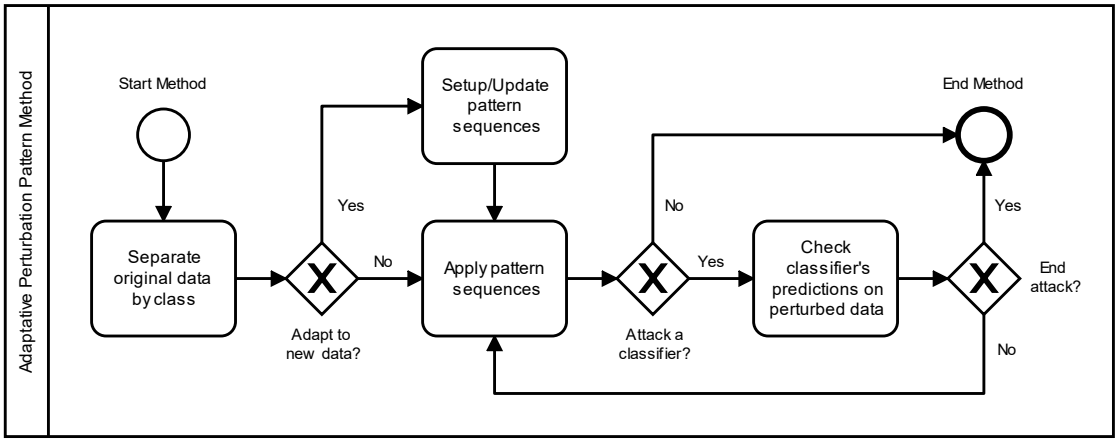

Figure 14. Base method workflow.

To use A2PM for adversarial training, it can be adapted to the characteristics of a regular training set. The method provides a function to add simple adversarial data perturbations to a

copy of each original sample within a regular training set, transforming it into an adversarial training set with perturbed samples. In the NID domain, this enables a model to learn not only from a recorded cyber-attack, but also from a simple variation of it. These perturbations could be performed manually by analyzing the entire dataset and adding modified samples according to the characteristics of each class. Nonetheless, to automate the process and prevent any bias, the randomness in feature choice and reliability of A2PM are preferred.

To use A2PM for adversarial attacks, it can be adapted to the characteristics of a regular holdout set. The method provides a function to perform full evasion attacks, generating as many perturbations as necessary in a copy of the regular holdout set until every data sample is misclassified or a specified maximum number of iterations is reached. This results in a model-specific adversarial holdout set containing the required perturbations to cause the most misclassifications in the attacked ML model. It is pertinent to highlight that even though A2PM can be used for both attack and defense strategies, it exhibits distinct behaviors in the simple perturbation function adapted to a training set and the full attack adapted to a holdout set, so it can be reliably used in a robustness analysis.

The performed attacks can be untargeted, to cause any misclassification, or targeted, seeking to reach a specific class. In the case of the NID domain, the untargeted attacks cause any misclassification of malicious flows to the benign or other cyber-attack classes, whereas the targeted attacks attempt to cause misclassifications of malicious flows solely into the benign class. An attack could generate data perturbations indefinitely, but it would be computationally expensive. Hence, early stopping is employed to end an attack when the latest iterations could not cause any further misclassifications.

In addition to static scenarios where the full data is available, the method is also suitable for scenarios where it is provided over time, accounting for changes in the data distributions. After the pattern sequences are created for an initial batch of data, these can be incrementally adapted to the characteristics of subsequent batches. Additionally, if novel classes are provided, the base configuration is used to automatically create their respective pattern sequences.

Even though A2PM could be applied in a black-box setting, it can only generate fully realistic examples in gray-box, with knowledge of the utilized features. This setting restriction is inherent to the feature analysis that is performed to setup and update the pattern sequences, but it also works as a safeguard against illegitimate use by actual attackers that do not have access to confidential system information about the utilized model and feature set.

3.2.2 Interval Pattern

One of the main concepts of the performed feature analysis is the interval of values that each numerical variable can have across different classes. To perturb uncorrelated numerical variables, an intra-feature constraint must be fulfilled by enforcing minimum and maximum values to a perturbed feature, according to its interval for a certain class.

The Interval pattern is built upon this concept and encapsulates a mechanism that records the valid intervals to create perturbations tailored to the characteristics of each feature (Figure 15). It has a configurable 'probability to be applied', in the $(0, 1]$ interval, which is used to randomly determine if an individual feature will be perturbed or not. For specific features that should only have whole numbers, it is also possible to specify enforce integer perturbations.

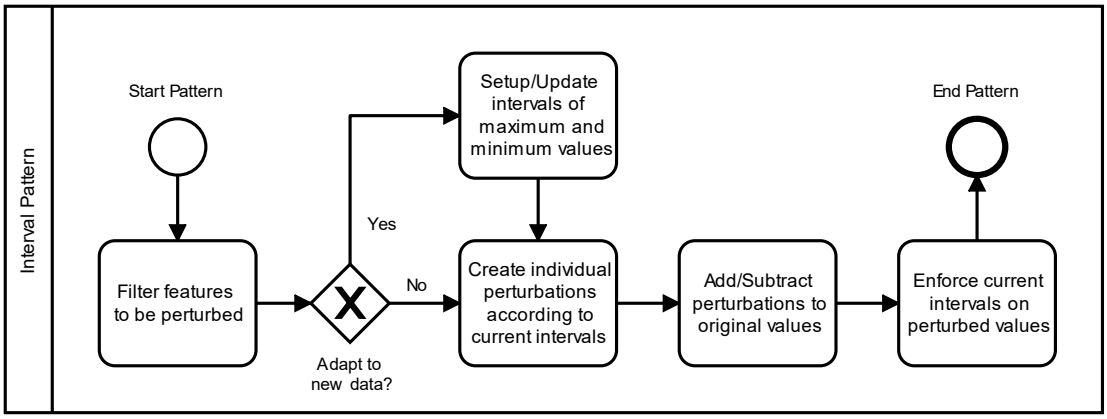

Figure 15. Interval pattern workflow.

Instead of a static interval, moving intervals can be utilized after the first batch to enable an incremental adaptation to new data, according to a configured momentum. For a given feature and a momentum $k \in [0, 1]$, the updated minimum m_i and maximum M_i of a batch i are mathematically defined as:

$$m_i = m_{i-1} * k + \min(x_i) * (1 - k) \tag{5}$$

$$M_i = M_{i-1} * k + \max(x_i) * (1 - k) \tag{6}$$

where $\min(x_i)$ and $\max(x_i)$ are the actual minimum and maximum values of the samples x_i of batch i.

Each perturbation is computed according to a randomly generated number and is affected by the current interval, which can be either static or moving. The random number $\varepsilon \in (0, 1]$ acts as a ratio to scale the interval. To restrict its possible values, it is generated within the standard range of $[0.1, 0.3]$, although other ranges can be configured. For a given feature, a perturbation P_i of a batch i can be represented as:

$$P_i = (M_i - m_i) * \varepsilon \tag{7}$$

After a perturbation is created, it is randomly added or subtracted to the original value. Exceptionally, if the original value is less or equal to the current minimum, it is always increased, and vice-versa. The resulting value is capped at the current interval to ensure it remains within the valid minimum and maximum values of that feature.

3.2.3 Combination Pattern

Regarding uncorrelated categorical variables, enforcing their limited set of qualitative values is the main intra-feature constraint. Therefore, the interval approach cannot be replicated even if they are encoded in a binary or numerical form, and a straightforward solution could be recording each value that can be present in a certain feature of a certain class.

Nonetheless, the most pertinent aspect of perturbing tabular data is the correlation between multiple variables. Since the value present in a variable may influence the values used for other variables, there can be several inter-feature constraints in both correlated numerical variables and correlated categorical variables. To improve beyond the previous solution and fulfill both types of constraints, several features can be combined into a single common record.

The Combination pattern records the valid value combinations to perform a simultaneous and coherent perturbation of multiple features (Figure 16). It can be configured with locked features, whose values are used to find combinations for other features without being modified themselves. Due to the consideration of multiple features, its 'probability to be applied', in the $(0, 1]$ interval, will affect several features.

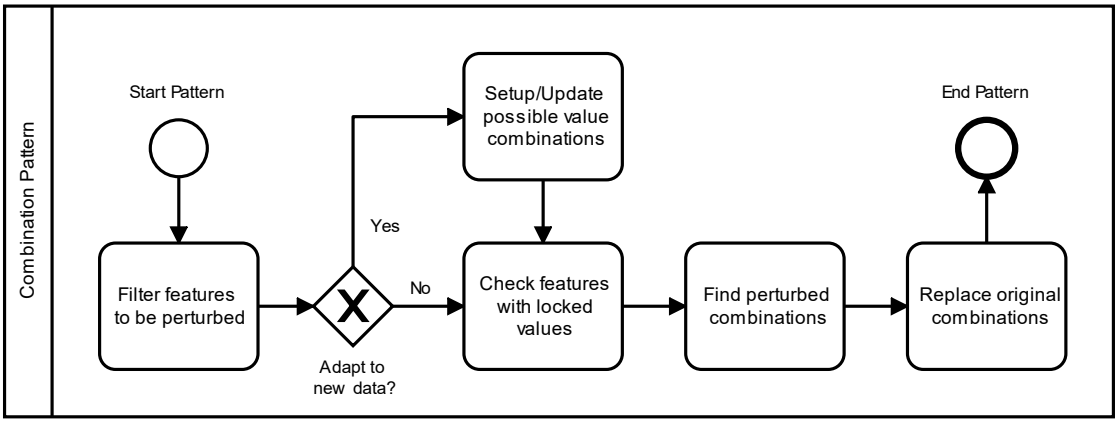

Figure 16. Combination pattern workflow.

Besides the initially recorded combinations, new data can provide additional possibilities. These can be merged with the previous or used as gradual updates. For a given feature and a momentum $k \in [0, 1]$, the number of updated combinations C_i of a batch i is expressed as:

$$C_i = C_{i-1} * k + \text{unique}(x_i) \tag{8}$$

where $\text{unique}(x_i)$ is the number of unique combinations of the samples x_i of batch i.

Each perturbation created by this pattern consists of a combination randomly selected from the current possibilities, considering the locked features. It directly replaces the original values, ensuring that the features remain coherent.

3.2.4 Pattern Sequences

A pattern sequence is an aggregation of several Interval and Combination patterns in a sequential order. The main advantage of applying the consecutive perturbations of a pattern sequence is that they enable the fulfillment of inter-feature constraints of greater complexity, which is required for complex domains that have diverse constraints that cannot be fulfilled by a single pattern. It is pertinent to note that all patterns in a sequence are independently adapted to the original data, to prevent any bias when recording its characteristics. Afterwards, the sequential order of patterns is enforced to create cumulative perturbations on that data.

An exemplification of the benefits of using these sequences is provided below, considering a small but relatively complex domain. The domain contains three nominal features, F0, F1 and F2, and two integer features, F3 and F4. For an adversarial example to be realistic within this domain, it must comply with the following constraints:

- F0 must always keep its original value.
- F1 and F4 can be modified but must have class-specific values.
- F2 and F3 can be modified but must have class-specific values, which are influenced by F0 and F1.

The base configuration corresponding to these constraints specifies the feature subsets that each pattern will analyze and perturb:

- Combination pattern – Modify {F1}.
- Combination pattern – Modify {F2, F3}, Lock {F0, F1}.
- Interval pattern – Modify {F3, F4}, Integer {F3, F4}.

A2PM will then assign each class to its own pattern sequence. For this example, the ‘probability to be applied’ will be 1.0 for all patterns, to demonstrate all three cumulative perturbations (Figure 17). The first perturbation created for each class is replacing F1 with another valid qualitative value, from ‘B’ to ‘C’. Then, without modifying the original F0 nor the new F1, a valid combination is found for F0, F1, F2 and F3. Since the original F2 and F3 were only suitable for ‘A’ and ‘B’, new values are found to match ‘A’ and ‘C’. Finally, the integer features F3 and F4 are perturbed according to their valid intervals. Regarding F3, to ensure it remains coherent with F0 and F1, the perturbation is created on the value of the new combination.

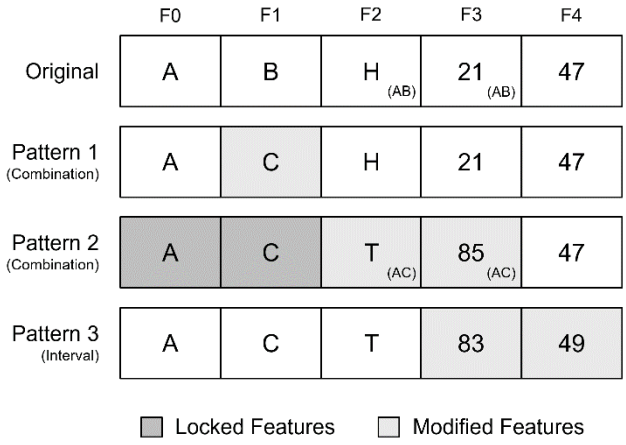

Figure 17. Consecutive perturbations of a pattern sequence.

3.3 Chapter Remarks

In this chapter, a methodology for an adversarial robustness analysis was defined, and an intelligent method for the generation of realistic adversarial examples was detailed. These two contributions are the proposed solution to tackle the lack of ML robustness in the NID domain, and were designed to be transferable to other complex tabular data domains.

The proposed methodology uses a realistic adversarial evasion attack vector and standardizes the evaluation and comparison of the generalization and robustness of multiple ML models. It was demonstrated that a successful adversarial attack is not guaranteed to be a successful cyber-attack, and that adversarial data perturbations can only be realistic if they are simultaneously valid and coherent, complying with the domain constraints of a real communication network and the class-specific constraints of a certain cyber-attack class.

The developed adversarial method, A2PM, relies on pattern sequences that are independently adapted to the characteristics of each class to generate constrained data perturbations and constrained adversarial examples that preserve both validity and coherence. It has a modular architecture and can be used for adversarial attacks in a gray-box setting, to iteratively cause misclassifications, and adversarial training, to augment a training set with more data variations. In addition to the source code, the documentation was also made publicly available.

4 Realism Case Study

This chapter presents the case study that analyzed the realism of the adversarial examples created by the targeted and untargeted attacks of the developed method. The study assessed if the generated perturbations complied with the constraints of Enterprise and IoT networks, preserving both validity and coherence, and if the method could be used for adversarial attacks and training with a low time consumption. The following sections describe the configuration of the study and present an analysis of the obtained results.

4.1 Study Configuration

Two scenarios were considered for the case study: Enterprise networks and IoT networks. For these scenarios, adversarial cyber-attack examples were created by A2PM using the original flows of the CIC-IDS2017 and the IoT-23 datasets, respectively. Assessments of adversarial realism and time consumption were performed by comparing the examples with the original flows and recording the time required for each iteration. To thoroughly analyze realism, the potential alternatives of the current literature were included: JSMA and OnePixel.

Since the internal reasoning of a deep learning algorithm based on an ANN and a tree-based ensemble are noticeably different, the susceptibility of both types of models to A2PM was analyzed by performing targeted and untargeted evasion attacks against Multilayer Perceptron (MLP) and RF. The attacks attempted to cause misclassifications for a maximum of 50 iterations, assessing the perturbations created by full adversarial attacks. In addition to evaluating the robustness of models created with regular training, models created with adversarial training with simple perturbations were also evaluated.

The study was conducted on common hardware: a machine with 16 gigabytes of random-access memory, an 8-core central processing unit, and a 6-gigabyte graphics processing unit. The implementation relied on the Python 3 programming language and several libraries: *numpy* and *pandas* for data preparation and manipulation, *tensorflow* for the MLP models, *scikit-learn* for the RF models, and *adversarial-robustness-toolbox* for the alternative methods.

4.1.1 Data Preprocessing

Multiple captures of the CIC-IDS2017 and IoT-23 datasets were selected to be used for their corresponding scenario. For CIC-IDS2017, the captures corresponding to Tuesday and Wednesday were merged, resulting in more than a million samples. For IoT-23, Capture-1-1 and Capture-34-1 were merged, also leading to a little over one million samples. Table 11 provides an overview of their characteristics, including the class proportions and the label of each class, either benign or a specific type of cyber-attack.

Table 11. Class proportions of realism case study datasets.

Scenario	Dataset (Captures)	Total Samples	Class Samples	Class Label
Enterprise computer network	CIC-IDS2017 (Tuesday and Wednesday)	1,138,612	873,066	Benign
			230,124	Hulk
			10,293	GoldenEye
			7,926	FTP-Patator
			5,897	SSH-Patator
			5,796	Slowloris
			5,499	Slowhttptest
			11	Heartbleed
IoT device network	IoT-23 (Capture-1-1 and Capture-34-1)	1,031,893	539,587	PortScan
			471,198	Benign
			14,394	DDoS
			6,714	C&C

Before their data was usable, both datasets required several data preprocessing steps. First, the features that did not provide any valuable information about a flow's benign or malicious purpose, such as timestamps and destination addresses, were discarded. Then, the categorical features were converted to numeric values by performing one-hot encoding. Due to the high cardinality of the categorical features, the very low frequency categories were aggregated into a single category designated as 'other', to avoid encoding qualitative values that were present in almost no samples and therefore had a small relevance.

Following the proposed analysis methodology, the holdout method was applied to randomly split the data into training and holdout sets with 70% and 30% of the samples, with stratification to perverse the original class proportions. The distinct characteristics of the resulting subsets were analyzed to identify the types of constraints required for each scenario and establish the base configurations for A2PM.

Regarding CIC-IDS2017, the sets were comprised of 8 imbalanced classes and 83 features, 58 numerical and 25 categorical. Most numerical features were continuous, but some had discrete

values that could only have integer perturbations. Due to the correlation between the encoded categorical features, they required combined perturbations to be compatible with a valid flow. Additionally, to guarantee the coherence of a generated flow with its cyber-attack class, the encoded features representing the utilized communication protocol and endpoint, designated as port, could not be modified. Hence, the following configuration was used for the Enterprise network scenario, after it was converted to the respective feature indices:

1. Interval pattern: Modify {numerical features}, Integer {discrete features}.

2. Combination pattern: Modify {categorical features}, Lock {port, protocol}.

Regarding IoT-23, the sets contained 4 imbalanced classes and approximately half the structural size of CIC-IDS2017, with 42 features, 8 numerical and 34 categorical. Despite the different features of IoT-23, it presented similar constraints. The main difference was that, in addition to the communication protocol, a generated flow also had to be coherent with the encoded features representing the application protocol, which was designated as service. The base configuration utilized for the IoT network scenario was:

1. Interval pattern: Modify {numerical features}, Integer {discrete features}.

2. Combination pattern: Modify {categorical features}, Lock {port, protocol, service}.

It is pertinent to note that, for the benign class, A2PM would only generate benign network traffic that could be misclassified as a cyber-attack. Therefore, the configurations were only applied to the cyber-attack classes, to generate examples compatible with their malicious purposes. Furthermore, since the examples should resemble the original flows as much as possible, the 'probability to be applied' was 0.6 and 0.4 for the interval and combination patterns, respectively. These values were established to slightly prioritize the small-scale modifications of individual numerical features over the more significant modifications of combined categorical features.

4.1.2 Model Fine-tuning

A total of 4 MLP and 4 RF models were created, one per scenario and per training approach: regular or adversarial training. The first approach used the original training sets, whereas the latter augmented the data with one adversarial example per malicious flow. The main characteristics of the models and their fine-tuning process are described below.

An MLP [235] is a feedforward ANN consisting of an input layer, an output layer and one or more hidden layers in between. Each layer can contain multiple nodes with forward connections to the nodes of the next layer. When utilized as a classifier, the number of input and output nodes correspond to the number of features and classes, respectively, and a prediction is performed according to the activations of the output nodes.

Due to the high computational cost of training an MLP, it was fine-tuned using a Bayesian optimization technique [236]. A validation set was created with 20% of a training set, which corresponded to 14% of the original samples. Since an MLP accounts for the loss of the training data, the optimization sought to minimize the loss of the validation data. To prevent overfitting, early stopping was employed to end the training when this loss stabilized. Additionally, due to the class imbalance present in both datasets, the assigned class weights were inversely proportional to their frequency.

The fine-tuning led to a four-layered architecture with a decreasing number of nodes for both training approaches. The hidden layers relied on the computationally efficient Rectified Linear Unit (ReLU) activation function and the dropout technique, which inherently helps prevent overfitting by randomly ignoring a certain percentage of the nodes of a layer during the model's training process. To address multi-class classification, the Softmax activation function was used to normalize the outputs to a class probability distribution. The MLP architecture for the Enterprise network scenario was:

1. Input layer: 83 nodes, 512 batch size.
2. Hidden layer: 64 nodes, ReLU activation, 10% dropout.
3. Hidden layer: 32 nodes, ReLU activation, 10% dropout.
4. Output layer: 8 nodes, Softmax activation.

A similar four-layered architecture was utilized for the IoT network scenario, with the ReLU activation function and a decreasing number of nodes, although it presented a decreased batch size and a higher dropout percentage:

1. Input layer: 42 nodes, 128 batch size.
2. Hidden layer: 32 nodes, ReLU activation, 20% dropout.
3. Hidden layer: 16 nodes, ReLU activation, 20% dropout.
4. Output layer: 4 nodes, Softmax activation.

The remaining MLP configuration was common to both scenarios because they represented similar multi-class cyber-attack classification tasks. Table 12 summarizes the utilized MLP configuration, obtained through the performed fine-tuning process.

Table 12. Multilayer Perceptron configuration for realism study.

Parameter	Value
Objective Loss	Categorical Cross-Entropy
Optimizer	Adam Algorithm
Learning Rate	0.001
Early Stopping	Enabled
Maximum Epochs	50
Class Weights	Balanced

On the other hand, an RF [237] is an ensemble of decision trees, which are decision support tools that use a tree-like structure. Each individual tree performs a prediction according to a different feature subset, and the most voted class across the ensemble is chosen to be the final prediction. It is based on the wisdom of the crowd, the concept that the collective decisions of multiple classifiers will be better than the decisions of just one.

Since training an RF has a substantially lower computational cost, a 5-fold cross-validated grid search was performed with well-established hyperparameter combinations for cyber-attack classification, following the proposed analysis methodology. Five stratified subsets were created, each with 20% of a training set. Then, five distinct iterations were performed, each training a model with four subsets and validating it with the remaining one. Hence, the MLP validation approach was replicated five times per combination. Table 13 summarizes the fine-tuned RF configuration, common to both scenarios and training approaches.

Table 13. Random Forest configuration for realism study.

Parameter	Value
Splitting Criterion	Gini Impurity
Number of Trees	100
Maximum Depth of a Tree	32
Minimum Samples in a Leaf	2
Maximum Features	$\sqrt{\text{Number of Features}}$

4.2 Results and Discussion

This section presents the results obtained by the MLP and RF models in the considered scenarios, as well as a comparative analysis of the realism of the examples, of the time consumption, and of the robustness of the models against adversarial cyber-attack examples.

4.2.1 Enterprise Network Scenario

In the Enterprise network scenario, adversarial cyber-attack examples were generated using the original flows of the CIC-IDS2017 dataset. The results obtained for the targeted and untargeted attacks were analyzed, and assessments of adversarial realism and time consumption were performed. To assess the realism of the created examples, these were analyzed and compared with the corresponding original flows, considering the functionalities and malicious purposes of the cyber-attacks. In addition to A2PM, the assessment included its potential alternatives: JSMA and OnePixel. To prevent any bias, a randomly generated number was used to select one example, detailed below.

The selected flow had the Slowloris class label, corresponding to a Denial-of-Service attack that attempts to overwhelm a web server by opening multiple connections and maintaining them as long as possible [230]. The data perturbations created by A2PM increased the total flow duration and the packet IAT, while reducing the number of packets transmitted per second and their size. It was observed that these modifications were mostly focused on enhancing time-related aspects of the cyber-attack to prevent its detection. Hence, in addition to being valid network traffic that can be transmitted through a computer network, the adversarial example also remained coherent with its class and IDT within the NID domain.

On the other hand, JSMA could not generate a realistic example for the selected flow. It created a major inconsistency in the encoded categorical features by assigning a single network flow to two distinct communication endpoints: destination ports number 80 and 88. Due to the unconstrained perturbations, the value of the feature representing port 88 was increased without accounting for its correlation with port 80, which led to an OOD example. In addition to the original Push flag (PSH) to keep the connection open, the method also assigned the flow to the Finished flag (FIN), which signals for connection termination and therefore contradicts the cyber-attack's malicious purpose. Even though two numerical features were also slightly modified, the adversarial example could only evade detection by using categorical features incompatible with real network traffic.

Likewise, OnePixel also generated an example that contradicted the Slowloris class. The feature selected to be perturbed represented the Reset flag (RST), which also causes termination. Since the method intended to perform solely one modification, it increased the value of a feature that no model learnt to detect because it is OOD, being incoherent with that cyber-attack's intended functionality. Consequently, neither JSMA nor OnePixel are adequate alternatives to A2PM for tabular data. Table 14 provides an overview of the modified features. The '--' character indicates that the original value was not perturbed.

Table 14. Modified features of an adversarial Slowloris example.

Feature	Original Value	A2PM Value	JSMA Value	OnePixel Value
Flow duration	109,034,141	119,046,064	109,034,140	--
Mean flow IAT	13,600,000	19,374,259	--	--
Flow packets per second	0.0825	0.0429	0.0824	--
Mean forward packet length	49.4	48.1	--	--
Min forward segment size	40	36	--	--
Connection flags	'PSH'	--	'PSH' + 'FIN'	'PSH' + 'RST'
Destination port	'80'	--	'80' + '88'	--

Regarding the targeted attacks performed by A2PM, the models created with regular training exhibited significant performance declines. Even though both MLP and RF achieved over 99% accuracy on the original holdout set, a single iteration lowered their scores by approximately 15% and 33%. In the subsequent iterations, more malicious flows gradually evaded MLP detection, whereas RF was quickly exploited. After 50 iterations, their very low accuracy evidenced their inherent susceptibility to adversarial examples. In contrast, the models created with adversarial training kept significantly higher scores, with fewer flows being misclassified as benign. By training with one perturbed sample per malicious flow, both classifiers successfully learned to detect most cyber-attack variations. RF stood out for preserving the 99.91% it obtained on the original holdout set throughout the entire attack, which highlighted its excellent generalization (Figure 18).

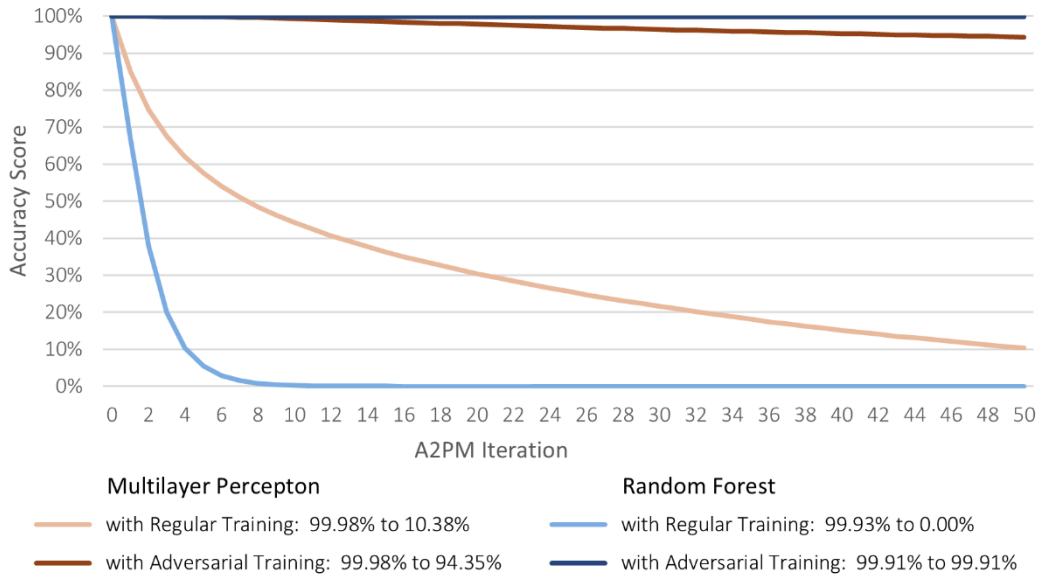

Figure 18. Targeted attack accuracy of Enterprise network scenario.

The untargeted attacks significantly lowered both evaluation metrics. The accuracy and macro-averaged F1-Score declines of the regularly trained models were approximately 99% and 79%, although RF was more affected in the initial iterations. The inability of both classifiers to distinguish between the different classes corroborated their high susceptibility to adversarial examples. Nonetheless, when adversarial training was performed, the models preserved considerably higher scores, with a gradual decrease of less than 2% per iteration. Despite some examples still deceiving them into predicting incorrect classes, both models were able to better distinguish each type of cyber-attack, which mitigated the impact of the generated perturbations. The adversarially trained RF consistently reached higher scores than MLP in both targeted and untargeted attacks, indicating a better robustness (Figures 19 and 20).

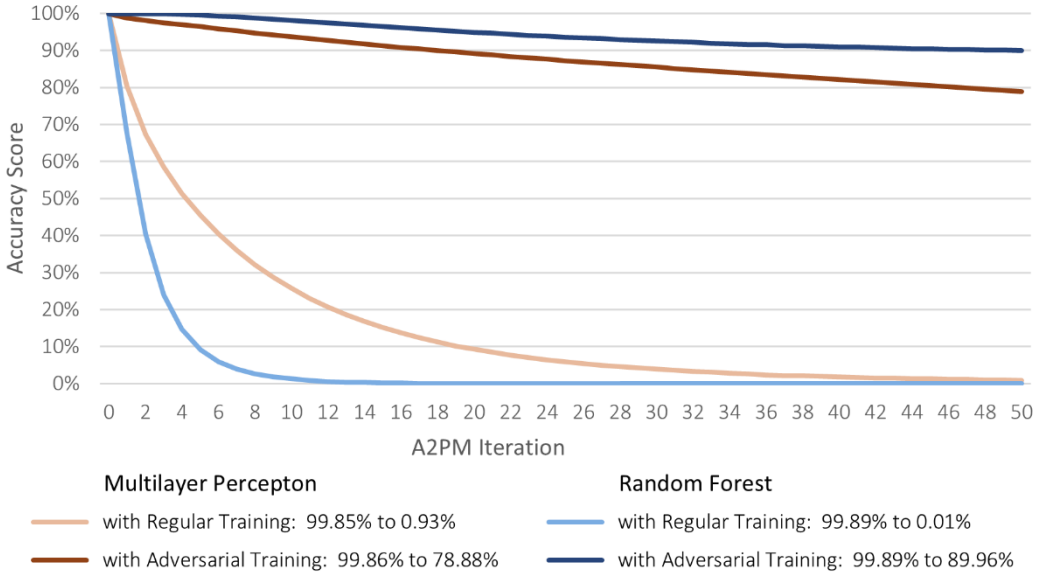

Figure 19. Untargeted attack accuracy of Enterprise network scenario.

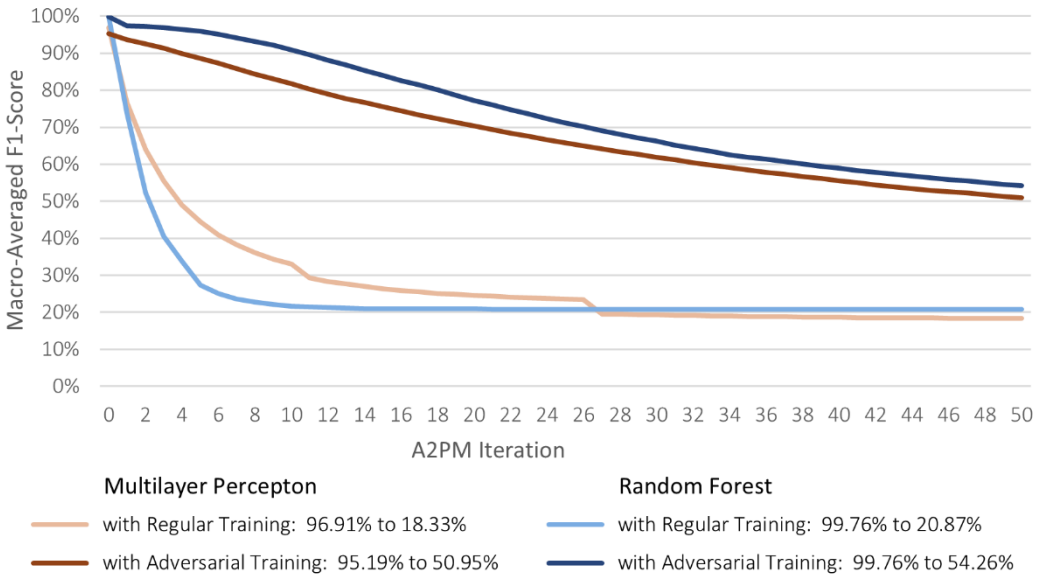

Figure 20. Untargeted attack F1-Score of Enterprise network scenario.

To analyze the time consumption of A2PM, the number of milliseconds required for each iteration was recorded and averaged, accounting for the decreasing quantity of new examples generated as an adversarial evasion attack progressed. The generation was performed at rate of 10 examples per 1.7 milliseconds on the utilized hardware, which evidenced the fast execution and scalability of the developed method when applied to adversarial evasion attacks and training in Enterprise computer networks.

4.2.2 IoT Network Scenario

In the IoT network scenario, the adversarial cyber-attack examples were generated using the original flows of the IoT-23 dataset. The analysis performed for the previous scenario was replicated to provide similar assessments, including the potential alternatives of the current literature: JSMA and OnePixel. To prevent any bias, a randomly generated number was also used to select one example, detailed below.

The selected flow for the realism assessment had the DDoS class label, which corresponds to a Distributed Denial-of-Service attack performed by the malware recorded in the IoT-23 dataset. A2PM replaced the encoded categorical features of the connection state and history with another valid combination, already used by other original flows of the DDoS class. Instead of an incomplete connection (OTH) with a bad packet checksum (BC), it became a connection attempt (S0) with a Synchronization flag (SYN). The crafted example was IDT within the NID domain because it remained valid and compatible with the DDoS class.

As in the previous scenario, both JSMA and OnePixel generated unrealistic examples. Besides the original OTH, both methods also increased the value of the feature representing an established connection with a termination attempt (S3). Since a flow with simultaneous OTH and S3 states is neither valid nor coherent with the cyber-attack's purpose, the methods still created OOD examples, remaining inadequate alternatives to A2PM for complex tabular data domains. In addition to the states, JSMA also assigned a single flow to two distinct communication protocols, TCP and Internet Control Message Protocol (ICMP), which further evidenced the inconsistency of the created data perturbations. Table 15 provides an overview of the modified features, with '--' indicating an unperturbed value.

Table 15. Modified features of an adversarial DDoS example.

Feature	Original Value	A2PM Value	JSMA Value	OnePixel Value
Connection state	'OTH'	'S0'	'OTH' + 'S3'	'OTH' + 'S3'
Connection history	'BC'	'SYN'	--	--
Communication protocol	'TCP'	--	'TCP' + 'ICMP'	--

Regarding the targeted attacks, A2PM caused much slower declines than in the previous scenario. The accuracy of the regularly trained MLP only started being lower than 50% at iteration 43, and RF stabilized with approximately 86%. These scores evidenced the decreased susceptibility of both classifiers, especially RF, to adversarial examples targeting the benign class. Furthermore, with adversarial training with simple perturbations, the models were able to preserve even higher scores during an attack. Even though many examples still evaded MLP detection, the number of malicious flows predicted to be benign by RF was significantly lowered, which enabled it to keep its accuracy above 99%. Hence, the latter successfully detected most cyber-attack variations (Figure 21).

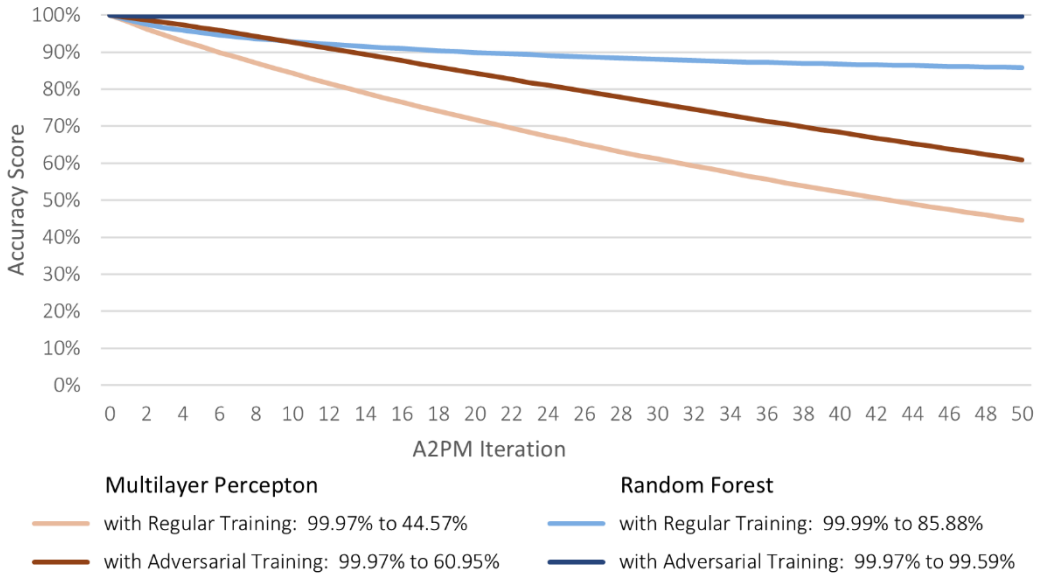

Figure 21. Targeted attack accuracy of IoT network scenario.

The untargeted attacks iteratively caused small decreases of both metrics. Despite RF starting to stabilize from the fifth iteration forward, MLP continued its decline for an additional 48% of accuracy and 17% of macro-averaged F1-Score. This difference in both targeted and untargeted attacks suggests that RF, and possibly tree-based algorithms in general, have a greater intrinsic robustness to adversarial examples of IoT network traffic. Unlike in the previous scenario, adversarial training did not provide considerable improvements. Nonetheless, the augmented training data still contributed to the creation of more adversarially robust models because they exhibited fewer incorrect class predictions throughout the attack (Figures 22 and 23).

A time consumption analysis was also performed, to further analyze the scalability of A2PM on common hardware like the one utilized in this study. The number of milliseconds required for each iteration was recorded and averaged, resulting in a rate of 10 examples per 2.4 milliseconds. By comparing the rates obtained in the two scenarios, it can be observed that it was 41% higher for IoT-23 than for CIC-IDS2017. Even though the former dataset had approximately half the structural size, with almost half the features, a greater number of locked categorical features were provided to the Combination pattern. Therefore, the increased time consumption suggests that the more complex inter-feature constraints are specified, the more

time will be required to apply A2PM and perform each iteration. Nonetheless, the time consumption was still reasonably low, which further evidenced the fast execution and scalability of the developed adversarial method.

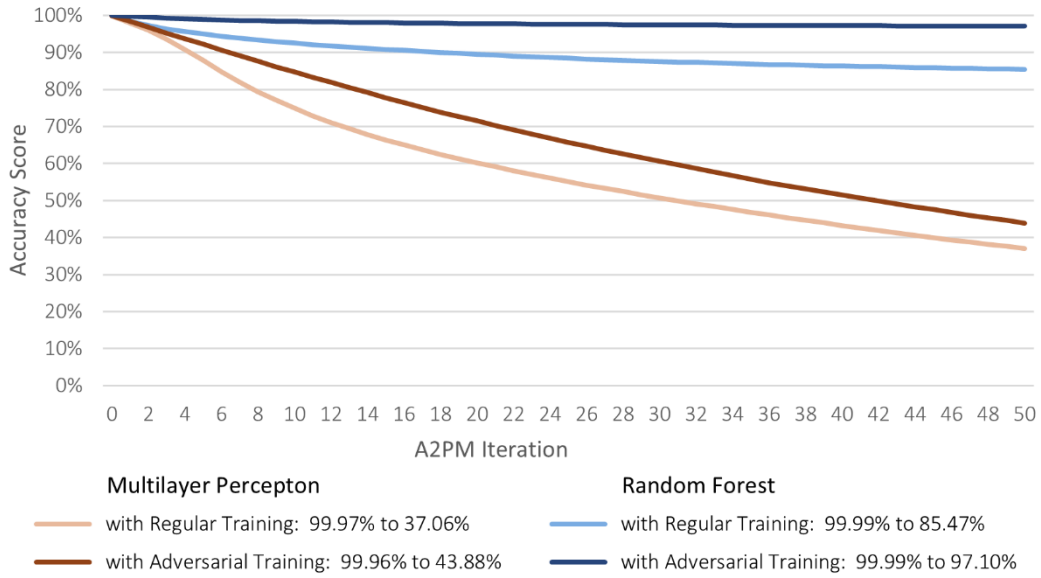

Figure 22. Untargeted attack accuracy of IoT network scenario.

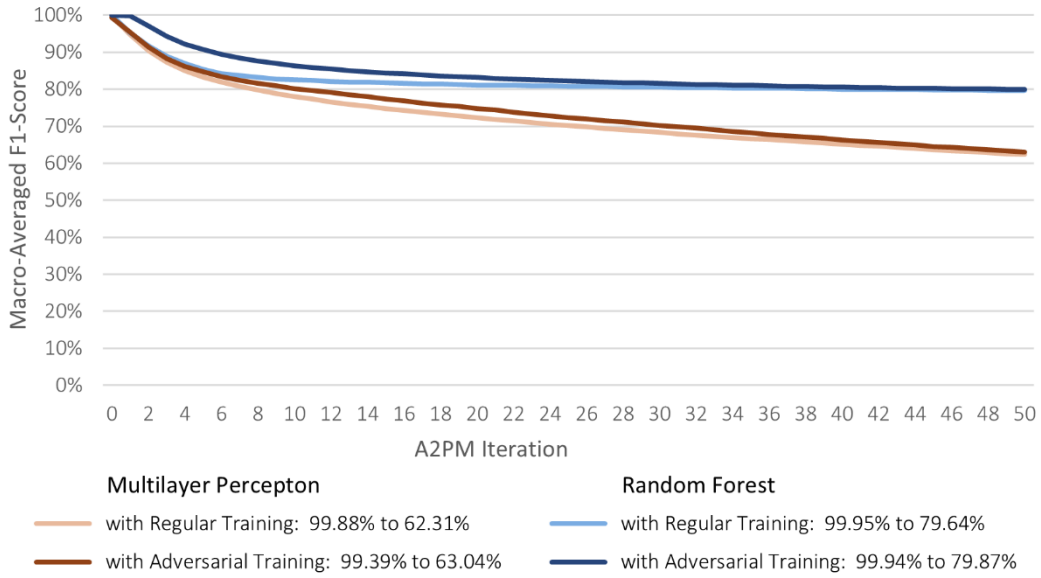

Figure 23. Untargeted attack F1-Score of IoT network scenario.

4.3 Chapter Remarks

In this chapter, the realism of the adversarial examples created by the developed method was assessed in Enterprise and IoT network scenarios. A total of 4 MLP and 4 RF models were created using the network traffic flows of the CIC-IDS2017 and IoT-23 datasets, and targeted and untargeted evasion attacks were performed against them. For each scenario, assessments of adversarial realism and time consumption were performed, and the impact of the attacks on the robustness of the models was analyzed.

The modular architecture of A2PM enabled the creation of pattern sequences adapted to each type of cyber-attack, according to the constraints of the utilized datasets. Both targeted and untargeted attacks successfully decreased the performance of all MLP and RF models, with significantly higher declines exhibited in the Enterprise network scenario. Nonetheless, the inherent susceptibility of these models to adversarial examples was mitigated by augmenting their training data with one crafted example with simple perturbations per malicious flow. Overall, the obtained results demonstrate that A2PM provides a scalable generation of valid and coherent examples for the NID domain.

5 Generalization Case Study

This chapter presents the case study that analyzed the generalization of adversarially trained ML models for binary and multi-class classification. The study assessed if performing adversarial training with simple data perturbations generated by the developed method enabled tree-based ensembles to retain a good generalization to regular IoT network traffic, in addition to being more robust to adversarial examples. The following sections describe the configuration of the study and present an analysis of the obtained results.

5.1 Study Configuration

Two scenarios were considered for the case study: IoT service networks and IoT device networks. For these scenarios, adversarial cyber-attack examples were created by A2PM using the original flows of the IoT-23 and the Bot-IoT datasets, respectively. The robustness and generalization of models created with regular training and with adversarial training were analyzed, to assess the effects of training with simple perturbations performed by A2PM.

Since tree-based algorithms and ensembles are well-established for classification tasks in the NID domain, targeted and untargeted evasion attacks were performed against RF, Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), and Isolation Forest (IFOR). The attacks attempted to cause misclassifications for a maximum of 30 iterations, assessing relatively fast attacks. In binary classification, the aim of a model was to detect that a network traffic flow was malicious, whereas in multi-class classification, a model had to correctly identify each cyber-attack class and distinguish between them. Even though IFOR can only perform anomaly detection, it was compared to the remaining models in binary classification.

The study was conducted on common hardware: a machine with 16 gigabytes of random-access memory, an 8-core central processing unit, and a 6-gigabyte graphics processing unit. The implementation relied on the Python 3 programming language and several libraries: *numpy* and *pandas* for data preparation and manipulation, *scikit-learn* for the implementation of RF and IFOR, *xgboost* for XGB, and *lightgbm* for LGBM.

5.1.1 Data Preprocessing

Multiple captures of the IoT-23 and Bot-IoT datasets were selected to be used for their corresponding scenario. For IoT-23, Capture-1-1 and Capture-34-1 were merged, whereas for Bot-IoT, the utilized capture was Full5pc-4. Table 16 provides an overview of their characteristics, including the class proportions and the label of each class, either benign or a specific type of cyber-attack.

Table 16. Class proportions of generalization case study datasets.

Scenario	Dataset (Captures)	Total Samples	Class Samples	Class Label
IoT service network	IoT-23 (Capture-1-1 and Capture-34-1)	1,031,893	539,587	PortScan
			471,198	Benign
			14,394	DDoS
			6,714	C&C
IoT device network	Bot-IoT (Full5pc-4)	668,522	576,884	DDoS
			91,082	Recon
			477	Benign
			79	Theft

Several data preprocessing steps were applied to both datasets. First, the features that did not provide valuable information about a flow's benign or malicious purpose, such as origin and destination addresses, were discarded. Then, one-hot encoding was employed to convert the categorical features to numeric values. Due to their high cardinality, low frequency categories were aggregated into a single category designated as 'other' to avoid encoding qualitative values that had a small relevance for the classification.

The data was randomly split into training and holdout sets with 70% and 30% of the samples, with stratification, according to the proposed analysis methodology. The types of constraints required for each scenario were identified and the base configurations for A2PM were established by analyzing the resulting subsets.

Regarding IoT-23, the sets were comprised of 4 imbalanced classes and 42 features, 8 numerical and 34 categorical, as in the previous case study. An equivalent configuration was used for the IoT service network scenario, after it was converted to the respective feature indices:

1. Interval pattern: Modify {numerical features}, Integer {discrete features}.

2. Combination pattern: Modify {categorical features}, Lock {port, protocol, service}.

Regarding Bot-IoT, the sets contained 4 imbalanced classes and 35 features, 15 numerical and 20 categorical. It required combined perturbations for the encoded categorical features and

presented some numerical features with discrete values for packet counts. Even though the sets did not contain services, the encoded features representing the utilized communication protocol and port could not be modified to guarantee the coherence of a generated flow with its cyber-attack class. The base configuration utilized for the IoT device network scenario was:

1. Interval pattern: Modify {numerical features}, Integer {discrete features}.

2. Combination pattern: Modify {categorical features}, Lock {port, protocol}.

As in the previous case study, it is pertinent to note that, for the benign class, A2PM would only generate benign network traffic that could be misclassified as a cyber-attack. Therefore, the configurations were only applied to the cyber-attack classes. The 'probability to be applied' was 0.6 and 0.4 for the interval and combination patterns, respectively, to slightly prioritize the small-scale modifications of individual numerical features over the more significant modifications of combined categorical features.

5.1.2 Model Fine-tuning

A total of 8 RF, 8 XGB, 8 LGBM, and 4 IFOR models were created, one per scenario, per regular or adversarial training approach, and per binary or multi-class classification task. The first approach created models with the original training sets, whereas the latter augmented the training set with one adversarial example per malicious flow. The main characteristics of the models and their fine-tuning process are described below.

The proposed analysis methodology was followed for all tree-based ensembles. Therefore, a grid search was performed with well-established hyperparameter combinations for NID and cyber-attack classification, and the optimal configuration for each model was determined through a 5-fold cross-validation. Five stratified subsets were created, each with 20% of a training set. Then, five distinct iterations were performed, each training a model with four subsets and validating it with the remaining one. After being fine-tuned, each model was retrained with its complete training set, so it was ready for the independent performance evaluations with the original holdout set and the adversarial holdout set.

For RF [237], which was described in the previous case study, the Gini Impurity criterion was used to measure the quality of the possible node splits, and the maximum number of features selected to build a tree was the square root of the total number of features of each dataset. The optimized value for the maximum depth of a tree was 32 and 16 for IoT-23 and Bot-IoT, respectively, and the minimum number of samples required to create a leaf node was 2 and 4. Table 17 summarizes the fine-tuned configuration.

Table 17. Random Forest configuration for generalization study.

Parameter	Value
Splitting Criterion	Gini Impurity
Number of Trees	100
Maximum Depth of a Tree	16 to 32
Minimum Samples in a Leaf	2 to 4
Maximum Features	$\sqrt{\text{Number of Features}}$

XGB [238] performs gradient boosting using a supervised ensemble of decision trees. A level-wise growth strategy is employed to split nodes level by level, seeking to minimize a loss function during its training. The acknowledged Cross-Entropy loss was used for both binary and multi-class classification, and the Histogram method was selected because it computes fast histogram-based approximations to choose the best splits. The key parameter of this model is the learning rate, which controls how quickly the model adapts its weights to the training data. It was optimized to relatively small values for each training set and scenario, ranging from 0.01 to 0.2. Table 18 summarizes the configuration.

Table 18. Extreme Gradient Boosting configuration for generalization study.

Parameter	Value
Method	Histogram
Objective Loss	Cross-Entropy
Learning Rate	0.01 to 0.2
Number of Trees	80 to 120
Maximum Depth of a Tree	8
Minimum Loss Reduction	0.01
Feature Subsample	0.7 to 0.8

LGBM [239] also utilizes a supervised ensemble of decision trees to perform gradient boosting. Unlike XGB, a leaf-wise strategy is employed, following a best-first approach. Hence, the leaf with the maximum loss reduction is directly split in any level. The key advantage of this model is its ability to use Gradient-based One-Side Sampling (GOSS) to build the decision trees, which is computationally lighter and therefore provides a faster training process. The Cross-Entropy loss was also used, and the minimum samples required to create a leaf was optimized to 16. To avoid fast convergences to suboptimal solutions, the learning rate was also kept at small values for the distinct datasets and scenarios. Table 19 summarizes the configuration.

Table 19. Light Gradient Boosting Machine configuration for generalization study.

Parameter	Value
Method	GOSS
Objective Loss	Cross-Entropy
Learning Rate	0.01 to 0.2
Number of Trees	80 to 120
Maximum Depth of a Tree	16
Maximum Leaves in a Tree	32
Minimum Loss Reduction	0.01
Minimum Samples in a Leaf	16
Feature Subsample	0.7 to 0.8

IFOR [240] isolates anomalies through an unsupervised ensemble of decision trees. The samples are repeatedly split by random values of random features until outliers are segregated from normal observations. Unlike the previous models, IFOR can only perform anomaly detection with unlabeled data. Nonetheless, it can be compared to the remaining models in binary classification, so cross-validation was also used to fine-tune it.

This model relies on the contamination ratio of a training set, which must not exceed 50%. Hence, the number of samples intended to be anomalies must be lower than the number of remaining samples, otherwise outliers cannot be detected. To reduce the contamination of the training data, each cyber-attack class was randomly subsampled with stratification. The optimized ratios of the total proportion of anomalies were 0.4 and 0.5 for IoT-23 and Bot-IoT, respectively. Therefore, the subsampled training data contained 40% and 50% of malicious samples. Table 20 summarizes the configuration.

Table 20. Isolation Forest configuration for generalization study.

Parameter	Value
Number of Trees	100
Contamination	0.4 to 0.5
Maximum Features	0.9
Maximum Samples	256

5.2 Results and Discussion

This section presents the results obtained by the four tree-based ensembles in binary and multi-class classification in the considered scenarios, as well as a comparative analysis of their generalization and robustness against adversarial cyber-attack examples.

5.2.1 IoT Service Network Scenario

In the IoT service network scenario, the models created with regular training exhibited reasonable performance declines in binary classification on the IoT-23 dataset. Even though all four models achieved over 99% accuracy on the original holdout set, numerous misclassifications were caused by the adversarial attacks. The lowest score on an adversarial holdout set, 68.35%, was obtained by XGB. In contrast, the models created with adversarial training kept significantly higher scores. By training with one realistically crafted example per malicious flow, all models successfully learnt to detect most cyber-attack variations. IFOR stood out for preserving the 99.98% accuracy it obtained on the original holdout set throughout the entire attack, which highlighted its excellent generalization (Figure 24).

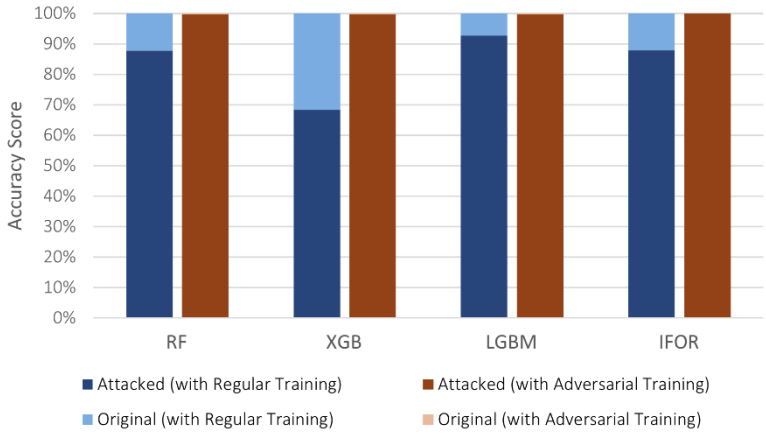

Figure 24. Attack accuracy of binary IoT service network scenario.

Regarding multi-class classification, the targeted and untargeted attacks had different impacts on a model's performance. The former caused malicious flows to be solely predicted as the benign class, whereas the latter caused malicious flows to be predicted as different classes, including other cyber-attack classes. Both attacks decreased the accuracy of the three supervised models on IoT-23, with LGBM being significantly more affected. Nonetheless, it can be observed that its targeted accuracy, 57.78%, was significantly higher than the untargeted, 32.11%, with more misclassifications occurring between different cyber-attack classes. Therefore, despite LGBM being susceptible, the benign class was more difficult to reach in multi-class cyber-attack classification. Even though performing adversarial training further increased the high scores of XGB, it was surpassed by RF on the targeted attack, which achieved 99.97% (Figures 25 and 26).

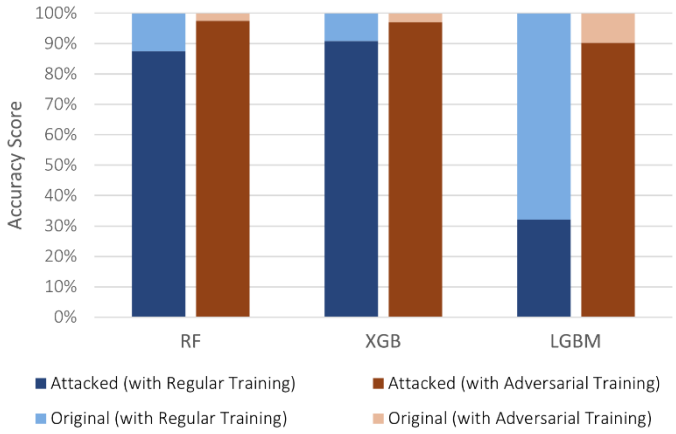

Figure 25. Untargeted attack accuracy of multi-class IoT service network scenario.

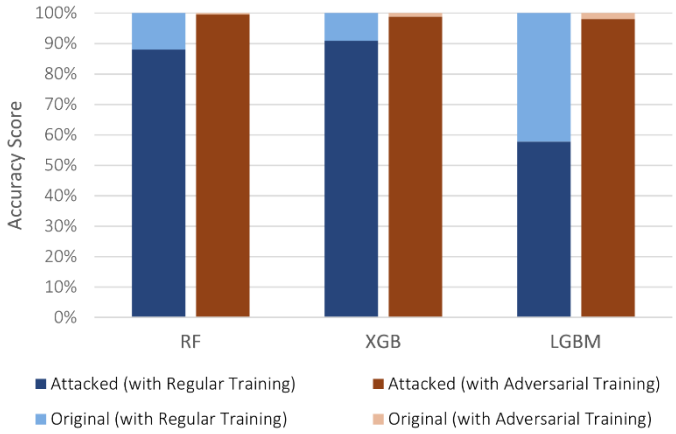

Figure 26. Targeted attack accuracy of multi-class IoT service network scenario.

5.2.2 IoT Device Network Scenario

In the IoT device network scenario with the Bot-IoT dataset, the declines were significantly higher. The inability of these tree-based algorithms to distinguish between the different classes evidenced their high susceptibility to adversarial examples. The score of LGBM dropped to 26.04%, followed by IFOR, with 34.31%. Regarding the latter, it could not reach 85% in the original holdout set, possibly due to the occurrence of overfitting. Despite some examples still deceiving them, the models created with adversarial training were able to learn the subtle nuances between each cyber-attack class, which mitigated the impact of the examples of the adversarial holdout set. Apart from IFOR, the remaining models consistently achieved scores over 97%, which indicated a good robustness (Figure 27).

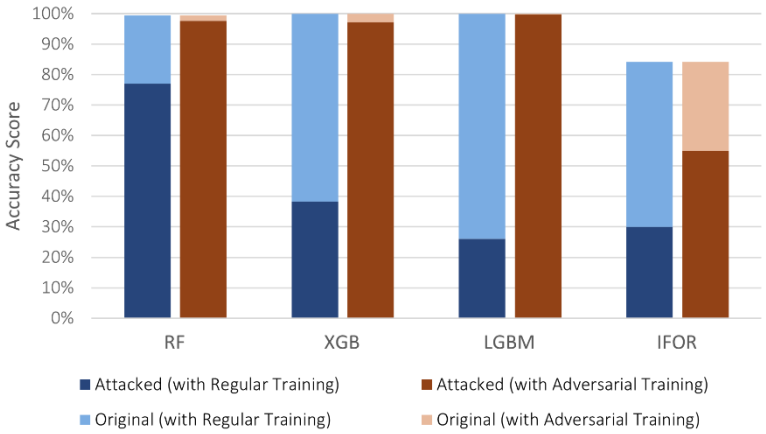

Figure 27. Attack accuracy of binary IoT device network scenario.

Higher declines were also exhibited in the multi-class classification task. The untargeted attacks performed by A2PM dropped the accuracy of RF and XGB to near 65%, although the targeted attacks only decreased it to 87.50% and 97.14%. Adversarial training contributed to the creation of more robust models, leading to fewer incorrect class predictions. Regarding RF, it could even preserve the 99.98% score it obtained on the original holdout set throughout the entire attack. Even though some malicious flows still evaded detection, the robustness of both XGB and LGBM was also successfully improved. Overall, the adversarial robustness of the analyzed tree-based algorithms was significantly improved by augmenting their training data with a simple variation of each cyber-attack (Figures 28 and 29).

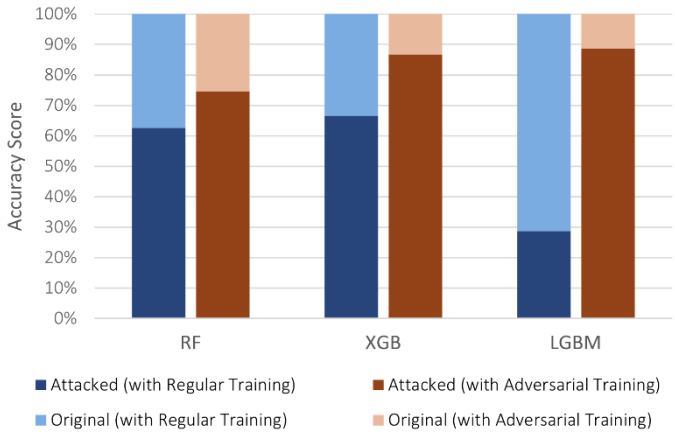

Figure 28. Untargeted attack accuracy of multi-class IoT device network scenario.

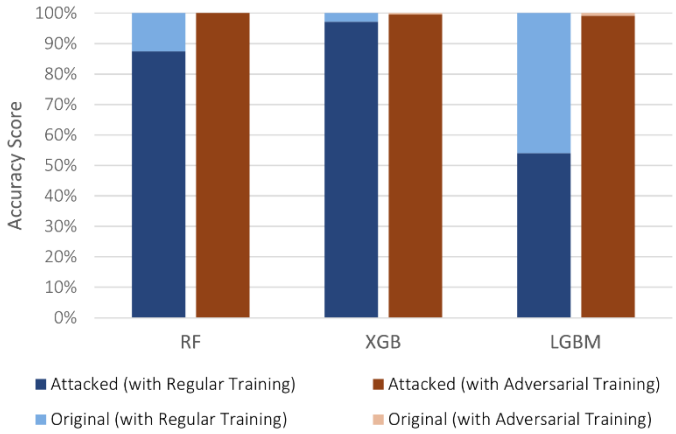

Figure 29. Targeted attack accuracy of multi-class IoT device network scenario.

5.3 Chapter Remarks

In this chapter, the generalization and robustness of tree-based ensembles with an adversarial training approach was assessed in IoT service and IoT device network scenarios. A total of 8 RF, 8 XGB, 8 LGBM, and 4 IFOR models were created using the network traffic flows of the IoT-23 and Bot-IoT datasets, and targeted and untargeted evasion attacks were performed against them. For each scenario, both regular and adversarial training approaches were evaluated in binary and multi-class classification tasks.

The models created with regular training exhibited significant performance declines, which were more prominent in the IoT device scenario. Even though RF was the least affected in the binary classification, XGB consistently achieved the highest accuracy in multi-class. Furthermore, when adversarial training was performed, all models successfully learnt to distinguish between most cyber-attack variations and kept significantly higher scores when attacked. The adversarially trained IFOR and RF stood out for preserving the highest accuracy throughout entire attacks, on binary IoT-23 and multi-class Bot-IoT, respectively. Regarding LGBM, the results suggest that it is highly susceptible to adversarial examples, especially on imbalanced multi-class classification. Nonetheless, this vulnerability can be tackled by augmenting its training data with one realistic adversarial example per malicious flow.

6 Conclusions

This chapter provides the main conclusions of this thesis, highlighting the accomplished objectives. The limitations of the proposed solution and possible improvements are also described, indicating research topics to be explored in the future.

6.1 Accomplished Objectives

This thesis addressed the lack of robustness of ML models and explored the realism of adversarial cyber-attack examples in the NID domain. All the initially established objectives were successfully accomplished, with a completion rate of 100%. A solution was developed to improve the security of ML in complex tabular data domains, which resulted in several scientific contributions. The main results for each objective were:

- **OB1**: A literature review of the state-of-the-art adversarial ML methods, constrained data generation approaches, and publicly available NID datasets. It was observed that most solutions were not designed to support the constraints of a communication network and of the utilized protocols, so they mostly lead to OOD data perturbations and unrealistic network traffic flow examples in the NID domain.

- **OB2**: A methodology for a trustworthy adversarial robustness analysis, with a realistic adversarial evasion attack vector and a comparison of multiple ML models. It was demonstrated that a successful adversarial attack is not guaranteed to be a successful cyber-attack, and that a robustness analysis must be performed with realistic attack vectors that account for the characteristics of the cyber-attack classes.

- **OB3**: An intelligent method for the generation of realistic adversarial examples in complex domains like NID. The developed method, A2PM, has a modular architecture and can be used for adversarial attacks, to iteratively cause misclassifications, and adversarial training, to augment a training set with more data variations. The source code and documentation are available at *GitHub*[1] and at *Read the Docs*[2].

- **OB4**: Two case studies that evaluated the suitability of A2PM for the NID domain. The first assessed the realism of the crafted examples, verifying that the perturbations preserved both validity and coherence and that the method could be used for adversarial attacks and training with a low time consumption. The second assessed the generalization of ML models, verifying that adversarial training with simple data perturbations enables tree-based ensembles to retain a good generalization to regular IoT network traffic, in addition to having a better robustness.

The obtained results evidence the inherent susceptibility of deep learning algorithms based on ANNs and of tree-based algorithms and ensembles to adversarial examples, and demonstrate that they can significantly benefit from adversarial training with simple data perturbations. The literature review, the proposed solution, and the use cases described in this thesis can help ML engineers and security practitioners improve both the robustness and the generalization of their ML models with realistically crafted adversarial examples.

6.2 Limitations and Future Work

Despite the benefits of the developed method to generate realistic examples according to a simple base configuration, it still requires previous knowledge about the utilized features, their data type, and which features are correlated. It is important to replicate the proposed analysis methodology in more case studies with other NID datasets and even live experiments in different types of communication networks, to start categorizing the constraints of different networks and help reduce the knowledge required to configure the developed method.

To further simplify the configuration of complex intra and inter-feature constraints, it would be very valuable to improve A2PM so it could automatically analyze all features and detect their correlations, creating the required interval and combination pattern sequences without the need for a base configuration. Additionally, due to its modular architecture, novel patterns may be added to account for any novel constraint that is required for a specific domain.

Regarding the method's ability to be incrementally adapted to new data, it is only possible after learning the characteristics of an initial dataset. If the initial dataset contains non-representative data with biased information or missing values, the method will not be able to generate realistic examples. Therefore, it is necessary to further enhance the adaptability of A2PM and improve the transferability of the crafted examples to different networks.

In the future, a major enhancement could be to use the characteristics recorded by the adaptative patterns of A2PM to check for concept drift and perform OOD detection. A new version of the method could be designed to address anomalous samples within each class and to detect changes in the initial data distributions. It could be beneficial to regularly perform OOD detection in deployed ML models, making modern organizations more aware of when their models stop being reliable and need to be retrained.

Another major enhancement could be to also use the adaptative patterns to create an explainable version of A2PM. In addition to robustness, explainability is also a highly desirable property because understanding the behavior of an ML model is essential for an organization to build confidence in it, especially if it is used in a cybersecurity solution. Since the method generates perturbations according to the characteristics of each class, the misclassifications caused by different perturbations could be used to provide insights of the internal reasoning of a model and of the decision boundaries between cyber-attack classes.

It is also pertinent to further contribute to robustness research and to ML security by exploring novel defense strategies throughout different stages of the lifecycle of an intelligent system, to provide a more reliable and robust NID and cyber-attack classification.

6.3 Final Remarks

This thesis presented the research and development work that led to the proposal of a methodology for an adversarial robustness analysis and to the creation of an intelligent method for the generation of realistic adversarial examples in complex tabular data domains like NID. It was aligned with the participation of GECAD in the SeCoIIA and VALU3S projects, and the developed method is starting to be used not only in the NID domain, but across several ongoing projects that tackle the lack of ML robustness in other complex domains.

Overall, this was a very interesting challenge. It provided me the opportunity to acquire a vast amount of knowledge and technical skills related to the AI field, while improving my problem-solving and organizational skills. Furthermore, it enabled me to put my prior knowledge of computer networking and software engineering into practice to develop an innovative solution to help other researchers analyze and improve the robustness of their models.

The key takeaway of this thesis is: ML models can be incredibly valuable to improve a cybersecurity system, but their own vulnerabilities must not be disregarded. It is essential to continue the research efforts to improve the security and trustworthiness of ML and of the intelligent systems that rely on it.

References

[1] European Commission and Directorate-General for Communications Networks Content and Technology, *A path to the digital decade : common governance and coordinated investment for the EU's digital transformation by 2030*. Publications Office of the European Union, 2021. doi: 10.2759/027045.

[2] European Union Agency for Cybersecurity, N. Christoforatos, I. Lella, E. Rekleitis, C. Van Heurck, and A. Zacharis, "Cyber Europe 2022: After Action Report," 2022. doi: 10.2824/397622.

[3] Verizon, "Data Breach Investigations Report 2022." https://www.verizon.com/business/resources/reports/dbir/ (accessed Jan. 13, 2023)

[4] IBM Security and Ponemon Institute, "Cost of a Data Breach Report 2022." https://www.ibm.com/reports/data-breach (accessed Jan. 13, 2023)

[5] S. Mansfield-Devine, "Sophos: The State of Ransomware 2022," *Comput. Fraud Secur.*, vol. 2022, no. 5, 2022, doi: 10.12968/s1361-3723(22)70573-8.

[6] European Union Agency for Cybersecurity *et al.*, "ENISA Threat Landscape 2022," 2022. doi: 10.2824/764318.

[7] European Commission and Directorate-General for Communications Networks Content and Technology, "The Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self assessment," Publications Office of the European Union, 2020. doi: 10.2759/002360.

[8] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014*, 2014, pp. 1–10. doi: 10.48550/ARXIV.1312.6199.

[9] European Union Agency for Cybersecurity, A. Malatras, and G. Dede, "AI Cybersecurity Challenges: Threat Landscape for Artificial Intelligence," 2020. doi: 10.2824/238222.

[10] R. S. Siva Kumar *et al.*, "Adversarial Machine Learning-Industry Perspectives," in *2020 IEEE Security and Privacy Workshops (SPW)*, 2020, pp. 69–75. doi: 10.1109/SPW50608.2020.00028.

[11] European Union Agency for Cybersecurity, A. Malatras, I. Agrafiotis, and M. Adamczyk, "Securing Machine Learning Algorithms," 2022. doi: 10.2824/874249.

[12] "GECAD Research Group." https://www.gecad.isep.ipp.pt/ (accessed Dec. 09, 2022).

[13] "SeCoIIA 871967 Project." doi: 10.3030/871967.

[14] "VALU3S 876852 Project." doi: 10.3030/876852.

[15] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Trans. neural networks Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, 2019, doi: 10.1109/TNNLS.2018.2886017.

[16] N. Martins, J. M. Cruz, T. Cruz, and P. Henriques Abreu, "Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review," *IEEE Access*, vol. 8, pp. 35403–35419, 2020, doi: 10.1109/ACCESS.2020.2974752.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, 2015,

pp. 1–11. doi: 10.48550/ARXIV.1412.6572.
[18] D. Stutz, M. Hein, and B. Schiele, “Disentangling Adversarial Robustness and Generalization,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6969–6980. doi: 10.1109/CVPR.2019.00714.
[19] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, “Backdoor Learning: A Survey,” *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–18, 2022, doi: 10.1109/TNNLS.2022.3182979.
[20] J. Vitorino, N. Oliveira, and I. Praça, “Adaptative Perturbation Patterns: Realistic Adversarial Learning for Robust Intrusion Detection,” *Future Internet*, vol. 14, no. 4, p. 108, 2022, doi: 10.3390/fi14040108.
[21] J. Vitorino, I. Praça, and E. Maia, “Towards Adversarial Realism and Robust Learning for IoT Intrusion Detection and Classification,” *Ann. Telecommun.*, 2023, doi: 10.1007/s12243-023-00953-y.
[22] J. Vitorino, T. Dias, T. Fonseca, E. Maia, and I. Praça, “Constrained Adversarial Learning and its applicability to Automated Software Testing: a systematic review,” *arXiv*, 2023, doi: 10.48550/arXiv.2303.07546.
[23] J. Vitorino, I. Praça, and E. Maia, “SoK: Realistic Adversarial Attacks and Defenses for Intelligent Network Intrusion Detection,” *arXiv*, 2023, doi: 10.48550/arXiv.2308.06819.
[24] J. Vitorino, R. Andrade, I. Praça, O. Sousa, and E. Maia, “A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection,” in *Foundations and Practice of Security*, 2022, pp. 191–207. doi: 10.1007/978-3-031-08147-7_13.
[25] R. Andrade, J. Vitorino, S. Wannous, E. Maia, and I. Praca, “LEMMAS: a secured and trusted Local Energy Market simulation system,” in *2022 18th International Conference on the European Energy Market (EEM)*, 2022, pp. 1–5. doi: 10.1109/EEM54602.2022.9921159.
[26] J. Vitorino, L. Rodrigues, E. Maia, I. Praça, and A. Lourenço, “Adversarial Robustness and Feature Impact Analysis for Driver Drowsiness Detection,” in *Artificial Intelligence in Medicine*, 2023, pp. 108–113. doi: 10.1007/978-3-031-34344-5_13.
[27] D. Moher *et al.*, “Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement,” *Syst. Rev.*, vol. 4, no. 1, p. 1, 2015, doi: 10.1186/2046-4053-4-1.
[28] “Elsevier ScienceDirect Search Source.” https://www.sciencedirect.com/search (accessed Dec. 09, 2022).
[29] “Association for Computing Machinery Digital Library Search Source.” https://dl.acm.org/search/advanced (accessed Dec. 09, 2022).
[30] “Institute of Electrical and Electronics Engineers Xplore Search Source.” https://ieeexplore.ieee.org/search/advanced (accessed Dec. 09, 2022).
[31] “Multidisciplinary Digital Publishing Institute Search Source.” https://www.mdpi.com/search (accessed Dec. 09, 2022).
[32] H. Liu and B. Lang, “Machine learning and deep learning methods for intrusion detection systems: A survey,” *Appl. Sci.*, vol. 9, no. 20, 2019, doi: 10.3390/app9204396.
[33] R. L. Alaoui and E. H. Nfaoui, “Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review,” *Future Internet*, vol. 14, no. 4, 2022, doi: 10.3390/fi14040118.
[34] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, “A review on machine learning–based approaches for Internet traffic classification,” *Ann. Telecommun.*, vol. 75, no. 11, pp. 673–710, 2020, doi: 10.1007/s12243-020-00770-7.
[35] A. Thakkar and R. Lohiya, *A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges*, no. 0123456789. Springer Netherlands, 2020. doi: 10.1007/s11831-020-09496-0.
[36] A. Fawzi, O. Fawzi, and P. Frossard, “Fundamental limits on adversarial robustness,” in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, 2015.
[37] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, “Adversarial Manipulation of Deep Representations,” in *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016*, 2016. doi: 10.48550/ARXIV.1511.05122.
[38] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, 2017, pp. 1–14. doi: 10.48550/ARXIV.1607.02533.

[39] N. Papernot, P. McDaniel, and I. Goodfellow, “Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples,” *arXiv*, 2016, doi: 10.48550/ARXIV.1605.07277.

[40] P. Tabacof and E. Valle, “Exploring the Space of Adversarial Images,” in *2016 IEEE International Joint Conference on Neural Networks (IJCNN)*, 2016. doi: 10.48550/ARXIV.1510.05328.

[41] K. Ren, T. Zheng, Z. Qin, and X. Liu, “Adversarial Attacks and Defenses in Deep Learning,” *Engineering*, vol. 6, no. 3, pp. 346–360, 2020, doi: 10.1016/j.eng.2019.12.012.

[42] J. Zhang and C. Li, “Adversarial Examples: Opportunities and Challenges,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 7, pp. 2578–2593, 2020, doi: 10.1109/TNNLS.2019.2933524.

[43] R. R. Wiyatno, A. Xu, O. Dia, and A. de Berker, “Adversarial Examples in Modern Machine Learning: A Review,” *arXiv*, 2019, doi: 10.48550/ARXIV.1911.05268.

[44] J. Li, Y. Liu, T. Chen, Z. Xiao, Z. Li, and J. Wang, “Adversarial Attacks and Defenses on Cyber-Physical Systems: A Survey,” *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5103–5115, 2020, doi: 10.1109/JIOT.2020.2975654.

[45] K. Eykholt *et al.*, “Robust Physical-World Attacks on Deep Learning Visual Classification,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634. doi: 10.1109/CVPR.2018.00175.

[46] T. Brown, D. Mane, A. Roy, M. Abadi, and J. Gilmer, “Adversarial Patch,” in *Advances in Neural Information Processing Systems*, 2017. doi: 10.48550/ARXIV.1712.09665.

[47] D. Edwards and D. B. Rawat, “Study of Adversarial Machine Learning with Infrared Examples for Surveillance Applications,” *Electronics*, vol. 9, no. 8, 2020, doi: 10.3390/electronics9081284.

[48] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, “Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain,” *ACM Comput. Surv.*, vol. 54, no. 5, 2021, doi: 10.1145/3453158.

[49] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal Adversarial Perturbations,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 86–94. doi: 10.1109/CVPR.2017.17.

[50] P. Papadopoulos, O. Thornewill von Essen, N. Pitropakis, C. Chrysoulas, A. Mylonas, and W. J. Buchanan, “Launching Adversarial Attacks against Network Intrusion Detection Systems for IoT,” *J. Cybersecurity Priv.*, vol. 1, no. 2, pp. 252–273, 2021, doi: 10.3390/jcp1020014.

[51] M. J. Hashemi, G. Cusack, and E. Keller, “Towards Evaluation of NIDSs in Adversarial Setting,” in *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, 2019, pp. 14–21. doi: 10.1145/3359992.3366642.

[52] M. A. Merzouk, F. Cuppens, N. Boulahia-Cuppens, and R. Yaich, “Investigating the practicality of adversarial evasion attacks on network intrusion detection,” *Ann. Telecommun.*, 2022, doi: 10.1007/s12243-022-00910-1.

[53] X. Peng, W. Huang, and Z. Shi, “Adversarial Attack Against DoS Intrusion Detection: An Improved Boundary-Based Method,” in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 1288–1295. doi: 10.1109/ICTAI.2019.00179.

[54] A. McCarthy, P. Andriotis, E. Ghadafi, and P. Legg, “Feature Vulnerability and Robustness Assessment against Adversarial Machine Learning Attacks,” in *Proceedings of the 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 2021, pp. 1–8. doi: 10.1109/CyberSA52016.2021.9478199.

[55] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, “Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems,” *Digit. Threat. Res. Pract.*, vol. 1, no. 1, 2021, doi: 10.1145/3469659.

[56] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, “A taxonomy and survey of attacks against machine learning,” *Comput. Sci. Rev.*, vol. 34, p. 100199, 2019, doi: 10.1016/j.cosrev.2019.100199.

[57] T. Gu, B. Dolan-Gavitt, and S. Garg, “BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain,” *arXiv*, 2017, doi: 10.48550/ARXIV.1708.06733.

[58] B. Wang *et al.*, “Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks,” in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 707–723. doi: 10.1109/SP.2019.00031.

[59] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted Backdoor Attacks on Deep Learning Systems

Using Data Poisoning," *arXiv*, 2017, doi: 10.48550/ARXIV.1712.05526.
[60] K. He, D. D. Kim, and M. R. Asghar, "Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 25, no. 1, pp. 538–566, 2023, doi: 10.1109/COMST.2022.3233793.
[61] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18. doi: 10.1109/SP.2017.41.
[62] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333. doi: 10.1145/2810103.2813677.
[63] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of Artificial Intelligence Adversarial Attack and Defense Technologies," *Appl. Sci.*, vol. 9, no. 5, 2019, doi: 10.3390/app9050909.
[64] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: model inversion attacks and data protection law," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 376, no. 2133, p. 20180083, 2018, doi: 10.1098/rsta.2018.0083.
[65] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning," *arXiv*, 2017, doi: 10.48550/ARXIV.1702.07464.
[66] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating Adversarial Evasion Attacks in the Context of Wireless Communications," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1102–1113, 2020, doi: 10.1109/TIFS.2019.2934069.
[67] J. Aiken and S. Scott-Hayward, "Investigating Adversarial Attacks against Network Intrusion Detection Systems in SDNs," in *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2019, pp. 1–7. doi: 10.1109/NFV-SDN47374.2019.9040101.
[68] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing Adversarial Attacks against Deep Learning for Intrusion Detection in IoT Networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9014337.
[69] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Trans. Intell. Technol.*, vol. 6, no. 1, pp. 25–45, 2021, doi: 10.1049/cit2.12028.
[70] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box Adversarial Attacks with Limited Queries and Information," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018, vol. 80, pp. 2137–2146. Available: https://proceedings.mlr.press/v80/ilyas18a.html
[71] M. C. Belavagi and B. Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," *Procedia Comput. Sci.*, vol. 89, pp. 117–123, 2016, doi: 10.1016/j.procs.2016.06.016.
[72] M. Pujari, Y. Pacheco, B. Cherukuri, and W. Sun, "A Comparative Study on the Impact of Adversarial Machine Learning Attacks on Contemporary Intrusion Detection Datasets," *SN Comput. Sci.*, vol. 3, no. 5, p. 412, 2022, doi: 10.1007/s42979-022-01321-8.
[73] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *Proceedings - IEEE Symposium on Security and Privacy*, 2017, pp. 39–57. doi: 10.1109/SP.2017.49.
[74] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, doi: 10.1109/CVPR.2016.282.
[75] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, "Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9310–9319, 2022, doi: 10.1109/JIOT.2021.3130434.
[76] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling Deep Structured Visual and Speech Recognition Models with Adversarial Examples," in *Advances in Neural Information Processing Systems*, 2017, pp. 6980–6990. Available: https://proceedings.neurips.cc/paper/2017/file/d494020ff8ec181ef98ed97ac3f25453-Paper.pdf
[77] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," in *2016 IEEE European Symposium on Security and Privacy*, 2016, pp. 372–387. doi: 10.1109/EuroSP.2016.36.
[78] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models

resistant to adversarial attacks," in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, 2018, pp. 1–28. doi: 10.48550/ARXIV.1706.06083.

[79] K. Xu *et al.*, "Structured adversarial attack: Towards general implementation and better interpretability," *Proc. 7th Int. Conf. Learn. Represent. ICLR 2019*, 2019.

[80] Z. Wang, M. Gao, J. Li, J. Zhang, and J. Zhong, "Gray-Box Shilling Attack: An Adversarial Learning Approach," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 5, 2022, doi: 10.1145/3512352.

[81] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection," in *Advances in Knowledge Discovery and Data Mining*, 2022, pp. 79–91.

[82] R. Chauhan, U. Sabeel, A. Izaddoost, and S. Shah Heydari, "Polymorphic Adversarial Cyberattacks Using WGAN," *J. Cybersecurity Priv.*, vol. 1, no. 4, pp. 767–792, 2021, doi: 10.3390/jcp1040037.

[83] S. Zhang, X. Xie, and Y. Xu, "A Brute-Force Black-Box Method to Attack Machine Learning-Based Systems in Cybersecurity," *IEEE Access*, vol. 8, pp. 128250–128263, 2020, doi: 10.1109/ACCESS.2020.3008433.

[84] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Black-box adversarial sample generation based on differential evolution," *J. Syst. Softw.*, vol. 170, 2020, doi: 10.1016/j.jss.2020.110767.

[85] J. Su, D. V. Vargas, and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, 2019, doi: 10.1109/TEVC.2019.2890858.

[86] H. Dai *et al.*, "Adversarial Attack on Graph Structured Data," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018, vol. 80, pp. 1115–1124. Available: https://proceedings.mlr.press/v80/dai18b.html

[87] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, 2017, vol. 70, pp. 214–223. Available: https://proceedings.mlr.press/v70/arjovsky17a.html

[88] P. Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C. J. Hsieh, "ZOO: Zeroth order optimization based black-box atacks to deep neural networks without training substitute models," in *AISec 2017 - Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, co-located with CCS 2017*, 2017, pp. 15–26. doi: 10.1145/3128572.3140448.

[89] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, 2018, pp. 1–12. doi: 10.48550/ARXIV.1712.04248.

[90] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv*, 2014, doi: 10.48550/ARXIV.1411.1784.

[91] K. Sohn, H. Lee, and X. Yan, "Learning Structured Output Representation using Deep Conditional Generative Models," in *Advances in Neural Information Processing Systems*, 2015. Available: https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf

[92] I. Rosenberg, A. Shabtai, L. Rokach, and Y. Elovici, "Generic Black-Box End-to-End Attack Against State of the Art API Call Based Malware Classifiers," in *Research in Attacks, Intrusions, and Defenses*, 2018, pp. 490–510. doi: 10.1007/978-3-030-00470-5_23.

[93] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A Query-Efficient Decision-Based Attack," in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 1277–1294. doi: 10.1109/SP40000.2020.00045.

[94] M. Cheng, H. Zhang, C. J. Hsieh, T. Le, P. Y. Chen, and J. Yi, "Query-efficient hard-label black-box attack: An optimization-based approach," in *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019*, 2019, pp. 1–12.

[95] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, 2014, vol. 27. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[96] "MITRE Adversarial Threat Landscape for Artificial-Intelligence Systems." https://atlas.mitre.org/ (accessed Dec. 09, 2022).

[97] B. Biggio, G. Fumera, and F. Roli, "Security Evaluation of Pattern Classifiers under Attack," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 984–996, 2014, doi: 10.1109/TKDE.2013.57.

[98] L. Smith and Y. Gal, "Understanding Measures of Uncertainty for Adversarial Example Detection," in *34th Conference on Uncertainty in Artificial Intelligence, UAI 2018 - Conference Track Proceedings*, 2018. doi: 10.48550/ARXIV.1803.08533.

[99] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks,” in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 582–597. doi: 10.1109/SP.2016.41.

[100] L. Schmidt, K. Talwar, S. Santurkar, D. Tsipras, and A. Madry, “Adversarially robust generalization requires more data,” *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, no. NeurIPS, 2018.

[101] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent Advances in Adversarial Training for Adversarial Robustness,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, {IJCAI-21}*, 2021, pp. 4312–4321. doi: 10.24963/ijcai.2021/591.

[102] R. A. Khamis, M. O. Shafiq, and A. Matrawy, “Investigating Resistance of Deep Learning-based IDS against Adversaries using min-max Optimization,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7. doi: 10.1109/ICC40277.2020.9149117.

[103] D. J. Miller, Z. Xiang, and G. Kesidis, “Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses Against Attacks,” in *Proceedings of the IEEE*, 2020, vol. 108, no. 3, pp. 402–433. doi: 10.1109/JPROC.2020.2970615.

[104] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, “Detecting Adversarial Samples from Artifacts,” in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, 2017. doi: 10.48550/ARXIV.1703.00410.

[105] Y. Ganin *et al.*, “Domain-adversarial training of neural networks,” *Adv. Comput. Vis. Pattern Recognit.*, no. 9783319583464, pp. 189–209, 2017, doi: 10.1007/978-3-319-58347-1_10.

[106] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “Ensemble adversarial training: Attacks and defenses,” in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, 2018, pp. 1–22. doi: 10.48550/ARXIV.1705.07204.

[107] E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, “Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems,” *J. Inf. Secur. Appl.*, vol. 58, no. February, p. 102717, 2021, doi: 10.1016/j.jisa.2020.102717.

[108] G. Apruzzese, M. Andreolini, M. Colajanni, and M. Marchetti, “Hardening Random Forest Cyber Detectors Against Adversarial Attacks,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 4, pp. 427–439, 2020, doi: 10.1109/TETCI.2019.2961157.

[109] A. Kantchelian, J. D. Tygar, and A. D. Joseph, “Evasion and hardening of tree ensemble classifiers,” in *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, 2016, vol. 5, pp. 3562–3573.

[110] Y. Chen, S. Wang, W. Jiang, A. Cidon, and S. Jana, “Cost-aware robust tree ensembles for security applications,” in *Proceedings of the 30th USENIX Security Symposium*, 2021, pp. 2291–2308.

[111] H. Chen, H. Zhang, D. Boning, and C. J. Hsieh, “Robust decision trees against adversarial examples,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, 2019. doi: 10.48550/ARXIV.1902.10660.

[112] D. Vos and S. Verwer, “Efficient Training of Robust Decision Trees Against Adversarial Examples,” in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, 2021, vol. 139, pp. 10586–10595. Available: https://proceedings.mlr.press/v139/vos21a.html

[113] A. Shafahi *et al.*, “Adversarial training for free!,” in *Advances in Neural Information Processing Systems*, 2019, vol. 32. Available: https://proceedings.neurips.cc/paper/2019/file/7503cfacd12053d309b6bed5c89de212-Paper.pdf

[114] M. Andriushchenko and N. Flammarion, “Understanding and Improving Fast Adversarial Training,” in *Advances in Neural Information Processing Systems*, 2020. Available: https://proceedings.neurips.cc/paper/2020/file/b8ce47761ed7b3b6f48b583350b7f9e4-Paper.pdf

[115] X. Fu, N. Zhou, L. Jiao, H. Li, and J. Zhang, “The robust deep learning–based schemes for intrusion detection in Internet of Things environments,” *Ann. Telecommun.*, vol. 76, no. 5, pp. 273–285, 2021, doi: 10.1007/s12243-021-00854-y.

[116] W. Zhao, S. Alwidian, and Q. H. Mahmoud, “Adversarial Training Methods for Deep Learning: A Systematic Review,” *Algorithms*, vol. 15, no. 8, 2022, doi: 10.3390/a15080283.

[117] A. Shafahi, M. Najibi, Z. Xu, J. Dickerson, L. S. Davis, and T. Goldstein, “Universal Adversarial Training,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, no. 04, pp. 5636–5643. doi: 10.1609/aaai.v34i04.6017.

[118] L. Cai and Y. Zhu, “The Challenges of Data Quality and Data Quality Assessment in the Big Data

Era," *Data Sci. J.*, vol. 14, p. 2, 2015, doi: 10.5334/dsj-2015-002.
[119] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014*, 2014, pp. 1–14. doi: 10.48550/ARXIV.1312.6114.
[120] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in Network Intrusion Detection Systems," *Expert Syst. Appl.*, vol. 186, p. 115782, 2021, doi: 10.1016/j.eswa.2021.115782.
[121] X. Gao, R. K. Saha, M. R. Prasad, and A. Roychoudhury, "Fuzz Testing Based Data Augmentation to Improve Robustness of Deep Neural Networks," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 1147–1158. doi: 10.1145/3377811.3380415.
[122] M. Fuentes Reyes, S. Auer, N. Merkle, C. Henry, and M. Schmitt, "SAR-to-Optical Image Translation Based on Conditional Generative Adversarial Networks—Optimization, Opportunities and Limits," *Remote Sens.*, vol. 11, no. 17, 2019, doi: 10.3390/rs11172067.
[123] Z. Chen, L. Tong, B. Qian, J. Yu, and C. Xiao, "Self-Attention-Based Conditional Variational Auto-Encoder Generative Adversarial Networks for Hyperspectral Classification," *Remote Sens.*, vol. 13, no. 16, 2021, doi: 10.3390/rs13163316.
[124] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, "CVA2E: A Conditional Variational Autoencoder With an Adversarial Training Process for Hyperspectral Imagery Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5676–5692, Aug. 2020, doi: 10.1109/TGRS.2020.2968304.
[125] C. J. Costa, S. Tiwari, K. Bhagat, A. Verlekar, K. M. C. Kumar, and S. Aswale, "Three-Dimensional Reconstruction of Satellite images using Generative Adversarial Networks," in *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, 2021, pp. 121–126. doi: 10.1109/ICTAI53825.2021.9673457.
[126] H. Mizuochi, Y. Iijima, H. Nagano, A. Kotani, and T. Hiyama, "Dynamic Mapping of Subarctic Surface Water by Fusion of Microwave and Optical Satellite Data Using Conditional Adversarial Networks," *Remote Sens.*, vol. 13, no. 2, 2021, doi: 10.3390/rs13020175.
[127] Y. Kim and S. Hong, "Deep Learning-Generated Nighttime Reflectance and Daytime Radiance of the Midwave Infrared Band of a Geostationary Satellite," *Remote Sens.*, vol. 11, no. 22, 2019, doi: 10.3390/rs11222713.
[128] C.-I. Cira, M. Kada, M.-Á. Manso-Callejo, R. Alcarria, and B. Bordel Sanchez, "Improving Road Surface Area Extraction via Semantic Segmentation with Conditional Generative Learning for Deep Inpainting Operations," *ISPRS Int. J. Geo-Information*, vol. 11, no. 1, 2022, doi: 10.3390/ijgi11010043.
[129] W. Han *et al.*, "Sample generation based on a supervised Wasserstein Generative Adversarial Network for high-resolution remote-sensing scene classification," *Inf. Sci. (Ny).*, vol. 539, pp. 177–194, 2020, doi: 10.1016/j.ins.2020.06.018.
[130] N. Hayatbini *et al.*, "Conditional Generative Adversarial Networks (cGANs) for Near Real-Time Precipitation Estimation from Multispectral GOES-16 Satellite Imageries—PERSIANN-cGAN," *Remote Sens.*, vol. 11, no. 19, 2019, doi: 10.3390/rs11192193.
[131] Y. Zhang, H. Sun, J. Zuo, H. Wang, G. Xu, and X. Sun, "Aircraft Type Recognition in Remote Sensing Images Based on Feature Learning with Conditional Generative Adversarial Networks," *Remote Sens.*, vol. 10, no. 7, 2018, doi: 10.3390/rs10071123.
[132] X. Pan, J. Zhao, and J. Xu, "Conditional Generative Adversarial Network-Based Training Sample Set Improvement Model for the Semantic Segmentation of High-Resolution Remote Sensing Images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7854–7870, 2021, doi: 10.1109/TGRS.2020.3033816.
[133] H. S. Munawar, F. Ullah, A. Heravi, M. J. Thaheem, and A. Maqsoom, "Inspecting Buildings Using Drones and Computer Vision: A Machine Learning Approach to Detect Cracks and Damages," *Drones*, vol. 6, no. 1, 2022, doi: 10.3390/drones6010005.
[134] W. Chen, Y. Li, and Z. Zhao, "InsulatorGAN: A Transmission Line Insulator Detection Model Using Multi-Granularity Conditional Generative Adversarial Nets for UAV Inspection," *Remote Sens.*, vol. 13, no. 19, 2021, doi: 10.3390/rs13193971.
[135] H. Cheng, W. Liao, M. Y. Yang, B. Rosenhahn, and M. Sester, "AMENet: Attentive Maps Encoder Network for trajectory prediction," *ISPRS J. Photogramm. Remote Sens.*, vol. 172, pp. 253–266,

2021, doi: 10.1016/j.isprsjprs.2020.12.004.
[136] J. Zhang, M. Zhu, and L. Peng, “Customized Parking Data Generation based on Multi-conditional GAN,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6. doi: 10.1109/ITSC45102.2020.9294436.
[137] J. Li, B. Zhao, K. Wu, Z. Dong, X. Zhang, and Z. Zheng, “A Representation Generation Approach of Transmission Gear Based on Conditional Generative Adversarial Network,” *Actuators*, vol. 10, no. 5, 2021, doi: 10.3390/act10050086.
[138] A. Jaafer, G. Nilsson, and G. Como, “Data Augmentation of IMU Signals and Evaluation via a Semi-Supervised Classification of Driving Behavior,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6. doi: 10.1109/ITSC45102.2020.9294496.
[139] M. Pöpperli, R. Gulagundi, S. Yogamani, and S. Milz, “Realistic Ultrasonic Environment Simulation Using Conditional Generative Adversarial Networks,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2278–2283. doi: 10.1109/IVS.2019.8814091.
[140] F. Falahatraftar, S. Pierre, and S. Chamberland, “A Conditional Generative Adversarial Network Based Approach for Network Slicing in Heterogeneous Vehicular Networks,” *Telecom*, vol. 2, no. 1, pp. 141–154, 2021, doi: 10.3390/telecom2010009.
[141] F. Kocayusufoglu, A. Silva, and A. K. Singh, “FlowGEN: A Generative Model for Flow Graphs,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 813–823. doi: 10.1145/3534678.3539406.
[142] R. Qin and J. Zhao, “High-Efficiency Generative Adversarial Network Model for Chemical Process Fault Diagnosis,” *IFAC-PapersOnLine*, vol. 55, no. 7, pp. 732–737, 2022, doi: 10.1016/j.ifacol.2022.07.531.
[143] Q.-X. Zhu, K.-R. Hou, Z.-S. Chen, Y. Xu, and Y.-L. He, “Research and Application of Virtual Sample Generation Method Based on Conditional Generative Adversarial Network,” in *2021 China Automation Congress (CAC)*, 2021, pp. 351–355. doi: 10.1109/CAC53003.2021.9728144.
[144] Y.-L. He, X.-Y. Li, J.-H. Ma, S. Lu, and Q.-X. Zhu, “A novel virtual sample generation method based on a modified conditional Wasserstein GAN to address the small sample size problem in soft sensing,” *J. Process Control*, vol. 113, pp. 18–28, 2022, doi: 10.1016/j.jprocont.2022.03.008.
[145] L. Zhang *et al.*, “Improved 1-km-Resolution Hourly Estimates of Aerosol Optical Depth Using Conditional Generative Adversarial Networks,” *Remote Sens.*, vol. 13, no. 19, 2021, doi: 10.3390/rs13193834.
[146] A. Chernikova and A. Oprea, “FENCE: Feasible Evasion Attacks on Neural Networks in Constrained Environments,” *ACM Trans. Priv. Secur.*, vol. 25, no. 4, 2022, doi: 10.1145/3544746.
[147] A. S. Dina, A. B. Siddique, and D. Manivannan, “Effect of Balancing Data Using Synthetic Data on the Performance of Machine Learning Classifiers for Intrusion Detection in Computer Networks,” *IEEE Access*, vol. 10, pp. 96731–96747, 2022, doi: 10.1109/ACCESS.2022.3205337.
[148] J. Wang, X. Yan, L. Liu, L. Li, and Y. Yu, “CTTGAN: Traffic Data Synthesizing Scheme Based on Conditional GAN,” *Sensors*, vol. 22, no. 14, 2022, doi: 10.3390/s22145243.
[149] Q. Liu, G. Yu, Y. Wang, and Z. Yi, “A Novel DGA Domain Adversarial Sample Generation Method By Geometric Perturbation,” in *2021 3rd International Conference on Advanced Information Science and System (AISS 2021)*, 2021. doi: 10.1145/3503047.3503080.
[150] M. Wan, H. Yao, and X. Yan, “Generation of malicious webpage samples based on GAN,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 864–869. doi: 10.1109/TrustCom50675.2020.00116.
[151] S. Kasarapu, S. Shukla, R. Hassan, A. Sasan, H. Homayoun, and S. M. PD, “CAD-FSL: Code-Aware Data Generation Based Few-Shot Learning for Efficient Malware Detection,” in *Proceedings of the Great Lakes Symposium on VLSI 2022*, 2022, pp. 507–512. doi: 10.1145/3526241.3530825.
[152] X. Guo, H. Okamura, and T. Dohi, “Automated Software Test Data Generation With Generative Adversarial Networks,” *IEEE Access*, vol. 10, pp. 20690–20700, 2022, doi: 10.1109/ACCESS.2022.3153347.
[153] P. Chonwiharnphan, P. Thienprapasith, and E. Chuangsuwanich, “Generating Realistic Users Using Generative Adversarial Network With Recommendation-Based Embedding,” *IEEE Access*, vol. 8, pp. 41384–41393, 2020, doi: 10.1109/ACCESS.2020.2976491.
[154] A. Esmaili and S. Farzi, “Effective synthetic data generation for fake user detection,” in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, 2021, pp. 1–5. doi:

10.1109/CSICC52343.2021.9420570.
[155] L. Chen, Y. Liu, W. Xiao, Y. Wang, and H. Xie, “SpeakerGAN: Speaker identification with conditional generative adversarial network,” *Neurocomputing*, vol. 418, pp. 211–220, 2020, doi: 10.1016/j.neucom.2020.08.040.
[156] Y.-Y. Ding, H.-J. Lin, L.-J. Liu, Z.-H. Ling, and Y. Hu, “Robustness of Speech Spoofing Detectors Against Adversarial Post-Processing of Voice Conversion,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3415–3426, 2021, doi: 10.1109/TASLP.2021.3124420.
[157] Y. Qian, H. Hu, and T. Tan, “Data augmentation using generative adversarial networks for robust speech recognition,” *Speech Commun.*, vol. 114, pp. 1–9, 2019, doi: 10.1016/j.specom.2019.08.006.
[158] S. R. Ram, V. K. M, B. Subramanian, N. Bacanin, M. Zivkovic, and I. Strumberger, “Speech enhancement through improvised conditional generative adversarial networks,” *Microprocess. Microsyst.*, vol. 79, p. 103281, 2020, doi: 10.1016/j.micpro.2020.103281.
[159] A. S. Khwaja, A. Anpalagan, and B. Venkatesh, “Smart Meter Data Masking Using Conditional Generative Adversarial Networks,” *Electr. Power Syst. Res.*, vol. 209, p. 108033, 2022, doi: 10.1016/j.epsr.2022.108033.
[160] Z. Liu, L. Meng, Y. Tan, J. Zhang, and H. Zhang, “Image compression based on octave convolution and semantic segmentation,” *Knowledge-Based Syst.*, vol. 228, p. 107254, 2021, doi: 10.1016/j.knosys.2021.107254.
[161] J. Yoon, L. N. Drumright, and M. van der Schaar, “Anonymization Through Data Synthesis Using Generative Adversarial Networks (ADS-GAN),” *IEEE J. Biomed. Heal. Informatics*, vol. 24, no. 8, pp. 2378–2388, 2020, doi: 10.1109/JBHI.2020.2980262.
[162] S. Athey, G. W. Imbens, J. Metzger, and E. Munro, “Using Wasserstein Generative Adversarial Networks for the design of Monte Carlo simulations,” *J. Econom.*, 2021, doi: 10.1016/j.jeconom.2020.09.013.
[163] D. Fang, X. Guan, B. Hu, Y. Peng, M. Chen, and K. Hwang, “Deep Reinforcement Learning for Scenario-Based Robust Economic Dispatch Strategy in Internet of Energy,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9654–9663, 2021, doi: 10.1109/JIOT.2020.3040294.
[164] J. Li, Y. Yang, J. S. Sun, K. Tomsovic, and H. Qi, “ConAML: Constrained Adversarial Machine Learning for Cyber-Physical Systems,” in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 52–66. doi: 10.1145/3433210.3437513.
[165] X. Gong, B. Tang, R. Zhu, W. Liao, and L. Song, “Data Augmentation for Electricity Theft Detection Using Conditional Variational Auto-Encoder,” *Energies*, vol. 13, no. 17, 2020, doi: 10.3390/en13174291.
[166] H. Liu, Z. Li, and Y. Li, “Noise Reduction Power Stealing Detection Model Based on Self-Balanced Data Set,” *Energies*, vol. 13, no. 7, 2020, doi: 10.3390/en13071763.
[167] J. Moon, S. Jung, S. Park, and E. Hwang, “Conditional Tabular GAN-Based Two-Stage Data Generation Scheme for Short-Term Load Forecasting,” *IEEE Access*, vol. 8, pp. 205327–205339, 2020, doi: 10.1109/ACCESS.2020.3037063.
[168] Y. Zhang, X. Deng, Y. Zhang, and Y. Zhang, “Generation of sub-item load profiles for public buildings based on the conditional generative adversarial network and moving average method,” *Energy Build.*, vol. 268, p. 112185, 2022, doi: 10.1016/j.enbuild.2022.112185.
[169] P. Qin, X. Wang, Z. Qiao, X. Li, Q. Hu, and W. Shu, “GAN-based Residential Load Data Generation Model Considering Users’ Privacy,” in *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE)*, 2022, pp. 838–843. doi: 10.1109/ACPEE53904.2022.9783676.
[170] S. Zhang, Z. Ji, J. Zhang, Y. Bao, and W. Wang, “Multi-dimensional Data Generation Method of Electric Vehicle Charging Behaviors Based on Improved Generative Adversarial Network,” in *2022 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia)*, 2022, pp. 1827–1832. doi: 10.1109/ICPSAsia55496.2022.9949855.
[171] D. Chang *et al.*, “Seismic Data Interpolation Using Dual-Domain Conditional Generative Adversarial Networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 10, pp. 1856–1860, 2021, doi: 10.1109/LGRS.2020.3008478.
[172] É. Guérin *et al.*, “Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks,” *ACM Trans. Graph.*, vol. 36, no. 6, 2017, doi: 10.1145/3130800.3130804.
[173] J. Zhang, C. Li, P. Zhou, C. Wang, G. He, and H. Qin, “Authoring multi-style terrain with global-to-

local control," *Graph. Models*, vol. 119, p. 101122, 2022, doi: 10.1016/j.gmod.2021.101122.
[174] S. Amirrajab, Y. Al Khalil, C. Lorenz, J. Weese, J. Pluim, and M. Breeuwer, "Label-informed cardiac magnetic resonance image synthesis through conditional generative adversarial networks," *Comput. Med. Imaging Graph.*, vol. 101, p. 102123, 2022, doi: 10.1016/j.compmedimag.2022.102123.
[175] N. Qiang *et al.*, "Learning brain representation using recurrent Wasserstein generative adversarial net," *Comput. Methods Programs Biomed.*, vol. 223, p. 106979, 2022, doi: 10.1016/j.cmpb.2022.106979.
[176] J. F. Teixeira, M. Dias, E. Batista, J. Costa, L. F. Teixeira, and H. P. Oliveira, "Adversarial Data Augmentation on Breast MRI Segmentation," *Appl. Sci.*, vol. 11, no. 10, 2021, doi: 10.3390/app11104554.
[177] M. Liu, W. Zou, W. Wang, C.-B. Jin, J. Chen, and C. Piao, "Multi-Conditional Constraint Generative Adversarial Network-Based MR Imaging from CT Scan Data," *Sensors*, vol. 22, no. 11, 2022, doi: 10.3390/s22114043.
[178] G. Silva, I. Domingues, H. Duarte, and J. A. M. Santos, "Automatic Generation of Lymphoma Post-Treatment PETs using Conditional-GANs," in *2019 Digital Image Computing: Techniques and Applications (DICTA)*, 2019, pp. 1–6. doi: 10.1109/DICTA47822.2019.8945835.
[179] X. Zhou, X. Zhu, K. Nakamura, and M. Noro, "Electrocardiogram Quality Assessment with a Generalized Deep Learning Model Assisted by Conditional Generative Adversarial Networks," *Life*, vol. 11, no. 10, 2021, doi: 10.3390/life11101013.
[180] S. E. Karabulut, M. M. Khorasani, and A. Pantanowitz, "Neurocartographer: CC-WGAN Based SSVEP Data Generation to Produce a Model toward Symmetrical Behaviour to the Human Brain," *Symmetry (Basel).*, vol. 14, no. 8, 2022, doi: 10.3390/sym14081600.
[181] J. L. Hagad, T. Kimura, K. Fukui, and M. Numao, "Learning Subject-Generalized Topographical EEG Embeddings Using Deep Variational Autoencoders and Domain-Adversarial Regularization," *Sensors*, vol. 21, no. 5, 2021, doi: 10.3390/s21051792.
[182] M. Ehrhart, B. Resch, C. Havas, and D. Niederseer, "A Conditional GAN for Generating Time Series Data for Stress Detection in Wearable Physiological Sensor Data," *Sensors*, vol. 22, no. 16, 2022, doi: 10.3390/s22165969.
[183] F. Zhang, Y. Zhang, X. Zhu, X. Chen, H. Du, and X. Zhang, "PregGAN: A prognosis prediction model for breast cancer based on conditional generative adversarial networks," *Comput. Methods Programs Biomed.*, vol. 224, p. 107026, 2022, doi: 10.1016/j.cmpb.2022.107026.
[184] N. Saffari *et al.*, "Fully Automated Breast Density Segmentation and Classification Using Deep Learning," *Diagnostics*, vol. 10, no. 11, 2020, doi: 10.3390/diagnostics10110988.
[185] A. Gür, "Deep Feature Synthesis for Accurate Breast Cancer Prediction," in *2022 Medical Technologies Congress (TIPTEKNO)*, 2022, pp. 1–4. doi: 10.1109/TIPTEKNO56568.2022.9960237.
[186] M. Havaei, X. Mao, Y. Wang, and Q. Lao, "Conditional generation of medical images via disentangled adversarial inference," *Med. Image Anal.*, vol. 72, p. 102106, 2021, doi: 10.1016/j.media.2021.102106.
[187] O. Bailo, D. Ham, and Y. M. Shin, "Red Blood Cell Image Generation for Data Augmentation Using Conditional Generative Adversarial Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1039–1048. doi: 10.1109/CVPRW.2019.00136.
[188] K. T. Chui, M. D. Lytras, and P. Vasant, "Combined Generative Adversarial Network and Fuzzy C-Means Clustering for Multi-Class Voice Disorder Detection with an Imbalanced Dataset," *Appl. Sci.*, vol. 10, no. 13, 2020, doi: 10.3390/app10134571.
[189] K. Chen, D. Zhu, J. Lu, and Y. Luo, "An Adversarial and Densely Dilated Network for Connectomes Segmentation," *Symmetry (Basel).*, vol. 10, no. 10, 2018, doi: 10.3390/sym10100467.
[190] M. Ahang, M. Jalayer, A. Shojaeinasab, O. Ogunfowora, T. Charter, and H. Najjaran, "Synthesizing Rolling Bearing Fault Samples in New Conditions: A Framework Based on a Modified CGAN," *Sensors*, vol. 22, no. 14, 2022, doi: 10.3390/s22145413.
[191] Y. Li, W. Zou, and L. Jiang, "Fault diagnosis of rotating machinery based on combination of Wasserstein generative adversarial networks and long short term memory fully convolutional network," *Measurement*, vol. 191, p. 110826, 2022, doi: 10.1016/j.measurement.2022.110826.
[192] Y. Liu, H. Jiang, Y. Wang, Z. Wu, and S. Liu, "A conditional variational autoencoding generative

adversarial networks with self-modulation for rolling bearing fault diagnosis," *Measurement*, vol. 192, p. 110888, 2022, doi: 10.1016/j.measurement.2022.110888.

[193] Y. Peng, Y. Wang, and Y. Shao, "A novel bearing imbalance Fault-diagnosis method based on a Wasserstein conditional generative adversarial network," *Measurement*, vol. 192, p. 110924, 2022, doi: 10.1016/j.measurement.2022.110924.

[194] S. Dixit, N. K. Verma, and A. K. Ghosh, "Intelligent Fault Diagnosis of Rotary Machines: Conditional Auxiliary Classifier GAN Coupled With Meta Learning Using Limited Data," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021, doi: 10.1109/TIM.2021.3082264.

[195] Y. Yu, B. Tang, R. Lin, S. Han, T. Tang, and M. Chen, "CWGAN: Conditional Wasserstein Generative Adversarial Nets for Fault Data Generation," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2019, pp. 2713–2718. doi: 10.1109/ROBIO49542.2019.8961501.

[196] C.-H. Chen, C.-K. Tsung, and S.-S. Yu, "Designing a Hybrid Equipment-Failure Diagnosis Mechanism under Mixed-Type Data with Limited Failure Samples," *Appl. Sci.*, vol. 12, no. 18, 2022, doi: 10.3390/app12189286.

[197] X. Guo *et al.*, "Damage Detection for Conveyor Belt Surface Based on Conditional Cycle Generative Adversarial Network," *Sensors*, vol. 22, no. 9, 2022, doi: 10.3390/s22093485.

[198] X. Wang and H. Liu, "Data supplement for a soft sensor using a new generative model based on a variational autoencoder and Wasserstein GAN," *J. Process Control*, vol. 85, pp. 91–99, 2020, doi: 10.1016/j.jprocont.2019.11.004.

[199] X. Liu, H. Ma, and Y. Liu, "A Novel Transfer Learning Method Based on Conditional Variational Generative Adversarial Networks for Fault Diagnosis of Wind Turbine Gearboxes under Variable Working Conditions," *Sustainability*, vol. 14, no. 9, 2022, doi: 10.3390/su14095441.

[200] Y. Wang, G. Sun, and Q. Jin, "Imbalanced sample fault diagnosis of rotating machinery using conditional variational auto-encoder generative adversarial network," *Appl. Soft Comput.*, vol. 92, p. 106333, 2020, doi: 10.1016/j.asoc.2020.106333.

[201] L. Zhang, H. Zhang, and G. Cai, "The Multiclass Fault Diagnosis of Wind Turbine Bearing Based on Multisource Signal Fusion and Deep Learning Generative Model," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022, doi: 10.1109/TIM.2022.3178483.

[202] S. Li and Y. Sung, "INCO-GAN: Variable-Length Music Generation Method Based on Inception Model-Based Conditional GAN," *Mathematics*, vol. 9, no. 4, 2021, doi: 10.3390/math9040387.

[203] K. N. Haque, R. Rana, and B. W. Schuller, "High-Fidelity Audio Generation and Representation Learning With Guided Adversarial Autoencoder," *IEEE Access*, vol. 8, pp. 223509–223528, 2020, doi: 10.1109/ACCESS.2020.3040797.

[204] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep Cross-Modal Audio-Visual Generation," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017, pp. 349–357. doi: 10.1145/3126686.3126723.

[205] Z. Geng, D. Johnson, and R. Fedkiw, "Coercing machine learning to output physically accurate results," *J. Comput. Phys.*, vol. 406, p. 109099, 2020, doi: 10.1016/j.jcp.2019.109099.

[206] E. Barsoum, J. Kender, and Z. Liu, "HP-GAN: Probabilistic 3D Human Motion Prediction via GAN," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1499–149909. doi: 10.1109/CVPRW.2018.00191.

[207] X. Chen, Y. Sun, X. Shu, and Q. Li, "Attention-aware conditional generative adversarial networks for facial age synthesis," *Neurocomputing*, vol. 451, pp. 167–180, 2021, doi: 10.1016/j.neucom.2021.04.068.

[208] Y. Li, T. Zhang, L. Duan, and C. Xu, "A Unified Generative Adversarial Framework for Image Generation and Person Re-Identification," in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018, pp. 163–172. doi: 10.1145/3240508.3240573.

[209] A. Marzouk, P. Barros, M. Eppe, and S. Wermter, "The Conditional Boundary Equilibrium Generative Adversarial Network and its Application to Facial Attributes," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–7. doi: 10.1109/IJCNN.2019.8852164.

[210] A. Belmonte-Hernández, G. Hernández-Peñaloza, D. Martín Gutiérrez, and F. Álvarez, "Recurrent Model for Wireless Indoor Tracking and Positioning Recovering Using Generative Networks," *IEEE Sens. J.*, vol. 20, no. 6, pp. 3356–3365, 2020, doi: 10.1109/JSEN.2019.2958201.

[211] L. Kang, P. Riba, M. Rusiñol, A. Fornés, and M. Villegas, "Content and Style Aware Generation of Text-Line Images for Handwriting Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44,

no. 12, pp. 8846–8860, 2022, doi: 10.1109/TPAMI.2021.3122572.
[212] B. Jahić, N. Guelfi, and B. Ries, “Software Engineering for Dataset Augmentation using Generative Adversarial Networks,” in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 2019, pp. 59–66. doi: 10.1109/ICSESS47205.2019.9040806.
[213] “Institute of Electrical and Electronics Engineers DataPort Search Source.” https://ieee-dataport.org/datasets (accessed Dec. 09, 2022).
[214] “Zenodo Search Source.” https://zenodo.org/ (accessed Dec. 09, 2022).
[215] “Kaggle Search Source.” https://www.kaggle.com/datasets (accessed Dec. 09, 2022).
[216] S. Samonas and D. L. Coss, “The CIA strikes back: Redefining confidentiality, integrity and availability in security,” *J. Inf. Syst. Secur.*, vol. 10, 2014.
[217] C. B. Simmons, C. Ellis, S. G. Shiva, D. Dasgupta, and Q.-S. Wu, “AVOIDIT: A Cyber Attack Taxonomy,” *CTIT Tech. reports Ser.*, 2009.
[218] I. Butun, P. Osterberg, and H. Song, “Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 616–644, 2020, doi: 10.1109/COMST.2019.2953364.
[219] A. C. Panchal, V. M. Khadse, and P. N. Mahalle, “Security Issues in IIoT: A Comprehensive Survey of Attacks on IIoT and Its Countermeasures,” in *Proceedings - 2018 IEEE Global Conference on Wireless Computing and Networking, GCWCN 2018*, 2019, pp. 124–130. doi: 10.1109/GCWCN.2018.8668630.
[220] S. M. Tahsien, H. Karimipour, and P. Spachos, “Machine learning based solutions for security of Internet of Things (IoT): A survey,” *J. Netw. Comput. Appl.*, vol. 161, no. April, 2020, doi: 10.1016/j.jnca.2020.102630.
[221] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116. doi: 10.5220/0006639801080116.
[222] Y. Meidan *et al.*, “N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders,” *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, 2018, doi: 10.1109/MPRV.2018.03367731.
[223] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, “Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset,” *Futur. Gener. Comput. Syst.*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.
[224] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, “Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset),” *Lect. Notes Networks Syst.*, vol. 180, no. June, pp. 73–84, 2021, doi: 10.1007/978-3-030-64758-2_6.
[225] S. Garcia, A. Parmisano, and M. J. Erquiaga, “IoT-23: A labeled dataset with malicious and benign IoT network traffic.” Zenodo, 2020. doi: 10.5281/zenodo.4743746.
[226] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, “Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic,” *Appl. Sci.*, vol. 11, no. 17, 2021, doi: 10.3390/app11177868.
[227] S. Samarakoon *et al.*, “5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network.” IEEE Dataport, 2022. doi: 10.21227/xtep-hv36.
[228] “CICFlowMeter Canadian Institute for Cybersecurity.” https://www.unb.ca/cic/research/applications.html#CICFlowMeter (accessed Dec. 09, 2022).
[229] A. Rosay, E. Cheval, F. Carlier, and P. Leroux, “Network Intrusion Detection: A Comprehensive Analysis of CIC-IDS2017,” in *8th International Conference on Information Systems Security and Privacy*, 2022, pp. 25–36. doi: 10.5220/0000157000003120.
[230] T. Shorey, D. Subbaiah, A. Goyal, A. Sakxena, and A. K. Mishra, “Performance Comparison and Analysis of Slowloris, GoldenEye and Xerxes DDoS Attack Tools,” *2018 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2018*, pp. 318–322, 2018, doi: 10.1109/ICACCI.2018.8554590.
[231] Z. Al-Qudah, M. Rabinovich, and M. Allman, “Web Timeouts and Their Implications,” in *Passive and Active Measurement*, 2010, pp. 211–221.
[232] N. Oliveira, I. Praça, E. Maia, and O. Sousa, “Intelligent cyber attack detection and classification for network-based intrusion detection systems,” *Appl. Sci.*, vol. 11, no. 4, pp. 1–21, 2021, doi: 10.3390/app11041674.

[233] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” *Cybersecurity*, vol. 2, no. 1, 2019, doi: 10.1186/s42400-019-0038-7.

[234] M. Hossin and M. N. Sulaiman, “A review on evaluation metrics for data classification evaluations,” *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, p. 1, 2015, doi: 10.5121/ijdkp.2015.5201.

[235] F. Murtagh, “Multilayer perceptrons for classification and regression,” *Neurocomputing*, vol. 2, no. 5, pp. 183–197, 1991, doi: 10.1016/0925-2312(91)90023-5.

[236] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” in *Advances in Neural Information Processing Systems*, 2012.

[237] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[238] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, vol. 13-17-Augu, pp. 785–794. doi: 10.1145/2939672.2939785.

[239] G. Ke *et al.*, “LightGBM: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, 2017, pp. 3147–3155.

[240] F. T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation forest,” in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2008, pp. 413–422. doi: 10.1109/ICDM.2008.17.