INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Robust Machine Learning Against Adversarial Attacks

MIGUEL DIOGO GAMEIRO SILVA
Junho de 2025

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Robust Machine Learning Against Adversarial Attacks

Miguel Silva

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Cybersecurity And Systems Administration

Advisor: Eva Maia
Co-Advisor: Isabel Praça
Supervisor: João Vitorino

Porto, June 29, 2025

Statement of Integrity

I hereby declare that I have conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore, the work presented in this document is original and authored by me, having not previously been used for any other end. The exceptions are explicitly recognised in the section "Ethical considerations" of the first chapter. This section also states how AI tools were used and for what purpose.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, June 29, 2025

Abstract

The growing interest in the application of Artificial Intelligence (AI) in various domains is evident, and the general public is becoming more accustomed to them. However, this interest is accompanied by significant challenges that threaten the security of AI-based systems. Rather than relying on AI as the sole decision-maker in critical scenarios, it should serve merely as a suportive tool, with its vulnerabilities carefully considered. One major concern is adversarial attacks, where bad actors introduce small, imperceptible perturbations to input data, causing AI models to misclassify. The resilience to this type of attack is called robustness, and is essential to ensure secure deployment, to avoid potential harmful consequences, such as bypassing AI-based security systems.

This dissertation presents a tool, Adversarial to Understand Robustness and Offensive Resilience Analysis (AURORA), for testing the robustness of Machine Learning (ML) models to adversarial threats, focusing on adversarial attacks. The results of the attacks are evaluated using state-of-the-art metrics identified through a systematic review. As most of the work on this topic is made around images, this tool focuses on a less explored field, tabular data. glsAURORA also presents a new adjustment to these metrics based on the distance between original and perturbed samples. Since effective adversarial examples should closely resemble the original input, those that differ significantly are considered less meaningful and have less influence on the evaluation. Therefore, attacks that generate completely unrelated samples are penalised, reducing their success rate. This adjustment also accounts for the validity of perturbed samples, since invalid data should not influence evaluation metrics as much as valid data.

Overall, from the results of the case study conducted using AURORA, as most of the existing methods are focused on image data, these methods do not create valid or realistic adversarial samples for tabular data. By adjusting the data to create valid samples, the attack success rate decreases. This highlights the need for testing models against the appropriate methods specific to the data used. By developing AURORA, this dissertation contributes to the advancement of adversarial robustness research in ML, particularly in the context of tabular data. AURORA provides a simple and effective framework for evaluating robustness, while considering the constraints and considerations related to tabular data. It provides two robustness scores perspectives: one suited for a general use, and another for high-stakes, real world scenarios where only the best performing adversarial attacks are considered in the evaluation.

A key takeway of this dissertation is the need to continue efforts to improve the robustness and trustworthiness of ML models, and to raise awareness of the inherent vulnerabilities of ML models, and the risks associated with their use.

Keywords: Adversarial Examples, Evaluation Metrics, Realism, Model Robustness, Validity

Resumo

O interesse crescente na aplicação de Inteligência Artificial (AI) em vários domínios é evidente, e o público em geral está cada vez mais habituado a estes sistemas. No entanto, este interesse é acompanhado por desafios significativos que ameaçam a segurança de sistemas baseados em AI. Ao invés de confiar na AI como único decisor em cenários críticos, esta deve servir apenas como uma ferramenta de apoio, com as suas vulnerabilidades cuidadosamente consideradas. Uma das principais preocupações são os ataques *adversarial*, em que os atacantes introduzem pequenas e imperceptíveis perturbações nos dados de entrada, fazendo com que os modelos de AI façam classificações erradas. A resiliência a este tipo de ataque é designada por robustez e é essencial para garantir uma implementação segura, de modo a evitar potenciais consequências nefastas, como contornar sistemas de segurança que se baseiam em AI.

Esta dissertação apresenta uma ferramenta, *Adversarial to Understand Robustness and Offensive Resilience Analysis* (AURORA), para testar a robustez de modelos de aprendizagem automática (ML) a ameaças, focando-se em ataques *adversarial*. Os resultados dos ataques são avaliados utilizando métricas identificadas através de uma revisão sistemática da literatura. Uma vez que a maioria do trabalho sobre este tema é feito em torno de imagens, esta ferramenta foca-se num campo pouco explorado, os dados tabulares. A AURORA também apresenta uma novo ajustamento a estas métricas baseado na distância entre as amostras originais e as perturbadas. Uma vez que os exemplos *adversarial* eficazes devem assemelhar-se ao máximo ao original, os que diferem significativamente são considerados menos relevantes e têm menos influência na avaliação. Por conseguinte, os ataques que geram amostras completamente não relacionadas são penalizados, reduzindo a sua taxa de sucesso. Este ajustamento também tem em conta a validade das amostras perturbadas, uma vez que os dados inválidos não devem influenciar as métricas de avaliação tanto quanto os dados válidos.

Em geral, os resultados do caso de estudo realizado utilizando a AURORA evidenciam que, como a maioria dos métodos existentes se centra em dados de imagem, estes métodos não criam amostras *adversarial* válidas ou realistas para dados tabulares. Ao ajustar os dados para criar amostras válidas, a taxa de sucesso do ataque diminui. Este facto realça a necessidade de testar os modelos com métodos adequados e específicos aos tipos de dados em questão. Ao desenvolver a AURORA, esta dissertação contribui para o avanço da investigação sobre robustez de modelos de ML contra ataques *adversarial*, particularmente no contexto de dados tabulares. A AURORA fornece uma estrutura simples e eficaz para avaliar a robustez, tendo em conta as restrições e considerações relacionadas com os dados tabulares. Ela fornece duas perspetivas de avaliação de robustez: uma adequada para uso geral e outra para cenários reais de alto risco, em que apenas os ataques com melhor desempenho são considerados na avaliação.

Uma das principais conclusões desta dissertação é a necessidade de continuar a encetar esforços para melhorar a robustez e a fiabilidade dos modelos de ML e de sensibilizar para os problemas apresentados neste trabalho, uma vez que a maioria dos utilizadores não está

ciente das vulnerabilidades inerentes a estes modelos e, por isso, não está consciente dos riscos associados à sua utilização.

Acknowledgement

I would like to express my sincere gratitude to GECAD for providing the necessary resources, environment, and challenges that made developing this work possible. I am also deeply thankful to everyone who supported me throughout this process. In particular:

- To my advisor, Eva Maia, for her continuous guidance, support, and critical feedback, which were essential throughout this work.
- To my co-advisor, Isabel Praça, for her valuable insights and suggestions, which significantly shaped the direction and quality of this work.
- To my supervisor, João Vitorino, for his mentorship and encouragement, both of which were vital to my development.
- To my teammates, especially José and Daniela, for their valuable suggestions, insightful discussions, and continuous support.
- To my family for their relentless support throughout my academic journey. I am especially grateful to my parents, Elsa and Pedro, for their dedication and sacrifices. I am also grateful to my sister Mariana, for her constant encouragement, companionship, and shared study sessions.
- To Inês, for her patience, and tireless support, always being there to listen, to motivate, and to believe in me.

Contents

List of Figures

List of Tables

List of Acronyms

A2PM	Adaptative Perturbation Pattern Method.
ACD	Average Confidence Different.
ACM	Association for Computing Machinery Digital Library Search Source.
AD	Attack Deterioration.
ADR	Average Defense Rate.
AI	Artificial Intelligence.
ANGRI	Antagonistic Network for Generating Rogue Images.
API	Application Programming Interface.
APS	Average Perturbation Strength.
ARS	Adversarial Robustness Score.
ART	Adversarial Robustness Toolbox.
ASR	Attack Success Rate.
AUAC	Area Under the Accuracy Curve.
AUC	Area Under the Curve.
AUC-ROC	Area Under the Receiver Operating Characteristic Curve.
AURORA	Adversarial to Understand Robustness and Offensive Resilience Analysis.
AWC	Average Worst-case Margin.
BFAM	Bruteforce Attack Method.
BIM	Basic Iterative Method.
BMI-FGSM	Black-box Momentum Iterative Fast Gradient Sign Method.
C&W	Carlini and Wagner.
CAR	Comprehensive Adversarial Robustness.
CLEVER	Cross Lipschitz Extreme Value for nEtwork Robustness.
CLPA	Clean-Label Poisoning Availability.
CM	Classification Margin.
CORR	Empirical Correlation Coefficient.
CVAE	Conditional Variational Auto-Encoder.
DA	Distance Adjustment.
DAI	Defocus Attack Intensity.
DET	Detection Error Tradeoff.
DL	Deep Learning.
DoS	Denial of Service.

DPR	Damage Prevention Ratio.
EBD	Empirical Boundary Distance.
ER	Empirical Robustness.
F1	F1 Score.
FGSM	Fast Gradient Sign Method.
FN	False Negative.
FNR	False Negative Rate.
FP	False Positive.
FPR	False Positive Rate.
FR	Fooling Ratio/Rate.
GAN	Generative Adversarial Network.
GeoA^3	Geometric-aware.
GoPs	Graph of Patterns.
HAA	Hierarchical Adversarial Attack.
HRS	Harmonic Robustness Score.
I-FGSM	Iterative Fast Gradient Sign Method.
IBP	Interval Bound Propagation.
IEEE	Institute of Electrical and Electronics Engineers.
IoU	Intersection over Union.
IPP	Instituto Politécnico do Porto.
ITA	Imperceptible Transfer Attack.
JGBA	Joint Gradient Based Attack.
JSMA	Jacobian-based Saliency Map Attack.
L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno.
LGBM	Light Gradient-Boosting Machine.
MAE	Mean Absolute Error.
MAP	Mean Average Precision.
MCC	Matthews Correlation Coefficient.
MDA	Min Distance Attack.
MHA	Metropolis Hastings Attack.
ML	Machine Learning.
MLP	Multilayer Perceptron.
MPAF	Model Poisoning Attack based on Fake clients.
MPG	Momentum-Enhanced Pointwise Gradient.
MR	Misclassification Rate.
MRE	Median Relative Error.
MSE	Mean Squared Error.
NIDS	Network Intrusion Detection Systems.

NS	Neuron Sensitivity.
NTE	Noise Tolerance Estimation.
NU	Neuron Uncertainty.
PGD	Project Gradient Descent.
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses.
PWWS	Probability Weighted Word Saliency.
RAE	Relative Absolute Error.
RF	Random Forest.
RM	Robustness Merit.
RMSE	Root Mean Squared Error.
ROC	Receiver Operating Characteristic.
RR	Robustness Radius.
RSE	Root Relative Squared Error.
RUA	Risk Under Attack.
SPA	Single Page Application.
TN	True Negative.
TNR	True Negative Rate.
TP	True Positive.
UPSET	Universal Perturbations for Steering to Exact Targets.
VSA	Variable Step-size Attack.
XAI	eXplainable Artificial Intelligence.
ZOO	Zeroth Order Optimization.

Chapter 1

Introduction

This chapter provides an overview of the context and problem under investigation, addresses the ethical considerations, and outlines the structure of the document.

1.1 Context and Problem

In today's world, the rapid growth and widespread influence of Artificial Intelligence (AI)-based applications is undeniable and shows no signs of slowing down. As AI becomes increasingly embedded in our daily lives, it is vital to ensure that security measures are in place to protect these systems. This is particularly important as AI applications extend into various sectors, including critical areas where decisions made by AI models can result in significant damage and loss [1–3].

Over time, there have been several incidents where AI models have caused unintended harm [4]. These failures are often due to inadequate testing or training datasets that lack sufficient information to ensure accurate predictions. In addition, vulnerabilities in the AI models themselves have been exploited by malicious actors, leading to the generation of inappropriate or harmful outputs [5].

AI models are trained on datasets with the objective of generalizing to new data and make predictions based on this learned patterns. A good performing model is typically defined by its ability to make accurate predictions on unseen data, however, when applied in real-world contexts, biases can emerge - particularly those related to sensitive personal attributes such as ethnicity or gender [6, 7]. These biases not only compromise the fairness of predictions, but can also cause significant harm outside the intended scope of the task, such as discrimination.

While some biases occur naturally in the data, others can be introduced deliberately. Malicious actors have learned to exploit these vulnerabilities by deliberately introducing perturbations into the data that cause the model to make incorrect predictions. These attacks, known as adversarial attacks, can be particularly damaging in the context of AI security. This is especially concerning in critical areas namely healthcare, finance, autonomous driving, and intrusion detection, where adversarial actors can introduce undetectable, corrupted data that compromises the performance of models and the reliability of their decisions.

Adversarial attacks can take several forms. Evasion attacks, for example, involve deceiving the model by providing it with misleading or modified inputs, causing it to make incorrect predictions [8]. Data poisoning and obfuscation attacks, on the other hand, focus on infecting the training data, leading to degraded model performance and, in some cases, compromising the availability of the system [9].

In addition to data-based attacks, there are also direct model-based attacks. Some attacks seek to steal the model itself, which not only compromises the model owner's intellectual property, but also exposes valuable data and computational resources used to train the model [10]. Other attacks focus on causing data breaches by extracting sensitive information, such as the identification of data used to train the model [11].

In the face of these diverse threats, it is clear that rigorous testing is essential before AI models are deployed in real-world applications. Ensuring that models are robust against adversarial attacks is critical to maintaining their security, reliability and ethical performance in high-stakes environments. To effectively assess a model's robustness, it is essential to use well-designed metrics and appropriate perturbations in testing.

As most existing research on adversarial attacks and robustness focuses on image data, the robustness of Machine Learning (ML) models against adversarial attacks in tabular data remains underexplored. Most commonly used adversarial attacks create perturbations under the assumption that the data consists solely of numerical values. This disregards other data types, which can result in samples that are unsuitable or unrealistic for the target domain. Furthermore, effective adversarial attacks should aim to produce perturbations that deceive the model while mantaining a high degree of realism, making the perturbed sample close to the original sample and difficult to detect. Taking into account these considerations, it is crucial to develop a solution that can effectively measure the robustness of ML, while taking into account the specific characteristics and challenges associated with the specific domain.

1.2 Ethical Considerations

Ethical considerations are essential in software engineering research, as they directly affect the quality of life worldwide and emphasize the need for honesty and integrity. In particular, the Instituto Politécnico do Porto (IPP) Code of Conduct [12] has been followed in the conduct of this work:

- Article 4, which outlines the duties of researchers to apply scientific rigor while respecting good research practices, ethical principles, honesty, and precision. This includes ensuring accurate referencing and citation of bibliographic sources.
- Article 6, which specifies the responsibilities of students to abstain from academic misconduct, such as plagiarism or any fraudulent activity related to the use, reproduction, alteration, or destruction of materials.
- Article 10, which defines good practices in research activities, including thorough and careful analysis and documentation, proper citation of work, consistent presentation of results to allow verification and reproduction, and respect for authorship by acknowledging the relevant work and intellectual contributions of others.

In addition, at the beginning of this document is Article 8, which contains the Student's Declaration of Commitment. This declaration is a commitment to P.PORTO's Code of Ethics, and affirms the integrity of the academic work, emphasizing its originality and respect for ethical standards.

AI writing tools were used solely to improve the grammar and readability of this document. All ideas and content in this dissertation are the result of original human thought and authorship.

As a master's student in computer engineering with a specialization in cybersecurity and systems administration, the Institute of Electrical and Electronics Engineers (IEEE) Code of Ethics [13] is also relevant to ensuring the highest standards of integrity, responsible behavior, and ethical conduct in professional activities. Because this work involves the development of a tool to test the robustness of models against adversarial attack scenarios, there is a risk that malicious individuals could misuse this tool. For this reason, the Association for Computing Machinery Digital Library Search Source (ACM) Code of Ethics [14] is also considered, in particular section 1.2, which emphasizes the obligation to avoid harm. Therefore, this work carefully considers its potential impact and implements responsible practices to ensure that the knowledge generated is beneficially used to promote safer and more resilient systems for society.

Since this work involves using data to train ML models and assess their robustness, the used dataset will have an open-source nature. To ensure privacy and compliance with data protection standards, an anonymization process will be applied to the dataset. This process will effectively remove all personally identifiable information, ensuring that no individuals can be identified from the data, maintaining confidentiality and privacy.

1.3 Objectives and Research Questions

The main goal of this dissertation was developing a solution to measure the robustness of ML models against adversarial attacks, with a focus on tabular data. With this goal in mind, five specific objectives were established:

- **OB1:** Investigate the state-of-the-art robustness metrics.
- **OB2:** Identify what types of perturbation methods are commonly used to generate adversarial attacks.
- **OB3:** Formulate an approach to measure the robustness of ML models against adversarial attacks.
- **OB4:** Formulate an approach to measure the validity and realism of generated adversarial data.
- **OB5:** Develop a tool to implement the proposed approach and evaluate the robustness of ML models against several adversarial attacks.
- **OB6:** Validate and test the developed tool in tabular data case study.

To guide the research performed in the scope of this dissertation and successfully accomplish the estabilished objectives, a primary research question was formulated: *"How can ML model robustness be assessed?"*. The main question was divided into three narrower sub-questions:

- **RQ1:** How can the robustness of ML models be measured?
- **RQ2:** What are the most common attacks on ML?
- **RQ3:** Which strategies are used to generate adversarial attacks?

1.4 Scientific Contributions

Throughout the development of this dissertation, significant reseach was performed, various concepts were introduced, and several experimental evaluations were conducted at GECAD.

The main scientific contributions of the performed research and developed work are as follows:

- A literature review on how AI models robustness can be tested.
- A methodology for assessing how feature selection impacts the robustness of ML models against adversarial attacks.
- The development of GeNIS [15], a tabular dataset consisting of both benign and different types of network attacks. This dataset was used in the case study, as an example of how attackers can use adversarial attacks to compromise the robustness of ML models, aiming to classify malicious traffic as benign traffic.

At the time of submission of this document, four peer-reviewd scientific publications related to the development of this dissertation were published. The publications included the study of feature impact on the robustness and performance of models with several datasets, and the development of a network intrusion detection dataset important as a tabular data case study to test the solution.

Regarding scientific journals, two open access articles were published in Q3 journals:

- **Miguel Silva**, Daniela Pinto, João Vitorino, José Gonçalves, Eva Maia, and Isabel Praça, "GeNIS: A Modular Dataset for Network Intrusion Detection and Classification", *Data in Brief*, volume 60, 2025, doi: 10.1016/j.dib.2025.111487 [15].
- João Vitorino, **Miguel Silva**, Eva Maia, and Isabel Praça, "Reliable feature selection for adversarially robust cyber-attack detection", *Annals of Telecommunications*, volume 80, 2024, doi: 10.1007/s12243-024-01047-z [16].

Regarding scientific conferences, two papers were presented and published in the proceedings of the respective conferences:

- **Miguel Silva**, João Vitorino, Eva Maia, and Isabel Praça, "Efficient Network Traffic Feature Sets for IoT Intrusion Detection", *presented in 21nd International Conference on Distributed Computing and Artificial Intelligence (DCAI)*, 2024, pages 3-13, 10.1007/978-3-031-76459-2_1 [17].
- João Vitorino, **Miguel Silva**, Eva Maia, and Isabel Praça, "An Adversarial Robustness Benchmark for Enterprise Network Intrusion Detection", *presented in 15th International Symposium on Foundations and Practice of Security (FPS)*, 2023, pages 3-17, 10.1007/978-3-031-57537-2_1 [18].

1.5 Document Structure

This document is divided into several chapters, each carefully organized to facilitate both a comprehensive reading of the entire work and easy exploration of individual chapters.

The work begins with this Chapter 1, which introduces the problem addressed in this research, setting the context and importance of the study. It also outlines the ethical considerations that were taken into account throughout the research process, ensuring adherence to responsible research practices.

Next, Chapter 2 presents a systematic literature review, detailing the current state of the art in the field. This chapter also presents the research questions that guided the study,

along with the methodology used to address them. In addition, it includes the findings and conclusions drawn from the investigation of each research question, providing a clear understanding of the study's findings.

The structure and design of the proposed solution, including its architecture and components, are detailed in Chapter 3. Building on this, Chapter 4 introduces the implementation of the proposed solution, covering the different adversarial perturbation methods used to generate adversarial samples, as well as the metrics used to measure the impact of adversarial attacks on ML models. This chapter also discusses how the tool can be extended to support new perturbation methods and metrics, and presents the proposed user interface design.

Chapter 5 focus on the concept of realism in adversarial samples for tabular data, discussing how it can be measured depending on data specific characteristics. It introduces a methodology for assessing the realism of generated adversarial samples, which is crucial for ensuring that the generated data is not only effective in testing model robustness but also valid and applicable in real-world scenarios. Without such validation, unrealistic data can lead to misleading conclusions and undermine the reliability of robustness evaluations.

To illustrate the pratical application of the solution, Chapter 6 presents a real-world case study. It includes an evaluation of model robustness, as well as a discussion of the results, highlighting key insights and the need for further refinement of adversarial evaluation techniques.

Finnaly, Chapter 7 provides the main conclusions of this dissertation, highlighting the key findings and contributions of the research. It also outlines the limitations of the study and suggests directions for future work.

Chapter 2

State of the art

The research was based on the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [19], a standard reporting guideline that aims to improve the transparency of literature reviews.

To guide the research in this work, a primary research question was raised: "How can ML model robustness be assessed?". This broader question was further broken down into three focused sub-questions, each addressing a critical component necessary to provide a comprehensive answer:

- **RQ1:** How can the robustness of ML models be measured?
- **RQ2:** What are the most common attacks on ML?
- **RQ3:** Which strategies are used to generate adversarial attacks?

Following the PRISMA guidelines, a systematic search was conducted using established bibliographic databases with specific inclusion and exclusion criteria to screen the identified publications. The databases searched included IEEE [20], Science Direct [21], provided by the publisher Elsevier, and ACM [22]. Publications were initially screened on the basis of titles and abstracts, excluding those that did not meet the defined criteria. After this initial stage, a more thorough review was carried out to fully assess the eligibility of the remaining publications. In addition, snowballing techniques were used to obtain further information on methods that required further explanation or to identify original sources for citation.

This chapter presents and discusses the findings of the systematic review, which was conducted by thoroughly investigating the formulated research questions using the research methodology described.

2.1 Machine Learning Robustness

Robustness in AI systems is the capacity of a ML model to mantain reliable performance when exposed to adversarial attacks. As a form of resilience, robustness is a essential for preserving system integrity and ensuring consistent behavior, even in the face of unexpected or malicious conditions [23].

These attacks exploit vulnerabilities in AI models by introducing carefully crafted noise into the inputs. While imperceptible to human judgement, this noise can significantly degrade model performance [9]. In high-stakes applications of ML-based systems, such as healthcare, adversarial attacks can have serious consequences. For instance, attackers could manipulate readings from medical devices, leading to incorrect patient diagnoses or inappropriate

treatment plans [24]. Similarly, ML models used in critical areas such as credit card fraud detection are vulnerable to adversarial examples, allowing attackers to bypass detection systems, resulting in significant financial loss and undermining system effectiveness [25].

As these vulnerabilities can have devastating results, robustness is not optional, it is a foundational requirement for trustworthy AI. Over time, researchers have developed a number of metrics to assess how models are affected by such attacks, however, defining and measuring robustness remains a complex challenge, highlighting the need for clear and effective evaluation methods. Therefore, this section intends to answer RQ1: "How can the robustness of ML models be measured?".

2.1.1 Research Methodology

To define a search query relevant to this research question, an initial analysis of the literature revealed that the terms 'robust*' (which includes both 'robust' and 'robustness') and 'resilience' are commonly associated with the ability of a model to withstand adversarial attacks. As such, these terms were combined with others such as ML, AI and Deep Learning (DL) to ensure comprehensive coverage of relevant publications that address model robustness. The search query also included terms related to measuring the impact of adversarial attacks, including 'indicator', 'metric' and 'adversarial'. The search was conducted on June 10, 2025 and targeted abstracts, titles and keywords to comprehensively identify relevant studies. Table 2.1 provides an overview of the terms used in each area, which were combined in a search query using AND operators.

Table 2.1: Search terms for RQ1.

Scope	**Terms**
Robustness	(*resilience* OR *robust**)
Model	(*"machine learning"* OR *"artificial intelligence"* OR *"deep learning"*)
Metric	(*indicator* OR *metric*)
Adversarial	*adversarial*

To keep the findings up tp date with recent advances, only publications from 2020 onwards were included in the query. As robustness refers to the ability of a model to withstand adversarial attacks, only studies that explicity evaluated the impact of such attacks on model performance were included. This performance comparison is critical to understanding how well a model can maintain its predictive accuracy and reliability under adversarial conditions, providing insight into its real-world applicability and resilience to potential threats. Studies that focused on unrelated topics such as data generation without attack evaluation, or those that did not include adversarial scenarios were excluded. In addition, studies dealing with missing data or system interruptions were only included if they were explicitly linked to adversarial attacks. Table 2.2 presents an overview of the defined inclusion and exclusion criteria.

Table 2.2: Inclusion and exclusion criteria for RQ1.

Inclusion Criteria	Exclusion Criteria
IC1: Articles published from 2020 onward. IC2: Must include evaluation methods for model performance post-attack. IC3: Available in English language	EC1: Duplicated publications. EC2: Studies unrelated to adversarial attacks. EC3: Studies on data generation or dataset creation without attack evaluation. EC4: Does not involve adversarial attack scenarios. EC5: Missing data or system disruptions unless linked to adversarial attacks. EC6: Full text not available.

2.1.2 Findings and Discussion

For RQ1, the query applied to the content in the selected databases initially identified 301 articles. After removal of duplicates and the screening process, 155 articles were carefully evaluated to determine if they contained the required content. Finally, 146 articles that met the inclusion criteria were included in this review (Figure 2.1).

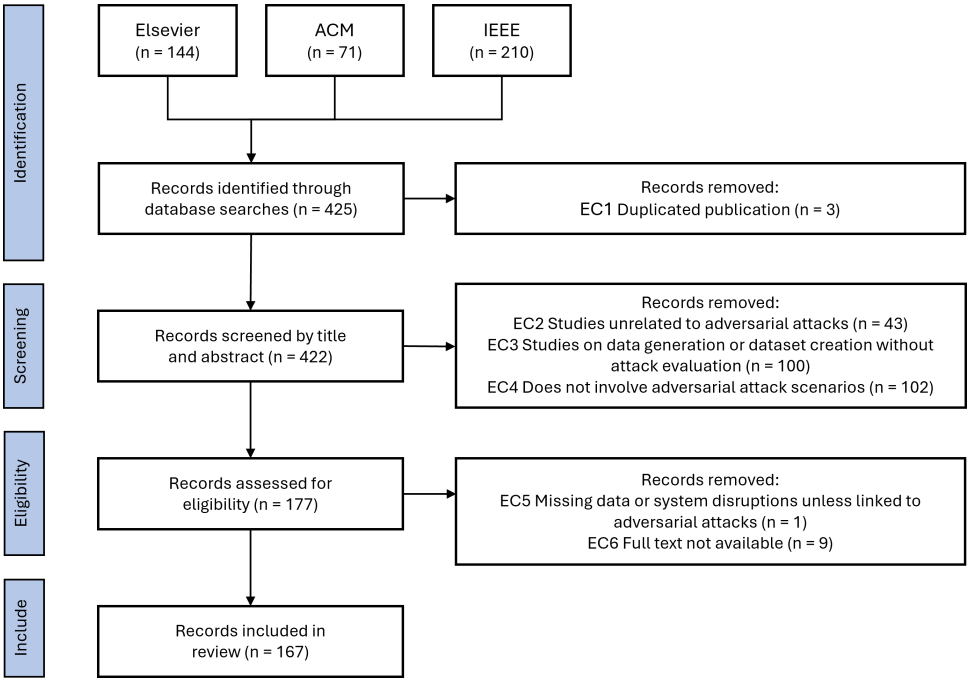

Figure 2.1: PRISMA search process for RQ1.

There are several domains that apply ML as a problem solver or to assist in choice-making. From these domains, computer vision is a broad one, where some of the applications of ML are used for object identification/detection, applied to fields such as health care [26] or even face id detection [27]. In Network Intrusion Detection Systems (NIDS), ML can be used to identify malicious behavior in the network [28] by scanning the network traffic searching for anomalies [29]. Text applications can also use ML which allows for speech and natural language processing [30], and is able to perform sentiment analysis [31]. As these

technologies are being increasingly deployed in high-risk environments [32], where failures can lead to accidents and casualties [33], ensuring the security and robustness of these models becomes critically important.

Adversarial attacks or adversarial examples [34] are inputs designed specifically by adversaries to change the output of ML algorithms in order to obtain a desired response or behaviour, such as misclassification [35, 36]. In the field of computer vision, for example, these inputs usually are perturbations made to be imperceptible to the human eye [37–39], as it can be seen in Figure 2.2. This can represent a security risk [27, 40, 41] depending on the aplication of the vulnerable ML model. An example of this can be a control access system that relies on ML to ensure access to only selected people [42], where an error could lead in a best case scenario to a lack of access (a false negative case), or in a worse case scenario, it can allow a malicious person access to a restricted asset. Although adversarial modifications are often designed to be subtle, it is possible for them to be clearly perceptible and obvious to the human eye while misleading the model, which does not recognize the the difference [43].

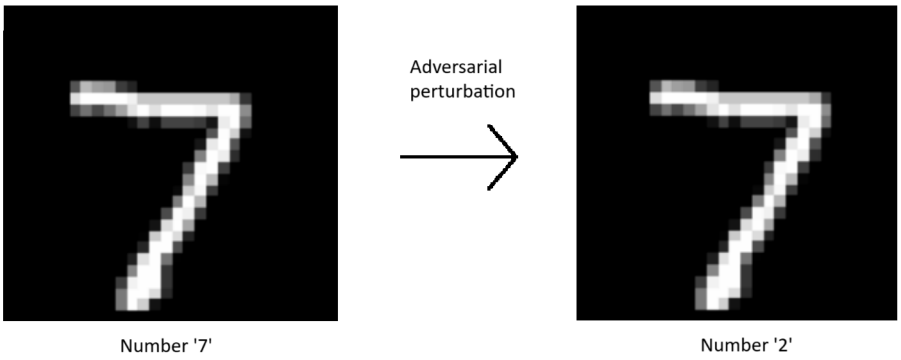

Figure 2.2: Model prediction before and after adversarial perturbation, based on [44].

These attacks can also be defined as either untargeted or targeted. An untargeted attack aims to cause any prediction errors by adding perturbations, while a targeted attack forces the model to predict a specific class selected by the attacker [45]. Apart from the objective, an adversarial attack can also be classified as either white-box or black-box, depending on the attacker's level of access and knowledge of target model. In a white-box attack, the attacker has full knowledge and access [46, 47], and in contrast, a black-box attack assumes no prior knowledge of the model, and the attacker can only interact with it by querying its outputs [47, 48].

As stated by Chen, Wang, and Chen [49], creating a robust model from a defender's perspective is challenging since the creator must protect against all possible adversarial attacks, whereas the attacker only needs to discover one effective attack. Different studies use distinct metrics to evaluate model robustness. A straightforward approach to assessing the robustness of a ML model is measuring the number of correct predictions, the sum of all correct positive predictions, known as True Positive (TP), and the correct negative predictions, known as True Negative (TN), relative to the total number of predictions made by the model, including the sum of TP, TN, and the incorrect predictions and the incorrect negative predictions, known as False Positive (FP) and False Negative (FN), respectively. This metric, commonly known as **Accuracy** Equation (2.1), is employed in various domains, including computer vision [49–68], NIDS [29], and natural language processing [69], although this approach is agnostic to the quality of the individual examples that contribute to the

measurement [70].

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP + TN + FP + FN}} = \frac{\text{Number of correctly classified examples}}{\text{Total number of examples}} \tag{2.1}$$

When used in the context of adversarial attacks, Accuracy can be divided into two categories: Adversarial Accuracy (also known as robustness accuracy [71, 72]) and Clean Accuracy (or natural accuracy). Clean Accuracy reflects the model's performance on unperturbed, original data, and indicates its ability to generalize to unseen inputs [73, 74]. Adversarial Accuracy, in contrast, measures the model's performance on adversarial examples in scenarios where an attacker is actively attempting to deceive it, and can be interpreted as robustness [30, 48, 75–79]. As Accuracy can be misleading in unbalanced datasets, where the model can achieve high accuracy by predicting the majority class, works as Bohachenko et al. [80] explore balanced Accuracy, which is a variant that accounts for class imbalance by averaging the Accuracy across all classes, ensuring that each class contributes equally to the overall metric. When assessing model robustness, studies ofte use Adversarial Accuracy as the primary metric, even though it is often referred to simply as Accuracy [81, 82]. This lack of clarity can be misleading and reflects a broader issue. Inconsistencies in metric naming and definitions not only affect accuracy, but also lead to confusion and misinterpretation of results, making it difficult to compare methods fairly [83]. As noted by Li and Li [84], the distinction between generalization and robustness lies in how the model detects inputs and responds to perturbed inputs, specifically whether the output changes as a result. An ideal model should be both accurate and robust, since a model that is accurate but not robust will fail to defend against adversarial attacks, while a robust model that lacks accuracy will produce unreliable predictions.

As Adversarial Accuracy is a very used metric to evaluate robustness, it is important to understand how it is defined. Most often, Adversarial Accuracy is defined with the same equation as standard accuracy, but some works propose alternative approaches to better measure robustness. For example, Gittens, Yener, and Yung [85] uses Equation (2.2), where the Adversarial Accuracy is defined as the expected value (E) of maximum loss ($\ell(f(X + \delta), Y)$) over all adversarial perturbations within a specified norm ball ($\|\delta\|_p \leq \epsilon$) around the original input. Other studies take different approaches, such as weighting both adversarial and natural accuracy, averaging them [86], or using the **Composite Robustness Score** Equation (2.3), which measures their differences [87]. In addition, **Adversarial Robustness** Equation (2.4) can be computed as the relative accuracy between adversarial and clean samples [88], which represents the ratio of accuracy. This ratio is more sensitive to changes than the simple accuracy difference used in the Composite Robustness Score, making it a more effective measure for evaluating model robustness.

$$\text{Adversarial Accuracy} = E \max_{\|\delta\|_p \leq \epsilon} \ell(f(X + \delta), Y) \tag{2.2}$$

$$\text{Composite Robustness Score} = \text{Clean Accuracy - Adversarial Accuracy} \tag{2.3}$$

$$\text{R}_{\text{adv}} = \frac{\text{Adversarial Accuracy}}{\text{Clean Accuracy}} \tag{2.4}$$

A closely related metric, **Attack Deterioration (AD)** [89], evaluates the ratio of accuracy degradation after an attack [90]. It is represented by Equation (2.5), and allows to compare the change in a value relative to its baseline. This metric essentially combines elements of both the Composite Robustness Score and Adversarial Robustness, providing a measure that is sensitive to changes like Adversarial Robustness, but also provides a direct quantification of accuracy loss.

$$\text{AD} = \frac{\text{Clean Accuracy - Adversarial Accuracy}}{\text{Clean Accuracy}} \tag{2.5}$$

To complement accuracy and provide a more detailed view of model performance. For instance, the **F1 Score (F1)**, which represents the trade-off between **Precision** Equation (2.6) and **Recall** Equation (2.7) [91], provide a more detailed perspective, highlighting the specific impact of adversarial attacks on model behavior, represented by Equation (2.8). Tariq, Le, and Woo [92] demonstrates the performance degradation caused by adversarial attacks, showing a nearly 45% decline for white-box attacks and approximately 25% for black-box attacks, as measured using F1, Recall and Precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.7}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.8}$$

As such, combining Accuracy with other metrics such as Recall, Precision, and F1 provides a more detailed perspective on model performance and robustness under adversarial conditions [93–101]. However, some studies focus solely on combining Accuracy and Recall [102], or even exclude Accuracy altogether, opting instead to combine Precision and Recall [103]. Incorporating the confusion matrix [104, 105], as represented in Figure 2.3, further enhances this analysis by providing a clear breakdown of TP, TN, FP and FN. This granular understanding deepens the assessment process and helps identify specific weaknesses.

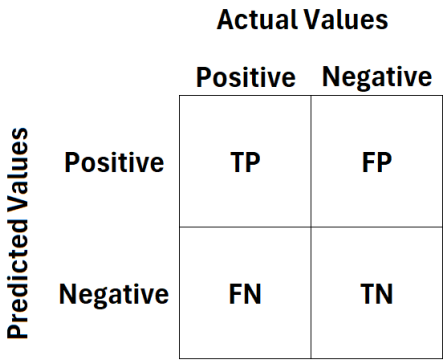

Figure 2.3: Confusion matrix for classification tasks.

In addition to these metrics, **Loss** functions provide further insight into model performance. Loss quantifies how well a model performs a given task by measuring the errors made during predictions, with lower values indicating better performance [106]. Using t-test to compare the Loss values before and after an attack, Chaddad et al. [107] demonstrate that adversarial perturbations lead to statistically significant increases in Loss, thereby confirming the vulnerability of the model. Combining different metrics such as Accuracy, Recall, Precision, and F1 with Loss provides a comprehensive assessment of how adversarial attacks compromise a model's reliability [108]. In particular, **Adversarial Loss**, which calculates the average worst case Loss induced by adversarial perturbations across the dataset, is used for assessing robustness of models and is represented by the Equation (2.9) [87]. This metric computes the maximum Loss ($L(Y_i, f(X_i + \delta(X_i)))$) for each sample ($X_i$) under perturbations ($\|\delta(X_i)\|_\infty \leq \epsilon$) constrained by bounds (ϵ), and averages these values over the dataset. Other studies, such as [75], also examine robustness through Loss functions, emphasizing their role in understanding performance under adversarial scenarios.

$$\text{Adversarial Loss} = \frac{1}{m}\sum_{i=1}^{m} \max_{\|\delta(X_i)\|_\infty \leq \epsilon} L(Y_i, f(X_i + \delta(X_i))) \tag{2.9}$$

Building on traditional metrics such as F1, Precision, and Recall, R.G., Sajjanhar, and Xiang [109] also incorporates **Cohen's kappa** (or kappa score), which quantifies the agreement between true and predicted values on a scale of 0 to 1, with values between 0.81 and 1 indicating near perfect agreement. In addition, they also use the **Matthews Correlation Coefficient (MCC)**, a metric that takes into account all four values of the confusion matrix and produces a score ranging from -1 to +1. By combining these metrics, the study provides a more comprehensive evaluation of model performance, allowing it to evaluate the impact of adversarial attacks across a broader set of evaluation criteria.

Additionally, the inclusion of **False Positive Rate (FPR)** Equation (2.11) can be important, especially in scenarios where false alarm minimization is a priority [110]. When coupled with Precision, it provides insight into the model's ability to maintain high accuracy while minimizing FP [111]. On the other hand, the **True Negative Rate (TNR)** Equation (2.10), or specificity, plays a crucial role in situations where correctly identifying negative instances is critical to avoiding FP [112]. To provide a more comprehensive view of misclassification, Barik and Misra [99] combines the FPR and **False Negative Rate (FNR)** Equation (2.12) to evaluate all types of errors made by the model.

$$\text{TNR} = \frac{TN}{FP + TN} \tag{2.10}$$

$$\text{FPR} = 1 - \frac{FP}{FP + TN} \tag{2.11}$$

$$\text{FNR} = 1 - \frac{FN}{FN + TP} \tag{2.12}$$

Using metrics specific to the automated driving systems domain, Yin et al. [113] and Zhang et al. [114] evaluate the robustness of vision-based models using domain specific metrics such as mean **Intersection over Union (IoU)** and average Precision with IoU values greater than 70%, respectively. These metrics demonstrate how robustness can be evaluated based on

the spatial overlap between the predicted and ground truth bounding boxes. However, since IoU is specific to object detection tasks, it cannot be applied to more general or non-visual scenarios.

Building on Precision, additional metrics such as **Mean Average Precision (MAP)** Equation (2.13) are also employed, as seen in [27, 115], which focus on person recognition tasks. This metric assesses model performance across multiple classes or scenarios, by averaging the Precision values. Specificaly, in the equation, MAP is calculated as the average of Precision values across all classes (C), weighted by the number of relevant items (R_c) in each class. The term Δr represents the change in rank, which is used to adjust the Precision value based on the relevance of the items. As such, it provides a more holistic view of model effectiveness, particularly when evaluating how well the model handles adversarial attacks over a range of conditions. Furthermore, MAP can be averaged across multiple models to provide an overall robustness measure. Building on this concept,[115] describe the mean MAP drop rate [116], which quantifies the average decline in MAP due to adversarial perturbations and effectively measures the vulnerability of multiple models.

$$\text{MAP} = \frac{1}{C} \sum_{c=1}^{C} \left(\frac{1}{R_c} \sum_{\text{relevant items}} \frac{\text{TP}}{\text{TP} + \text{FP}} \cdot \Delta r \right) \tag{2.13}$$

Another approach to measuring robustness that takes into account both the perturbation strength and the performance of the model is to use **Comprehensive Adversarial Robustness (CAR)** Equation (2.14), introduced by Bao et al. [117]. This metric combines two key factors: the **Average Perturbation Strength (APS)**, which measures the average difference between clean and perturbed samples over a set of data, and the model's accuracy on both original and adversarial examples, which translates to Clean and Adversarial Accuracy, respectively. A similar approach is taken by Guo et al. [118], who propose a metric for evaluating model robustness across varying perturbation strengths by measuring the normalized confidence scores of worst-case adversarial examples.

$$\text{CAR} = \frac{\text{APS}}{\text{Clean Accuracy} - \text{Adversarial Accuracy}} \cdot (1 + \text{Clean Accuracy}) \tag{2.14}$$

While using Clean Accuracy and Adversarial Accuracy to measure the performance of the models, Devaguptapu et al. [119] also introduces the **Harmonic Robustness Score (HRS)** metric Equation(2.15), which captures the trade-off between these two accuracies. In addition, a weighted version of HRS, called HRS_β Equation (2.16), is proposed to further refine this balance, where β is the importance over of Adversarial Accuracy over Clean Accuracy. These metrics help to evaluate the performance of the model on unperturbed inputs, while also assessing its robustness to adversarial attacks. The HRS metric takes a similar approach to F1 since both use the harmonic mean to balance two key metrics. In the case of HRS, the metrics are clean and Adversarial Accuracy, while F1 balances Precision and Recall. This structure allows HRS to effectively evaluate the balance between a model's performance on clean inputs and its robustness to adversarial attacks.

$$\text{HRS} = 2 \cdot \frac{\text{Clean Accuracy} \cdot \text{Adversarial Accuracy}}{\text{Clean Accuracy} + \text{Adversarial Accuracy}} \tag{2.15}$$

$$\text{HRS}_\beta = (\beta^2 + 1) \cdot \frac{\text{Clean Accuracy} \cdot \text{Adversarial Accuracy}}{\beta^2 \cdot \text{Clean Accuracy} + \text{Adversarial Accuracy}} \tag{2.16}$$

As another approach to evaluate the robustness or resilience of a ML model, the measurement of the success of an adversarial attack can be used [70, 120–124]. By assessing this effectiveness, the **Attack Success Rate (ASR)** Equation(2.17), also refered as Fooling Ratio/Rate (FR) [125, 126] or Evasion Rate [127, 128], measures the percentage of adversarial examples where the predicted label differs from the original label [129, 130].

$$\text{ASR} = \frac{\text{Number of successful attacks}}{\text{Total number of attacks}} \cdot 100 \tag{2.17}$$

This metric is used across different attack contexts, such as images [59, 60, 67, 131–134], tabular data [99, 135, 136], text [30, 31, 137–139] or even audio data [140, 141]. ASR can be represented by an equation similar to Adversarial Accuracy, although it represents the success of the attacks, not the success of the predictions made by the model.

As was the case for Accuracy, some studies use ASR as the sole measure of robustness, as is the case of [129, 142]. However, other works combine it with additional metrics to provide a more comprehensive evaluation. For instance, the **Detection Error Tradeoff (DET)**, which represents the trade-off between the FP Identification Rate and the FN Identification Rate [143], to properly evaluate and compare performaces of models. Some other studies pair ASR with Adversarial Accuracy [136], while others employ problem-specific metrics, which essentially measure the evasion rate of adversarial attacks, as is the case of [144].

Although ASR is commonly used and presented in its generic form, some studies build on it to present robustness in different ways. For instance, Park et al. [42] categorizes ASR results into five risk levels, each covering a 20% range, to indicate varying degrees of vulnerability. Similarly, Sun et al. [90] introduced **Average Defense Rate (ADR)**, a metric derived from ASR, to evaluate model defense performance by comparing results before and after defense measures are applied.

A metric similar to ASR is the **Failure Rate** [145], also referred to as Success Rate when viewed from the defender's perspective [146, 147]. Unlike ASR, which measures the proportion of adversarial attacks that successfully deceive the model, Failure Rate quantifies the opposite, how often adversarial attacks fail, and the model correctly classifies the input. This perspective provides insight into the model's ability to resist adversarial perturbations. The metric can be represented mathematically by an equation similar to Equation (2.18). While this is the most commonly used definition of Failure Rate, some studies define it as the ratio of false classifications per unit time [148].

$$\text{Failure rate} = 100 - ASR \tag{2.18}$$

Another metric similar to ASR, **Misclassification Rate (MR)** [70, 149] quantifies the percentage of successful adversarial examples. The primary distinction is their application, as MR is used to evaluate success in untargeted attacks, while ASR focuses on targeted attacks [150, 151]. This distinction is important because a failed attack in a targeted setting does not necessarily indicate that the model predicted the correct class [45]. Furthermore, MR serves as the inverse of Accuracy, reflecting the proportion of incorrect predictions made by

the model, as formalized in Equation (2.19). However, because it essentially reflects Accuracy, this metric provides limited additional insight. Therefore, other metrics that provide a different perspective on model performance may be more valuable.

$$\text{MR} + \text{Accuracy} = 1 \tag{2.19}$$

A related approach to MR is the evaluation of average MR, called **Robustness Analysis** by Zhang et al. [152], which assesses the consistency of a model's output under perturbations. Another perspective on measuring MR and evaluating the effectiveness of adversarial examples in remaining imperceptible to humans is provided by **Noise Tolerance Estimation (NTE)** [153]. This metric quantifies the difference between the misclassification probability and the maximum probability of other classes [153]. In addition, **Risk Under Attack (RUA)** extends this concept by calculating the misclassification risk over a set of samples subjected to adversarial perturbations [154].

While metrics such as MR evaluate misclassification rates under adversarial conditions as a measure of model robustness, other approaches aim to provide a broader perspective on robustness. For example, Jaiswal, Gollapudi, and Susma [155] introduced a framework specifically designed for numerical data. This framework uses a smart score calculator to compute the absolute differences between the chosen metric and the derived metric over six modified datasets Equation(2.20). These values are then scaled Equation (2.21) to assign a robustness level to the model, ranging from one, indicating the lowest robustness, to five, representing the highest.

$$\text{AvgSumDiff} = \sum_{k=1}^{6} \frac{|\text{ReferenceMetric} - \text{DerivedMetric}_k|}{6} \tag{2.20}$$

$$\text{Model robustness level} = (5 - (\text{AvgSumDiff} \times 5)) \times \text{ReferenceMetric} \tag{2.21}$$

Similarly, other works have focused on providing additional insight into how adversarial examples differ from the original data [27]. For instance, Sharma et al. [156] proposes alternative methods for measuring robustness, such as the **Robustness Index**, represented in Equation(2.22), which evaluates the average distance of data points from the decision boundary. Expanding on boundary-based evaluations, Gibert, Zizzo, and Le [157] studies certified robustness, where models are formally proven to withstand attacks within a defined boundary or domain. In a related approach, Chen et al. [158] analyzes how perturbations affect boundary accuracy and regional similarity, offering further insight into robustness under adversarial conditions.

$$l_{\text{robust}} = E_X[d(X, B)] \tag{2.22}$$

In addition to the Robustness Index, a similar approach is the **Classification Margin (CM)** 2.23, which focuses on the label space by measuring the difference in confidence between the predicted class and the most likely incorrect class [90, 159]. As with the Robustness Index, a higher CM indicates greater robustness, suggesting that the model is more confident in its predictions and less susceptible to adversarial manipulation. In line with this, the **Average Confidence Different (ACD)** [160] builds on the concept of CM by evaluating a model's defense performance by the average difference in CMs before and after the application of

adversarial defense techniques [90]. This metric provides an additional layer of insight into how defenses affect model performance under adversarial conditions.

$$CM(t) = \log \frac{p_{t,c_t}}{\max_{c \neq c_t} p_{t,c}} \tag{2.23}$$

Also related to CM, metrics such as **Adversarial Risk** and **Adversarial Gap** probabilistically assess a model's vulnerability by evaluating its robustness to small perturbations across the entire dataset [159]. These metrics, like CM, use confidence values to measure how well the model can withstand adversarial manipulation, providing a more complete understanding of the model's susceptibility to perturbations [159].

Confidence scores, which reflect a model's confidence in its predictions, are another crucial indicator of performance under adversarial conditions. A decrease in confidence scores typically correlates with a decrease in the proportion of correct decisions, revealing a model's vulnerability to an attack [161].

To further analyze the effects of adversarial perturbations on model behavior, several specialized metrics have been proposed. These include the mean confidence in the adversarial class for successful attacks **MeanConf**; the distance between the adversarial and correct benign classes **DistAATA**; the gap between the adversarial and true class predictions in corresponding benign examples **DistAATB**; and the difference between the confidence of the adversarial class and the next most likely predicted class **DistACOC** [45].

Shifting focus from metrics such as adversarial risk and confidence score based metrics, the **Receiver Operating Characteristic (ROC)** and its **Area Under the Curve (AUC)** are commonly used to evaluate the performance of binary classifiers [162]. These metrics are particularly effective in assessing both accuracy and recall [163], with AUC proving particularly valuable for unbalanced datasets [164].

For example, Roshan, Zafar, and Ul Haque [165] and Roshan and Zafar [166] evaluated the performance of a NIDS model under adversarial attack, where a significant drop in performance was observed as measured by F1, Precision, Recall, Accuracy, confusion matrix, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Although according to [162], AUC is typically used for binary classification, it can also be adapted for multiclass classifiers, allowing performance evaluation across multiple categories [167].

In this context, Yang et al. [168] analyzed the behavior of AUC alongside Accuracy, while Yerlikaya and Bahtiyar [169] included the F1 to gain a deeper understanding of metric performance under adversarial conditions. Other studies have paired AUC with Accuracy [138] and AUC-ROC [125].

To further evaluate model performance over different levels of perturbation, Bouniot, Audigier, and Loesch [163] introduced the **Area Under the Accuracy Curve (AUAC)** Equation (2.24), which evaluates model performance from low to high levels of perturbation. Similarly, Richards, Raff, and Matuszek [170] proposed the AUC for samples with different attack budgets (or perturbation budgets ϵ), which here are the intervals between the minimum and maximum perturbation allowed for each sample (ϵ_{min} and ϵ_{max}), and the resulting accuracy, AUC_acc, as represented by the equation 2.25.

Extending this approach, Bao et al. [171] proposed **AUCPro**, a framework that evaluates adversarial robustness using the AUC metric. By smoothing the model with isotropic Gaussian

noise, AUCPro estimates a proxy AUC that is provably stable under ℓ_2-bounded adversarial perturbations, offering both empirical and certified guarantees of performance.

$$\text{AUAC}_{\epsilon_{\max}} = \frac{1}{\epsilon_{\max}} \int_0^{\epsilon_{\max}} \text{Accuracy}(\epsilon)\, d\epsilon \tag{2.24}$$

$$\text{AUC}_{\text{acc}}(X, Y, s) = \int_{\epsilon_{\min}}^{\epsilon_{\max}} \frac{|f(X_s + \delta\epsilon) = Y_s|}{|X_s|}\, d\epsilon \tag{2.25}$$

Following the discussion of classification metrics such as ROC and AUC, it is important to acknowledge that ML tasks can be broadly divided into regression and classification, each of which requires different performance evaluation methods. Classification models produce discrete outputs and therefore rely on metrics designed to compare categorical labels, such as Accuracy, Precision, Recall, F1 and AUC. In contrast, regression models produce continuous outputs and are evaluated using metrics that measure the discrepancy between predicted and true values, such as **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, and **Mean Absolute Error (MAE)** [84, 172]. In addition, prediction deviation can be used to assess how adversarial perturbations affect predictions, providing valuable insight into the robustness of models under attack [173].

An example of regression tasks using these metrics can be found in the Quality-of-Service domain [174], where MAE, RMSE, and **Median Relative Error (MRE)** were used to evaluate model performance. The study emphasized that metrics based on relative error are better suited for assessing robustness in datasets with high variance. Moreover, MAE was specifically used to analyze the impact of adversarial attacks on model predictions.

In time series analysis, metrics such as **Relative Absolute Error (RAE)**, **Root Relative Squared Error (RSE)**, and **Empirical Correlation Coefficient (CORR)** are used to evaluate performance in adversarial contexts, such as in [175]. Lower error values and higher correlation coefficients indicate better model performance. Similarly, MAE-based metrics are used in fusion models, where two images are combined to produce clearer results. For example, Jin et al. [176] introduced **Defocus Attack Intensity (DAI)**, a metric that measures the area of an image altered by an attack and calculates the MAE between the original and altered images.

Despite the usefulness of these metrics, some researchers have argued that existing indicators do not fully capture all aspects of model robustness. In response, novel approaches have been developed to fill these gaps. One such approach is the Adversarial Robustness Score (ARS) Equation (2.26), introduced by Hartl et al. [177] in the context of NIDS. **ARS** quantifies how easily a classifier can be defeated, providing a more focused and nuanced assessment of robustness, where N is the number of samples, S_{adv} is the set of adversarial samples, and $d_{s_{\text{adv}}}$ is the distance between the adversarial sample and the original sample, measured using l_1 norm.

$$ARS = \frac{1}{\lfloor N/2 \rfloor} \sum_{s_{\text{adv}} \in S_{\text{adv}}} d_{s_{\text{adv}}} \tag{2.26}$$

Other studies have introduced different metrics to evaluate model robustness, each focusing on different aspects of adversarial vulnerability. One such metric is **Pointwise Robustness**, which examines perturbations small enough that a human cannot distinguish them from the

original input [178]. Another is **Adversarial Frequency** Equation (2.27), which measures how often a model fails to demonstrate robustness, effectively corresponding to its accuracy on adversarial examples [178]. Complementary to this is **Adversarial Severity** Equation (2.28), which assesses the magnitude of a model's failure when confronted with adversarial examples, providing a different perspective from Adversarial Frequency [178]. For both these metrics, x represents the input, h is the model, δ is the perturbation size, X is the dataset, and $\rho(h, x)$ is the model's prediction error for input x.

$$\hat{\phi}(h, \delta, X) = \frac{|\{x \in X \mid \rho(h, x) \leq \delta\}|}{|X|} \tag{2.27}$$

$$\hat{\mu}(h, \delta, X) = \frac{\sum_{x \in X} \rho(h, x) I[\rho(h, x) \leq \delta]}{|\{x \in X \mid \rho(h, x) \leq \delta\}|} \tag{2.28}$$

The name of Pointwise Robustness metric has been associated with other nomenclature, local adversarial robustness by Wang et al. [179]. These authors also present other metrics under a Global Robustness, these being Adversarial Frequency, Adversarial Severity, and **Relative Robustness**. Relative Robustness is a more known metric, although it is often referred to by other studies as Average Robustness, **Empirical Robustness (ER)** [180, 181], or Fooling Rate [182–184]. In particular, ER Equation (2.29), represents the average ratio of the perturbation sizes of minimal adversarial examples to their corresponding original inputs [185]. In this equation, $\hat{r}(x)$ represents the minimal perturbation required to change the model's prediction for input x, and D is the dataset used for evaluation.

$$\hat{\rho}_{\text{adv}}(f) = \frac{1}{|D|} \sum_{x \in D} \frac{\|\hat{r}(x)\|_2}{\|x\|_2} \tag{2.29}$$

Building on the minimal perturbation, another approach to assessing a model's resilience to adversarial attacks is the **Robustness Radius (RR)**. This metric considers the minimum perturbation required to change a model's predicted label, and can be formally expressed as shown in Equation (2.30) [67]. For this equation, $d(x, x')$ represents the distance between the original input x and the perturbed input x', $f_i(x')$ is the model's output for class i given input x', and $f_y(x')$ is the model's output for the true class y.

$$\text{RR} = \min_{x' \in [0,1]^n} d(x, x') \quad \text{subject to } \max_{i \neq y} f_i(x') \geq f_y(x') \tag{2.30}$$

Evaluating Relative Robustness poses significant challenges, as it is not bound to a specific metric. In the context of studying the transferability of adversarial examples, Menéndez [186] proposed a different equation to quantify robustness in terms of a model's ability to resist changes in prediction caused by transformations Equation (2.31). For this approach to measure robustness using Relative Robustness, $\delta(f(\phi(x_i)), f(x_i))$ represents the difference between the model's prediction for the transformed input $\phi(x_i)$ and the original input x_i, N representing the total number of samples in the dataset. The same study also introduced the concept of Absolute Robustness Equation (2.32), which evaluates a model's ability to correctly classify data after it has undergone transformations between different domains. For this metric, $\delta(f(\phi(x_i)), y_i)$ represents the difference between the model's prediction for the transformed input $\phi(x_i)$ and the true label y_i of the original input x_i.

$$\text{ReR}(M, \phi) = 1 - \frac{\sum_1^N \delta(f(\phi(x_i)), f(x_i))}{N} \tag{2.31}$$

$$\text{AbR}(M, \phi) = 1 - \frac{\sum_1^N \delta(f(\phi(x_i)), y_i)}{N} \tag{2.32}$$

While some metrics, such as Fooling Rate and FNR, are often used in robustness studies, Nguyen et al. [187] argue that these metrics should be interpreted as measures of attack strength, not model robustness. According to their framework, proper robustness evaluation should rely on metrics such as Robust Task Performance, which measures how well a model performs with adversarial inputs, Effective Robustness, which measures the difference between expected performance with clean inputs and actual performance with adversarial inputs, and Relative Robustness, which is the difference in performance between two models (for example, with and without a defense).

Global Robustness (or Adversarial Frequency [179]) is also not associated with any particular metric and the solution of the robustness equation is computationally intensive [75, 84]. To overcome these limitations, practical metrics have been proposed to provide feasible approximations. Examples include local robustness [188] and the **Cross Lipschitz Extreme Value for nEtwork Robustness (CLEVER)** metric [189]. CLEVER, which being attack-agnostic, calculates the minimum perturbation required to misclassify a benign example [59, 84]. It is represented by the Equation (2.33) for untargeted attacks and the Equation (2.34) for targeted attacks [84]. The **Lipschitz constant** is a key component of CLEVER, and measures the sensitivity of a model to changes in input [190]. For CLEVER, $f_c(x_0)$ is the model's output for the correct class c at input x_0, $f_j(x_0)$ is the model's output for the incorrect class j, and L^j_{q,x_0} is the Lipschitz constant for class j at input x_0. The parameter R represents a predefined perturbation budget, which limits the maximum perturbation allowed in the adversarial example.

$$\|\delta\|_p \leq \min\left\{\min_{j \neq c} \frac{f_c(x_0) - f_j(x_0)}{L^j_{q,x_0}}, R\right\} \tag{2.33}$$

$$\|\delta\|_p \leq \min\left\{\frac{f_c(x_0) - f_t(x_0)}{L^j_{q,x_0}}, R\right\} \tag{2.34}$$

The relationship between CLEVER and ASR has been explored in various studies. For instance, Jankovic and Mayer [132] concluded that CLEVER scores can be indicative of the success of adversarial attacks, but they are not sufficient on their own to predict attack effectiveness. Similarly, Jin et al. [180] evaluated their proposed metric, ROBY, against CLEVER, ER, ASR and Adversarial Accuracy over a range of datasets, where ASR and ROBY represented a better average robustness.

While also dwelling in boundaries to evaluate model robustness, **Interval Bound Propagation (IBP) Bounds Tightness** assesses the quality of the upper and lower bounds within a defined bound that adversarial examples cannot exceed [70]. While this metric is valuable for determining verifiable robustness, it should not be relied upon as a standalone indicator of certified robustness. Instead, it should be paired with complementary metrics, such as standard Accuracy and training accuracy, as recommended by Omar et al. [70]. Similarly,

Shen and Li [191] propose a verification method that defines an upper bound on adversarial perturbation power, which serves as a certificate of robustness within their specific context.

According to Wang et al. [192], several model representation-based metrics are also used to assess adversarial robustness. These include **Neuron Sensitivity (NS)** [193], which defines robustness as the global insensitivity of a model to adversarial examples, and **Neuron Uncertainty (NU)** [194], which is derived from the variance of neuron activations and used to measure uncertainty in safety-critical applications. Other notable metrics include **Empirical Boundary Distance (EBD)** [195], which calculates the minimum distance to the decision boundary, and **EBD-2** [194], an extension that includes the minimum distance to the decision boundary for each class. In addition, **Empirical Noise Sensitivity** [190] measures the response of the model to random noise. While these metrics provide valuable insights, Wang et al. [192] argued that these are not tailored to improve adversarial robustness through model fine-tuning and proposed the Graph of Patterns (GoPs) framework, which evaluates differences in average occurrences within a given class.

Other metrics highlighted by Sun et al. [90] include **Concealment Measures** [196], which examine the effectiveness of hiding nodes or communities in graphs, and the **Similarity Score** [197], which measures the similarity between pairs of instances and reflects the attacker's ability to change this score for a target pair. Some other metrics include **Average Worst-case Margin (AWC)** [198], which computes the average minimum value of the classification margin across a batch of data; **Robustness Merit (RM)** [89], which compares the post-attack accuracy of a proposed method to that of a baseline model; and **Damage Prevention Ratio (DPR)** [199], which quantifies the amount of damage mitigated by a defense, as defined in Equation (2.35), where L_A is the loss under an attack, L_D is the loss when making queries according to a defense strategy, and L_0 is the defender loss when no attack is applied.

$$\mathrm{DRP}^{\mathrm{Defence}}_{Adversarial} = \frac{L_A - L_D}{L_A - L_0} \tag{2.35}$$

Adding hidden messages to images, known as steganography, is explored by Liu et al. [200], where these messages are embedded in images after adversarial perturbations have been applied. They apply the **Reed-Solomon Bits Per Pixel** metric to evaluate robustness against distortions, where a higher value indicates greater redundancy and improved error correction capabilities.

In the context of certified robustness, **Certified Accuracy** evaluates the reliability of certification methods for robust community detection models under adversarial attacks [90]. Meanwhile, **Practical Effect** serves as a more abstract metric, assessing the impact of attacks and defenses on broader factors such as revenue and reputation [90].

Beyond metrics that evaluate performance under different levels of perturbation, other approaches focus on quantifying the effort required for a successful attack. The **Attacker Budget** [201] is the minimum perturbation required for an attacker to achieve his objective [90]. While not explicitly defined as a direct metric of robustness, it is important to be considered since an attacker only needs to find a single perturbation to succeed. Similarly, the **Average Modified Links** [202] metric builds on this concept by calculating the average number of links that need to be modified to achieve the attacker's goal Equation (2.36) [90].

$$\text{Average Modified Links} = \frac{\text{Number of modified links}}{\text{All attacks}} \tag{2.36}$$

In a similar vein, the time required to execute a successful adversarial attack can be used as a measure of the effort required by a malicious actor, providing an approach to assessing robustness similar to that used to assess cipher robustness. An implementation of this concept is the **Dynamic Absolute Robustness** [186], as defined in equation Equation (2.37), with **Adversarial Convergence** represented in Equation (2.38). In this context, M is the model, $\phi(t)$ is the perturbation function, N is the number of samples, δ is the difference between the model's prediction and the true label, and y_i is the true label for input x_i. For Adversarial Convergence, $\arg\min_t$ represents the time at which the model's performance is optimized under adversarial conditions.

$$\text{DyR}(M, \phi(t)) = 1 - \frac{\sum_1^N \delta(M(\phi(t, x_i)), y_i)}{N} \tag{2.37}$$

$$t^* = \arg\min_t \text{DyR}(M, \phi(t)) \tag{2.38}$$

Metrics should generally take into account the need to ensure that perturbations applied to inputs are as minimal as possible while still changing the model's predictions, especially in the context of adversarial examples. While these metrics do not directly measure model robustness, they assess the quality of adversarial examples [203] under the premise that a robust model should withstand perturbations that are nearly indistinguishable from the original input. Commonly used metrics for quantifying the distance between the original and adversarial data include the l_p-norm, (with typical values of 1, 2, or ∞) [47], as well as cosine similarity [27]. In the domain of tabular data, where perturbation strategies are often adapted from image-based methods, a more nuanced evaluation is required. He et al. [204] address this by introducing the concept of imperceptibility tailored to tabular features. The authors define seven properties — proximity, sparsity, deviation, sensitivity, immutability, feasibility, and feature interdependency — to evaluate how well adversarial examples maintain the semantic and structural integrity of the data. This expanded framework moves beyond success rates alone, emphasizing that effective adversarial evaluation should consider both the attack's impact and the realism of the perturbations.

Among the metrics discussed, some of the most meaningful for evaluating overall model robustness are accuracy-based metrics such as Adversarial Accuracy and AD. These metrics provide a comparison of model performance against both adversarial examples, and inputs that sits between real and adversarial examples. Other important metrics include ASR and MR, which are used to evaluate performance in targeted and untargeted attack scenarios. While MR provides limited additional information beyond Accuracy, it is still valuable for distinguishing the success of attacks in different scenarios. In addition, the attacker budget provides valuable insight by estimating the effort required for an attacker to generate an input that changes the model's predictions.

Although not explicitly designed as robustness metrics, fairness and diversity are critical considerations that provide complementary insights into model security and performance. Fairness ensures that a model's predictions are unbiased with respect to sensitive attributes [85] and prevents discrimination due to uneven representation in the training data [70]. For instance, Fukuchi, Hara, and Maehara [205] demonstrates how adversarial attacks can

compromise fairness by poisoning training data to introduce bias. Diversity, on the other hand, emphasizes the inclusion of diverse examples within the training data [70], which can increase model robustness to perturbations [206].

In evaluating improvements in model robustness, sustainability is emerging as a significant concern. For example, Hasan, Shahid, and Imteaj [207] [208] proposes the robust carbon trade-off index, a metric that evaluates robustness in the context of carbon emissions resulting from energy consumption, thereby aligning robustness with environmental concerns.

Recent research also explores the use of eXplainable Artificial Intelligence (XAI) techniques to identify features most influential in model decision making and to analyze adversarial attack mechanisms [96, 209–213]. Techniques such as LIME [214], SHAP [215], and IG [216] are often used in these studies. For example, Vaccari et al. [209] and Salah et al. [217] analyzes attack detection using ratios of FP, TP, TN, and FN, while De Aguiar, Traina, and Traina [212] evaluates adversarial robustness using Accuracy and ASR.

Among the metrics discussed in this research, Accuracy and ASR stand out due to their widespread use, as both rely on the classification of adversarial data, either in terms of correct classification or the ability to fool the model. However, it is evident that numerous other metrics have been proposed to address the challenge of measuring robustness, with recent studies increasingly incorporating XAI techniques. A key takeaway from this research is that while these metrics are often used in combination, direct comparisons between them are rare. Exploring such comparisons could provide valuable insights and help researchers better understand the strengths and limitations of each metric. This, in turn, could refine model robustness evaluation methods and contribute to the development of more reliable, consistent measures for assessing adversarial performance and model resilience.

2.2 Adversarial Attacks

As is widely recognized in the field, not all attacks on ML models are alike. These attacks vary significantly in nature, and understanding this variation is critical to assessing the vulnerability and robustness of ML systems. Adversarial attacks can differ in terms of their objectives, and the level of knowledge or capabilities of the attacker [218]. This variety makes it essential to categorize and classify the different types of attacks, as this allows researchers and practitioners to identify potential vulnerabilities and devise appropriate countermeasures.

To provide a more organized and systematic understanding of these attacks, researchers have developed various taxonomies. These taxonomies categorize adversarial attacks according to specific criteria, such as the characteristics of the attack, its impact on the system, and the approach used to execute it. The goal of these taxonomies is to provide a structured framework for evaluating different types of attacks so that the ML community can more effectively address and mitigate each specific threat. Therefore, this section aims to answer RQ2: "What are the most common attacks on ML?".

2.2.1 Research Methodology

To ensure a comprehensive categorization of the different types used to classify adversarial attacks in the literature, a search query was carefully structured based on the terms presented in Table 2.3. This approach results in different a systematic organization of knowledge about adversarial attacks on ML models. The search was performed on June 10, 2025, using the terms specifically queried in the abstracts, keywords and titles of the documents. To refine

the search scope and ensure a thorough coverage of the topic, search terms were combined using AND operators.

Table 2.3: Search terms for RQ2.

Scope	**Terms**
Model	(*"machine learning"* OR *"artificial intelligence"* OR *"deep learning"*)
Adversarial	(*adversarial*)
Systematization	(*taxonomy* OR *systematization*)
Attack	(*"attack"*)

To ensure that the literature reviewed reflects the most recent and relevant information on adversarial attacks, the inclusion and exclusion criteria outlined in Table 2.4 were established. Specifically, this research will focus on studies published from 2019 onwards, with the aim of capturing key developments and emerging trends in the field. Additionally, only articles that present a clear categorization of adversarial attacks against ML systems will be considered, allowing for a structured understanding of the different types of attacks targeting these systems.

Table 2.4: Inclusion and exclusion criteria for RQ2.

Inclusion Criteria	**Exclusion Criteria**
IC1: Articles published from 2019 onward	EC1: Duplicated publication
IC2: Available in English language	EC2: Does not present a categorization of adversarial attacks against machine learning
IC3: Survey or review of adversarial attacks	EC3: Full text not available

2.2.2 Findings and Discussion

Regarding RQ2, the initial search of the selected databases identified 54 articles. After removing duplicates and performing a screening process, 32 articles remained for further evaluation. These articles were carefully reviewed to determine if they met the inclusion criteria. In the end, 30 articles were included in this review (Figure 2.4).

Based on the analysis of existing literature, these findings have been organized into two distinct sections: Threat Model and Attack Type. The Threat Model focuses on the adversary's capabilities and knowledge, while the Attack Type focuses on the strategies based on their objectives and methods of the attacks. This division allows for a clearer understanding of how different attack strategies are shaped by the adversary's resources and intentions. Furthermore, this organization provides a comprehensive approach to identifying and categorizing the most prevalent attacks on ML, while highlighting which attacks are more prevalent under varying conditions.

Threat Model

According to Rosenberg et al. [219], a threat model is characterised by the adversary's known knowledge and capabilities prior to the attack, with a critical focus on identifying the extent

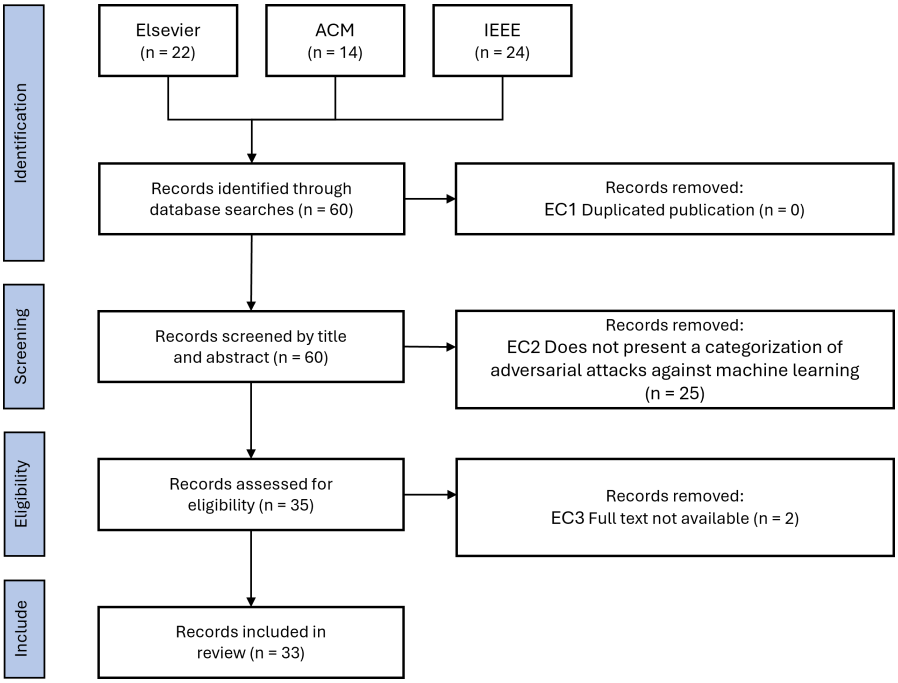

Figure 2.4: PRISMA search process for RQ2.

of their understanding of a ML system.

Adversary knowledge is typically divided into two main classes: black-box and white-box [220–224]. Besides these two main classes, some works choose to also add more specific classes to better define the attacker. Another classe, transparent-box, are characterized by the adversary's complete knowledge of the system, including both white-box knowledge and insight into the defender's methods, which helps the attacker to choose an attack capable of bypassing the defenses [219]. To define a more limited degree of knowledge, some works such as [8, 225–236] use gray-box (or partial white-box), where the attacker has partial knowledge of the system, such as access to either the ground truth or the model, but not both [237].

An attacker's capabilities are closely tied to his level of knowledge. Instead of accessing the model directly, an adversary may target the training data used by the model. Access to training data can be categorized into levels such as no access, read access (partial or full), the ability to inject new samples, or the ability to modify existing samples [219]. In some works, these levels are simplified into three broader levels: no access, read access, and write access [226]. These capabilities allow for an attacker to change the training data of a model, which allows the attacker to modify/add samples, causing the model to misbehave, and tampering with its training integrity.

To address these risks, especially those related to access to training data, Federated Learning has been suggested as a ML framework to preserve privacy. It aims to create a global learning model without requiring the sharing of raw data. In this approach, a coordinating agent is often required to manage the exchange of information between data owners (clients) and to ensure the efficient training of the global model [232].

Despite its privacy benefits, Federated Learning introduces new attack surfaces. Adversaries

can exploit the distributed nature of the system through different capabilities, typically categorized as collusive and non-collusive scenarios. In collusion, two cases can occur where the attacker controls multiple clients and gains greater power in the distributed system. Sever-participant collusion occurs when both the server and benign clients are compromised. Participant-participant collusion, on the other hand, occurs when a subset of clients is used to infer information about others or to compromise the model [232].

Attack Type

An attack type is characterized by the specific features and methods employed in its implementation [219]. These characteristics include the nature of the attacker objective, the time of the attack, and the strategy used to carry out the attack.

Adversarial attacks can be categorized into the objective of the attacker. Usually, these attacks are either classified as targeted or non-targeted (or indiscriminate) based on their specificity [223, 227, 228, 238–240]. A targeted attack aims to manipulate the prediction of a ML model for a specific sample or batch of samples to match a label chosen by the attacker. Non-targeted attacks, by contrast, cause general errors to the ML output without the attacker specifying a particular desired output, meaning that the output just needs to be different from the original and unperturbed [220, 225, 237]. Besides these two main classifications, according to Assion et al. [224], some other sub-type of target attacks include static target, where the attacker forces the model to produce the same output regardless of the input; dynamic target, where specific target classes are selectively removed while the rest of the output remains unaffected; and confusion target, where the overall output distribution remains unchanged but the representation or appearance of specific target classes is manipulated. The attacker can also aim to reduce the confidence of the model regardless of the prediction output, which is known as confidence reduction [241].

The goal of an attacker instead of being based on the output of the model, can often align with security violations that compromise the integrity, availability, or privacy (confidentiality) of a ML-based system. For instance, adversarial attacks can degrade performance (integrity violation), render the model unusable (availability violation), or gain unauthorized access to sensitive information (privacy violation) [8, 225, 242].

Another important aspect that can be used to classify adversarial attacks is the timing of an attack. These attacks usually occur eihter during the training phase or during the inference phase of a ML model. Training-time attacks are further divided into passive (honest-but-curious) attacks, in which the attacker infers knowledge without interfering with the process, and active (malicious) attacks, where the attacker directly manipulates the training process [231, 232].

In order to carry out adversarial attacks, attackers often adopt strategies designed to evade detection. A key factor in these strategies is attack frequency, which can be categorised into two main types: one-step attacks and iterative attacks [243]. One-step attacks optimize adversarial examples in a single interaction, offering efficiency and reduced computational requirements. However, these attacks are usually less reliable against robust models [244]. In contrast, iterative attacks involve multiple interactions, progressively refining adversarial examples for greater effectiveness, but at the cost of increased computational resources [220, 225, 229]. Having this in mind, attacks can be individual or collaborative. Collaborative attacks enhance efficiency and allow attackers to better conceal their actions, although these scenarios are less common [237].

Adversarial attacks exploit ML through a variety of strategies, each tailored to specific objectives. Among these, evasion attacks, also known as exploratory attacks, are commonly employed. These attacks target a model's weaknesses to induce target or indiscriminate prediction errors during the inference phase without altering the training process, thereby compromising the integrity. Due to their exploratory nature, they are often used to gather information about the target model. A frequent example is the input/output attack, where adversarial inputs are provided, and the model's outputs are analyzed to recreate a surrogate model [219, 221, 225, 237]. Creating this model is typically foundational for black-box attacks and facilitates adversarial transferability, which is the ability to craft adversarial examples that can successfully attack different models.

In contrast, poisoning attacks (or causative attacks) involve tampering with training data to disrupt the learning process. This category includes inserting adversarial examples into the training dataset to degrade the model's classification accuracy or alter its predictions to favor the attacker's goals. Within poisoning attacks, backdoor or trojan attacks are notable: these involve embedding a trigger (often named backdoor key) in the data to ensure normal performance on standard data but deliberate misbehavior when the trigger is present. These attacks, which also function as evasion attacks, highlight the dynamic nature of some adversarial methods [225, 226]. Poisoning attacks are further classified into training data modification/injection, label manipulation/flipping, and input feature manipulation [230, 239]. As noted by Rosenberg et al. [219], an add or modify permissions to the training data is a requirement for conducting poisoning attacks.

Another type of attack is logic corruption. Instead of targeting the model's inputs or training data, this attack focuses on the model's architecture itself, and it can modify the model's parameters and hyper-parameters. According to [241], this type of attack is hard to detect and can be very damaging, but it is not very common.

Specific to Federated Learning, Byzantine attacks involve malicious clients submitting invalid updates to derail collaborative training, potentially causing divergence or delays [245]. Similarly, Sybil attacks manipulate the system by allowing a single entity to create multiple active identities, increasing its influence and easing manipulation of the global model [246].

On the privacy front, attackers employ methods such as model inversion, wich can be classified as either membership inference attack, or property inference attack [222]. Membership inference attacks, a prevalent form of privacy attack, aim to determine whether a specific sample was part of the training dataset. This approach shares similarities with exploratory attacks. A similar attack, reconstruction attacks, attempt to recreate training samples and/or their labels, leveraging partial or full knowledge about the model [231]. Property inference attacks seek to uncover secondary attributes unintentionally learned by the model, revealing information about individuals or populations not intended for inference [232].

Model extraction attacks, a type of black-box attack, aim to replicate a target model's behavior or accuracy by extracting its knowledge. These attacks can serve as precursors to other threats, such as adversarial or membership inference attacks, with the attacker often seeking to minimize query complexity during the extraction process [222, 231].

When it comes to generating the adversarial examples that are critical to many of these attacks, attackers use various techniques, including gradient-based, transferability/scoring-based, decision-based, approximation-based, and other specialized approaches. Gradient-based attacks exploit the model's gradients to create perturbations that maximize loss,

making them effective in white-box scenarios [8, 225, 228, 234, 236, 239, 240, 247]. Optimization techniques refine these perturbations, aiming for minimal perturbation while degrading performance of ML models [227, 228, 236, 247]. Search-based attacks, in contrast, do not rely on gradient information, making them suitable for black-box environments [236].

Adversarial perturbations can be categorized into individual, universal, and contextual types. Individual perturbations are tailored to specific inputs, while universal perturbations are dataset-agnostic, facilitating real-world application [220]. Contextual perturbations, as noted by Assion et al. [224], focus on generating agnostic modifications that consistently alter output labels.

Transferability attacks leverage adversarial examples generated on one model to exploit another, while score-based attacks use prediction scores to approximate gradients and create perturbations [225, 228, 229, 234, 240, 247]. Decision-based attacks rely only on output labels and adjust perturbations iteratively using rejection sampling, generating smaller perturbations [225, 228, 234, 236, 240, 247]. Approximation-based attacks, on the other hand, use differentiable functions to approximate the outputs of non-differentiable or randomised layers, allowing gradient-based attacks for evasion [225]. Specialized methods such as generative model-based attacks, geometric transformations, and signal processing-based attacks further expand the arsenal of adversaries, demonstrating the breadth of strategies available [227, 228, 236, 247].

For textual data, perturbations are classified into character-level, word-level, and sentence-level changes, reflecting the unique challenges of adversarial attacks in non-visual domains [223].

Based on the findings of this research, attackers can be categorized according to their level of knowledge about the target model and their capabilities. This classification directly influences which types of attacks - such as evasion, poisoning, model extraction, and privacy attacks - can be effectively carried out. Furthermore, the wide variety of methods used to generate adversarial examples highlight the numerous ways these attacks can be executed.

Therefore, rigorously testing and evaluating the robustness of ML models against a broad spectrum of attacks while considering the diverse objectives and capabilities of adversaries is essencial for ensuring the security and reliability of these systems.

2.3 Adversarial Methods

The ability to generate adversarial examples is critical to the detection of potential threats to any ML model. By simulating these threats in a controlled environment, it is possible to study their impact on the model's behavior and establish the limits of the model's robustness.

Based on the previous research, adversarial attacks can be classified into several categories, depending on the knowledge and capabilities of the attacker and the purpose of the attack. Therefore, it is necessary to conduct research to identify the most common methods for generating these attacks, as well as to understand how these methods are made available for use. This section therefore aims to answer RQ3: "Which strategies are used to generate adversarial attacks?".

2.3.1 Research Methodology

To ensure comprehensive coverage of the various algorithms and methods used to generate perturbations for attacking ML models, as well as to identify their corresponding toolbox implementations, the search query was constructed using the terms listed in Table 2.5. This search was conducted on June 10, 2025, and the terms were queried in the abstracts, keywords and titles of the documents. To identify methods relevant to testing defenses against adversarial perturbations, the search terms included "attack" or "defense". This ensured that selected studies not only addressed these topics but also provided details on how the methods operate, which is crucial for undertanding their differences and effectiveness. Each scope was combined into a search query using AND operators.

Table 2.5: Search terms for RQ3.

Scope	**Terms**
Model	(*"machine learning"* OR *"artificial intelligence"* OR *"deep learning"*)
Purpose	(*attack* OR *defense*)
Strategy	(*strategy* OR *tool**)
Adversarial Perturbation	(*"adversarial example"* OR *"adversarial sample"*)

Since perturbation methods for attacks can be implemented in a variety of ways, inclusion and exclusion criteria were established to ensure that the information collected would highlight the most relevant methods used in the literature, while also providing descriptions or names of these methods. Table 2.6 provides an overview of the defined inclusion and exclusion criteria. These criteria were estabilished in order to find works that summarize adversarial methods and their applications in toolboxes, where the perturbations performed where done in attack/defense scenarios.

Table 2.6: Inclusion and exclusion criteria for RQ3.

Inclusion Criteria	**Exclusion Criteria**
IC1: Articles published from 2020 onward IC2: Available in English language	IC1: Duplicated publications EC2: Does not describes/present adversarial methods/toolboxes EC3: Does not generate perturbations in attack/defenses scenarios EC4: Does not present a summarization of adversarial methods EC5: Full text not available

2.3.2 Findings and Discussion

For RQ3, a total of 196 articles were initially obtained by applying the query to the content published in the selected databases. After removing duplicates and performing the screening phase, 13 articles were considered as they met the defined inclusion criteria (Figure 2.5).

Adversarial methods pose a significant threat to the effectiveness of ML-based applications, as the perturbations they generate are often imperceptible, but are capable of misleading the model into making incorrect classifications [248].

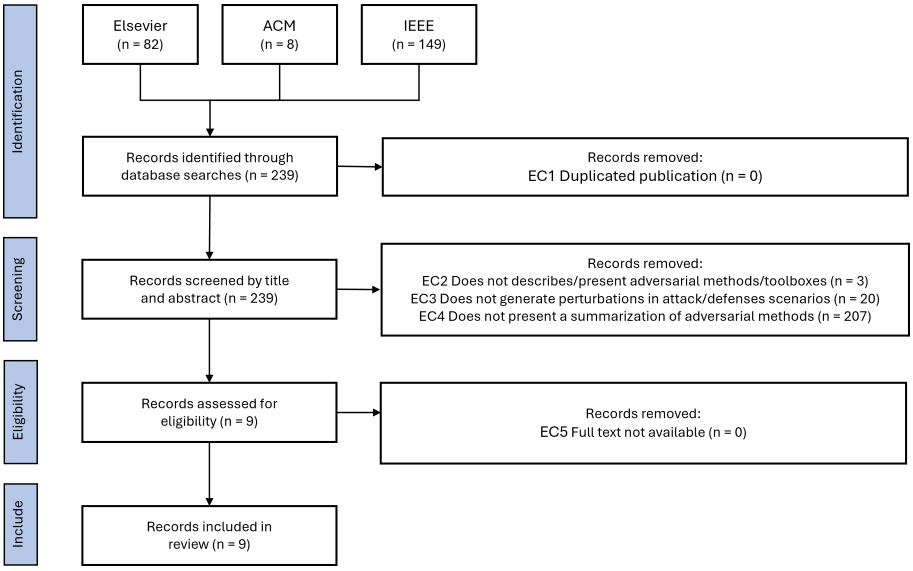

Figure 2.5: PRISMA search process for RQ3.

For text-related perturbations, there are several methods for modifying data for adversarial purposes, as summarized by Qiu et al. [249]. These methods aim to manipulate text while achieving specific adversarial goals, often by preserving semantics or introducing subtle modifications that challenge model predictions.

A common method is to replace words in a sentence with alternatives that retain the original semantics and meaning. This technique ensures that the text remains comprehensible to humans while misleading models. Attacks that implement this method include AdvGen [250], iAdv-Text [251], RewritingSampler [252], DeepWordBug [253], Probability Weighted Word Saliency (PWWS) [254], and Explain2Attack [255].

An alternative method is to replace identified keywords with arbitrary words, which can effectively disrupt model predictions. Similarly, another approach is to append arbitrary words to the original phrase and place them in different positions within the text. These methods are used in attacks such as Metropolis Hastings Attack (MHA) [256], ADDANY [257], ADDANY-KBEST [258], and ADDENT-DIVERSE [259].

One other method of perturbation is to insert or remove words to change the structure of a sentence. Word removal simplifies the text, while word insertion can expand the text to introduce perturbing effects. For example, TextFool [260] uses both insertion and removal to generate adversarial examples, while AdvExpander [261] focuses on expanding phrases through word insertion.

Generating visually similar words is another method used to disguise adversarial attacks, making them more difficult for humans to detect. Attacks such as Textbugger [262] and ADDSENT [257] use this strategy to fool models while maintaining a natural appearance.

Building on the idea of perturbing model predictions with word-level changes, some methods ensure that no matching words from the original text remain in the perturbed version, or require the presence of specific keywords. Attacks such as Seq2Sick [263] use this approach to create adversarial text that meets predefined constraints.

In addition to word-level modifications, character-level methods involve swapping, shuffling, or otherwise altering individual characters to introduce errors into models. For example, DISTFLIP [264] and HotFlip [265] swap characters or tokens to generate adversarial examples. Other character-level techniques include shuffling letters within words (inner or full shuffle), adding symbols (intrusion), removing vowels (disemvoweling), truncating letters, joining words (segmentation), introducing typographical errors, or applying phonetic modifications that preserve pronunciation while changing spelling. These methods are implemented in the attack Zéroe [266].

Another method for generating adversarial examples involves the adding triggers that act as adversarial perturbations. These triggers are designed to generalize across multiple inputs, and often are made to be natural in fluent text. An attack that uses this trigger is Natural Universal Trigger Search [267].

Unlike text-based data, perturbations applied to images and tabular data do not require highly specific modifications because these types of data can often be slightly altered without changing human perception. For example, with image perturbations, individual pixels can be minimally adjusted without being noticeable to the human eye. In the case of tabular data, a small change in a value can lead to prediction errors, but the effect may go unnoticed because the meaning of the data points is not always immediately clear. Several methods for generating such perturbations have been developed, each tailored to different domains and goals. As such, Gao et al. [268], Naderi and Bajić [269], and Priya and Dinesh [270] provide in-depth analyses of image-based adversarial techniques. Techniques for creating adversarial examples through data poisoning are explored by Zhang et al. [271], Guo, Yang, and Song [272] presents attacks specific to the communication domain, and methods specifically targeting the NIDS domain are discussed by Roshan, Zafar, and Ul Haque [165] and Vitorino, Praça, and Maia [273]. For a more general overview of adversarial strategies, studies such as [241, 248, 274] provide a comprehensive perspective. In addition, Vitorino, Praça, and Maia [273] emphasizes the importance of applying constraints to perturbations in tabular data to ensure that these modifications retain realistic properties. The following synthesis provides an overview of these findings and summarizes the various methods and their applications.

One approach to generating adversarial perturbations is to adjust the input data to maximize the model's loss function. This method relies on computing the gradient of the loss function with respect to the input and making adjustments in the direction that maximizes loss. By iteratively updating the input, this method can generate adversarial examples designed to confuse or mislead the target model. A wide range of attack techniques have used this gradient-based method, including Fast Gradient Sign Method (FGSM) [275], Iterative Fast Gradient Sign Method (I-FGSM), also known as Basic Iterative Method (BIM) [276], Project Gradient Descent (PGD) [277], Auto-PGD [278], Fast Adaptive Boundary Attack [279], Momentum-Enhanced Pointwise Gradient (MPG) [280], Black-box Momentum Iterative Fast Gradient Sign Method (BMI-FGSM) [281], Joint Gradient Based Attack (JGBA) [282], Minimal attack [283], Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [34], and Variable Step-size Attack (VSA) [284]. Each method builds on the basic gradient approach in specific ways, enhancing its effectiveness, efficiency, and adaptability to particular challenges and constraints.

Another method for generating these perturbations focuses on identifying and adjusting the most influential features, data points, or pixels in the input to achieve a specific adversarial goal while minimizing the magnitude of the perturbation. Attacks that employ this method,

while also using gradient information are Jacobian-based Saliency Map Attack (JSMA) [285], Carlini and Wagner (C&W) [286], Hierarchical Adversarial Attack (HAA) [287], DeepFool [288], One-Point Attack [289], Houdini [290].

Using this approach to find the most significant features, some attacks can be performed without gradient information, relying in gradient approximations or estimations. Some of these attacks include Zeroth Order Optimization (ZOO) [291], GADGET [292], Optimization attack [293], HopSkipJumpAttack [294], Boundary attack [295], and the attack proposed in [296]. This method is also used in attacks such as One Pixel attack [297] and StrAttack [298], although these attack uses different ways to find the most representative pixel. Some attacks can even use this type of approach without gradient information, such as Advpc [299], Bruteforce Attack Method (BFAM) [300] and the algorithm developed by Guo et al. [301]. Point-detachment attack [302] is another example of an attack that uses this method, but it does not rely on gradient information. Instead, it uses a greedy strategy to remove points that correspond to the true class, in order of their importance. Attacks such as Square Attack [303] also uses a similar approach by adding a random perturbation in each iteration.

Although the previous methods focus on finding the most significant perturbation, to mantain imperceptible, some other methods focus on creating realistic samples that are compatible with the original data. These methods generate perturbations based on the patters of the unperturbed samples, while implementing constraints to ensure that the new data remains compatible with the domain. Adaptative Perturbation Pattern Method (A2PM) [304] and DosBoundary [305] are two of these cases, which were used originally to produce traffic that is both realistic and compatible with the attack scenario.

Regarding adversarial examples specific to images, some other methods can also be used. Instead of changing slightly the pixels, a model can also be tricked by changing or deforming the structure of objects. Some attacks that apply this method are Mesh attack [306], Geometric-aware (GeoA^3) [307], Imperceptible Transfer Attack (ITA) [308], and NormalAttack [309], Shape Prior Guided Attack [310], ShapeAdv [311], Manifold Attack [312]. Similarly, another approach to generate adversarial examples involves overlaying additional information onto the original input, such as patch-based method proposed by Saha, Subramanya, and Pirsiavash [313], which adds adversarial patches on top of the clean images.

Another method to create adversarial examples from images, instead of finding the most significant pixel or using a gradient to understand the model direction, it to apply an universal perturbation. These universal perturbations are made to be added to any image, and that image will be classified as an attacker choosen class. Examples of these methods in attacks can be found in Universal Perturbations for Steering to Exact Targets (UPSET) [314], Autoregressive Perturbations [315], and in the algorithm proposed by Moosavi-Dezfooli et al. [316].

Generative models are another method for generating adversarial data. In this method, a generator network is tasked with generating adversarial samples designed to fool a target model. The generator's loss function is designed to penalize it if a secondary component, a classifier or discriminator, fails to predict the adversary's desired target. This iterative process allows the generator to produce increasingly effective adversarial examples. Several attacks use this approach, including Antagonistic Network for Generating Rogue Images (ANGRI) [314], which generates adversarial images that are designed to induce specific misclassifications while maintaining visual similarity. Generative Adversarial Network (GAN)

[317] are often used in this context, as in GSA-GAN [318], IDS-GAN [319], C-GAN [320], Clean-Label Poisoning Availability (CLPA) [321], and Polymorphic attack [322], based on a Wasserstein GAN. These attacks rely on the interaction between the generator and the discriminator to produce adversarial data that closely resembles the real input, in order to make them effective. Similar to GANs, Conditional Variational Auto-Encoder (CVAE) [323] is a model in which input data affects the distribution of variables, which in turn are used to generate outputs.

For graph-base data, the methods for generating adversarial data can be acomplished by modifying the graph structure itself, such as adding/removing the edges between the nodes, or even changing the node features. The goal of this perturbation is to make the graph appear similar to the original, but causing the model to make incorrect predictions. One attack that supports such method is RL-S2V [324].

As explained in the previous research question, data poisoning involves manipulating the training data. A method to carry out data poisoning may involve switching the labels from the original dataset, which cause the model to make incorrect predictions [271]. Therefore, incorporating the results of the previously mentioned adversarial attacks into the training dataset can lead to a poisoning attack. Another method to carry data poisoning is to create a backdoor, which adds a trigger that is activated during inference when a specific characteristic defined by the attacker is detected by the model [271]. These backdoors are often imperceptible, and some attacks such as those described in [325, 326], exploit image-scaling mechanisms within certain frameworks to enhance stealthiness.

In Federated Learning, as a model is not centralized, the methods for generating perturbations to cause model erros are different from the ones previously described. A method that can be used in this specific case is to perform perturbations in the local model, to affect the global model. Some of these methods are LIE [327], Min Distance Attack (MDA) [328] and Model Poisoning Attack based on Fake clients (MPAF) [329], although MPAF works differently by also generating fake local updates using fake clients.

Several of the attacks described have been implemented in toolboxes, making it easier for anyone interested in generating adversarial examples to use these attacks. As listed by Hu [248], well-known and widely used toolboxes include CleverHans [330], FoolBox [331], AdvBox [332], and Adversarial Robustness Toolbox (ART) [333]. All of these toolboxes are implemented in Python and provide a variety of attack methods, making them valuable tools for both developers and researchers exploring model vulnerabilities and evaluating defenses. Among them, ART stands out for including the largest number of methods. However, all of them provide the most commonly used attacks, such as FGSM, BIM, JSMA, DeepFool, and C&W [248]. Each toolbox has specific functionalities, allowing users to choose based on their needs, whether for simplicity, flexibility, or the range of attack options available.

Adversarial attacks are generated through a various of strategies, including gradient-based methods that modify inputs based on gradients, optimization-based attacks that minimize perturbations while causing misclassifications, and feature-based approaches that target influential input components. Since each method exploits different model weaknesses in the model, testing the model performance against multiple types of perturbations is essential for a comprehensive evaluation of its robustness.

2.4 Chapter Remarks

In this chapter, the research questions have been addressed by means of a systematic review of the scientific literature. Given the variety of approaches used to evaluate the post-attack performance of machine learning models, RQ1 was addressed by conducting a thorough review of the metrics used and analysing their similarities and differences. As different types of attacks provide a broader perspective on model vulnerabilities, RQ2 was addressed by examining the different attack strategies used in the literature to ensure a comprehensive evaluation of model robustness. RQ3 was addressed by investigating the different strategies used to generate adversarial attacks, with a focus on understanding the different perturbation techniques used to generate adversarial examples and how these strategies can affect the model judgement.

The main findings highlight the common use of simple metrics such as Accuracy and ASR (which is closely related to adversarial accuracy). In addition, some studies include metrics that aim to define certified accuracy by identifying decision boundaries beyond which models begin to make prediction errors, such as Interval Bound Propagation IBP Bounds Tightness. Attack types are often categorised based on the knowledge and capabilities of the attacker, as these factors determine which methods the attacker can employ. Perturbations are generated in a multitude of ways, but the most effective attacks typically involve small changes to the original input, targeting the features most relevant to the specific model.

Overall, while some frameworks exist for evaluating models after adversarial attacks, they often lack diversity in the perturbations and metrics used. Therefore, a tool that can generate a range of perturbations and assess model performance across multiple metrics would provide more granular insights. This gap in the current literature is the subject of this dissertation.

Chapter 3

AURORA Design

This chapter presents the design of Adversarial to Understand Robustness and Offensive Resilience Analysis (AURORA), a tool developed to evaluate the robustness of ML models against adversarial attacks.To provide a clear understanding of the system's main components, how they interact, and how the tool is deployed, it outlines the system's architecture and internal logic using the C4 model, which covers logical, sequential, and deployment views. Additionally, the chapter presents the functional and non-functional requirements that guided the development of AURORA, ensuring that the tool effectively meets domain-specific needs.

3.1 Requirements

To understand AURORA, it is important to first comprehend the domain and requirements that shaped its development. AURORA is designed to assess the adversarial robustness of ML models and operates at the intersection of ML and adversarial attack analysis. Its core functionality involves generating adversarial examples through various perturbation techniques and evaluating how well models withstand these attacks.

The tool must accept the dataset, the model to be evaluated, and, optionally, the target labels as input. To create adversarial examples, attacks require access to the model's predictions and generate perturbations around clean data that cause the model to misclassify inputs. The evaluation compares the model's performance/predictions on the perturbed data with its performance on the original data. Some evaluation metrics also require the original labels, which may differ from the model's predicted labels for clean inputs.

Therefore, the backend must efficiently handle these interactions to ensure the accurate execution and evaluation of adversarial attacks. To guarantee consistent and reproducible results across multiple runs, a fixed random seed is used whenever possible. Despite the complexity of adversarial attacks, AURORA is designed to apply these concepts in a straightforward, user-friendly manner without requiring complex input formats or structures.

The system's functional and non-functional requirements can be categorized using the FURPS model, which stands for Functionality, Usability, Reliability, Performance, and Supportability. These requirements are as follows:

- **Functionality:** AURORA must reliably evaluate adversarial attacks using different perturbation techniques, models, and metrics. It should allow users to upload models, clean data, and target labels. It should also provide access to detailed evaluation reports and visual representations of results and perturbed data, and provide the perturbed datasets.

- **Usability:** The Single Page Application (SPA) frontend should provide an intuitive and accessible interface that allows users to easily submit data, initiate evaluations, monitor progress, and download outputs without requiring extensive training or documentation.
- **Reliability:** The system must effectively handle errors and unexpected inputs to ensure accurate and consistent evaluation results. It must persist evaluation data until new analyses are performed and handle attack failures gracefully by notifying users without disrupting ongoing processes.
- **Performance:** Evaluations should be optimized to minimize runtime and make efficient use of model queries. In real time, users should have clear visibility into the status of each adversarial attack and the overall evaluation pipeline.
- **Supportability:** The software architecture must support straightforward maintenance and future enhancements. This includes the ability to easily add new attack methods, evaluation metrics, and reporting formats. Users should also be able to dynamically configure attack parameters during runtime.

Although AURORA includes a SPA frontend and backend, the design primarily focuses on the backend since it handles the core logic and evaluation process, while the frontend was designed to provide a user-friendly interface. Since the backend is composed of a single component, lower-level representations of the system were omitted as they would not add value to understanding the system's architecture.

3.2 Logical View

The logical view illustrates the components of the system and how they are related, providing a comprehensive understanding of its structure and organization. The proposed system, AURORA, is depicted in Figure 3.1. This level provides a high-level overview of the system, which is divided into two main components: the user interface, which is as an SPA, and the backend, which is the Robustness Module.

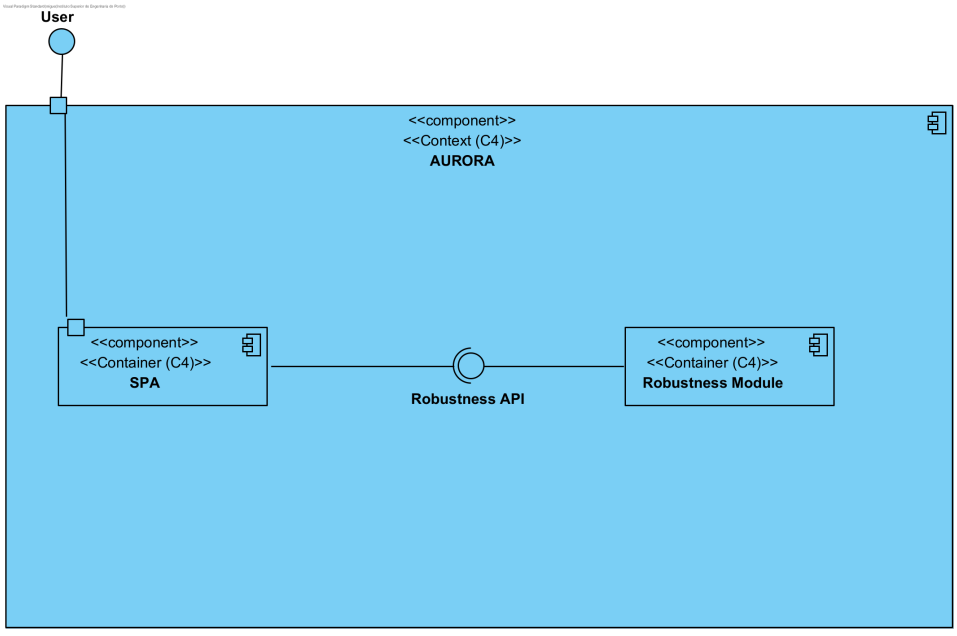

Figure 3.1: Level 2 Logic View.

The backend is structured according to the principles of the Onion Architecture, chosen based on prior software engineering experience. Among the design models considered, it was deemed the most suitable for meeting the system's requirements and ensuring maintainability and scalability. The Onion Architecture is a software design pattern that emphasizes a strong separation of concerns and reduces coupling by organizing the system into concentric layers. Each layer has specific responsibilities, including an Application Business Rules layer that defines the core business logic. However, the system does not have a dedicated domain layer, as it does not require complex domain modeling or entities typically associated with Domain-Driven Design. The focus is instead on the application business rules that define the interactions and operations of the backend component. This layered approach promotes modularity and maintainability, allowing for easier testing and development. Although the system can handle tasks such as saving images and reports, the layers communicate directly using objects rather than data transfer objects. Figure 3.2 illustrates this architectural structure.

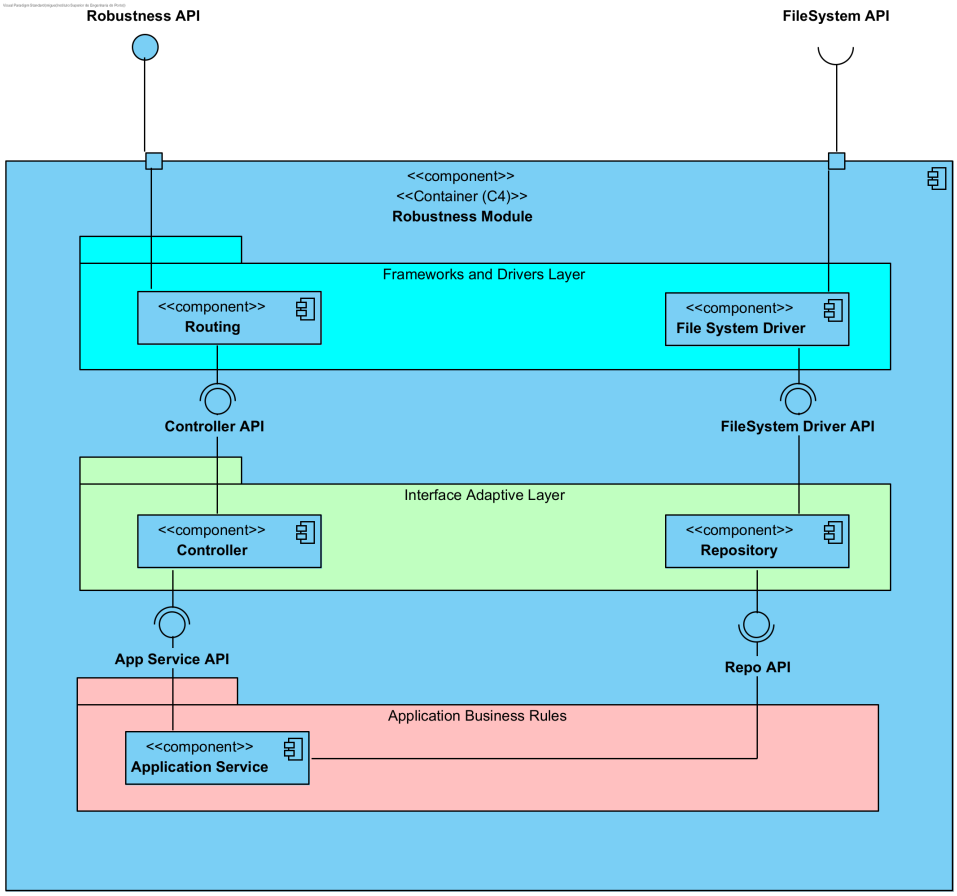

Figure 3.2: Level 3 Logic View.

By adopting standard software design patters, the backend is structured into three distinct layers: Frameworks and Drivers Layer, which handles the routing, user interface communication, and implements the file system driver; the Interface Adaptive Layer, which is responsible for the mediating between routing and the Application Business Rules using a controller pattern, and mediating between Application Business Rules and the file system using a repository pattern; and the Application Business Rules Layer, which contains the core logic of the system, including adversarial attack implementation, evaluation techniques,

and report generation. The backend component is totally independent of the user interface, and can be used directly from its Application Programming Interface (API), allowing users to integrate it into their own applications or systems. The sequential interactions between layers are going to be detailed in the next section.

3.3 Sequential View

The sequential view illustrates the interactions between the user and the system, providing a detailed representation of communication between system components at various levels. Also referred to as the process view, it focuses on the system's dynamic behaviour by depicting how components interact and exchange information over time to perform specific tasks or processes. In this way, the sequential view helps identify the flow of control, the timing of events, the sequence of operations, and the relationships between different components during runtime.

Since the backend of AURORA is designed to be RESTful, the sequential view is represented as a continuous flow of interactions with the users, which can be performed directly through the backend's API or through the more user-friendly SPA component. As each interaction with ML models and adversarial attacks may take some time to complete, the SPA will not require the user to wait for the process to finish, but instead will provide a success or error message when each interaction changes state. The sequential view is divided into three levels, each of which provides a different perspective on the system's interactions.

Level 1 (Figure 3.3) represents the highest level of abstraction, focusing solely on the user's perspective. It illustrates the minimal interaction with the system, specifically highlighting the SPA. This level does not delve into the backend or any other components, providing a clear view of what the user sees and interacts with.

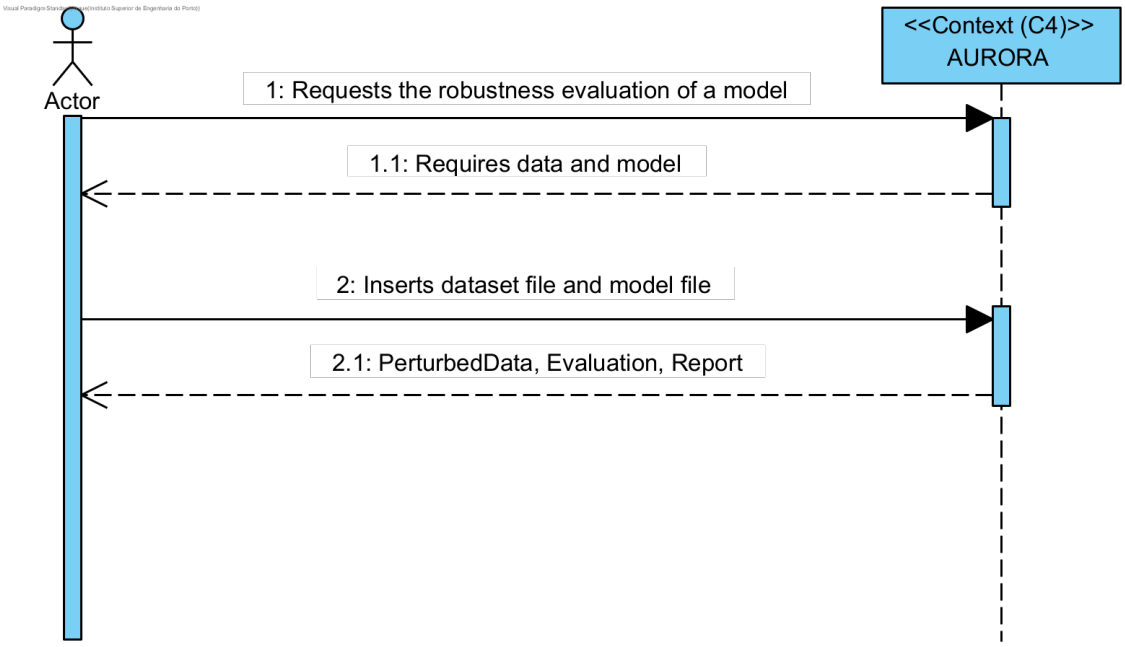

Figure 3.3: Level 1 Sequential View.

Level 2, shown in Figure 3.4, adds more context by illustrating the communication between the SPA and the Robustness Module, the backend of the system. This level provides a

representation of the interactions between system modules, showing how the user interface communicates with the backend to perform tasks such as the evaluation of a ML model robustness. It is also represented the necessary input information, the model and clean data, and the expected output, which is a report containing the results of the evaluation.

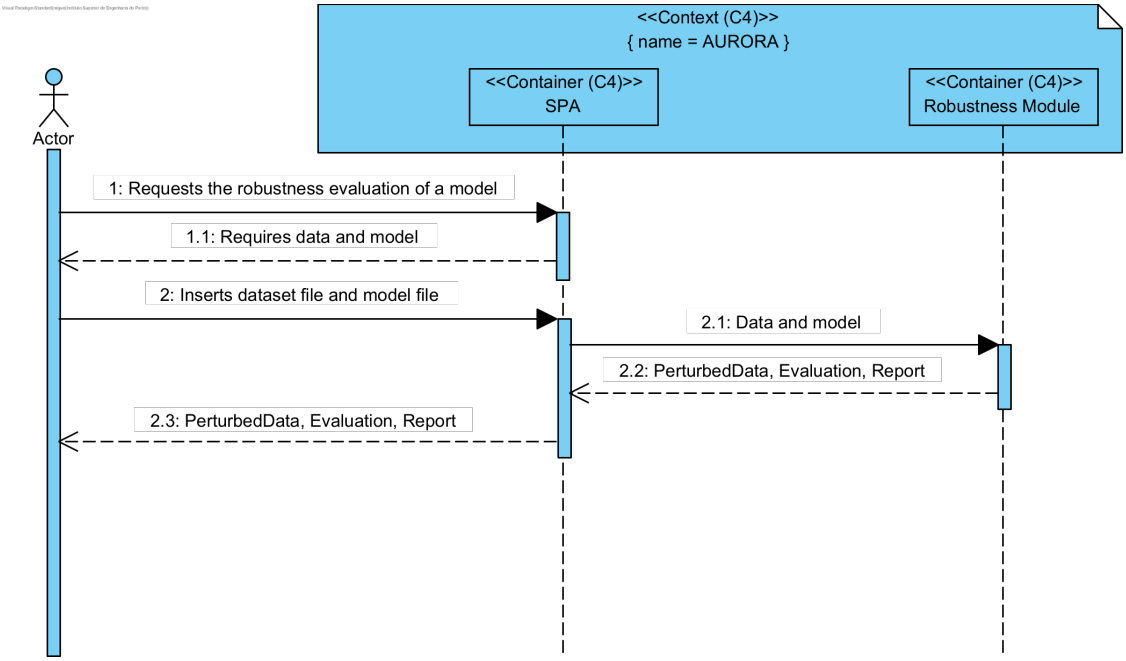

Figure 3.4: Level 2 Sequential View.

Level 3 (Appendix A) provides a more detailed view, illustrating specific interactions between individual components within each container. The controller component mediates between services, and is in charge to pass the necessary information to the service, which in turn performs the required operations.

First, the model, clean data, and target (if applicable) are sent to the service responsible for performing the adversarial attack. Once complete, the service returns the perturbed data to the controller.The controller then forwards the model, clean data, target, and perturbation data to the service responsible for evaluating the model. This evaluation service also applies Distance Adjustment (DA) techniques based on the clean and perturbed data, influencing the model evaluation, which will be addressed in the following chapters. The results are then returned to the controller, which sends them to the service responsible for generating the reports and images. These outputs are then saved in the file system, before being returned to the controller, which sends them back to the user if desired.

As previously stated, these diagrams depict an interrupted sequence, as the backend is designed to be RESTful, but in the actual implementation, the user later is able to receive the report and images once the process has finished, under a different endpoint.

3.4 Deployment View

The implementation follows a client-server architecture, where the backend is an independent, scalable component designed to run on a remote server accessible within a private network. To ensure secure access and data handling, the backend was deployed on a private

server at GECAD for internal use and validation by the team. The SPA resides on the client side and serves as a proof of concept for a user-developed interface that communicates directly with the backend. Figure 3.5 shows a representation of this deployment setup.

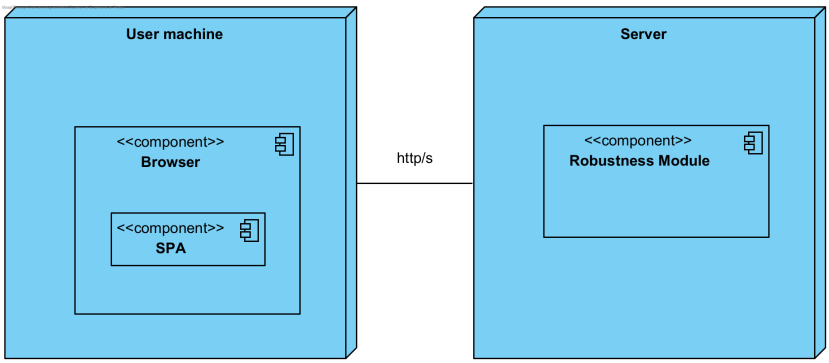

Figure 3.5: Level 2 Deployment View.

3.5 Chapter Remarks

This chapter presented the architectural design of AURORA, emphasizing its simple, modular structure and the functional and non-functional requirements that guided its development. The system comprises two main components: a user interface and a backend, with the latter serving as the core element responsible for executing evaluations and handling all adversarial logic. To promote maintainability, scalability, and clear separation of concerns, the backend architecture follows the Onion Architecture principles , decoupling business logic from external concerns such as frameworks, storage, and interfaces. This makes testing, extension, and integration into other systems easier.

The frontend was primarily developed as a proof of concept with a strong focus on usability and clarity. It enables users to engage with the backend effectively by offering a user-friendly interface for uploading models and data, configuring attack parameters, and visualizing results. Meanwhile, the backend offers a flexible, self-contained API that enables potential use beyond the provided interface.

Overall, the system's design promotes a clean, organized architecture that enables straightforward interactions between components and adaptability for future expansions. AURORA fulfills all the requirements outlined in this chapter, delivering a robust, extensible, and accessible tool for evaluating the adversarial robustness of ML models.

Chapter 4

Model's Adversarial Robustness and Resilience

This chapter introduces Adversarial to Understand Robustness and Offensive Resilience Analysis (AURORA) [1], a open-source tool for assessing the robustness of ML models. AURORA accomplishes this by implementing a variety of adversarial attack methods and testing ML models against the resulting perturbations. The outcomes of these attacks are assessed using multiple evaluation metrics, which enable a comprehensive analysis of model performance under adversarial conditions.

While these perturbations are designed to mislead the model into making incorrect predictions, they are meant to remain imperceptible to humans. To account for this, AURORA uses a distance-based evaluation approach that adjusts the evaluation metrics according to how much the adversarial samples deviate from the original inputs. Specifically, if an attack generates adversarial samples significantly distant from the original data, the evaluation metrics for that attack are penalized. This ensures that attacks relying on unrealistic or easily detectable perturbations have less impact on the overall assessment of the model, resulting in a more meaningful evaluation.

Furthermore, AURORA evaluates the logical consistency of the data. Adversarial samples that break underlying constraints, such as those related to categorical feature logic, are considered farther from the original than valid samples with similar characteristics.

This chapter details the adversarial attack methods employed by AURORA, highlighting their limitations and requirements. The chapter also introduces the evaluation metrics used and provides representative examples. Additionally, it describes briefly a metric adjustment based on the distance between adversarial samples and the original data, a feature integrated into AURORA. Furthermore, the chapter showcases the final report generated by AURORA, which summarizes the results of each perturbation method, provides statistics on the adversarial samples, and includes the model's robustness score. Finally, it discusses the solution's scalability and user interface design, both of which are critical factors for real-world applicability.

4.1 Perturbation Methods

As discussed in Section 2, the evaluation of ML models is a complex task, especially when it comes to assessing their robustness against adversarial perturbations. Considering the wide

[1] `https://github.com/msilva2002/AURORA`

variety of identified perturbation methods, the perturbation methods selected for AURORA were chosen based on the following criteria to ensure meaningful and consistent evaluations:

- The perturbation method must generate a perturbation for each sample individually.
- The perturbation method must return adversarial samples in the form of an array that preserves the original data distribution, that is, the output must maintain the same ordering as the original dataset.
- The perturbation method must be applicable to tabular data in a logical way, ensuring the changes remain meaningful within the context of structured datasets.
- A targeted variant of the perturbation method must be available.

Given these requirements, certain methods, such as those based on GANs, which generate new samples rather than perturbing existing ones, were ruled out. In addition to meeting the specified criteria, methods that employed different approaches to perturbation were selected to ensure methodological diversity. These include model-agnostic approaches, also known as black-box methods, as well as gradient-based techniques suitable for white-box attacks. However, the latter are not universally applicable since not all ML models provide gradient information, so the priority was given to methods that could be applied to any model.

The following sections describe the characteristics of each selected method. For each method, the corresponding toolbox is specified, along with a description of the perturbation technique and its requirements and limitations.

Carlini and Wagner

The C&W method, introduced by Carlini and Wagner [286], is a perturbation technique designed to solve an optimization problem, finding the minimal perturbation necessary to modify a model's prediction while ensuring that the sample remains valid. Although originally designed for image data, this method can be adapted to tabular datasets by treating each feature as a pixel. An implementation of this method is available in the ART [333] toolbox. This method requires full access to the target model and relies on the model providing gradient information, so, consequently, it is incompatible with models such as Light Gradient-Boosting Machine (LGBM) or Random Forest (RF), which lack this feature.

Adaptative Perturbation Pattern Method

Introduced by Vitorino, Oliveira, and Praça [304], A2PM was designed to generate adversarial examples for tabular data. It is unique among other methods as it imposes constraints that are both domain and class specific, particularly with regard to the structure of one-hot encoded categorical variables, an aspect generally neglected by other methods.

The requirement of a configuration pattern by A2PM ensures that the adversarial samples remain within the bounds of the original dataset. Each pattern reflects the feature distribution and includes user-defined parameters, such as the probability of a feature being altered, as well as the ratio and maximum ratio by which continuous features may fluctuate. These ratios are bounded by the minimum and maximum values of the original dataset.

This method operates based on model outputs but requires knowledge of the model's data structure, making it partially black-box. However, this makes it adaptable to API-based models, allowing users to query the model without needing direct access, but rather through

HTTP requests. Furthermore, A2PM is highly configurable and generally agnostic to the model type, as long as tabular input is used. Nevertheless, not all models are supported, and the adaptation to an unknown model, specifically the output format requires some user knowledge.

Relative to other methods, A2PM generates perturbations more efficiently by returning the first successful adversarial sample. However, it does not optimize for the smallest perturbation, and if a successful perturbation is not found, the last attempt is returned. As a result, failed attempts must be reverted back to the original sample. This method is available as a stand-alone perturbation method in [304].

Boundary Attack

A decision based method developed by Brendel, Rauber, and Bethge [295], the Boundary Attack is a black-box method for generating adversarial samples. The process starts with a highly perturbed adversarial input and reduces the perturbation iteratively while maintaining its adversarial behavior.

As a decision-based method, it does not require access to the model's gradients or internal parameters. It relies solely on the model's final predictions, making it fully compatible with API-based systems. The Boundary Attack is effective at generating adversarial examples close to the model's decision boundary, requiring minimal perturbations to cause misclassification. This method is available in the ART library.

One identified limitation of using ART was its lack of support for binary LGBM models. These were handled as black-box classifiers on account of their inaccessible gradient information, which allows them to be used in methods designed specifically for black-box scenarios.

HopSkipJumpAttack

HopSkipJumpAttack, proposed by Chen, Jordan, and Wainwright [294], is a black-box method similar in concept to the Boundary Attack. It differs by estimating the gradient direction using only the model's decision outputs, and so as Boundary Attack, HopSkipJumpAttack can be applied to any model, including those accessed via APIs. This allows it to approximate the decision boundary and apply the minimum necessary perturbation.

Although it is model-agnostic and adaptable to API-based models, due to the boundary mapping process, HopSkipJumpAttack is computationally expensive and very time-consuming, particularly when working with medium to large datasets. Despite this, HopSkipJumpAttack is one of the most effective implemented methods, achieving a consistently high success rate. An implementation is available in the ART library.

Zeroth Order Optimization

Introduced by Chen et al. [291], the ZOO method is inspired by the C&W perturbation method. As with the C&W, it aims to solve an optimization problem that minimizes perturbation while altering the model's prediction. However, unlike C&W, ZOO operates in a black-box setting and relies only on the model's predictions. This method is also similar to HopSkipJumpAttack since it computes an approximation of the model's gradient, but ZOO requires the model confidence scores for each class, rather than just the predicted class.

ZOO approximates the model's gradient using finite differences, similar to techniques used in surrogate modeling for transfer attacks. This makes ZOO suitable for scenarios where gradient access is unavailable. In the case where the outputs are decisions rather than confidence scores, the method can still be applied by treating the decision as a confidence score of 1 for the predicted class and 0 for all others, allowing ZOO to be adaptable to API-based models. An implementation of this method is available in the ART library.

4.2 Evaluation Metrics

As stated earlier, one of the major challenges in measuring the robustness of ML models is selecting the appropriate evaluation metric. Section 2 outlines the wide range of metrics commonly adopted, and so, the most representative and widely used metrics were selected. From these, the final set of metrics are required to provide an independent, objective view of how the model behaves when confronted with adversarial examples, or to provide a basis for comparing the model's performance before and after perturbation. Metrics such as Adversarial Accuracy, MR, ASR, AD were chosen by their relevance and applicability to the problem at hand, while others such as Clean Accuracy, Confusion Matrix, and Time Required were selected for their general utility in evaluating model performance, while providing a baseline for comparison. The following subsections explain how to calculate each metric and any specific requirements for usage.

Clean Accuracy

Often referred to only as Accuracy, Clean Accuracy measures how well a model distinguishes between correct and incorrect class predictions with original, unaltered samples. This metric establishes a performance baseline, which is helpful for evaluating the impact of any adversarial perturbations applied later.

As shown in Table 4.1, where red cells indicate misclassified samples, the model incorrectly classifies two labels. Using the formula introduced in Equation 2.1, the number of correctly classified samples (3) is divided by the total number of samples (5), resulting in a Clean Accuracy of 0.6, which is equivalent to 60%. This metric requires knowledge of the actual/true labels of the input samples, which are not the same as the model's predicted labels. As a general rule, the closer the predictions are to the true labels, the higher this metric will be.

Table 4.1: Clean Accuracy - Actual vs Predicted Labels.

Actual	0	1	2	1	2
Predicted	1	1	2	1	0

Adversarial Accuracy

Conceptually similar to Clean Accuracy, Adversarial Accuracy is calculated using samples that have been modified by an adversarial attack. Much like Clean Accuracy, it depends on the true labels of the original data. For instance, Table 4.2 shows that the model misclassifies three out of five samples, and so the Adversarial Accuracy is calculated as the number of correct predictions (2) divided by the total number of samples (5), resulting in an Adversarial Accuracy of 0.4, or 40%.

Table 4.2: Adversarial Accuracy - Actual vs Predicted Labels after Perturbation.

Actual	0	1	2	1	2
Predicted after perturbation	1	0	2	1	0

However, this metric has a notable setback, as shown in Table 4.3. In this example, the perturbation affects only the first sample, resulting in a correct prediction that improves accuracy rather than reducing it, improving it from 60% to 80% after the perturbation. Though rare, this situation shows that adversarial accuracy can sometimes be higher than clean accuracy. For this reason, Adversarial Accuracy alone does not always reflect the success of an attack. Nevertheless, it remains a useful metric, especially in scenarios such as adversarial training, where post-attack performance is of interest.

Table 4.3: Edge Case: Accuracy Improvement After Adversarial Perturbation.

Actual	0	1	2	1	2
Predicted	1	1	2	1	0
Predicted after perturbation	0	1	2	1	0

Misclassification Rate

Since accuracy alone does not fully capture the effects of perturbation, MR serves as a more representative measure. In practical ML applications, achieving perfect scores on accuracy or other metrics, such as F1, is rare, therefore, the final prediction made by the model often determines how new, unseen information is classified.

The MR metric functions similarly to Accuracy, with one key difference: instead of measuring the proportion of correct predictions, it measures the proportion of predictions that differ between the original and perturbed inputs. This metric only requires the model's original and adversarial predictions (not true labels). The formula for calculating MR is given by Equation 4.1.

$$\text{MR} = \frac{\text{Number of different predictions}}{\text{Total number of examples}} \tag{4.1}$$

As such, MR does not consider whether the perturbed prediction is correct (as shown in Table 4.3), it simply accounts for how many predictions changed. In the example in Table 4.4, only one out of five predictions is different, so, the number of different predictions (1) divided by the total number of examples (5) results in a MR of 0.2, or 20%.

Table 4.4: Misclassification Rate - Before and After Attack Predictions.

Predicted	0	1	2	1	2
Predicted after perturbation	1	1	2	1	0

Attack Success Rate

A targeted attack is a specific type of adversarial strategy in which the attacker attempts to alter the model's output to a specific target class. As such, ASR is similar to the MR, but with an important distinction: ASR measures the success rate of changing predictions to the attacker's specified target class, rather than measuring any change in prediction.

Table 4.5 provides an example of this, where the goal is set to change the model's predicted output to class 1. In this scenario, only three of the five samples can be changed to the target class, and since only one of these was successfully perturbed into class 1, the number of successful perturbations (1) is divided by the total number of possible perturbations (3), resulting in an ASR of 0.33, or 33%.

Table 4.5: Attack Success Rate - Before and After Attack Predictions.

Predicted	0	1	2	1	2
Predicted after perturbation	1	1	2	1	0

Confusion Matrix

A confusion matrix provides a mutitude of representation of a model's classification results, as it serves as the starting point for various derived metrics, such as FP and F1. However, in AURORA, the confusion matrix is specifically designed to reflect the model's predictions before and after adversarial perturbations, rather than requiring the true labels of the samples. This allows for a more intuitive and comprehensive interpretation of how each class's predictions are affected by adversarial attacks.

Time Required

Although it is not directly related to the success or failure of an attack, the time required to generate adversarial samples (or batch) is a valuable metric for evaluating the practicality of different perturbation methods. This metric allows us to compare methods in terms of not only their effectiveness but also their computational cost and efficiency. In time-sensitive scenarios, faster methods may be preferable, even if their success rate is slightly lower. Thus, this metric provides an additional layer of insight when evaluating and comparing perturbation techniques.

Attack Deterioration

As previously stated, AD is a metric derived from Clean Accuracy and Adversarial Accuracy. It reflects the model's performance drop resulting from an adversarial attack, where a larger decrease in Accuracy indicates greater AD, which signifies more impactful perturbations. Like the accuracy metrics it is based on, this metric requires the true labels of the input samples.

For instance, Tables 4.6 and 4.7 illustrate a scenario with a low performance drop, where the Clean Accuracy is 80% and the Adversarial Accuracy is 60%. In this case, the AD is calculated as follows: the difference between Clean Accuracy (0.8) and Adversarial Accuracy (0.6), divided by Clean Accuracy, resulting in an AD of 0.25, or 25%. This indicates a moderate impact on the model's performance due to the adversarial perturbations.

Table 4.6: Low Attack Degradation - Actual vs Predicted

Actual	0	1	2	1	2
Predicted	0	1	2	1	0

Table 4.7: Low Attack Degradation - Before and After Attack Predictions.

Predicted	0	1	2	1	0
Predicted after perturbation	1	1	2	1	1

For the example show in Tables 4.8 and 4.9, the Clean Accuracy of the model is 0.8 and the Adversarial Accuracy is 0.2, resulting in an AD of 75%. As demonstrated, AD provides a more balanced and informative metric by integrating Clean and Adversarial Accuracy, offering a comprehensive view of how the model's performance degrades under adversarial perturbations. While relying solely on the difference between Clean and Adversarial Accuracy may highlight the performance gap, it lacks the context needed to interpret the severity of the degradation. In contrast, AD uses the model's peak clean performance as a reference point, enabling a more meaningful evaluation of the impact of adversarial examples on the model.

Table 4.8: High Attack Degradation - Actual vs Predicted.

Actual	0	1	2	1	2
Predicted	0	1	2	1	0

Table 4.9: High Attack Degradation - Before and After Attack Predictions.

Predicted	0	1	2	1	0
Predicted after perturbation	1	0	2	2	1

By combining two widely used metrics, AD provides useful means for comparing different attack methods. However, when a model achieves a perfect clean accuracy score of 100%, AD becomes equivalent to MR, offering no additional insight in that scenario.

4.3 Distance Adjustment

As previously stated, the most effective adversarial attacks introduce minimal perturbations that successfully change the model's prediction while remaining nearly imperceptible. However, since most existing perturbation methods are designed for image data, the criteria used to assess whether resulting adversarial examples are realistic and imperceptible do not directly apply to tabular data. This is because tabular data has feature-specific constraints and intrinsic properties that require different evaluation approaches.

As such, an adversarial example that is far from the original sample may not be considered good, even if it causes misclassification, as it could be unrealistic or represent an entirely different sample. In these cases, distant examples can be treated as outliers and should not carry the same weight or influence to the final metrics as closer, more realistic examples do.

Figure 4.1 illustrates this concept. The original sample is on the left, and two perturbed versions are on the right: one similar to the original and one substantially different. Both successfully change the model's prediction, from sunny to cloudy, but they are evaluated equally under standard metrics, even though the lower example clearly deviates from the original and is far from imperceptible. This highlights the need for a more precise evaluation approach that considers the distance between original and perturbed samples, ensuring that the evaluation metrics accurately reflect the quality of the adversarial examples.

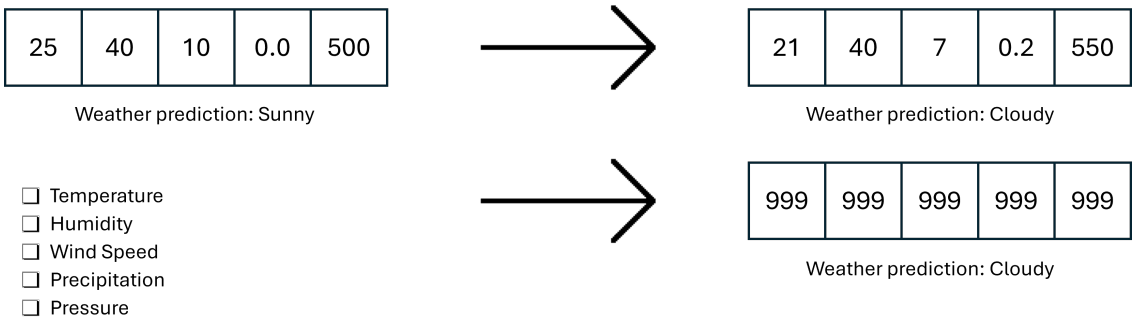

Figure 4.1: Perturbed data example with different distances.

To address this issue, AURORA introduces a Distance Adjustment (DA) method that can be applied to any numerical metric, such as MR and ASR, as this results in more accurate and realistic evaluation scenarios for tabular adversarial examples. Additionally, since some attack methods do not consider feature properties, such as minimum and maximum values, means, or feature-specific logic, this adjustment helps ensure that generated perturbations remain valid. Consequently, the resulting evaluations provide a more trustworthy and realistic assessment of model robustness against adversarial attacks.

The DA computes the distance between the original and adversarial samples using the Euclidean distance formula for numerical features and the Hamming distance formula for categorical features. A threshold derived from the dataset's characteristics is used to determine if an adversarial example is an outlier, remaining consistent when the same data is subjected to different perturbation method. If an adversarial example's combined distance exceeds this threshold, it is considered less realistic and its contribution to the evaluation metric is reduced proportionally. The farther an adversarial example is from the original sample, the greater the penalty applied to the metric, ensuring that overly unrealistic adversarial examples have a reduced impact on the final evaluation. A more detailed explanation of this adjustment is provided in Section 5.

4.4 Report of Robustness

To provide a comprehensive overview of the results, AURORA generates a Markdown-formatted report that includes all metrics calculated during the evaluation process, as well as statistics for the original and perturbed samples. The report is designed to be easily readable and interpretable, allowing users to quickly understand the effect of adversarial perturbations on the model's performance.

The report includes visual representations of the original and perturbed samples for each perturbation method. These images help users understand the nature of the perturbations by showing which features were modified and how. The visualization also aids in determining

whether the modified values remain valid. Additionally, the report presents the calculated distance between each original and perturbed sample, the threshold used to identify outliers, and the penalty applied to invalid perturbations, providing a complete and transparent view of the evaluation process.

Lastly, the report presents two distinct robustness evaluation scenarios based on three metrics and objectives: ASR, MR, and AD. ASR measures the proportion of samples successfully perturbed to reach the target class. MR captures the number of misclassified samples, regardless of the target class. AD offers a more balanced view of the model's performance degradation under adversarial perturbations. Since ASR and MR can be adjusted based on distance to account for perturbation realism, their distance-aware versions, DA-ASR and DA-MR, are used to provide a more reliable robustness evaluation.

The first scenario, referred to as the standard robustness evaluation, averages the metrics calculated across all perturbation methods. This approach provides an overview of how robust the model is against adversarial attacks and reflects its typical performance under everyday adversarial conditions. It serves as the most common method for evaluating overall robustness.

The second scenario, the worst-case robustness evaluation, identifies the most effective perturbation method for each metric, providing a more pessimistic assessment of the model's robustness. While this view may be less realistic, by simulating a critical situation, it provides a more cautious and comprehensive understanding of the model's vulnerabilities, recommended in contexts involving critical decision-making.

The robustness evaluation is divided into five different robustness levels, ranging from 0 to 100, where 0 represents the worst possible robustness and 100 represents the best possible robustness. The levels are defined as follows:

- **80-100**: Very robust.
- **60-80**: Robust.
- **40-60**: Moderately robust.
- **20-40**: Weakly robust.
- **0-20**: Not robust.

4.5 User Interface

The application's user interface is designed to be intuitive and user-friendly, enabling users to effortlessly navigate through its features. The main menu (Figure 4.2) provides quick access to essential functionalities, including executing various adversarial attacks, adjusting configurations, and viewing detailed evaluation metrics related to applied perturbations. Users can also download a detailed report - described in the previous section - which summarizes the evaluation results, and track the progress of the evaluation process step by step.

Using AURORA to execute adversarial attacks requires uploading up to three files: a model saved in joblib format, a dataset to be perturbed, and a target file (Figure 4.3). Only the dataset is mandatory, as it enables the application to use the API to perform the perturbation methods. If a model is provided, then the attacks are directed at it. Omitting the target file, however, will prevent targeted variants of the perturbation methods from being executed, which may lead to an inaccurate final robustness evaluation.

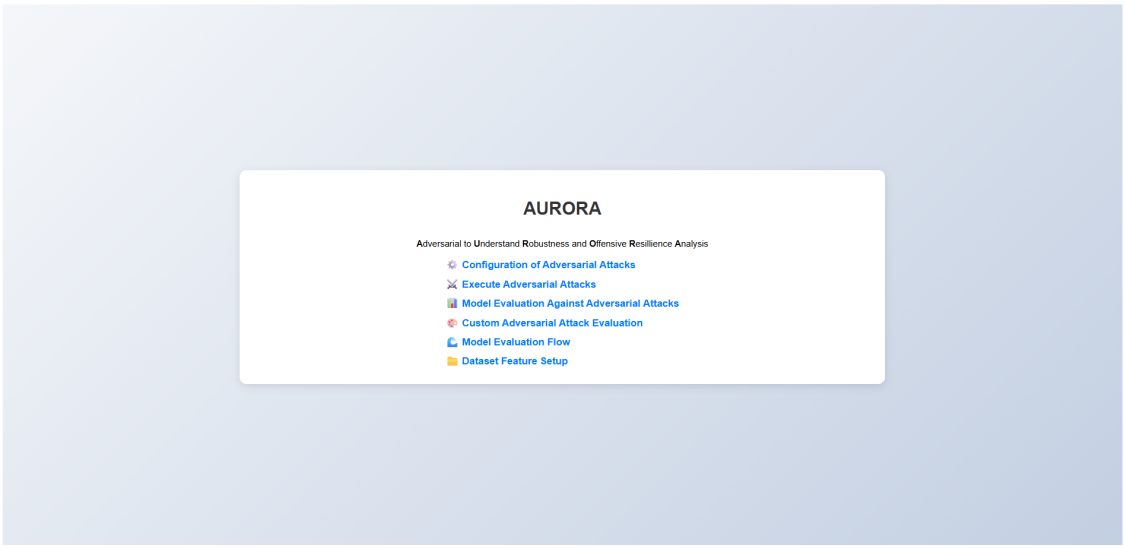

Figure 4.2: AURORA Main Menu.

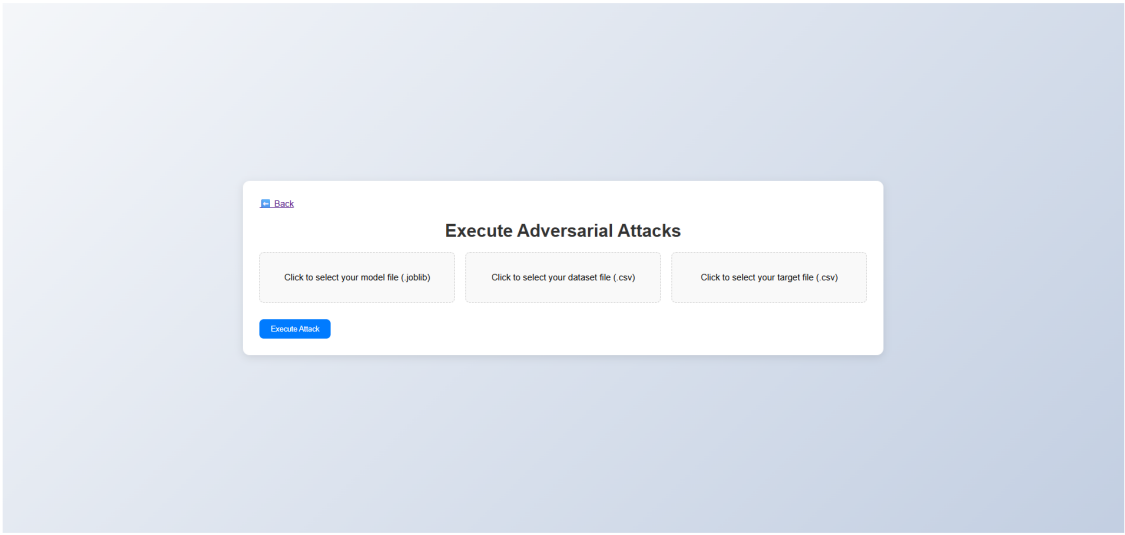

Figure 4.3: Execution of Adversarial Methods.

Different adversarial methods and datasets require distinct configurations with specific parameters. The application features an intuitive interface that allows users to quickly modify these settings (Figure 4.4). When users save their changes, the new configuration replaces the current one in memory, ensuring that subsequent evaluations use the updated parameters. Users can reset the configuration to its default values, which reloads the original settings from the local configuration file. The interface supports uploading and downloading configuration files in JSON format, making it easy to import and export configurations as needed.

As shown in Figure 4.5, the users can monitor the status of each method. The possible states include: "Loaded", "Running", "Finished", and "Evaluated", and "Evaluated and Adjusted". There are also error messages, which detail if the model is incompatible with a particular method/configuration. Once all methods have been executed and the metrics

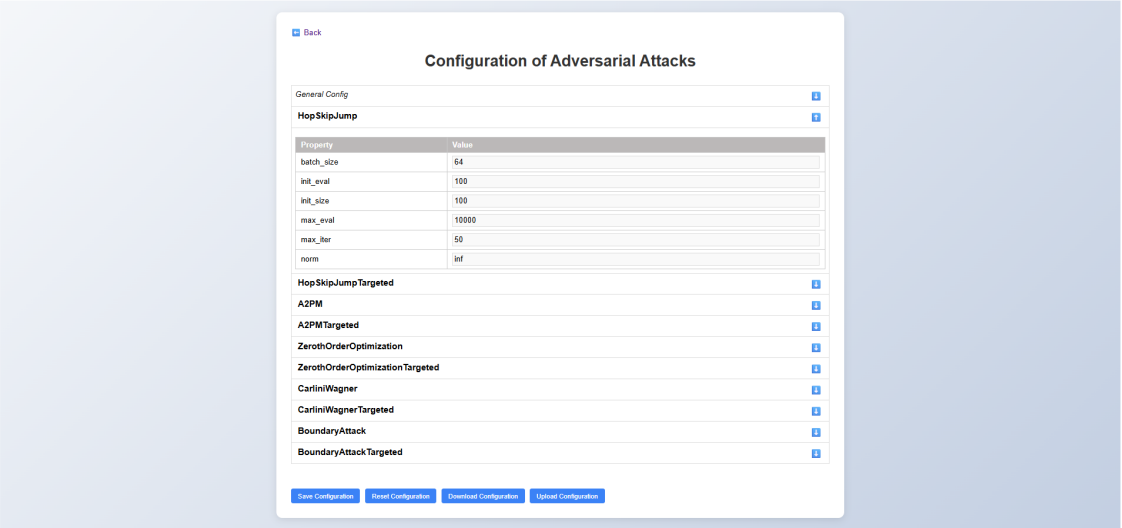

Figure 4.4: Configuration menu.

adjusted, the report detailing the statistics of the generated data and the robustness evaluation is available for download. Additionally, data generated during a method's execution can be downloaded once the method reaches the "Running" state.

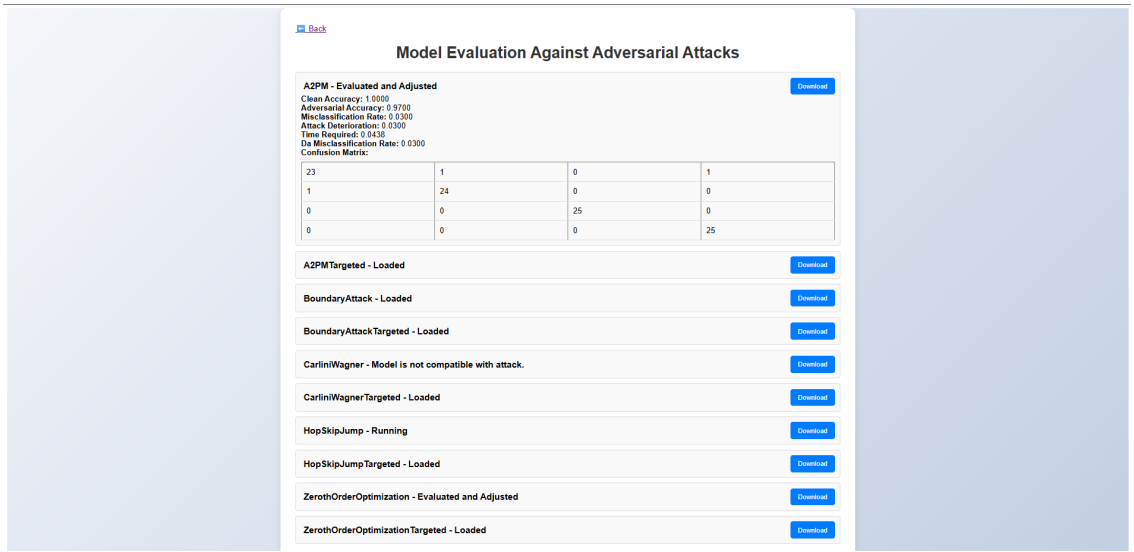

Figure 4.5: Model Evaluation Against Adversarial Attacks.

Users who prefer to use their own perturbation methods can evaluate their data via the menu shown in Figure 4.6. In addition to the usual perturbation information, users must provide the perturbed dataset, created by the user custom perturbation. This allows for the evaluation of submitted data, with the results displayed under the "Custom" section of the evaluation menu.

Since these custom perturbations can differ significantly from other methods, this evaluation can only be performed when no other evaluations are running. Consequently, no report is generated since a single method is insufficient for a comprehensive and thorough robustness assessment, besides not necessarily respecting the requirements set previously. If the custom

method is relevant to the overall evaluation and has not yet been included in the tool, users can add it by following the procedure described in the next section, on how to extend the solution with new perturbation methods and evaluation metrics.

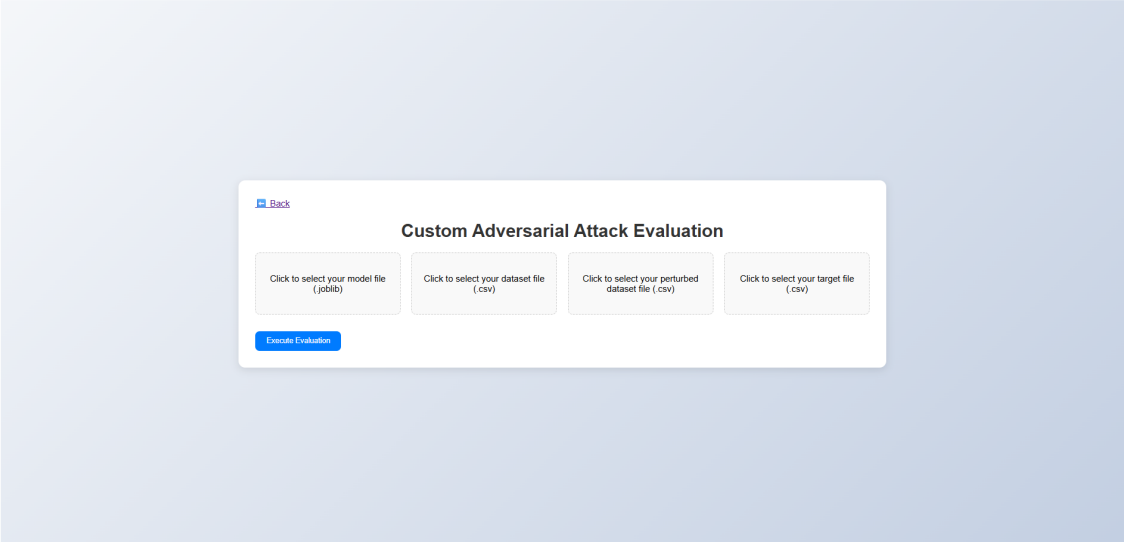

Figure 4.6: Custom Adversarial Attack Evaluation.

The AURORA evaluation methodology is structured into several key phases, each designed to assess the model's robustness against adversarial attacks. These phases include identifying the model's initial predictions, generating adversarial samples under various attacker knowledge (white-box, gray-box, and black-box), and evaluating attack objectives, such as integrity, availability, and privacy. Additional considerations, such as maintaining data quality in tabular formats and adjusting evaluation metrics, are also addressed. Figure 4.7 provides a visual summary of this process and helps contextualize the overall flow of the evaluation pipeline.

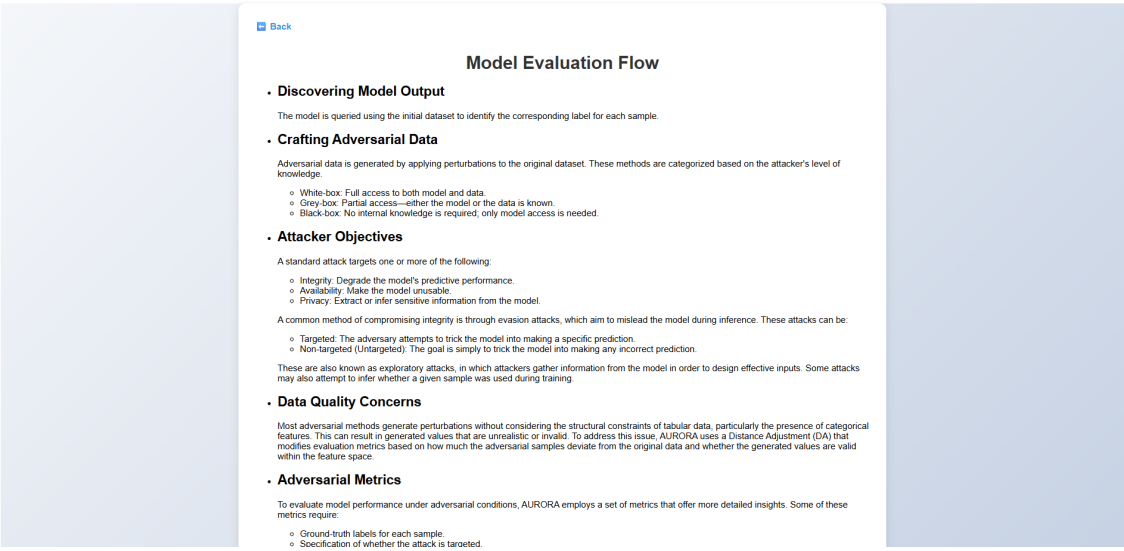

Figure 4.7: Model Evaluation Flow.

Figure 4.8 shows the dataset setup, which is used to specify the categorical features within the dataset. If multiple features share a common prefix in their names and are represented

across different columns, the user can use the wildcard "*" for autocomplete. This indicates that all features with that prefix are grouped together and associated with the same categorical feature.

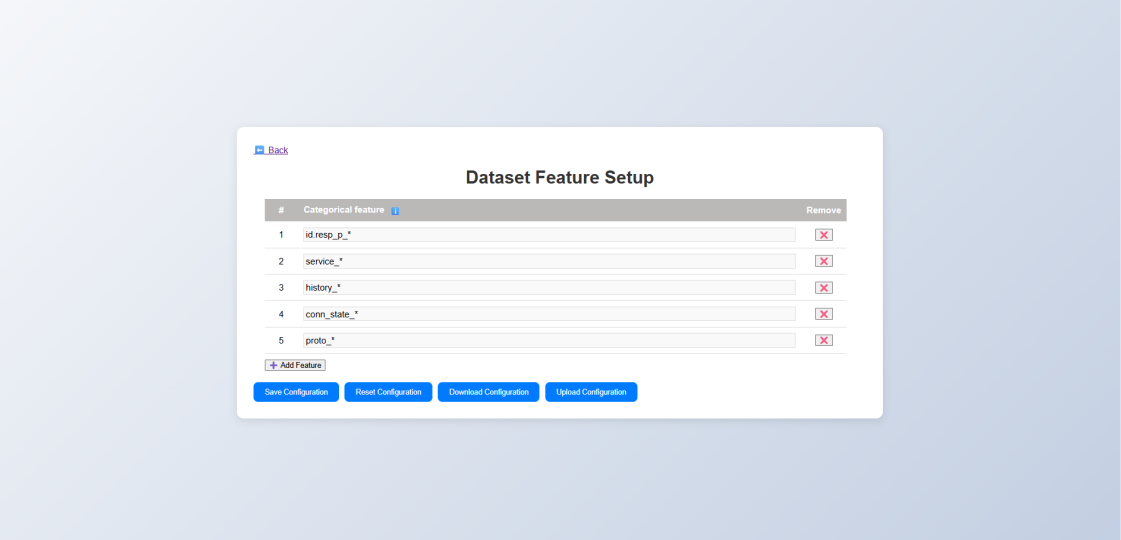

Figure 4.8: Dataset Feature Setup.

4.6 Scalability of the Solution

Since robustness evaluationin AURORA is based on the model's performance against adversarial perturbations, it is essential to ensure that the system supports easy integration of new adversarial attacks and evaluation metrics. This flexibility is especially important, considering the variability among model domains and the possibility that new methods could offer more relevant assessments of robustness. As illustrated in Figures 4.9 and 4.10, both the evaluations and attacks folders contain template files within the project structure. These templates implement a common interface that defines the expected input and output formats, ensuring consistent communication between components and so guaranteeing that the application will continue to function correctly even when new methods or metrics are added.

This design allows users to easily add any methods missing from the current solution. In addition to implementing the perturbation method, users should include the corresponding default configuration in the config/config.json file to ensure the method is available and functional by default.

When adding a new evaluation method, it is essential to perform thorough testing, such as unit testing, to verify its correctness and compatibility with the rest of the system. Figure 4.11 shows an example of the required testing process. After adding a new evaluation or perturbation method, it must be imported into the controller (Figure 4.12). Once imported, it will be passed to the corresponding service for initialization.

The solution can also be used to test remote models based on APIs, where only the output is accessible. This restricts the range of applicable methods to fully black-box, although the user should know the input data required to a successful model prediction. Figure 4.13 illustrates a simple example of how this can be implemented. Users should modify this class

src > attacks > attack_template.py > AttackTemplate > execute

```python
import zope.interface
from attacks.attack_error_handler import AttackErrorHandler
from attacks.attack_interface import AttackI
from domain_data.model_data import ModelData
from domain_data.perturbed_data import PerturbedData
from config.configuration import Configuration
import time

@zope.interface.implementer(AttackI)
class AttackTemplate:
    _attackName = "Attack Name"
    def execute(self, modeldata: ModelData) -> PerturbedData:
        # retrive neccessary data from ModelData (columns, dataset, target, model)
        try:
            # convert data to numpy array if necessary
            # get configuration for the attack
            # initiate the attack and pass the configurations if necessary
            # start the timer
            start = time.time()
            # execute the attack, passing the target if necessary
            #adversarial_data = attack.generate(x=data)
            # end the timer
            end = time.time()
            # return the perturbed data
            #return PerturbedData(attackName=self._attackName, perturbations=pd.DataFrame(adversarial_data), run_time=end-start)
        except Exception as e:
            # in case of an error, handle it using the error handler
            return AttackErrorHandler().handle_error(attackName=self._attackName, targeted=False, error=e)

```

Figure 4.9: Implementation of adversarial perturbation method template.

src > evaluations > evaluation_template.py > ...

```python
import zope.interface
from evaluations.evaluation_interface import EvaluationI
from domain_data.evaluation_data import EvaluationData


@zope.interface.implementer(EvaluationI)
class EvaluationTemplate:

    evaluationName = "Evaluation Name"
    adjustable = False

    def execute(self, evaluationData : EvaluationData) -> EvaluationData:
        #try:
            # check if perturbation was targeted or not (can be necessary in some evaluations)
            # retrive necessary data from evaluationData
            # perform the evaluation
            # add the evaluation to the evaluationData
            #evaluationData.add_evaluation(self.evaluationName, evaluationResult)
        #except Exception as e:
        #    pass
        return evaluationData
```

Figure 4.10: Implementation of evaluation template.

to include necessary connection constraints, such as authentication or data formatting, to tailor the solution to specific system requirements.

4.7 Chapter Remarks

This chapter presented AURORA, a tool designed to evaluate the robustness of ML models against adversarial attacks in tabular data. It outlined the constraints applied to the perturbation methods, described the implemented attack techniques, and detailed the evaluation metrics used to assess model performance under adversarial conditions. Most of the selected methods are available through ART, although some have alternative versions in other

```python
import unittest
from domain_data.perturbed_data import PerturbedData
from evaluations.evaluation_attack_success_rate import AttackSuccessRate
from domain_data.evaluation_data import EvaluationData
import pandas as pd

class TestAttackSuccessRate(unittest.TestCase):
    def setUp(self):
        predicted_perturbed_labels = pd.DataFrame([0, 1, 0, 3, 0, 1, 0, 3, 0, 1])
        target = pd.DataFrame([1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
        perturbed_clean_labels = pd.DataFrame([0, 1, 0, 3, 0, 2, 0, 3, 0, 3])

        perturbedData = PerturbedData(None, None, None, True, "", False)
        self.evalData = EvaluationData(perturbedData, None, perturbed_clean_labels, predicted_perturbed_labels, target)

    def test_attack_success_rate_correct(self):
        print("Testing Attack Success Rate")
        evaluationData : EvaluationData = AttackSuccessRate().execute(self.evalData)
        self.assertEqual(evaluationData.get_evaluation().Evaluation_Name[0], "Attack Success Rate")
        self.assertEqual(evaluationData.get_evaluation().Value[0], 0.2)

        self.assertNotEqual(evaluationData.get_evaluation().Evaluation_Name[0], "Clean Accuracy")
        self.assertNotEqual(evaluationData.get_evaluation().Value[0], 0.7)

    def test_attack_success_rate_no_target(self):
        print("Testing Attack Success Rate with no target")
        self.evalData.perturbedData.targeted = False
        evaluationData : EvaluationData = AttackSuccessRate().execute(self.evalData)
        # evaluationData.get_evaluation() should be an empty DataFrame
        self.assertEqual(evaluationData.get_evaluation().empty, True)
```

Figure 4.11: Attack Success Rate unity test.

```
src > controllers > robustness_controller.py > start_attack_sequence
14  from attacks.zoo_attack import ZerothOrderOptimizationAttack
15  from attacks.zoo_targeted_attack import ZerothOrderOptimizationAttackTargeted
16  from attacks.cw_attack import CarliniWagnerAttack
17  from attacks.cw_targeted_attack import CarliniWagnerAttackTargeted
18  from attacks.a2pm_attack import A2PMAttack
19  from attacks.a2pm_targeted_attack import A2PMAttackTargeted
20  from attacks.pgd_attack import ProjectedGradientDescentAttack
21  from attacks.pgd_targeted_attack import ProjectedGradientDescentAttackTargeted
22
23  from evaluations.evaluation_clean_accuracy import CleanAccuracy
24  from evaluations.evaluation_adversarial_accuracy import AdversarialAccuracy
25  from evaluations.evaluation_attack_success_rate import AttackSuccessRate
26  from evaluations.evaluation_misclassification_rate import MisclassificationRate
27  from evaluations.evaluation_confusion_matrix import ConfusionMatrix
28  from evaluations.evaluation_time import EvaluationTime
29  from evaluations.evaluation_attack_deterioration import AttackDeterioration
30
31  from reports.report import ReportCreator
32
33  attackClasses = [ZerothOrderOptimizationAttack, CarliniWagnerAttack, A2PMAttack, HopSkipJumpAttack, ProjectedGradientDescentAttack]
34  attackClassesTargeted = [ZerothOrderOptimizationAttackTargeted,CarliniWagnerAttackTargeted, A2PMAttackTargeted,
35                           HopSkipJumpAttackTargeted, ProjectedGradientDescentAttackTargeted]
36
37  evaluationClasses = [CleanAccuracy, AdversarialAccuracy, AttackSuccessRate, MisclassificationRate, AttackDeterioration, ConfusionMatrix, EvaluationTime]
```

Figure 4.12: Evaluation and perturbation methods import.

libraries. The ART implementations were preferred due to their active maintenance and widespread adoption within the research community.

Due to most of the perturbation methods being designed for image data, a domain constraint version of these methods was implemented to ensure that the generated adversarial examples remain valid for tabular data. Besides preserving the data's integrity, this approach also shows the adaptability of the tool to implement new perturbation methods, as it can be easily extended to include additional methods or evaluation metrics.

Although not all perturbation methods, some of those implemented do not handle boolean values correctly. To address this issue, AURORA converts boolean columns to integers, which does not affect the dataset's integrity but ensures compatibility with these methods. This conversion prevents errors that would otherwise arise when applying perturbation methods designed for image data, which often do not expect boolean values.

src > repositories > query_repository.py > ...

```python
import zope.interface
from repositories.query_interface import QueryModelI
import pandas as pd
import requests

@zope.interface.implementer(QueryModelI)
class QueryModel:

    def predict(self, df: pd.DataFrame):
        predictions = []
        for _, row in df.iterrows():
            try:
                features = row.values.tolist()
                response = requests.post(
                    "http://127.0.0.1:8080/predict",
                    json={'features': features},
                    timeout=10
                )

                if response.status_code == 200:
                    pred = response.json()['prediction'][0]
                    predictions.append(pred)
                else:
                    predictions.append(None)
                    print(f"Error for row {_}: {response.json().get('error')}")

            except Exception as e:
                predictions.append(None)
                print(f"Failed to process row {_}: {str(e)}")

        return predictions
```

Figure 4.13: API testing example.

Overall, the presented solution provides a comprehensive framework for evaluating the robustness of ML models against adversarial attacks in tabular data. The implemented perturbation methods, while effective in generating subtle changes, often struggle to maintain the validity of tabular data due to their origins in image-based domains. This highlights the need for careful adaptation when applying these methods to different data types.

Chapter 5

Realism of data

As most of the existing perturbation methods were created to perturb images, they do not take into account the specific constraints of tabular data. This results in adversarial samples that are not valid, as they do not respect the intrinsic meaning of the features, or either the perturbation size is too large, making the generated samples unrealistic. For instance, a perturbation method thought to perturb pixels in an image might generate a value too far off from the original value in a numerical feature, or it might change a categorical feature to a value that is not present in the dataset, change multiple columns in the same row, or even change a column to a value that is not binary, such as -1 or 2. These issues can lead to adversarial samples that are not valid and do not represent realistic dat. This can significantly impact the performance of ML models evaluated on such samples, as models should be tested using possible and realistic data; otherwise, the results will not be representative of the model's performance in real-world scenarios.

To address this issue, a distance adjustment metric is proposed, which ensures that the generated adversarial samples are realistic and valid while also considering the distance of the perturbation to the original sample. This adjustment is applied to the existing metrics, allowing them to still be used while ensuring that invalid or unrealistic perturbations do not influence negatively the evaluation of the model's performance.

This chapter delves into the proposed DA approach implemented in AURORA, which addresses the challenge of generating adversarial samples that effectively deceive ML models while remaining realistic within the constraints of tabular data. The images presented in this chapter are generated using AURORA and are created for each perturbation method, showcasing the changes made to the original data and are included in the robustness report.

5.1 Distance of Numerical Features

As numerical features do not have specific constrains, the distance can be measured with a simple approach. A common way to measure the distance between two numeric points in space is to use the Euclidian distance [334], which in the case of the distance between two columns (the original column and the perturbed), results in a 1 dimension perspective, calculated by the difference between the two values. As the final perturbed value can be lower than the original, the distance is calculated using the raised square difference, to ensure an absolute value, as it is shown in Equation 5.1.

$$\text{Distance}_{\text{numerical}} = \sqrt{(\text{Original value} - \text{Perturbed value})^2} \tag{5.1}$$

Using this formula makes it is possible to understand how much each numerical feature changed from its original value, resulting in a measure of distance. AURORA also generates images such as Figure 5.1, which helps visualize the changes by showing the original values as a blue dashed line and the changed values as a solid green line. Both sets of values are adjusted to the same scale, which makes it easier to compare differences between features and highlighting the most affected ones.

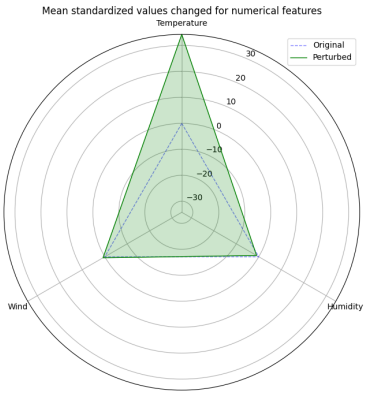

Figure 5.1: Numerical standardized feature variation.

5.2 Distance of Categorical Features

Categorical features have specific contraints and usually cannot/should not be used directly in ML, so a common pre-processing technique, one-hot encoding, is used [335]. This technique converts the categorical variable into a set of binary indicators, where the variable that matches the category in that entry will be set to 1, while all the others will be set to 0, as it can be seen in Figure 5.2.

In some applications, all indicator values may be 0, which usually implies that the corresponding category is missing from the one-hot encoding. In other words, the observation belongs to a category that is not represented by any of the encoded columns. As this case limits both the perturbation methods and the understanding of the dataset, it is assumed that the samples given have all the columns, and this case does not happen.

Weather
Rain
Clear
Cloudy
Rain

→

Weather_Rain	Weather_Clear	Weather_Cloudy
1	0	0
0	1	0
0	0	1
1	0	0

Figure 5.2: One-hot encoding technique.

An approach to calculate the distance between the original and the perturbed sample of a categorical feature, is to use the Hamming distance algorithm [334]. This algorithm allows for a simple computation and results on the understanding of how many columns changed their category value. As this algorithm takes into account the number of columns existing

for a specific categorical feature, but in reality the category should only atmost be changed once, independently of how many options it has, a slight adjustment was made to make sure that, for example, a categorical feature with 5 possibilities will return the same distance as a categorical feature with 30 possibilities, assuming that both were sucessfully changed and the new value is indeed possible. In the example shown in Table 5.1, two values change, Weather_Rain from 0 to 1 and Weather_Cloudy from 1 to 0, resulting in a distance of 2.

Table 5.1: Hamming distance algorithm.

	Weather_Rain	Weather_Clear	Weather_Cloudy
Original	0	0	1
Perturbed	1	0	0

As previously stated, most of the perturbation methods were created to perturb images, and as images do not have the logic of categorical features, these methods when applied to tabular data can create invalid perturbations. The identified invalid cases are:

- The category is not binary, not 0 or 1, including negative values.
- More than one column can have a value different of 0, for the same row.
- All columns are set to 0. Although this can happen, the perturbation methods implemented do not account for such cases and therefore they are treated as invalid.

To ensure that the data used to evaluate the models is valid, a penalty system is employed, where for each invalidity on each sample, it is multiplied for a custom set penalty value, which is added to the final Hamming distance between the two samples.

As an example, in Figure 5.3 it is possible to observe some statistics of a given dataset, where the maximum value for each column is 1 and the minimum is 0 (if this value would be invalid it would be highlighted in red). Regarding the average values of one-hot encoded columns, the sum of the average values for each categorical feature column must equal to 1 in order to be considered valid. Additionally, the total value counts across these columns must match the number of rows in the dataset. When these conditions are met, the average values and the value counts will have similar proportions, as shown in Figure 5.3. In this example, the sum of the average values equals 1, and the total value counts equal 7, corresponding to the number of rows in the dataset since it represents the original, unperturbed data.

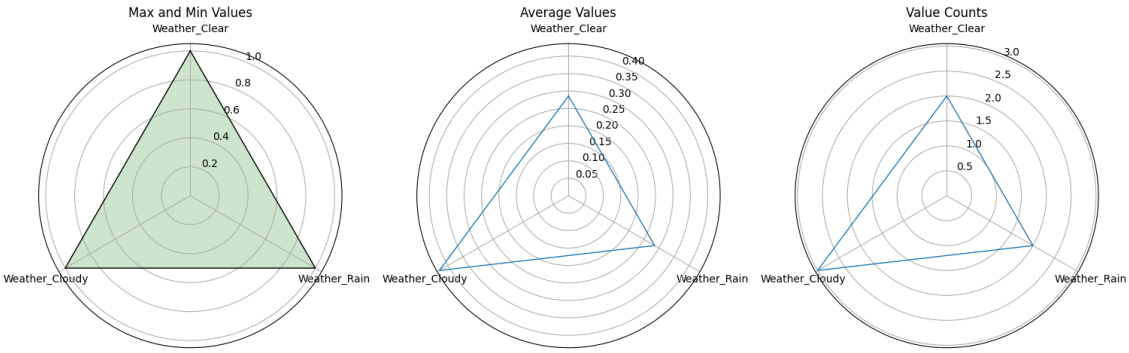

Figure 5.3: Original feature data properties.

Figure 5.4 shows a perturbed version of the dataset that has the same valid statistics as the one in Figure 5.3. Although the feature values have shifted to different columns, the overall

structure is still valid. The total value counts across the one-hot encoded columns sum to 7, which corresponds to the number of rows in the dataset. Similarly, the average values sum to 1 for each column group, reflecting the same proportions as the value counts despite having shifted positions. These results demonstrate that the perturbed dataset maintains the expected one-hot encoding properties while representing a modified, yet structurally valid, input.

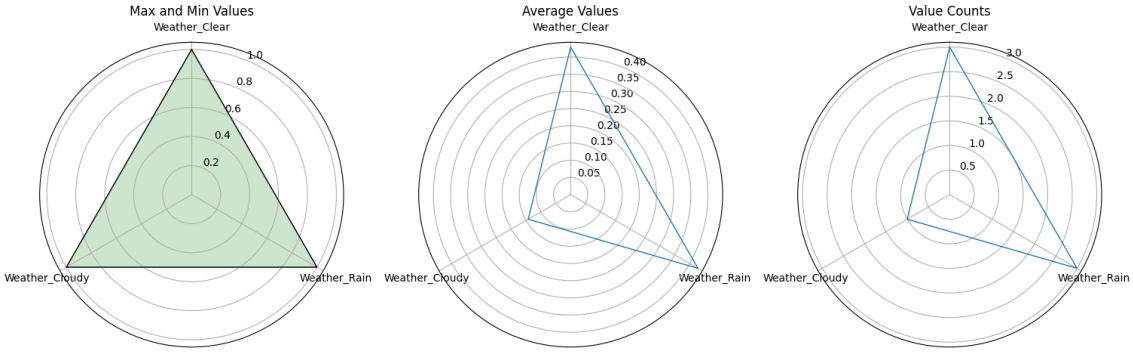

Figure 5.4: Perturbed feature data properties.

Figure 5.5 shows an invalid example of a perturbed dataset. In this case, the minimum and maximum values are no longer binary, as indicated by the red highlighting. Additionally, more than one column within the same categorical feature contains non-zero values, contradicting the one-hot encoding constraint. This inconsistency is evident in the mismatch between the value counts and the average values, as the proportions no longer align.

Furthermore, the total sum of the value counts is 11, which is greater than the number of rows in the dataset and clearly indicates a discrepancy. The presence of values greater than or less than 0 or 1 across the adversarial samples also causes the total average across columns to deviate from 1.

This example illustrates an invalid perturbation that disregards the structural constraints of categorical features by treating them as continuous numerical variables. Consequently, while the perturbed values may appear close to the original data, they do not conform to the conditions necessary for valid categorical encoding. Therefore, such perturbations are considered invalid.

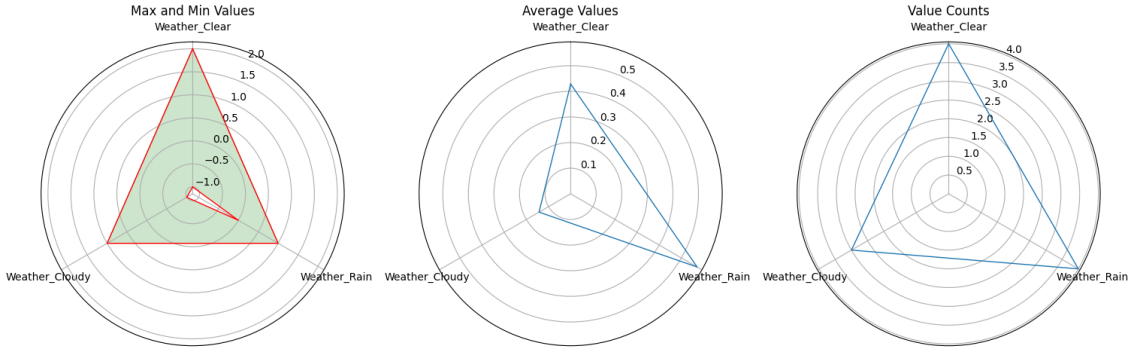

Figure 5.5: Invalid perturbed feature data properties.

The images above illustrate three states of the dataset: the original data (Figure 5.3), a valid perturbed version that preserves encoding constraints (Figure 5.4), and an invalid perturbed version that lacks these constraints (Figure 5.7). However, these examples do

not illustrate cases in which categorical values shift between columns, that is, when a value changes from one category to another or vice versa. Figures 5.6 and 5.7 illustrate how many times each column changed due to the perturbations. For instance, changing the value of the Weather_Rain column from 0 to 1 counts as one change, as does changing it from 1 to 0. These changes are calculated using the Hamming distance algorithm, however, the penalization for invalid perturbations is not applied in this case, since the focus is solely on counting changes rather than their validity.

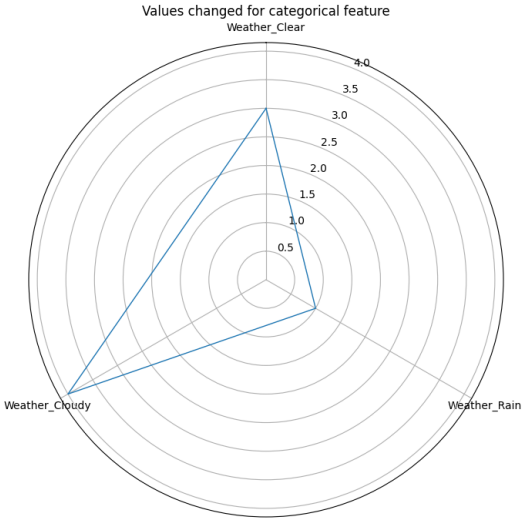

Figure 5.6: Categorical feature variation.

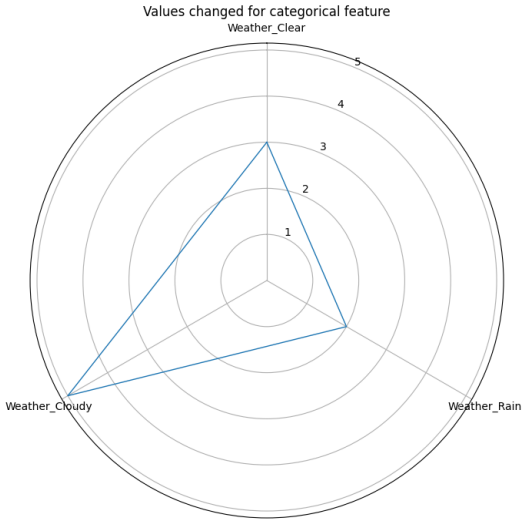

Figure 5.7: Categorical feature variation for invalid perturbation.

For the valid perturbation (Figure 5.6), the total number of changed values is 8. Since the Hamming distance counts each change in a column as one modification, whether it changes from 0 to 1 or from 1 to 0, the sum of 8 changes corresponds to modifications in 4 out of 7 rows. In contrast, the invalid perturbation (Figure 5.7) has 11 changes, exceeding the number of rows. This indicates that multiple columns in the same row have non-zero values, which breaks the one-hot encoding constraint. Furthermore, the fact that the total is an odd number highlights the inconsistency, as such a sum cannot occur with valid one-hot encoded data.

5.3 Metric Adjustment

With the distance measured from the original to the adversarial samples properly calculated, it is now possible to adjust any existing numerical metrics, such as MR and ASR, to ensure that these represent accurate results of the models, by using possible and realistic data. To this extent the following requirements were placed:

- The final metric value cannot be reduced to zero if the original value was not zero. This prevents the complete loss of meaningful information where it originally existed.
- There must be a defined threshold distance beyond which the original metric starts to become affected. This ensures the metric behaves reliably in the presence of small adversarial perturbations, preserving the original value when the input remains close to the unperturbed data.
- The value of the original metric should remain unchanged or decrease as a result of adversarial perturbations and must never increase. This reflects the principle that adversarial changes are intended to degrade the input or reveal model vulnerabilities, not improve performance. Allowing the metric to increase in such cases would undermine its reliability in evaluating robustness.
- The metric should degrade gradually as the distance of the perturbation increases. This allows for a smooth, interpretable transition from unaltered to fully affected values rather than sudden drops.

Keeping these requirements in mind, the threshold was set to allow one change per categorical feature (a Hamming distance of two per feature), and to match the median perturbation for each numerical feature. Doing this allows the threshold to account for some room to change; otherwise, a successful perturbation might not be possible.

As previously mentioned, the Euclidean distance is calculated for each sample across all numerical features and the Hamming distance is computed for each categorical feature to measure the distance between the original data and the successful perturbations. To reduce the influence of outliers, the median distance is determined for each feature across all samples. Finally, the median distances across all features are summed to produce an overall distance score.

The overall distance is compared to a predefined threshold specific to the original data. If the difference between the distance and threshold, Equation (5.2), is equal to or less than the threshold, the metric remains unchanged. When the distance exceeds the threshold, two penalty cases are applied, as shown in Equation (5.3). For moderate exceedances, where the difference is less than or equal to the threshold, the metric is reduced linearly in proportion to the excess. For larger exceedances, where the difference is greater than the threshold, a gradually slowing penalty is applied that approaches an asymptote at 90% of the original metric value. This approach ensures that the metric reflects the degree of perturbation while preserving information about the original data and complying with established requirements. The final adjusted metric, referred to as the DA Metric, is calculated as shown in Equation (5.4), where Metric represents the original numerical value of the metric before applying the distance adjustment.

$$x = \max(0, \text{distance} - \text{threshold}) \tag{5.2}$$

$$\text{Distance adjustment} = \begin{cases} 1, & \text{if } x = 0 \\ 1 - \dfrac{0.9x}{\text{threshold}}, & \text{if } 0 < x \leq \text{threshold} \\ 1 - \left(1 - \dfrac{\text{threshold}}{20\left(x - \dfrac{\text{threshold}}{2}\right)}\right), & \text{if } x > \text{threshold} \end{cases} \tag{5.3}$$

$$\text{DA Metric} = \text{Metric} * \text{Distance adjustment} \tag{5.4}$$

Figure 5.8 shows a visual representation of Equation (5.3), with the threshold set to 3 as an example. The adjustment factor remains at 1 when the perturbation distance is within the threshold, indicating no penalty. When the distance exceeds the threshold but is less than or equal to twice the threshold, the adjustment factor decreases linearly down to 0.1 (10% of the original metric). For larger perturbations beyond twice the threshold, the adjustment approaches an asymptote near 0.1, ensuring that the metric never reaches zero.

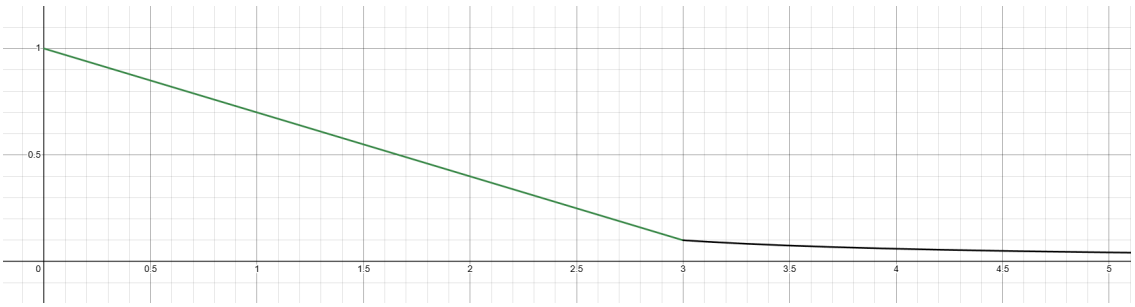

Figure 5.8: Representation of metric penalty based on the distance.

5.4 Chapter Remarks

This chapter introduces a method for ensuring the validity and realism of generated tabular data. The proposed Distance Adjustment (DA) method improves upon existing metrics by taking into account the size of the perturbations and how well they align with valid data constraints.

The DA relies on detailed knowledge of the dataset's structure, including the data type of each feature. To provide accurate adjustments, it also requires a representative dataset that covers the full range of possible values for each feature, as well as sufficient variability. Without it, the adjustment may not fully capture the realism of adversarial samples based on perturbation size.

The DA depends on having complete knowledge of the dataset's structure, including the type of each feature. It also requies the user to provide a representative dataset that covers the range of possible values for each feature to ensure a accurate adjustment. Without sufficient data variability, the adjustment may not accurately reflect the realism of adversarial samples.

Overall, the DA provides a more comprehensive and reliable evaluation of model robustness by considering both perturbation size and data validity. Integrating this adjustment into existing metrics leads to more meaningful and accurate assessments of ML model performance against adversarial perturbations, ensuring that the results reflect the model's robustness in real-world scenarios.

Chapter 6

Robustness Case Study

This chapter presents a case study on the robustness of two ML models trained for multiclass classification using the same dataset. The study evaluates the performance of each model under adversarial perturbations using AURORA to measure their robustness. In addition to performance metrics, we present and discuss the robustness score of each model provided by AURORA.

6.1 Study Configuration

Two models were considered for the case study: a CatBoost and a Multilayer Perceptron (MLP). These models were chosen to showcase the results on a standard ML tree-based model and a Neural Network model.

The study was conducted on common hardware: a machine with 16 gigabytes of random-access-memory, an 12-core central processing unit, and a 12-gigabyte graphics processing unit. The pre-process and model training implementation relied on the Python 3 programming language and several libraries: *numpy*, *pandas* and *scikit-learn* for data preparation and manipulation, including the fine tuning, *catboost* and *tensorflow* for the implementation of the CatBoost and MLP respectively, and *joblib* to serialize the models.

6.1.1 Data Pre-processing

The requirements for the dataset used in this study are simple. The dataset must be tabular, appropriate for classification tasks, and contain categorical features. The GeNIS dataset [15] meets these requirements, as it is a tabular dataset designed for network intrusion detection, containing both numerical and categorical features. Additionally, GeNIS supports multiclass classification, covering multiple types of cyber-attacks as well as benign traffic, enabling models to distinguish between different attack categories and typical network behavior.

This dataset was developed to accurately represent network traffic in small and medium-sized organizations, which are often targeted by cyber-attacks but may lack extensive security resources. It includes various types of cyber-attacks such as Denial of Service (DoS), Reconnaissance, and Bruteforce attacks, as well as their specific subcategories. The dataset captures different aspects of network flows through features such as ports and protocols, as well as a variety of statistical measures, making it well-suited for training and evaluating machine learning models for NIDS.

Since the dataset includes multiple types of labels (BinaryLabel, CategoryLabel, and SubCategoryLabel), it was necessary to select one for model training. For this study, CategoryLabel was chosen as the prediction target. This decision was based on two main considerations.

First, the CategoryLabel supports multiclass classification, offering a more challenging and informative evaluation of model robustness. Second, several classes in the SubCategoryLabel have very few samples, which can lead to class imbalance and hinder training and evaluation. Table 6.1 presents the traffic statistics for the 60-second version of the dataset, illustrating the distribution and characteristics of the data used in this study.

Table 6.1: Statistics of GeNIS dataset for 60 seconds flows.

Label	Flows	SubCategory labels
DoS	295640	DoS-udp, DoS-icmp, DoS-pushack, DoS-slowloris, DoS-hulk.
Benign	27150	Benign-background, Benign-users, Benign-admin.
Recon	27733	Recon-nmap, Recon-dns.
Bruteforce	18033	Bruteforce-ssh, Bruteforce-smb, Bruteforce-ftp.

As datasets often have noisy data, missing values, and a lack of value diversity [16], feature selection techniques are widely used to identify and retain the most informative and relevant features. This process can reduce dimensionality and often leads to faster training times and simpler models. However, this efficiency gain may result in a slight reduction in predictive accuracy or model performance [17]. Due to the large number of features in the GeNIS dataset, a feature selection process was applied as a pre-processing step to streamline the models before evaluating their robustness against adversarial attacks. Five techniques were used: Information Gain, Dispersion Ratio, Mean Absolute Deviation, Recursive Feature Elimination, and Chi-Squared.

The dataset was split into training and testing sets in the standard 70/30 ratio, and the feature selection methods were applied exclusively to the training set. During pre-processing, features containing negative values or representing random noise were removed, and non-target labels were excluded, resulting in a final set containing 79 features, with each column of a categorical feature treated as an individual feature.

For each feature selection method, the features were ranked based on their importance scores. These scores were then normalized to a range between 0 and 1, ensuring that the total sum of the scores across all features equaled 1. Due to one-hot encoding, the categorical variables were initially split across multiple columns. To allow for an easier interpretation, the scores from these columns were summed for each original categorical feature. The top ten features identified by each method are listed in Table 6.2. Of the five methods, only Recursive Feature Elimination considered categorical features relevant since the others, such as Information Gain and Dispersion Ratio, rely on statistical properties like variance or distribution, which, one-hot encoded categorical variables do not express meaningfully. The top five features were selected from each method to train the models, and after removing duplicates, the resulting set of unique features is shown in Table 6.3.

6.1.2 Model Configuration

As previously mentioned, a CatBoost model and an MLP model were selected for this study. The CatBoost is a gradient boosting algorithm designed to efficiently handle categorical features, often delivering strong performance with minimal pre-processing. In contrast, an MLP is a type of feedforward neural network that can learn complex, nonlinear patterns from data.

Table 6.2: Feature selection methods top 10 features.

Information Gain	Dispersion Ratio	Mean Absolute Deviation	Recursive Feature Elimination	Chi-Squared
Sdaddr	DstWin	Offset	State	Load
TotBytes	SrcWin	DstWin	Flgs	Offset
SrcBytes	DstLoad	SrcWin	Proto	SrcLoad
Dport	Ssaddr	Load	DstLoss	DstLoad
SIntPktMax	SrcLoad	SrcLoad	SrcLoss	DstWin
RunTime	Load	DstLoad	pLoss	SrcWin
Mean	DIntPktMax	SrcJitter	Loss	Sdaddr
Sum	SIntPktIdl	Ssaddr	DstJitAct	SrcJitter
Dur	SrcJitter	Sport	Sport	Dport
Max	DIntPktAct	Sdaddr	SrcJitAct	Ssaddr

Table 6.3: Selected features.

Sdaddr	Load	SrcWin	Dport	Proto
Ssaddr	DstLoss	SrcBytes	Offset	
DstLoad	SrcLoss	SIntPktMax	State	
SrcLoad	DstWin	TotBytes	Flgs	

The CatBoost model's configuration is detailed in Table 6.4. When evaluated on the test dataset using macro-averaged metrics, it achieved perfect scores for accuracy, precision, recall, and F1, as shown in Table 6.5. These results highlight CatBoost's effectiveness for this classification task, although they also raise suspicions about potential overfitting, given the perfect scores across all metrics.

Table 6.4: CatBoost model configuration.

Parameter	Value
Learning rate	0.1
Depth	6
Iterations	1000

Table 6.5: CatBoost model results.

Metric	Value (%)
Accuracy	100.0
Precision	100.0
Recall	100.0
F1-score	100.0

The MLP model is a type of neural network composed of multiple fully connected layers. This implementation consists of two hidden layers with ReLU activation functions and a

softmax-activated output layer designed for multiclass classification. The MLP model's detailed configuration is summarized in Table 6.6.

Table 6.6: MLP model configuration.

Parameter	Value
Hidden layers	2
Neurons per layer	64
Activation function	ReLU (hidden), Softmax (output)
Optimizer	Adam
Loss function	Sparse Categorical Crossentropy
Epochs	10
Batch size	32

Since neural networks typically require properly scaled input data, the numerical features were standardized using the StandardScaler prior to training. The rounded, macro-averaged metrics for the MLP model are presented in Table 6.7. These results indicate that the MLP model also achieved near-perfect performance, with accuracy, precision, recall, and F1-score all exceeding 99%. This suggests that the model is well-suited for the classification task at hand, although similar to the CatBoost model, it raises concerns about potential overfitting due to the high performance across all metrics.

Table 6.7: CatBoost model results.

Metric	Value (%)
Accuracy	99.97
Precision	99.99
Recall	99.98
F1-score	99.97

6.2 Results and Discussion

This section presents the results of subjecting the models to adversarial attacks, along with the robustness score for each model, provided by AURORA. The models were subjected to adversarial perturbations using various methods, including A2PM, Boundary Attack, C&W, HopSkipJumpAttack, and ZOO, including constrained and unconstrained versions of the attacks. A balanced sample of 25 examples per class was extracted from the test set to perform the attacks, for a total of 100 examples for perturbation. Since class 0 represents benign flows, targeted attacks were designed with this class as the objective.

CatBoost

The configuration setting for each method were standard, except for the A2PM method. For this method, the probability of perturbation was adjusted to 0.8 for every feature with exception of Offset. As the feature values ranged from low to high values, the ratios were set from 0.001 to 0.2 in each numerical feature. The C&W perturbation method was not applicable to this model as it requires gradient information, which is unavailable for this model. As the clean samples had specific constraints regarding their features, the

integer features were rounded to the closest integer value, and the categorical features were turned into valid values. Due to the extremely high median feature values in the samples, the threshold for this dataset was set high, at 1277308. The results obtained from each perturbation method are summarized in Table 6.8, where C is tabular data constrain, T is targeted, CA is Clean Accuracy, AA is Adversarial Accuracy, TR is Time Required, D is Distance, and DA ASR and DA MR are the distance adjustment ASR and MR.

Table 6.8: Adversarial perturbations results for CatBoost model.

	T	C	CA	AA	ASR	MR	AD	TR	DA ASR	DA MR	D
A2PM	✗	✓	1.0	0.99	-	0.01	0.01	0.26	-	0.0063	1800154
BoundaryAttack	✗	✗	1.0	0.27	-	0.73	0.73	112.91	-	0.73	474
BoundaryAttack	✗	✓	1.0	0.91	-	0.09	0.09	106.92	-	0.09	4334
HopSkipJump	✗	✗	1.0	0.24	-	0.76	0.76	83.66	-	0.76	802
HopSkipJump	✗	✓	1.0	0.95	-	0.05	0.05	83.42	-	0.05	7954
Zoo	✗	✗	1.0	1.0	-	0.0	0.0	0.67	-	0.0	nan
Zoo	✗	✓	1.0	1.0	-	0.0	0.0	0.67	-	0.0	nan
A2PM	✓	✓	1.0	1.0	0.0	0.0	0.0	0.09	0.0	0.0	nan
BoundaryAttack	✓	✗	1.0	0.97	0.04	0.03	0.03	10.01	0.04	0.03	470
BoundaryAttack	✓	✓	1.0	0.99	0.0	0.01	0.01	8.70	0.0	0.01	27882
HopSkipJump	✓	✗	1.0	0.97	0.04	0.03	0.03	9.13	0.04	0.03	482
HopSkipJump	✓	✓	1.0	1.0	0.0	0.0	0.0	8.11	0.0	0.0	nan
Zoo	✓	✗	1.0	1.0	0.0	0.0	0.0	1.32	0.0	0.0	nan
Zoo	✓	✓	1.0	1.0	0.0	0.0	0.0	1.52	0.0	0.0	nan

Out of the four untargeted types of perturbation methods, ZOO was the only adversarial attack unable to generate any successful adversarial example. This is likely due to the perturbation method itself that required a large number of queries to the model in order to make an approximation of the gradients, which was not possible in this case. The other three methods, Boundary Attack, HopSkipJump, and A2PM, were able to generate adversarial samples, but with varying degrees of success.

Regarding A2PM, this method generated only one adversarial example (as MR is 0.01 from 100 samples), but with a high perturbation distance of 1800154, which is significantly larger than the threshold of 1277308. This suggests that although the perturbation created a valid sample, it was not realistic. That is because the method was designed to generate valid perturbations of tabular data.

The Boundary Attack and HopSkipJump methods without constraints achieved high levels of perturbation success, with MR of 73% and 76% respectively, with low distance values of 474 and 802, which are significantly lower than the threshold. Due to the high threshold value, the penalty for unvalid perturbations were not very meaningful, not showing any difference in the DA metrics. However, the constrained versions of these methods (which apply the data constraints to the generated examples), achieved much lower MR values of 9% and 5% respectively, with perturbation distances of 4334 and 7954, which although still low, are much higher than the non-constrained versions. This indicates that the approximations to create the valid perturbations not only showed that the model is being tricked by impossible values, as the approximation of the numerical features carry a higher perturbation distance.

The targeted attacks, on the other hand, did not show the same results. A2PM was not able to generate any adversarial example, which indicates that the untargeted attack effectively

changed a sample with a benign class to a sample with an attack class. ZOO was not able to generate adversarial examples either, which is likely due to the same reasons as in the untargeted attack.

Regarding the Boundary Attack and HopSkipJump targeted attacks, both achieved much lower results than the untargeted attacks, with MR of 3% and 4% respectively, although similar distances. As with the untargeted attacks, the constrained versions achieved a lower ASR, while having a higher distance. The HopSkipJump constrained was not able to generate any adversarial example, which indicates that the untargeted attack followed the same pattern as A2PM and perturbed benign labeled samples into attack labeled samples.

When comparing the two top performance methods, Boundary Attack and HopSkipJump, the Boundary Attack achieved higher results in the constrained version, while HopSkipJump achieved almost always better results in the non-constrained version. Although sharing similar results, the Boundary Attack created samples with smaller perturbations, as indicated by the distance metric.

Overall, the results indicate that the CatBoost model is specifically vulnerable to untargeted adversarial attacks, particularly when the perturbations are not constrained, but it is more resilient to targeted attacks.

As stated earlier, the robustness of the model was evaluated based on three primary objectives: resilience to adversarial perturbations, resilience to targeted adversarial attacks, and overall deterioration of model performance. Regular resilience was assessed using the MR metric after the DA, targeted resilience was measured using the DA ASR metric, and performance deterioration was quantified with AD, which was set to not adjustable. Using the average values obtained in a standard scenario, the model achieved a robustness score of 88 (Figure 6.1), which is considered to be very robust.

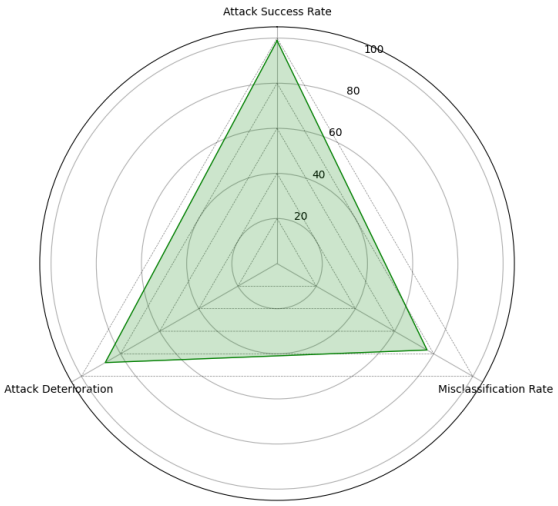

Figure 6.1: CatBoost robustness score.

In a worst case scenario, where it is considered the model to be subjected to the most effective adversarial attacks (only the better performing perturbation method for each objective was considered), the model obtained a 48 robustness score (Figure 6.2). Although this score indicates that the model is robust, it borders on being only moderately robust. For this evaluation, only the result with the greatest impact for each objective was considered.

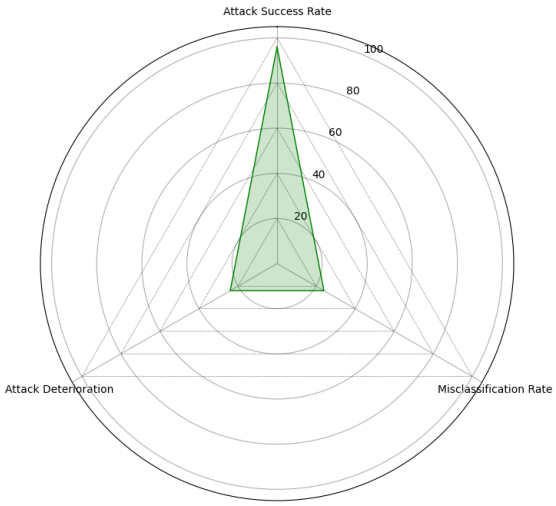

Figure 6.2: CatBoost worst case robustness score.

Multilayer Perceptron

The MLP model was subjected to the same adversarial perturbation methods as the CatBoost model. As the data used to train this model was subjected to a scaler, the dataset to perturbed was also scaled using the same scaler. The configuration for each perturbation method was the same as for the CatBoost model, with the exception of A2PM, which was configured to use the ratios between 0.0001 and 0.0005 for all numerical features, and 0.8 for the probability of perturbation. As the scaled data does not keep integer features as integers, the constrained versions of the adversarial attacks only focused on turning the categorical features into valid values. Due to the data scaling, the median value of the each feature was very low, resulting in a threshold of 12.08. The results obtained from each perturbation method are summarized in Table 6.9.

From the results, it is evident that ZOO was unable to generate any successful adversarial examples, for both targeted and untargeted attacks, with and without constraints. As with the CatBoost model, this is likely due to the perturbation method's requirement for a large number of queries to approximate the gradients, which was not feasible in this case. The other three methods, Boundary Attack, HopSkipJump, and A2PM, were able to generate adversarial samples, but with varying degrees of success.

Regarding A2PM, this method achieved better results in the untargeted attack, generating 11 adversarial examples, than in the targeted attack, where it generated only 6 adversarial examples. The distance required to perturbed these samples was the lowest for all the implemented targeted attacks, and the second lowest for the untargeted attacks, only surpassed by the C&W constrained.

The Boundary Attack and HopSkipJump methods without constraints achieved high levels of perturbation success, with MR of 81% and 100%, and ASR of 85% and 100% respectively for untarged and targeted attacks. The distances obtained by these methods were very high, surpassing by far the threshold set, which resulted in low DA metrics. C&W too achieved such distance, although with a lower success in generating adversarial examples. From these attacks, only the constrained version of C&W untargeted was able to generate adversarial examples, with a MR of 0.02, and the lowest distance of all the methods. The comparison between constrained and non-constrained versions of this method shows that

Table 6.9: Adversarial perturbations results for MLP model.

	T	C	CA	AA	ASR	MR	AD	TR	DA ASR	DA MR	D
A2PM	✗	✓	1.0	0.89	-	0.11	0.11	1.17	-	0.11	2
BoundaryAttack	✗	✗	1.0	0.19	-	0.81	0.81	301.43	-	0.0011	459
BoundaryAttack	✗	✓	1.0	1.0	-	0.0	0.0	305.81	-	0.0	nan
CarliniWagner	✗	✗	1.0	0.87	-	0.13	0.13	374.86	-	0.0002	459
CarliniWagner	✗	✓	1.0	0.98	-	0.02	0.02	375.76	-	0.02	1
HopSkipJump	✗	✗	1.0	0.0	-	1.0	1.0	458.79	-	0.0014	460
HopSkipJump	✗	✓	1.0	1.0	-	0.0	0.0	463.57	-	0.0	nan
Zoo	✗	✗	1.0	1.0	-	0.0	0.0	25.64	-	0.0	nan
Zoo	✗	✓	1.0	1.0	-	0.0	0.0	25.78	-	0.0	nan
A2PM	✓	✓	1.0	0.95	0.07	0.05	0.05	0.68	0.07	0.05	2
BoundaryAttack	✓	✗	1.0	0.36	0.85	0.64	0.64	223.79	0.0012	0.0009	459
BoundaryAttack	✓	✓	1.0	1.0	0.0	0.0	0.0	228.37	0.0	0.0	nan
CarliniWagner	✓	✗	1.0	0.91	0.12	0.09	0.09	370.71	0.0002	0.0001	459
CarliniWagner	✓	✓	1.0	1.0	0.0	0.0	0.0	385.25	0.0	0.0	nan
HopSkipJump	✓	✗	1.0	0.25	1.0	0.75	0.75	370.35	0.0014	0.0010	460
HopSkipJump	✓	✓	1.0	1.0	0.0	0.0	0.0	394.14	0.0	0.0	nan
Zoo	✓	✗	1.0	1.0	0.0	0.0	0.0	54.66	0.0	0.0	nan
Zoo	✓	✓	1.0	1.0	0.0	0.0	0.0	52.75	0.0	0.0	nan

the non-constrained version achieved a much higher MR, although not realistic and valid, while the constrained version achieved a lower MR but successfully created realistic and valid samples.

Overall, the results show that the MLP model is more vulnerable to untargeted adversarial attacks than to targeted attacks, particularly when the perturbations are not constrained, although the results are very similar.

In a standard scenario, where the robustness evaluation of the model is based on the average values obtained, the model achieved a robustness score of 93 (Figure 6.3), which is considered to be very robust. The model's resilience to adversarial perturbations, targeted attacks, and performance deterioration were all evaluated, with the results indicating that the model is highly robust against adversarial attacks.

Regarding the worst case scenario, where the model is subjected to the most effective adversarial attacks for each objective, the model obtained a robustness score of 61 (Figure 6.4). This score indicates that the model is still robust, although it is very close to being moderately robust. The evaluation considered only the results with the greatest impact for each objective, which highlights the model's vulnerability to adversarial attacks under extreme conditions.

6.3 Chapter Remarks

This chapter presents a case study which analyzes the robustness of two multiclass classification models trained on the same dataset. Both models demonstrated near-perfect performance metrics after training, however, their resilience changed considerably when subjected to adversarial attacks. The CatBoost model achieved robustness scores of 88 and 48 in standard and worst-case scenarios, respectively, while the MLP model scored 93 and

Figure 6.3: MLP robustness score.

Figure 6.4: MLP worst case robustness score.

61. The threshold setting was a key factor influencing these results, as the CatBoost model had a very high threshold due to the large median value of the Offset feature in the dataset. In contrast, the MLP model had a much lower threshold, enabling it to achieve higher robustness scores, since the generated samples were required to be closer to the original samples.

The study also highlighted that the outcomes of robustness testing differ depending on whether data scaling is applied. The MLP model, which used scaled data, had a lower threshold, resulting in better adversarial sample generation and robustness scores. In contrast, the CatBoost model, which did not use scaled data, had a higher threshold, resulting in lower robustness scores. The scale of the numerical features was also a decisive factor in influencing the threshold and distance of adversarial samples. On the scaled dataset, the numerical features had a much narrower range of values, making the penalty for invalid categorical features more impactful. While CatBoost outperformed MLP on standard evaluation metrics, MLP demonstrated greater robustness under adversarial conditions. These results underscore the importance of thoroughly evaluating model robustness before deployment.

The configuration of adversarial perturbation methods, perturbed data and the report generated by AURORA is available online, in Github [1].

[1] https://github.com/msilva2002/MastersCaseStudy

Chapter 7

Conclusion

This chapter provides the main conclusion of this dissertation, highligting the accomplished objectives. The limitations of the proposed solution and possible improvements are also described, indicating the next steps to be explored in the future.

7.1 Accomplised Objectives

This dissertation addressed the lack of realism and validity in the generated data used to evaluate ML models, more specifically in tabular data. All the initial estabilished objectives were successfully accomplished, with a completion rate of 100%. A solution was developed to evaluate the robustness of models, while taking into account the constrains and specifications related to tabular data. The main results for each objective were:

- **OB1:** A systematic literature review of the state-of-the-art of adversarial ML attacks and robustness metrics. It was found that most of the existing methods are focused on image data, and there is a lack for tabular data specific robustness metrics, and adequate adversarial samples generation.
- **OB2:** A methodology for adversarial robutness testing using different scenarios, based on the threat model and the adversarial attack type.
- **OB3:** Two main approaches have been proposed to measure the robustness of ML models against adversarial attacks. The first approach is based on the average success rate of multiple attacks and provides an overall view of the model's vulnerability under a range of adversarial conditions. The second approach focuses on the worst-case scenario, evaluating the model's robustness using only the most effective attacks - those that cause the greatest performance degradation.
- **OB4:** A Distance Adjustment was proposed to adjust the commonly used metrics to ensure that the generated data is both realistic and valid, based on the distance between the original and perturbed data. This adjustment allows for a more accurate evaluation of the robustness of ML models, while mantaining the adopted metrics.
- **OB5:** A tool developed to intuitively test the robustness of ML models against adversarial attacks, adjusting the scenarios to the characteristics of the data and the model. The tool is also easly scalable and user friendly to allow users to test their own models and data without the need for extensive knowledge in ML security.
- **OB6:** A case study that evaluated the robustness of two ML models using AURORA. The first case evaluated the robustness of a Catboost model and the second case evaluated the robustness of a MLP model, both trained on the GeNIS dataset. Both

cases were able to demonstrate the effectiveness of the proposed solution in evaluating the robustness of ML models, where it was shown that the better performing model was also the less robust. It was also shown that the Distance Adjustment is highly dependent on the calculated threshold of the dataset, which translates to how much perturbation is allowed in the data.

The obtained results evidence the effectiveness of the proposed solution in evaluating the robustness of ML models, while taking into account the constrains and specifications related to tabular data. The systematic literature review, the proposed solution, and the case study described in this dissertation can be used as a reference for future research in this area, and the developed tool can be used by researchers and practitioners to evaluate the robustness of their own ML models, which could raise the standard for model testing. The tool also represents a step in the certification and compliance process for models under the scope of the AI Act and AI trustworthiness legislation.

7.2 Limitations and Future Work

During the development, a known vulnerability was identified in the library used for saving and loading ML models in this implement - *joblib*. This vulnerability, documented as CVE-2024-34997 [336], concerns the library's handling of model serialization and deserialization. According to the National Institute of Standards and Technology (NIST), the vulnerability is associated with CWE-502: Deserialization of Untrusted Data [337], which is recognized as a critical security risk. Therefore, it is strongly recommended that users avoid using unknown or untrusted models with this implementation. Despite the known vulnerability, *joblib* was chosen due to its popularity and widespread use for this type of task. However, if a new library that allows for saving and loading ML models is developed in the near future and is unaffected by this vulnerability, *joblib* will be replaced with it.

The tool is also planned to be released in a Docker container, enabling it to be used in a more controlled environment and removing the need for users to install dependencies. This approach also enables a more controlled access to the toolbox and constitutes a step towards mitigating the detected vulnerability.

As previously stated, most of the generation methods adopted by researchers are image-based. In order to evaluate the model more thoroughly, it is important to incorporate more table-oriented methods, which use different approaches to generate those perturbations.

Despite the attained considerations to classify generated data as both realistic and valid, there is still a need for a perturbation method that takes this into account when generating the perturbations. Although methods such as A2PM are effective into generating data suitable for realistic scenarios, it lacks the optimization problem used by other methods, such as C&W and ZOO to generate data that is close/similar to the original. The complex configuration of A2PM poses as a set back for its use and requires users to trial and error the configuration to achieve successful perturbations while remaining realistic.

In the future, a major enhancement could be the development of a perturbation method that uses the proposed distance adjustment as a budget. Additionally, adapting the implemented methods, such as ZOO and HopSkipJumpAttack, to ensure the data constraint is ideal, as these are well-known and widely used methods, although they require this enforcement to be used with greater confidence in tabular data.

Another significant contribution would be to implement more black-box attacks in the tool to try to map the model's decisions. Researching and incorporating exploratory attacks that require no initial information would also significantly contribute to evaluating these models.

A further contribution could be the study of a distance metric that is not point-to-point, but rather a more general distance between datasets. This could be used to evaluate the robustness of the model in a more global way, rather than just on individual data points. This would allow for a more comprehensive understanding of the model's performance and its ability to generalize across different datasets. Another approach would be to measure the distance between each feature independently. This would allow for a more accurate measurement of the distance between the original and the perturbed data. The result would be a real value that could be used to identify, from a batch, how many data points are realistic and valid and how many are not.

It is also pertinent to further expand this study by exploring different novel defense strategies, to validate whether these are effective to provide a more reliable and robust ML model.

7.3 Final Remarks

This dissertation presented the reasearch and development work that led to the proposal of a metric adjustment for robustness evaluation, where metrics are adjusted in unrealistic and/or invalid data is present during the evaluation. It was aligned with the participation of GECAD in the CYDERCO (Grant Agreement No. 101128052) and SAFE (Grant Agreement No. 101190370) european projects.

Overall, this was a very interesting and challenging project, as it provided me with the opportunity to explore the field of cybersecurity of AI in greater depth, while improving my problem-solving, organisational and critical thinking skills. Furthermore, it enabled me to learn more about ML models and how they work, as well as practise my cybersecurity skills.

The main takeaway of this dissertation is that, although ML models are receiving more and more attention and being adopted by the general public, they should only be used as an assistant in critical decision-making and their own vulnerabilities must not be underestimated. It is essential to continue researching ways to improve the security and trustworthiness of ML, and to raise awareness of the issues presented in this work.

Bibliography

[1] Roman Yampolskiy. "Incident Number 52: Tesla on AutoPilot Killed Driver in Crash in Florida while Watching Movie". In: *AI Incident Database* (2016). Ed. by Sean McGregor. Retrieved December 2024 from https://incidentdatabase.ai/cite/52. url: https://incidentdatabase.ai/cite/52.

[2] Catherine Olsson. "Incident Number 67: Sleeping Driver on Tesla AutoPilot". In: *AI Incident Database* (2018). Ed. by Sean McGregor. Retrieved December 2024 from https://incidentdatabase.ai/cite/67. url: https://incidentdatabase.ai/cite/67.

[3] Florian Tambon et al. "How to certify machine learning based safety-critical systems? A systematic literature review". In: *Automated Software Engineering* 29.2 (Apr. 2022). issn: 1573-7535. doi: 10.1007/s10515-022-00337-x.

[4] Roman Yampolskiy. "Incident Number 1: Google's YouTube Kids App Presents Inappropriate Content". In: *AI Incident Database* (2015). Ed. by Sean McGregor. Retrieved December 2024 from https://incidentdatabase.ai/cite/1. url: https://incidentdatabase.ai/cite/1.

[5] Daniel Atherton. "Incident Number 545: Chatbot Tessa gives unauthorized diet advice to users seeking help for eating disorders". In: *AI Incident Database* (2023). Ed. by Daniel Atherton. Retrieved December 2024 from https://incidentdatabase.ai/cite/545. url: https://incidentdatabase.ai/cite/545.

[6] Anonymous. "Incident Number 37: Female Applicants Down-Ranked by Amazon Recruiting Tool". In: *AI Incident Database* (2016). Ed. by Sean McGregor. Retrieved December 2024 from https://incidentdatabase.ai/cite/37. url: https://incidentdatabase.ai/cite/37.

[7] Ingrid Dickinson. "Incident Number 135: UT Austin GRADE Algorithm Allegedly Reinforced Historical Inequalities". In: *AI Incident Database* (2012). Ed. by Sean McGregor. Retrieved December 2024 from https://incidentdatabase.ai/cite/135. url: https://incidentdatabase.ai/cite/135.

[8] Shunyao Wang et al. "Evasion Attack and Defense on Machine Learning Models in Cyber-Physical Systems: A Survey". In: *IEEE Communications Surveys & Tutorials* 26.2 (2024), pp. 930–966. doi: 10.1109/COMST.2023.3344808.

[9] Antonio Emanuele Cinà, Alessandro Torcinovich, and Marcello Pelillo. "A black-box adversarial attack for poisoning clustering". In: *Pattern Recognition* 122 (2022), p. 108306. issn: 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.108306.

[10] Hailong Hu and Jun Pang. "Stealing Machine Learning Models: Attacks and Countermeasures for Generative Adversarial Networks". In: *Proceedings of the 37th Annual Computer Security Applications Conference*. ACSAC '21. Virtual Event, USA: Association for Computing Machinery, 2021, pp. 1–16. isbn: 9781450385794. doi: 10.1145/3485832.3485838.

[11] Hongsheng Hu et al. "Membership Inference Attacks on Machine Learning: A Survey". In: *ACM Comput. Surv.* 54.11s (Sept. 2022). issn: 0360-0300. doi: 10.1145/3523273.

[12] Instituto Politécnico do Porto. *Regulamento do Código de Boas Práticas e de Conduta do Instituto Politécnico do Porto*. Diário da República, 2.ª série PARTE E Artigo 2.º. Retrieved December 2024 from https://www.iscap.ipp.pt/regulamentos/CodigoboaspraticasedecondutaIPP.pdf. url: https://www.iscap.ipp.pt/regulamentos/CodigoboaspraticasedecondutaIPP.pdf.

[13] IEEE. *IEEE Code of Ethics*. Retrieved December 8, 2024 from https://www.ieee.org/about/corporate/governance/p7-8.html. n.d. url: https://www.ieee.org/about/corporate/governance/p7-8.html.

[14] ACM. *ACM Code of Ethics and Professional Conduct*. Retrieved December 8, 2024 from https://www.acm.org/code-of-ethics. n.d. url: https://www.acm.org/code-of-ethics.

[15] Miguel Silva et al. "GeNIS: A modular dataset for network intrusion detection and classification". In: *Data in Brief* 60 (2025), p. 111487. issn: 2352-3409. doi: https://doi.org/10.1016/j.dib.2025.111487.

[16] João Vitorino et al. "Reliable feature selection for adversarially robust cyber-attack detection". In: *Annals of Telecommunications* 80.3 (Apr. 2025), pp. 341–355. issn: 1958-9395. doi: 10.1007/s12243-024-01047-z.

[17] Miguel Silva et al. "Efficient Network Traffic Feature Sets for IoT Intrusion Detection". In: *Distributed Computing and Artificial Intelligence, Special Sessions I, 21st International Conference*. Ed. by Rashid Mehmood et al. Cham: Springer Nature Switzerland, 2025, pp. 3–13. isbn: 978-3-031-76459-2.

[18] João Vitorino et al. "An Adversarial Robustness Benchmark for Enterprise Network Intrusion Detection". In: *Foundations and Practice of Security*. Ed. by Mohamed Mosbah et al. Cham: Springer Nature Switzerland, 2024, pp. 3–17. isbn: 978-3-031-57537-2.

[19] David Moher et al. "Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement". In: *Systematic Reviews* 4.1 (Jan. 2015), p. 1. issn: 2046-4053. doi: 10.1186/2046-4053-4-1.

[20] *Institute of Electrical and Electronics Engineers Xplore Search Source*. Accessed: 2024-10-27. url: https://ieeexplore.ieee.org/search/advanced.

[21] *ScienceDirect*. Accessed: 2024-10-27. url: https://www.sciencedirect.com/search.

[22] *Association for Computing Machinery Digital Library Search Source*. Accessed: 2024-10-27. url: https://dl.acm.org/search/advanced.

[23] European Union Agency for Cybersecurity et al. *Standardisation in support of the cybersecurity of AI*. Ed. by E. Magonara et al. European Union Agency for Cybersecurity, 2023. doi: 10.2824/277479.

[24] Akm Iqtidar Newaz et al. "Adversarial Attacks to Machine Learning-Based Smart Healthcare Systems". In: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. 2020, pp. 1–6. doi: 10.1109/GLOBECOM42002.2020.9322472.

[25] Nishant Kumar et al. "Evolutionary Adversarial Attacks on Payment Systems". In: *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2021, pp. 813–818. doi: 10.1109/ICMLA52953.2021.00134.

[26] Chao-Han Huck Yang et al. "Treatment Learning Causal Transformer for Noisy Image Classification". In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 6128–6139. doi: 10.1109/WACV56688.2023.00608.

[27] Quentin Bouniot, Romaric Audigier, and Angélique Loesch. "Vulnerability of Person Re-Identification Models to Metric Adversarial Attacks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 3450–3459. doi: 10.1109/CVPRW50498.2020.00405.

[28] Jiahui Chen et al. "FedDef: Defense Against Gradient Leakage in Federated Learning-Based Network Intrusion Detection Systems". In: *IEEE Transactions on Information Forensics and Security* 18 (2023), pp. 4561–4576. doi: 10.1109/TIFS.2023.3297369.

[29] Madeleine Schneider, David Aspinall, and Nathaniel D. Bastian. "Evaluating Model Robustness to Adversarial Samples in Network Intrusion Detection". In: *2021 IEEE International Conference on Big Data (Big Data)*. 2021, pp. 3343–3352. doi: 10.1109/BigData52589.2021.9671580.

[30] Hassan Ali et al. "Tamp-X: Attacking explainable natural language classifiers through tampered activations". In: *Computers & Security* 120 (Sept. 2022), p. 102791. issn: 0167-4048. doi: 10.1016/j.cose.2022.102791.

[31] Ivan Fursov et al. "A Differentiable Language Model Adversarial Attack on Text Classifiers". In: *IEEE Access* 10 (2022), pp. 17966–17976. doi: 10.1109/ACCESS.2022.3148413.

[32] Yijing Zhou et al. "Bit Attacking Deep Neural Networks Based on Complex Networks Theory". In: *2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*. 2024, pp. 260–268. doi: 10.1109/QRS-C63300.2024.00042.

[33] Haipeng Wang et al. "Context-Aware Fuzzing for Robustness Enhancement of Deep Learning Models". In: *ACM Trans. Softw. Eng. Methodol.* (July 2024). Place: New York, NY, USA Publisher: Association for Computing Machinery. issn: 1049-331X. doi: 10.1145/3680464.

[34] Christian Szegedy et al. *Intriguing properties of neural networks*. 2014. arXiv: 1312.6199 [cs.CV]. url: https://arxiv.org/abs/1312.6199.

[35] József Sándor, Roland Nagy, and Levente Buttyán. "Increasing the Robustness of a Machine Learning-based IoT Malware Detection Method with Adversarial Training". In: *Proceedings of the 2023 ACM Workshop on Wireless Security and Machine Learning*. WiseML'23. event-place: Guildford, United Kingdom. New York, NY, USA: Association for Computing Machinery, 2023, pp. 3–8. isbn: 9798400701337. doi: 10.1145/3586209.3591401.

[36] Emad Efatinasab et al. "Adversarially Robust Fault Zone Prediction in Smart Grids With Bayesian Neural Networks". In: *IEEE Access* 12 (2024), pp. 121169–121184. doi: 10.1109/ACCESS.2024.3452476.

[37] Max Lennon, Nathan Drenkow, and Phil Burlina. "Patch Attack Invariance: How Sensitive are Patch Attacks to 3D Pose?" In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021, pp. 112–121. doi: 10.1109/ICCVW54120.2021.00018.

[38] Wenguan Wang et al. "Salient Object Detection in the Deep Learning Era: An In-Depth Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.6 (2022), pp. 3239–3259. doi: 10.1109/TPAMI.2021.3051099.

[39] Qiancheng Yang, Yong Luo, and Bo Du. "Training-Free Robust Neural Network Search Via Pruning". In: *2024 IEEE International Conference on Multimedia and Expo (ICME)*. 2024, pp. 1–6. doi: 10.1109/ICME57554.2024.10687950.

[40] Kaikang Zhao et al. "Ensemble Adversarial Defense via Integration of Multiple Dispersed Low Curvature Models". In: *2024 International Joint Conference on Neural Networks (IJCNN)*. 2024, pp. 1–9. doi: 10.1109/IJCNN60899.2024.10651354.

[41] Hongyu Zhu et al. "Reliable Model Watermarking: Defending against Theft without Compromising on Evasion". In: *Proceedings of the 32nd ACM International Conference on Multimedia*. MM '24. event-place: Melbourne VIC, Australia. New York, NY, USA: Association for Computing Machinery, 2024, pp. 10124–10133. isbn: 9798400706868. doi: 10.1145/3664647.3681610.

[42] Seong Hee Park et al. "A Comprehensive Risk Analysis Method for Adversarial Attacks on Biometric Authentication Systems". In: *IEEE Access* 12 (2024), pp. 116693–116710. doi: 10.1109/ACCESS.2024.3439741.

[43] Azuka Chiejina et al. "System-level Analysis of Adversarial Attacks and Defenses on Intelligence in O-RAN based Cellular Networks". In: *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec '24. event-place: Seoul, Republic of Korea. New York, NY, USA: Association for Computing Machinery, 2024, pp. 237–247. isbn: 9798400705823. doi: 10.1145/3643833.3656119.

[44] Neha Nagarkar et al. "Energy-Efficient and Adversarially Robust Machine Learning with Selective Dynamic Band Filtering". In: *Proceedings of the 2021 Great Lakes Symposium on VLSI*. GLSVLSI '21. event-place: Virtual Event, USA. New York, NY, USA: Association for Computing Machinery, 2021, pp. 195–200. isbn: 978-1-4503-8393-6. doi: 10.1145/3453688.3461756.

[45] Wenqi Wei et al. "Adversarial Deception in Deep Learning: Analysis and Mitigation". In: *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. 2020, pp. 236–245. doi: 10.1109/TPS-ISA50397.2020.00039.

[46] Ramtin Hosseini, Xingyi Yang, and Pengtao Xie. "DSRNA: Differentiable Search of Robust Neural Architectures". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 6192–6201. doi: 10.1109/CVPR46437.2021.00613.

[47] Zhe Zhao et al. "Attack as Detection: Using Adversarial Attack Methods to Detect Abnormal Examples". In: *ACM Trans. Softw. Eng. Methodol.* 33.3 (Mar. 2024). Place: New York, NY, USA Publisher: Association for Computing Machinery. issn: 1049-331X. doi: 10.1145/3631977.

[48] Yinghua Zhang et al. "Two Sides of the Same Coin: White-box and Black-box Attacks for Transfer Learning". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '20. event-place: Virtual Event, CA, USA. New York, NY, USA: Association for Computing Machinery, 2020, pp. 2989–2997. isbn: 978-1-4503-7998-4. doi: 10.1145/3394486.3403349.

[49] Jiyu Chen, David Wang, and Hao Chen. "Explore the Transformation Space for Adversarial Images". In: *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*. CODASPY '20. event-place: New Orleans, LA, USA. New York, NY, USA: Association for Computing Machinery, 2020, pp. 109–120. isbn: 978-1-4503-7107-0. doi: 10.1145/3374664.3375728.

[50] Benjamin Appiah et al. "Decision tree pairwise metric learning against adversarial attacks". In: *Computers & Security* 106 (July 2021), p. 102268. issn: 0167-4048. doi: 10.1016/j.cose.2021.102268.

[51] Yao Yu chen. "Dog and Cat Classification with Deep Residual Network". In: *Proceedings of the 2020 European Symposium on Software Engineering*. ESSE '20.

event-place: Rome, Italy. New York, NY, USA: Association for Computing Machinery, 2020, pp. 137–141. isbn: 978-1-4503-7762-1. doi: 10.1145/3393822.3432321.

[52] Tommaso Zoppi and Andrea Ceccarelli. "Detect Adversarial Attacks Against Deep Neural Networks With GPU Monitoring". In: *IEEE Access* 9 (2021), pp. 150579–150591. doi: 10.1109/ACCESS.2021.3125920.

[53] Camilo Pestana et al. "Defense-friendly Images in Adversarial Attacks: Dataset and Metrics for Perturbation Difficulty". In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 556–565. doi: 10.1109/WACV48630.2021.00060.

[54] Tianyu Pang et al. "Two Coupled Rejection Metrics Can Tell Adversarial Examples Apart". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 15202–15212. doi: 10.1109/CVPR52688.2022.01479.

[55] Keji Han, Bin Xia, and Yun Li. "(AD)2: Adversarial domain adaptation to defense with adversarial perturbation removal". In: *Pattern Recognition* 122 (Feb. 2022), p. 108303. issn: 0031-3203. doi: 10.1016/j.patcog.2021.108303.

[56] Longxin Lin et al. "Understanding the impact on convolutional neural networks with different model scales in AIoT domain". In: *Journal of Parallel and Distributed Computing* 170 (Dec. 2022), pp. 1–12. issn: 0743-7315. doi: 10.1016/j.jpdc.2022.07.011.

[57] Ricardo Bigolin Lanfredi, Joyce D. Schroeder, and Tolga Tasdizen. "Quantifying the preferential direction of the model gradient in adversarial training with projected gradient descent". In: *Pattern Recognition* 139 (July 2023), p. 109430. issn: 0031-3203. doi: 10.1016/j.patcog.2023.109430.

[58] Jaehyuk Heo, Seungwan Seo, and Pilsung Kang. "Exploring the differences in adversarial robustness between ViT- and CNN-based models using novel metrics". In: *Computer Vision and Image Understanding* 235 (Oct. 2023), p. 103800. issn: 1077-3142. doi: 10.1016/j.cviu.2023.103800.

[59] Haibo Jin et al. "Excitement surfeited turns to errors: Deep learning testing framework based on excitable neurons". In: *Information Sciences* 637 (Aug. 2023), p. 118936. issn: 0020-0255. doi: 10.1016/j.ins.2023.118936.

[60] Deepak Ravikumar et al. "TREND: Transferability-Based Robust ENsemble Design". In: *IEEE Transactions on Artificial Intelligence* 4.3 (2023), pp. 534–548. doi: 10.1109/TAI.2022.3175172.

[61] Chenhao Lin, Xingliang Zhang, and Chao Shen. "DeepLogic: Priority Testing of Deep Learning Through Interpretable Logic Units". In: *Chinese Journal of Electronics* 33.4 (2024), pp. 948–964. doi: 10.23919/cje.2022.00.451.

[62] James Kihara MWANGI, Jane KURIA, and John WANDETO. "Block Switching: Defying Fast Gradient Sign Resistance". In: *2024 IST-Africa Conference (IST-Africa)*. 2024, pp. 1–12. doi: 10.23919/IST-Africa63983.2024.10569920.

[63] Jeonghun Kim et al. "Camouflaged Adversarial Attack on Object Detector". In: *2021 21st International Conference on Control, Automation and Systems (ICCAS)*. 2021, pp. 613–617. doi: 10.23919/ICCAS52745.2021.9650004.

[64] Ziming Zhao et al. "Poster: Detecting Adversarial Examples Hidden under Watermark Perturbation via Usable Information Theory". In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. CCS '23. event-place: Copenhagen, Denmark. New York, NY, USA: Association for Computing Machinery, 2023, pp. 3636–3638. isbn: 9798400700507. doi: 10.1145/3576915.3624396.

[65] Anshuman Chhabra and Prasant Mohapatra. "Moving Target Defense against Adversarial Machine Learning". In: *Proceedings of the 8th ACM Workshop on Moving Target Defense*. MTD '21. event-place: Virtual Event, Republic of Korea. New York, NY, USA: Association for Computing Machinery, 2021, pp. 29–30. isbn: 978-1-4503-8658-6. doi: 10.1145/3474370.3485662.

[66] Xuan-Ming Zhang et al. "Optimized L2 Norm Loss for Adversarial Robustness". In: *2022 IEEE 2nd International Conference on Computer Communication and Artificial Intelligence (CCAI)*. 2022, pp. 8–16. doi: 10.1109/CCAI55564.2022.9807767.

[67] Hengyue Liang et al. "Optimization for Robustness Evaluation Beyond lp Metrics". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. doi: 10.1109/ICASSP49357.2023.10095871.

[68] Yunseong Kim and Seunghyun Yoon. "Similarity-based Filtering for Defending Against Malicious Clients in Federated Learning". In: *2024 IEEE International Conference on Big Data (BigData)*. 2024, pp. 8728–8730. doi: 10.1109/BigData62323.2024.10825131.

[69] Guoqin Chang et al. "TextGuise: Adaptive adversarial example attacks on text classification model". In: *Neurocomputing* 529 (Apr. 2023), pp. 190–203. issn: 0925-2312. doi: 10.1016/j.neucom.2023.01.071.

[70] Marwan Omar et al. "Robust Natural Language Processing: Recent Advances, Challenges, and Future Directions". In: *IEEE Access* 10 (2022), pp. 86038–86056. doi: 10.1109/ACCESS.2022.3197769.

[71] Hamid Bostani et al. "Level Up with ML Vulnerability Identification: Leveraging Domain Constraints in Feature Space for Robust Android Malware Detection". In: *ACM Trans. Priv. Secur.* 28.2 (Feb. 2025). issn: 2471-2566. doi: 10.1145/3711899.

[72] Mengdie Huang et al. "Dimensional Robustness Certification for Deep Neural Networks in Network Intrusion Detection Systems". In: *ACM Trans. Priv. Secur.* (Apr. 2025). Just Accepted. issn: 2471-2566. doi: 10.1145/3715121.

[73] Farhan Ahmed et al. "Ares: A System-Oriented Wargame Framework for Adversarial ML". In: *2022 IEEE Security and Privacy Workshops (SPW)*. 2022, pp. 73–79. doi: 10.1109/SPW54247.2022.9833895.

[74] Nathalie Baracaldo et al. "Benchmarking the Effect of Poisoning Defenses on the Security and Bias of Deep Learning Models". In: *2023 IEEE Security and Privacy Workshops (SPW)*. 2023, pp. 45–56. doi: 10.1109/SPW59333.2023.00010.

[75] Jingyi Wang et al. "RobOT: Robustness-Oriented Testing for Deep Learning Systems". In: *Proceedings of the 43rd International Conference on Software Engineering*. ICSE '21. Place: Madrid, Spain. IEEE Press, 2021, pp. 300–311. isbn: 978-1-4503-9085-9. doi: 10.1109/ICSE43902.2021.00038.

[76] Mahsa Paknezhad et al. "Explaining adversarial vulnerability with a data sparsity hypothesis". In: *Neurocomputing* 495 (July 2022), pp. 178–193. issn: 0925-2312. doi: 10.1016/j.neucom.2022.01.062.

[77] J. DeMarchi et al. "Evaluation of Robustness Metrics for Defense of Machine Learning Systems". In: *2023 International Conference on Military Communications and Information Systems (ICMCIS)*. 2023, pp. 1–12. doi: 10.1109/ICMCIS59922.2023.10253593.

[78] Bingzhi Chen et al. "Stay Focused is All You Need for Adversarial Robustness". In: *Proceedings of the 32nd ACM International Conference on Multimedia*. MM '24. event-place: Melbourne VIC, Australia. New York, NY, USA: Association for Computing Machinery, 2024, pp. 6482–6491. isbn: 9798400706868. doi: 10.1145/3664647.3681676.

[79] Salam Omar Alo et al. "Automated Detection of Cybersecurity Threats Using Generative Adversarial Networks (GANs)". In: *2024 36th Conference of Open Innovations Association (FRUCT)*. 2024, pp. 566–577. doi: 10.23919/FRUCT64283.2024.10749874.

[80] Anastasiia Bohachenko et al. "Mitigating Filter-Based Adversarial Attacks in BCIs through Model Compression". In: *Proceedings of the 2025 ACM Southeast Conference*. ACMSE 2025. Southeast Missouri State University, Cape Girardeau, MO, USA: Association for Computing Machinery, 2025, pp. 299–300. isbn: 9798400712777. doi: 10.1145/3696673.3723093.

[81] Samy Abd El-Nabi et al. "Driver Drowsiness Detection Using Swin Transformer and Diffusion Models for Robust Image Denoising". In: *IEEE Access* 13 (2025), pp. 71880–71907. doi: 10.1109/ACCESS.2025.3561717.

[82] Xun Wang, Zhaoming Yao, and Hang Wei. "Intelligent characterization and robustness quantification of frozen soil strength images using a multi-module fusion strategy". In: *Cold Regions Science and Technology* 231 (2025), p. 104384. issn: 0165-232X. doi: https://doi.org/10.1016/j.coldregions.2024.104384.

[83] Avilash Rath et al. "When AI Meets Code Analysis: A Study of Adversarial Attacks on Deep Learning-based Code Models via Program Transformation". In: *2024 Annual Computer Security Applications Conference Workshops (ACSAC Workshops)*. 2024, pp. 85–96. doi: 10.1109/ACSACW65225.2024.00017.

[84] Jingyang Li and Guoqiang Li. "The Triangular Trade-off between Robustness, Accuracy and Fairness in Deep Neural Networks: A Survey". In: *ACM Comput. Surv.* (Feb. 2024). Place: New York, NY, USA Publisher: Association for Computing Machinery. issn: 0360-0300. doi: 10.1145/3645088.

[85] Alex Gittens, Bülent Yener, and Moti Yung. "An Adversarial Perspective on Accuracy, Robustness, Fairness, and Privacy: Multilateral-Tradeoffs in Trustworthy ML". In: *IEEE Access* 10 (2022), pp. 120850–120865. doi: 10.1109/ACCESS.2022.3218715.

[86] Marine Picot et al. "Adversarial Robustness Via Fisher-Rao Regularization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.3 (2023), pp. 2698–2710. doi: 10.1109/TPAMI.2022.3174724.

[87] Mohammed Rajhi and Niki Pissinou. "Adversarial Training on Limited-Resource Devices Through Asymmetric Disturbances". In: *2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. 2024, pp. 27–34. doi: 10.1109/DCOSS-IoT61029.2024.00015.

[88] Krishnakant Singh et al. "Is Synthetic Data all We Need? Benchmarking the Robustness of Models Trained with Synthetic Images". In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2024, pp. 2505–2515. doi: 10.1109/CVPRW63382.2024.00257.

[89] Ming Jin et al. *Power up! Robust Graph Convolutional Network via Graph Powering*. 2021. arXiv: 1905.10029 [cs.LG]. url: https://arxiv.org/abs/1905.10029.

[90] Lichao Sun et al. "Adversarial Attack and Defense on Graph Data: A Survey". In: *IEEE Transactions on Knowledge and Data Engineering* 35.8 (2023), pp. 7693–7711. doi: 10.1109/TKDE.2022.3201243.

[91] Senthil Murugan Nagarajan et al. "Adversarial Deep Learning based Dampster–Shafer data fusion model for intelligent transportation system". In: *Information Fusion* 102 (Feb. 2024), p. 102050. issn: 1566-2535. doi: 10.1016/j.inffus.2023.102050.

[92] Shahroz Tariq, Binh M. Le, and Simon S. Woo. "Towards an Awareness of Time Series Anomaly Detection Models' Adversarial Vulnerability". In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

CIKM '22. event-place: Atlanta, GA, USA. New York, NY, USA: Association for Computing Machinery, 2022, pp. 3534–3544. isbn: 978-1-4503-9236-5. doi: 10.1145/3511808.3557073.

[93] Tao Wu et al. "ERGCN: Data enhancement-based robust graph convolutional network against adversarial attacks". In: *Information Sciences* 617 (Dec. 2022), pp. 234–253. issn: 0020-0255. doi: 10.1016/j.ins.2022.10.115.

[94] Mst Shapna Akter et al. "Exploring the Vulnerabilities of Machine Learning and Quantum Machine Learning to Adversarial Attacks Using a Malware Dataset: A Comparative Analysis". In: *2023 IEEE International Conference on Software Services Engineering (SSE)*. 2023, pp. 222–231. doi: 10.1109/SSE60056.2023.00037.

[95] Haiwen Chen et al. "Robustness Analysis and Evaluation Study of Chinese Text Event Detection Models". In: *2023 9th International Conference on Big Data and Information Analytics (BigDIA)*. 2023, pp. 632–639. doi: 10.1109/BigDIA60676.2023.10429355.

[96] Manh-Dung Nguyen et al. "A deep learning anomaly detection framework with explainability and robustness". In: *Proceedings of the 18th International Conference on Availability, Reliability and Security*. ARES '23. event-place: Benevento, Italy. New York, NY, USA: Association for Computing Machinery, 2023. isbn: 9798400707728. doi: 10.1145/3600160.3605052.

[97] Mst Shapna Akter et al. "Quantum Adversarial Attacks: Developing Quantum FGSM Algorithm". In: *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*. 2024, pp. 1073–1079. doi: 10.1109/COMPSAC61105.2024.00145.

[98] Rashid Amin et al. "A hybrid approach for adversarial attack detection based on sentiment analysis model using Machine learning". In: *Engineering Science and Technology, an International Journal* 58 (Oct. 2024), p. 101829. issn: 2215-0986. doi: 10.1016/j.jestch.2024.101829.

[99] Kousik Barik and Sanjay Misra. "Adversarial attack defense analysis: An empirical approach in cybersecurity perspective". In: *Software Impacts* 21 (Sept. 2024), p. 100681. issn: 2665-9638. doi: 10.1016/j.simpa.2024.100681.

[100] Khushnaseeb Roshan and Aasim Zafar. "Black-box adversarial transferability: An empirical study in cybersecurity perspective". In: *Computers & Security* 141 (June 2024), p. 103853. issn: 0167-4048. doi: 10.1016/j.cose.2024.103853.

[101] Sadaf Hina, Qaiser Abbas, and Kashan Ahmed. "Adversarial attacks on artificial Intelligence of Things-based operational technologies in theme parks". In: *Internet of Things* 32 (2025), p. 101654. issn: 2542-6605. doi: https://doi.org/10.1016/j.iot.2025.101654.

[102] Washington Garcia et al. "Brittle Features of Device Authentication". In: *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*. CODASPY '21. event-place: Virtual Event, USA. New York, NY, USA: Association for Computing Machinery, 2021, pp. 53–64. isbn: 978-1-4503-8143-7. doi: 10.1145/3422337.3447842.

[103] Anyuan Sang et al. "Obfuscating Provenance-Based Forensic Investigations with Mapping System Meta-Behavior". In: *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*. RAID '24. event-place: Padua, Italy. New York, NY, USA: Association for Computing Machinery, 2024, pp. 248–262. isbn: 9798400709593. doi: 10.1145/3678890.3678916.

[104] Burhan Ul Haque Sheikh. "Mitigating adversarial threats in deep CT image diagnosis models via a dual-stage inference-time defense". In: *Applied Soft Computing* 163 (Sept. 2024), p. 111909. issn: 1568-4946. doi: 10.1016/j.asoc.2024.111909.

[105] Zhangying He, Houman Homayoun, and Hossein Sayadi. "Beyond Conventional Defenses: Proactive and Adversarial-Resilient Hardware Malware Detection using Deep Reinforcement Learning". In: *Proceedings of the 61st ACM/IEEE Design Automation Conference*. DAC '24. event-place: San Francisco, CA, USA. New York, NY, USA: Association for Computing Machinery, 2024. isbn: 9798400706011. doi: 10.1145/3649329.3658252.

[106] Rakesh Podder and Sudipto Ghosh. "Impact of White-Box Adversarial Attacks on Convolutional Neural Networks". In: *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*. 2024, pp. 1–9. doi: 10.1109/ETNCC63262.2024.10767521.

[107] Ahmad Chaddad et al. "EAMAPG: Explainable Adversarial Model Analysis via Projected Gradient Descent". In: *Computers in Biology and Medicine* 188 (2025), p. 109788. issn: 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2025.109788.

[108] Hemashree P and Padmavathi G. "Resilience in Remote Sensing Image Classification: Evaluating Deep Learning Models Against Adversarial Attacks". In: *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2024, pp. 1–7. doi: 10.1109/ICCCNT61001.2024.10724534.

[109] Gayathri R.G., Atul Sajjanhar, and Yong Xiang. "Hybrid deep learning model using SPCAGAN augmentation for insider threat analysis". In: *Expert Systems with Applications* 249 (Sept. 2024), p. 123533. issn: 0957-4174. doi: 10.1016/j.eswa.2024.123533.

[110] Neha Gupta et al. "An Efficient Distributed Intrusion Detection System in IoT: GAN-Based Attacks and a Countermeasure". In: *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2023, pp. 1824–1829. doi: 10.1109/ICCWorkshops57953.2023.10283577.

[111] Enis Kara, Mustafa Sinasi Ayas, and Selen Ayas. "Analysis of CNN-LSTM Model Performance Under Targeted Adversarial Attacks in Water Treatment System". In: *2024 47th International Conference on Telecommunications and Signal Processing (TSP)*. 2024, pp. 107–110. doi: 10.1109/TSP63128.2024.10605932.

[112] Md Nazmul Kabir Sikder et al. "Deep H2O: Cyber attacks detection in water distribution systems using deep learning". In: *Journal of Water Process Engineering* 52 (Apr. 2023), p. 103568. issn: 2214-7144. doi: 10.1016/j.jwpe.2023.103568.

[113] Huilin Yin et al. "On Adversarial Robustness of Semantic Segmentation Models for Automated Driving". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. 2022, pp. 867–873. doi: 10.1109/IV51971.2022.9827460.

[114] Jindi Zhang et al. "Evaluating Adversarial Attacks on Driving Safety in Vision-Based Autonomous Vehicles". In: *IEEE Internet of Things Journal* 9.5 (2022), pp. 3443–3456. doi: 10.1109/JIOT.2021.3099164.

[115] Yuan Bian et al. "Modality Unified Attack for Omni-Modality Person Re-Identification". In: *IEEE Transactions on Information Forensics and Security* 20 (2025), pp. 5577–5587. doi: 10.1109/TIFS.2025.3566993.

[116] Wenjie Ding et al. "Beyond Universal Person Re-Identification Attack". In: *IEEE Transactions on Information Forensics and Security* 16 (2021), pp. 3442–3455. doi: 10.1109/TIFS.2021.3081247.

[117] Zhida Bao et al. "OATGA: Optimizing Adversarial Training via Genetic Algorithm for Automatic Modulation Classification". In: *GLOBECOM 2023 - 2023 IEEE Global*

Communications Conference. 2023, pp. 6073–6078. doi: 10.1109/GLOBECOM54140.2023.10437810.

[118] Ping Guo et al. "Exploring the Adversarial Frontier: Quantifying Robustness via Adversarial Hypervolume". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 9.2 (2025), pp. 1367–1378. doi: 10.1109/TETCI.2025.3535656.

[119] Chaitanya Devaguptapu et al. "On Adversarial Robustness: A Neural Architecture Search perspective". In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021, pp. 152–161. doi: 10.1109/ICCVW54120.2021.00022.

[120] Yingzhe He et al. "Towards Security Threats of Deep Learning Systems: A Survey". In: *IEEE Transactions on Software Engineering* 48.5 (2022), pp. 1743–1770. doi: 10.1109/TSE.2020.3034721.

[121] Afia Afrin and Omid Ardakanian. "Adversarial Attacks on Machine Learning-Based State Estimation in Power Distribution Systems". In: *Proceedings of the 14th ACM International Conference on Future Energy Systems*. e-Energy '23. event-place: Orlando, FL, USA. New York, NY, USA: Association for Computing Machinery, 2023, pp. 446–458. isbn: 9798400700323. doi: 10.1145/3575813.3597352.

[122] Haider Ali et al. "A Survey on Attacks and Their Countermeasures in Deep Learning: Applications in Deep Neural Networks, Federated, Transfer, and Deep Reinforcement Learning". In: *IEEE Access* 11 (2023), pp. 120095–120130. doi: 10.1109/ACCESS.2023.3326410.

[123] Elif Değirmenci, İlker Özçelik, and Ahmet Yazici. "Adversarial Attack Detection Approach for Intrusion Detection Systems". In: *IEEE Access* 12 (2024), pp. 195996–196009. doi: 10.1109/ACCESS.2024.3520406.

[124] Keane Lucas et al. "Training Robust ML-based Raw-Binary Malware Detectors in Hours, not Months". In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. CCS '24. Salt Lake City, UT, USA: Association for Computing Machinery, 2024, pp. 124–138. isbn: 9798400706363. doi: 10.1145/3658644.3690208.

[125] Mengting Xu et al. "Towards evaluating the robustness of deep diagnostic models by adversarial attack". In: *Medical Image Analysis* 69 (Apr. 2021), p. 101977. issn: 1361-8415. doi: 10.1016/j.media.2021.101977.

[126] Ronghui Mu et al. "Enhancing robustness in video recognition models: Sparse adversarial attacks and beyond". In: *Neural Networks* 171 (Mar. 2024), pp. 127–143. issn: 0893-6080. doi: 10.1016/j.neunet.2023.11.056.

[127] Abderrahmen Amich and Birhanu Eshete. "EG-Booster: Explanation-Guided Booster of ML Evasion Attacks". In: *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*. CODASPY '22. event-place: Baltimore, MD, USA. New York, NY, USA: Association for Computing Machinery, 2022, pp. 16–28. isbn: 978-1-4503-9220-4. doi: 10.1145/3508398.3511510.

[128] Orel Lavie, Asaf Shabtai, and Gilad Katz. "Cost effective transfer of reinforcement learning policies". In: *Expert Systems with Applications* 237 (Mar. 2024), p. 121380. issn: 0957-4174. doi: 10.1016/j.eswa.2023.121380.

[129] Xiaofeng Qiu and Shuya Zhou. "Generating adversarial examples with input significance indicator". In: *Neurocomputing* 394 (June 2020), pp. 1–12. issn: 0925-2312. doi: 10.1016/j.neucom.2020.01.040.

[130] Lianguang Liu et al. "Saliency-Aware Generation of Adversarial Point Clouds". In: *2023 15th International Conference on Advanced Computational Intelligence (ICACI)*. 2023, pp. 1–8. doi: 10.1109/ICACI58115.2023.10146196.

[131] Zhe Zhao et al. "Attack as defense: characterizing adversarial examples using robustness". In: *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. ISSTA 2021. event-place: Virtual, Denmark. New York, NY, USA: Association for Computing Machinery, 2021, pp. 42–55. isbn: 978-1-4503-8459-9. doi: 10.1145/3460319.3464822.

[132] Aleksandar Jankovic and Rudolf Mayer. "An Empirical Evaluation of Adversarial Examples Defences, Combinations and Robustness Scores". In: *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*. IWSPA '22. event-place: Baltimore, MD, USA. New York, NY, USA: Association for Computing Machinery, 2022, pp. 86–92. isbn: 978-1-4503-9230-3. doi: 10.1145/3510548.3519370.

[133] Jiachun Li, Yuchao Hu, and Fei Xia. "A variable adversarial attack method based on filtering". In: *Computers & Security* 134 (Nov. 2023), p. 103431. issn: 0167-4048. doi: 10.1016/j.cose.2023.103431.

[134] Zisheng Xu and Qiao Yan. "Boosting the transferability of adversarial CAPTCHAs". In: *Computers & Security* 145 (Oct. 2024), p. 104000. issn: 0167-4048. doi: 10.1016/j.cose.2024.104000.

[135] Peyman Rasouli and Ingrid Chieh Yu. "Analyzing and Improving the Robustness of Tabular Classifiers using Counterfactual Explanations". In: *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2021, pp. 1286–1293. doi: 10.1109/ICMLA52953.2021.00209.

[136] Zhida Bao et al. "Threat of Adversarial Attacks on DL-Based IoT Device Identification". In: *IEEE Internet of Things Journal* 9.11 (2022), pp. 9012–9024. doi: 10.1109/JIOT.2021.3120197.

[137] Xiaoning Du et al. "Marble: model-based robustness analysis of stateful deep learning systems". In: *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*. ASE '20. event-place: Virtual Event, Australia. New York, NY, USA: Association for Computing Machinery, 2021, pp. 423–435. isbn: 978-1-4503-6768-4. doi: 10.1145/3324884.3416564.

[138] Jiahao Zhao, Wenji Mao, and Daniel Dajun Zeng. "Disentangled Text Representation Learning With Information-Theoretic Perspective for Adversarial Robustness". In: *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 32 (Jan. 2024). Publisher: IEEE Press, pp. 1237–1247. issn: 2329-9290. doi: 10.1109/TASLP.2024.3358052.

[139] Lei Xu et al. "SNN-GST: Gradient-Based Security Testing Method for Spiking Neural Networks". In: *2025 7th International Conference on Software Engineering and Computer Science (CSECS)*. 2025, pp. 1–6. doi: 10.1109/CSECS64665.2025.11009396.

[140] Xiaoliang Wu and Ajitha Rajan. "Catch Me If You Can: Blackbox Adversarial Attacks on Automatic Speech Recognition using Frequency Masking". In: *2022 29th Asia-Pacific Software Engineering Conference (APSEC)*. 2022, pp. 169–178. doi: 10.1109/APSEC57359.2022.00029.

[141] Xinyu Zhang et al. "A Highly Stealthy Adaptive Decay Attack Against Speaker Recognition". In: *IEEE Access* 10 (2022), pp. 118789–118805. doi: 10.1109/ACCESS.2022.3220639.

[142] Qi Liang, Qiang Li, and Song Yang. "LP-GAN: Learning perturbations based on generative adversarial networks for point cloud adversarial attacks". In: *Image and Vision Computing* 120 (Apr. 2022), p. 104370. issn: 0262-8856. doi: 10.1016/j.imavis.2021.104370.

[143] Fabio Valerio Massoli, Fabrizio Falchi, and Giuseppe Amato. "Cross-resolution face recognition adversarial attacks". In: *Pattern Recognition Letters* 140 (Dec. 2020), pp. 222–229. issn: 0167-8655. doi: 10.1016/j.patrec.2020.10.008.

[144] The Duy Phan et al. "Leveraging Reinforcement Learning and Generative Adversarial Networks to Craft Mutants of Windows Malware against Black-box Malware Detectors". In: *Proceedings of the 11th International Symposium on Information and Communication Technology*. SoICT '22. event-place: Hanoi, Vietnam. New York, NY, USA: Association for Computing Machinery, 2022, pp. 31–38. isbn: 978-1-4503-9725-4. doi: 10.1145/3568562.3568636.

[145] Fei Ren et al. "ADVRET: An Adversarial Robustness Evaluating and Testing Platform for Deep Learning Models". In: *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. 2021, pp. 9–14. doi: 10.1109/QRS-C55045.2021.00012.

[146] Zigang Chen et al. "A method for recovering adversarial samples with both adversarial attack forensics and recognition accuracy". In: *Computers & Security* 144 (Sept. 2024), p. 103987. issn: 0167-4048. doi: 10.1016/j.cose.2024.103987.

[147] Junfan Zhou et al. "Attributed Scattering Center Guided Adversarial Attack for DCNN SAR Target Recognition". In: *IEEE Geoscience and Remote Sensing Letters* 20 (2023), pp. 1–5. doi: 10.1109/LGRS.2023.3235051.

[148] Charles Meyers et al. "A Training Rate and Survival Heuristic for Inference and Robustness Evaluation (Trashfire)". In: *2024 International Conference on Machine Learning and Cybernetics (ICMLC)*. 2024, pp. 613–623. doi: 10.1109/ICMLC63072.2024.10935101.

[149] Marwan Omar et al. "Quantifying the Performance of Adversarial Training on Language Models with Distribution Shifts". In: *Proceedings of the 1st Workshop on Cybersecurity and Social Sciences*. CySSS '22. event-place: Nagasaki, Japan. New York, NY, USA: Association for Computing Machinery, 2022, pp. 3–9. isbn: 978-1-4503-9177-1. doi: 10.1145/3494108.3522764.

[150] Shahroz Tariq, Sowon Jeon, and Simon S. Woo. "Am I a Real or Fake Celebrity? Evaluating Face Recognition and Verification APIs under Deepfake Impersonation Attack". In: *Proceedings of the ACM Web Conference 2022*. WWW '22. event-place: Virtual Event, Lyon, France. New York, NY, USA: Association for Computing Machinery, 2022, pp. 512–523. isbn: 978-1-4503-9096-5. doi: 10.1145/3485447.3512212.

[151] Harsh Kasyap and Somanath Tripathy. "Beyond data poisoning in federated learning". In: *Expert Systems with Applications* 235 (Jan. 2024), p. 121192. issn: 0957-4174. doi: 10.1016/j.eswa.2023.121192.

[152] Huangzhao Zhang et al. "Towards Robustness of Deep Program Processing Models—Detection, Estimation, and Enhancement". In: *ACM Trans. Softw. Eng. Methodol.* 31.3 (Apr. 2022). Place: New York, NY, USA Publisher: Association for Computing Machinery. issn: 1049-331X. doi: 10.1145/3511887.

[153] Bingjun He et al. "A Convenient Deep Learning Model Attack and Defense Evaluation Analysis Platform". In: *2023 8th International Conference on Computer and Communication Systems (ICCCS)*. 2023, pp. 1109–1116. doi: 10.1109/ICCCS57501.2023.10151180.

[154] Hamid Eghbal-zadeh et al. "Rethinking data augmentation for adversarial robustness". In: *Information Sciences* 654 (Jan. 2024), p. 119838. issn: 0020-0255. doi: 10.1016/j.ins.2023.119838.

[155] Shruti Jaiswal, Krishna Chaitanya Gollapudi, and R Susma. "Comprehensive Framework for Robustness evaluation on Numeric data classification". In: *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2024, pp. 1–6. doi: 10.1109/ICCCNT61001.2024.10725606.

[156] Shubham Sharma et al. "FaiR-N: Fair and Robust Neural Networks for Structured Data". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '21. event-place: Virtual Event, USA. New York, NY, USA: Association for Computing Machinery, 2021, pp. 946–955. isbn: 978-1-4503-8473-5. doi: 10.1145/3461702.3462559.

[157] Daniel Gibert, Giulio Zizzo, and Quan Le. "Certified Robustness of Static Deep Learning-based Malware Detectors against Patch and Append Attacks". In: *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. AISec '23. event-place: Copenhagen, Denmark. New York, NY, USA: Association for Computing Machinery, 2023, pp. 173–184. isbn: 9798400702600. doi: 10.1145/3605764.3623914.

[158] Ying Chen et al. "Black-box Attack against Self-supervised Video Object Segmentation Models with Contrastive Loss". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 20.2 (Oct. 2023). Place: New York, NY, USA Publisher: Association for Computing Machinery. issn: 1551-6857. doi: 10.1145/3617502.

[159] Jiarong Xu et al. "Robustness of deep learning models on graphs: A survey". In: *AI Open* 2 (Jan. 2021), pp. 69–78. issn: 2666-6510. doi: 10.1016/j.aiopen.2021.05.002.

[160] Jinyin Chen et al. *Can Adversarial Network Attack be Defended?* 2019. arXiv: 1903.05994 [cs.SI]. url: https://arxiv.org/abs/1903.05994.

[161] Sergei Chuprov et al. "Are Industrial ML Image Classifiers Robust to Withstand Adversarial Attacks on Videos?" In: *2023 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*. 2023, pp. 1–4. doi: 10.1109/WNYISPW60588.2023.10349595.

[162] Omer Faruk Tuna, Ferhat Ozgur Catak, and M. Taner Eskil. "Closeness and uncertainty aware adversarial examples detection in adversarial machine learning". In: *Computers and Electrical Engineering* 101 (July 2022), p. 107986. issn: 0045-7906. doi: 10.1016/j.compeleceng.2022.107986.

[163] Quentin Bouniot, Romaric Audigier, and Angelique Loesch. "Optimal Transport as a Defense Against Adversarial Attacks". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. 2021, pp. 5044–5051. doi: 10.1109/ICPR48806.2021.9413327.

[164] Chenkang Zhang et al. "Doubly Robust AUC Optimization against Noisy and Adversarial Samples". In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '23. event-place: Long Beach, CA, USA. New York, NY, USA: Association for Computing Machinery, 2023, pp. 3195–3205. isbn: 9798400701030. doi: 10.1145/3580305.3599316.

[165] Khushnaseeb Roshan, Aasim Zafar, and Shiekh Burhan Ul Haque. "Untargeted white-box adversarial attack with heuristic defence methods in real-time deep learning based network intrusion detection system". In: *Computer Communications* 218 (Mar. 2024), pp. 97–113. issn: 0140-3664. doi: 10.1016/j.comcom.2023.09.030.

[166] Ms Khushnaseeb Roshan and Aasim Zafar. "Boosting robustness of network intrusion detection systems: A novel two phase defense strategy against untargeted white-box optimization adversarial attack". In: *Expert Systems with Applications* 249 (Sept. 2024), p. 123567. issn: 0957-4174. doi: 10.1016/j.eswa.2024.123567.

[167] Jinhan Kim, Robert Feldt, and Shin Yoo. "Evaluating Surprise Adequacy for Deep Learning System Testing". In: *ACM Trans. Softw. Eng. Methodol.* 32.2 (Mar. 2023). Place: New York, NY, USA Publisher: Association for Computing Machinery. issn: 1049-331X. doi: 10.1145/3546947.

[168] Mengyue Yang et al. "Specify Robust Causal Representation from Mixed Observations". In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '23. event-place: Long Beach, CA, USA. New York, NY, USA: Association for Computing Machinery, 2023, pp. 2978–2987. isbn: 9798400701030. doi: 10.1145/3580305.3599512.

[169] Fahri Anıl Yerlikaya and Şerif Bahtiyar. "Data poisoning attacks against machine learning algorithms". In: *Expert Systems with Applications* 208 (Dec. 2022), p. 118101. issn: 0957-4174. doi: 10.1016/j.eswa.2022.118101.

[170] Luke E. Richards, Edward Raff, and Cynthia Matuszek. "Measuring Equality in Machine Learning Security Defenses: A Case Study in Speech Recognition". In: *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. AISec '23. event-place: Copenhagen, Denmark. New York, NY, USA: Association for Computing Machinery, 2023, pp. 161–171. isbn: 9798400702600. doi: 10.1145/3605764.3623911.

[171] Shilong Bao et al. "AUCPro: AUC-Oriented Provable Robustness Learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47.6 (2025), pp. 4579–4596. doi: 10.1109/TPAMI.2025.3545639.

[172] Masoumeh Mohammadi and Insoo Sohn. "Adversarial defense for battery state-of-health prediction models". In: *ICT Express* 11.3 (2025), pp. 436–441. issn: 2405-9595. doi: https://doi.org/10.1016/j.icte.2025.03.011.

[173] Hsin-Yi Lin, Huan-Hsin Tseng, and Yu Tsao. "On the Robustness of Non-Intrusive Speech Quality Model by Adversarial Examples". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. doi: 10.1109/ICASSP49357.2023.10097261.

[174] Wei Liang et al. "QoS Prediction and Adversarial Attack Protection for Distributed Services Under DLaaS". In: *IEEE Transactions on Computers* 73.3 (2024), pp. 669–682. doi: 10.1109/TC.2021.3077738.

[175] Aidong Xu et al. "Adversarial Attacks on Deep Neural Networks for Time Series Prediction". In: *2021 10th International Conference on Internet Computing for Science and Engineering*. ICICSE 2021. event-place: Guilin, China. New York, NY, USA: Association for Computing Machinery, 2022, pp. 8–14. isbn: 978-1-4503-8495-7. doi: 10.1145/3485314.3485316.

[176] Xin Jin et al. "Adversarial attacks on multi-focus image fusion models". In: *Computers & Security* 134 (Nov. 2023), p. 103455. issn: 0167-4048. doi: 10.1016/j.cose.2023.103455.

[177] Alexander Hartl et al. "Explainability and Adversarial Robustness for RNNs". In: *2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService)*. 2020, pp. 148–156. doi: 10.1109/BigDataService49289.2020.00030.

[178] Osbert Bastani et al. "Measuring neural net robustness with constraints". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 2621–2629. isbn: 9781510838819.

[179] Jie Wang et al. "The Quantitative Relationship between Adversarial Training and Robustness of CNN Model". In: *2020 7th International Conference on Dependable Systems and Their Applications (DSA)*. 2020, pp. 543–549. doi: 10.1109/DSA51864.2020.00092.

[180] Haibo Jin et al. "ROBY: Evaluating the adversarial robustness of a deep model by its decision boundaries". In: *Information Sciences* 587 (Mar. 2022), pp. 97–122. issn: 0020-0255. doi: 10.1016/j.ins.2021.12.021.

[181] Hardhik Mohanty, Arousha Haghighian Roudsari, and Arash Habibi Lashkari. "Robust stacking ensemble model for darknet traffic classification under adversarial settings". In: *Computers & Security* 120 (Sept. 2022), p. 102830. issn: 0167-4048. doi: 10.1016/j.cose.2022.102830.

[182] Wen Tang et al. "Deep transform and metric learning network: Wedding deep dictionary learning and neural network". In: *Neurocomputing* 509 (Oct. 2022), pp. 244–256. issn: 0925-2312. doi: 10.1016/j.neucom.2022.08.069.

[183] Jon Vadillo and Roberto Santana. "On the human evaluation of universal audio adversarial perturbations". In: *Computers & Security* 112 (Jan. 2022), p. 102495. issn: 0167-4048. doi: 10.1016/j.cose.2021.102495.

[184] Shawqi Al-Maliki et al. "Defending Emotional Privacy with Adversarial Machine Learning for Social Good". In: *2023 International Wireless Communications and Mobile Computing (IWCMC)*. 2023, pp. 345–350. doi: 10.1109/IWCMC58020.2023.10182780.

[185] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2574–2582. doi: 10.1109/CVPR.2016.282.

[186] Héctor D. Menéndez. "Measuring Machine Learning Robustness in front of Static and Dynamic Adversaries*". In: *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*. 2022, pp. 174–181. doi: 10.1109/ICTAI56018.2022.00033.

[187] Khoi Nguyen Tiet Nguyen et al. "A Survey and Evaluation of Adversarial Attacks in Object Detection". In: *IEEE Transactions on Neural Networks and Learning Systems* (2025), pp. 1–17. doi: 10.1109/TNNLS.2025.3561225.

[188] Xiaowei Huang et al. "Safety Verification of Deep Neural Networks". In: July 2017, pp. 3–29. isbn: 978-3-319-63386-2. doi: 10.1007/978-3-319-63387-9_1.

[189] Tsui-Wei Weng et al. "Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach". In: (Jan. 2018). doi: 10.48550/arXiv.1801.10578.

[190] Huan Xu and Shie Mannor. "Robustness and generalization". In: *Machine Learning* 86.3 (Mar. 2012), pp. 391–423. issn: 1573-0565. doi: 10.1007/s10994-011-5268-1.

[191] Kaijie Shen and Chengju Li. "A Method to Verify Neural Network Decoders Against Adversarial Attacks". In: *IEEE Communications Letters* 29.4 (2025), pp. 843–847. doi: 10.1109/LCOMM.2025.3545068.

[192] Ke Wang et al. "Uncovering Hidden Vulnerabilities in Convolutional Neural Networks through Graph-based Adversarial Robustness Evaluation". In: *Pattern Recognition* 143 (Nov. 2023), p. 109745. issn: 0031-3203. doi: 10.1016/j.patcog.2023.109745.

[193] Chongzhi Zhang et al. "Interpreting and Improving Adversarial Robustness of Deep Neural Networks With Neuron Sensitivity". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 1291–1304. doi: 10.1109/TIP.2020.3042083.

[194] Jun Guo et al. "A comprehensive evaluation framework for deep model robustness". In: *Pattern Recognition* 137 (2023), p. 109308. issn: 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109308.

[195] Aishan Liu et al. "Training Robust Deep Neural Networks via Adversarial Noise Propagation". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 5769–5781. doi: 10.1109/TIP.2021.3082317.

[196] Marcin Waniek et al. "Hiding individuals and communities in a social network". In: *Nature Human Behaviour* 2.2 (Feb. 2018), pp. 139–147. issn: 2397-3374. doi: 10.1038/s41562-017-0290-3.

[197] Mingjie Sun et al. *Data Poisoning Attack against Unsupervised Node Embedding Methods*. 2018. arXiv: 1810.12881 [cs.LG]. url: https://arxiv.org/abs/1810.12881.

[198] Aleksandar Bojchevski and Stephan Günnemann. "Certifiable robustness to graph perturbations". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[199] Kai Zhou, Tomasz P. Michalak, and Yevgeniy Vorobeychik. "Adversarial Robustness of Similarity-Based Link Prediction". In: *2019 IEEE International Conference on Data Mining (ICDM)*. 2019, pp. 926–935. doi: 10.1109/ICDM.2019.00103.

[200] Xiaolong Liu et al. "Image steganography with high embedding capacity based on multi-target adversarial attack". In: *Engineering Applications of Artificial Intelligence* 156 (2025), p. 111341. issn: 0952-1976. doi: https://doi.org/10.1016/j.engappai.2025.111341.

[201] Benjamin A Miller et al. "Improving robustness to attacks against vertex classification". In: *MLG Workshop*. 2019.

[202] Jinyin Chen et al. *Fast Gradient Attack on Network Embedding*. 2018. arXiv: 1809.02797 [physics.soc-ph]. url: https://arxiv.org/abs/1809.02797.

[203] Sicheng Zhang et al. "Channel-Robust Class-Universal Spectrum-Focused Frequency Adversarial Attacks on Modulated Classification Models". In: *IEEE Transactions on Cognitive Communications and Networking* 10.4 (2024), pp. 1280–1293. doi: 10.1109/TCCN.2024.3382126.

[204] Zhipeng He et al. "Investigating imperceptibility of adversarial attacks on tabular data: An empirical analysis". In: *Intelligent Systems with Applications* 25 (2025), p. 200461. issn: 2667-3053. doi: https://doi.org/10.1016/j.iswa.2024.200461.

[205] Kazuto Fukuchi, Satoshi Hara, and Takanori Maehara. "Faking Fairness via Stealthily Biased Sampling". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (Apr. 2020), pp. 412–419. doi: 10.1609/aaai.v34i01.5377.

[206] Xi Chen et al. "Diversity supporting robustness: Enhancing adversarial robustness via differentiated ensemble predictions". In: *Computers & Security* 142 (July 2024), p. 103861. issn: 0167-4048. doi: 10.1016/j.cose.2024.103861.

[207] Syed M. Hasan, Abdur R. Shahid, and Ahmed Imteaj. "Evaluating Sustainability and Social Costs of Adversarial Training in Machine Learning". In: *IEEE Consumer Electronics Magazine* (2024), pp. 1–6. doi: 10.1109/MCE.2024.3458350.

[208] Syed Mhamudul Hasan, Abdur R. Shahid, and Ahmed Imteaj. "Towards Sustainable SecureML: Quantifying Carbon Footprint of Adversarial Machine Learning". In: *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2024, pp. 1359–1364. doi: 10.1109/ICCWorkshops59551.2024.10615723.

[209] Ivan Vaccari et al. "eXplainable and Reliable Against Adversarial Machine Learning in Data Analytics". In: *IEEE Access* 10 (2022), pp. 83949–83970. doi: 10.1109/ACCESS.2022.3197299.

[210] Hubert Baniecki and Przemyslaw Biecek. "Adversarial attacks and defenses in explainable artificial intelligence: A survey". In: *Information Fusion* 107 (July 2024), p. 102303. issn: 1566-2535. doi: 10.1016/j.inffus.2024.102303.

[211] Prithwijit Chowdhury et al. "Are Objective Explanatory Evaluation Metrics Trustworthy? An Adversarial Analysis". In: *2024 IEEE International Conference on Image Processing (ICIP)*. 2024, pp. 3938–3944. doi: 10.1109/ICIP51287.2024.10647779.

[212] Erikson J. De Aguiar, Caetano Traina, and Agma J. M. Traina. "RADAR-MIX: How to Uncover Adversarial Attacks in Medical Image Analysis through Explainability". In: *2024 IEEE 37th International Symposium on Computer-Based Medical Systems (CBMS)*. 2024, pp. 436–441. doi: 10.1109/CBMS61543.2024.00078.

[213] Shen Wang et al. "Towards Accountable and Resilient AI-Assisted Networks: Case Studies and Future Challenges". In: *2024 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. 2024, pp. 818–823. doi: 10.1109/EuCNC/6GSummit60053.2024.10597060.

[214] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144. isbn: 9781450342322. doi: 10.1145/2939672.2939778.

[215] Scott M. Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777. isbn: 9781510860964.

[216] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 3319–3328.

[217] Rghda Salah et al. "Efficient Detection of Black Box Adversarial Attacks in Machine Learning Systems Using Cluster-Based and Class-Based Input Patterns". In: *2024 34th International Conference on Computer Theory and Applications (ICCTA)*. 2024, pp. 158–164. doi: 10.1109/ICCTA64612.2024.10974889.

[218] Anirban Chakraborty et al. "A survey on adversarial attacks and defences". In: *CAAI Transactions on Intelligence Technology* 6.1 (Mar. 2021), pp. 25–45. issn: 2468-2322. doi: 10.1049/cit2.12028.

[219] Ishai Rosenberg et al. "Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain". In: *ACM Comput. Surv.* 54.5 (May 2021), 108:1–108:36. issn: 0360-0300. doi: 10.1145/3453158.

[220] Xiaoyong Yuan et al. "Adversarial Examples: Attacks and Defenses for Deep Learning". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.9 (Sept. 2019). Conference Name: IEEE Transactions on Neural Networks and Learning Systems, pp. 2805–2824. issn: 2162-2388. doi: 10.1109/TNNLS.2018.2886017.

[221] Fatemeh Vakhshiteh, Ahmad Nickabadi, and Raghavendra Ramachandra. "Adversarial Attacks Against Face Recognition: A Comprehensive Study". In: *IEEE Access* 9 (2021). Conference Name: IEEE Access, pp. 92735–92756. issn: 2169-3536. doi: 10.1109/ACCESS.2021.3092646.

[222] Muhammad Tayyab et al. "A comprehensive review on deep learning algorithms: Security and privacy issues". In: *Computers & Security* 131 (Aug. 2023), p. 103297. issn: 0167-4048. doi: 10.1016/j.cose.2023.103297.

[223] Chenyu Zhang et al. "Adversarial attacks and defenses on text-to-image diffusion models: A survey". In: *Information Fusion* 114 (Feb. 2025), p. 102701. issn: 1566-2535. doi: 10.1016/j.inffus.2024.102701.

[224] Felix Assion et al. "The Attack Generator: A Systematic Approach Towards Constructing Adversarial Attacks". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. ISSN: 2160-7516. June 2019, pp. 1370–1379. doi: 10.1109/CVPRW.2019.00177.

[225] Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. "Adversarial Machine Learning in Image Classification: A Survey Toward the Defender's Perspective". In: *ACM Comput. Surv.* 55.1 (Nov. 2021), 8:1–8:38. issn: 0360-0300. doi: 10.1145/3485133.

[226] Fatimah Aloraini et al. "Adversarial machine learning in IoT from an insider point of view". In: *Journal of Information Security and Applications* 70 (Nov. 2022), p. 103341. issn: 2214-2126. doi: 10.1016/j.jisa.2022.103341.

[227] Jiahe Lan et al. "Adversarial attacks and defenses in Speaker Recognition Systems: A survey". In: *Journal of Systems Architecture* 127 (June 2022), p. 102526. issn: 1383-7621. doi: 10.1016/j.sysarc.2022.102526.

[228] Teng Long et al. "A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions". In: *Computers & Security* 121 (Oct. 2022), p. 102847. issn: 0167-4048. doi: 10.1016/j.cose.2022.102847.

[229] Damilola Adesina et al. "Adversarial Machine Learning in Wireless Communications Using RF Data: A Review". In: *IEEE Communications Surveys & Tutorials* 25.1 (2023). Conference Name: IEEE Communications Surveys & Tutorials, pp. 77–100. issn: 1553-877X. doi: 10.1109/COMST.2022.3205184.

[230] Panagiotis Bountakas et al. "Defense strategies for Adversarial Machine Learning: A survey". In: *Computer Science Review* 49 (Aug. 2023), p. 100573. issn: 1574-0137. doi: 10.1016/j.cosrev.2023.100573.

[231] Maria Rigaki and Sebastian Garcia. "A Survey of Privacy Attacks in Machine Learning". In: *ACM Comput. Surv.* 56.4 (Nov. 2023), 101:1–101:34. issn: 0360-0300. doi: 10.1145/3624010.

[232] Nuria Rodríguez-Barroso et al. "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges". In: *Information Fusion* 90 (Feb. 2023), pp. 148–173. issn: 1566-2535. doi: 10.1016/j.inffus.2022.09.011.

[233] Anjan K Koundinya, S S Patil, and Chandu B R. "Data Poisoning Attacks in Cognitive Computing". In: *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*. Apr. 2024, pp. 1–4. doi: 10.1109/I2CT61223.2024.10544345.

[234] Mayra Macas, Chunming Wu, and Walter Fuertes. "Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems". In: *Expert Systems with Applications* 238 (Mar. 2024), p. 122223. issn: 0957-4174. doi: 10.1016/j.eswa.2023.122223.

[235] Parya Haji Mirzaee et al. "Smart Grid Security and Privacy: From Conventional to Machine Learning Issues (Threats and Countermeasures)". In: *IEEE Access* 10 (2022). Conference Name: IEEE Access, pp. 52922–52954. issn: 2169-3536. doi: 10.1109/ACCESS.2022.3174259.

[236] Jia Wang et al. "Adversarial attacks and defenses in deep learning for image recognition: A survey". In: *Neurocomputing* 514 (Dec. 2022), pp. 162–181. issn: 0925-2312. doi: 10.1016/j.neucom.2022.09.004.

[237] Nikolaos Pitropakis et al. "A taxonomy and survey of attacks against machine learning". In: *Computer Science Review* 34 (Nov. 2019), p. 100199. issn: 1574-0137. doi: 10.1016/j.cosrev.2019.100199.

[238] Koosha Sadeghi, Ayan Banerjee, and Sandeep K. S. Gupta. "A System-Driven Taxonomy of Attacks and Defenses in Adversarial Machine Learning". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 4.4 (Aug. 2020). Conference Name: IEEE Transactions on Emerging Topics in Computational Intelligence, pp. 450–467. issn: 2471-285X. doi: 10.1109/TETCI.2020.2968933.

[239] Daniel Machooka, Xiaohong Yuan, and Albert Esterline. "A Survey of Attacks and Defenses for Deep Neural Networks". In: *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*. July 2023, pp. 254–261. doi: 10.1109/CSR57506.2023.10224947.

[240] John Mulo et al. "Towards an Adversarial Machine Learning Framework in Cyber-Physical Systems". In: *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*. ISSN: 2770-8209. May 2023, pp. 138–143. doi: 10.1109/SERA57763.2023.10197774.

[241] Kshitiz Aryal et al. "A Survey on Adversarial Attacks for Malware Analysis". In: *IEEE Access* 13 (2025), pp. 428–459. doi: 10.1109/ACCESS.2024.3519524.

[242] Jerzy Surma. "Hacking Machine Learning: Towards The Comprehensive Taxonomy of Attacks Against Machine Learning Systems". In: *Proceedings of the 2020 the 4th International Conference on Innovation in Artificial Intelligence*. ICIAI '20. New York, NY, USA: Association for Computing Machinery, June 2020, pp. 1–4. isbn: 978-1-4503-7658-7. doi: 10.1145/3390557.3394126.

[243] Hailong Xi et al. "Adversarial Attacks: Key Challenges for Security Defense in the Age of Intelligence". In: *2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)*. 2024, pp. 41–46. doi: 10.1109/ICAIRC64177.2024.10900089.

[244] Nicholas Dietrich, Bo Gong, and Michael N. Patlas. "Adversarial artificial intelligence in radiology: Attacks, defenses, and future considerations". In: *Diagnostic and Interventional Imaging* (May 2025). issn: 2211-5684. doi: 10.1016/j.diii.2025.05.006.

[245] Ruisi Zhang et al. "Systemization of Knowledge: Robust Deep Learning using Hardware-software co-design in Centralized and Federated Settings". In: *ACM Trans. Des. Autom. Electron. Syst.* 28.6 (Oct. 2023), 88:1–88:32. issn: 1084-4309. doi: 10.1145/3616868.

[246] Mohamed Amine Ferrag et al. "Edge Learning for 6G-Enabled Internet of Things: A Comprehensive Survey of Vulnerabilities, Datasets, and Defenses". In: *IEEE Communications Surveys & Tutorials* 25.4 (2023). Conference Name: IEEE Communications Surveys & Tutorials, pp. 2654–2713. issn: 1553-877X. doi: 10.1109/COMST.2023.3317242.

[247] Bakary Badjie, José Cecílio, and Antonio Casimiro. "Adversarial Attacks and Countermeasures on Image Classification-based Deep Learning Models in Autonomous Driving Systems: A Systematic Review". In: *ACM Comput. Surv.* 57.1 (Oct. 2024), 20:1–20:52. issn: 0360-0300. doi: 10.1145/3691625.

[248] Qingyuan Hu. "A Survey of Adversarial Example Toolboxes". In: *2021 2nd International Conference on Computing and Data Science (CDS)*. Jan. 2021, pp. 603–608. doi: 10.1109/CDS52072.2021.00109.

[249] Shilin Qiu et al. "Adversarial attack and defense technologies in natural language processing: A survey". In: *Neurocomputing* 492 (July 2022), pp. 278–307. issn: 0925-2312. doi: 10.1016/j.neucom.2022.04.020.

[250] Yong Cheng, Lu Jiang, and Wolfgang Macherey. "Robust Neural Machine Translation with Doubly Adversarial Inputs". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4324–4333. doi: 10.18653/v1/P19-1425.

[251] Motoki Sato et al. "Interpretable adversarial perturbation in input embedding space for text". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. Stockholm, Sweden: AAAI Press, 2018, pp. 4323–4330. isbn: 9780999241127.

[252] Lei Xu, Ivan Ramirez, and Kalyan Veeramachaneni. *Rewriting Meaningful Sentences via Conditional BERT Sampling and an application on fooling text classifiers*. 2022. arXiv: 2010.11869 [cs.CL]. url: https://arxiv.org/abs/2010.11869.

[253] Ji Gao et al. "Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers". In: *2018 IEEE Security and Privacy Workshops (SPW)*. 2018, pp. 50–56. doi: 10.1109/SPW.2018.00016.

[254] Shuhuai Ren et al. "Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1085–1097. doi: 10.18653/v1/P19-1103.

[255] M. Hossam et al. "Explain2Attack: Text Adversarial Attacks via Cross-Domain Interpretability". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. 2020 25th International Conference on Pattern Recognition (ICPR). 2021, pp. 8922–8928. doi: 10.1109/ICPR48806.2021.9412526.

[256] Huangzhao Zhang et al. *Generating Fluent Adversarial Examples for Natural Languages*. 2020. arXiv: 2007.06174 [cs.CL]. url: https://arxiv.org/abs/2007.06174.

[257] Robin Jia and Percy Liang. *Adversarial Examples for Evaluating Reading Comprehension Systems*. 2017. arXiv: 1707.07328 [cs.CL]. url: https://arxiv.org/abs/1707.07328.

[258] Naveen Jafer Nizar and Ari Kobren. *Leveraging Extracted Model Adversaries for Improved Black Box Attacks*. 2020. arXiv: 2010.16336 [cs.LG]. url: https://arxiv.org/abs/2010.16336.

[259] Yicheng Wang and Mohit Bansal. *Robust Machine Comprehension Models via Adversarial Training*. 2018. arXiv: 1804.06473 [cs.CL]. url: https://arxiv.org/abs/1804.06473.

[260] Bin Liang et al. "Deep text classification can be fooled". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. Stockholm, Sweden: AAAI Press, 2018, pp. 4208–4215. isbn: 9780999241127.

[261] Zhihong Shao et al. *AdvExpander: Generating Natural Language Adversarial Examples by Expanding Text*. 2020. arXiv: 2012.10235 [cs.CL]. url: https://arxiv.org/abs/2012.10235.

[262] Jinfeng Li et al. "TextBugger: Generating Adversarial Text Against Real-world Applications". In: *Proceedings 2019 Network and Distributed System Security Symposium*. NDSS 2019. Internet Society, 2019. doi: 10.14722/ndss.2019.23138.

[263] Minhao Cheng et al. "Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.04 (Apr. 2020), pp. 3601–3608. doi: 10.1609/aaai.v34i04.5767. url: https://ojs.aaai.org/index.php/AAAI/article/view/5767.

[264] Yotam Gil et al. *White-to-Black: Efficient Distillation of Black-Box Adversarial Attacks*. 2019. arXiv: 1904.02405 [cs.LG]. url: https://arxiv.org/abs/1904.02405.

[265] Javid Ebrahimi et al. "HotFlip: White-Box Adversarial Examples for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 31–36. doi: 10.18653/v1/P18-2006.

[266] Steffen Eger and Yannik Benz. *From Hero to Zéroe: A Benchmark of Low-Level Adversarial Attacks*. Oct. 2020. doi: 10.48550/arXiv.2010.05648.

[267] Liwei Song et al. *Universal Adversarial Attacks with Natural Triggers for Text Classification*. Apr. 2020. doi: 10.48550/arXiv.2005.00174.

[268] Yansong Gao et al. "Security threats to agricultural artificial intelligence: Position and perspective". In: *Computers and Electronics in Agriculture* 227 (Dec. 2024), p. 109557. issn: 0168-1699. doi: 10.1016/j.compag.2024.109557.

[269] Hanieh Naderi and Ivan V. Bajić. "Adversarial Attacks and Defenses on 3D Point Cloud Classification: A Survey". In: *IEEE Access* 11 (2023). Conference Name: IEEE Access, pp. 144274–144295. issn: 2169-3536. doi: 10.1109/ACCESS.2023.3345000.

[270] K V Priya and Peter J Dinesh. "A Detailed Study on Adversarial Attacks and Defense Mechanisms on Various Deep Learning Models". In: *2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*. Jan. 2023, pp. 1–6. doi: 10.1109/ACCTHPA57160.2023.10083378.

[271] Jiale Zhang et al. "PoisonGAN: Generative Poisoning Attacks Against Federated Learning in Edge Computing Systems". In: *IEEE Internet of Things Journal* 8.5 (Mar. 2021). Conference Name: IEEE Internet of Things Journal, pp. 3310–3322. issn: 2327-4662. doi: 10.1109/JIOT.2020.3023126.

[272] Jianpeng Guo, Chengwu Yang, and Guangning Song. "A Survey of Recognition Model Attack Algorithms in Communication Countermeasure". In: *2024 10th International Conference on Big Data and Information Analytics (BigDIA)*. 2024, pp. 619–625. doi: 10.1109/BigDIA63733.2024.10808651.

[273] João Vitorino, Isabel Praça, and Eva Maia. "SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection". In: *Computers & Security* 134 (Nov. 2023), p. 103433. issn: 0167-4048. doi: 10.1016/j.cose.2023.103433.

[274] Vishakha Sehgal et al. "Navigating The Battleground: An Analysis Of Adversarial Threats And Protections In Deep Neural Networks". In: *2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG)*. 2024, pp. 1–9. doi: 10.1109/ICTBIG64922.2024.10911402.

[275] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML]. url: https://arxiv.org/abs/1412.6572.

[276] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. *Adversarial examples in the physical world*. 2017. arXiv: 1607.02533 [cs.CV]. url: https://arxiv.org/abs/1607.02533.

[277] Aleksander Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2019. arXiv: 1706.06083 [stat.ML]. url: https://arxiv.org/abs/1706.06083.

[278] Francesco Croce and Matthias Hein. *Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks*. 2020. arXiv: 2003.01690 [cs.LG]. url: https://arxiv.org/abs/2003.01690.

[279] Francesco Croce and Matthias Hein. *Minimally distorted Adversarial Examples with a Fast Adaptive Boundary Attack*. 2020. arXiv: 1907.02044 [cs.LG]. url: https://arxiv.org/abs/1907.02044.

[280] Yinpeng Dong et al. "Boosting Adversarial Attacks with Momentum". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9185–9193. doi: 10.1109/CVPR.2018.00957.

[281] Junyu Lin et al. "Black-box adversarial sample generation based on differential evolution". In: *Journal of Systems and Software* 170 (Dec. 2020), p. 110767. issn: 0164-1212. doi: 10.1016/j.jss.2020.110767.

[282] Chengcheng Ma et al. "Efficient Joint Gradient Based Attack Against SOR Defense for 3D Point Cloud Classification". In: *Proceedings of the 28th ACM International Conference on Multimedia*. MM '20. Seattle, WA, USA: Association for Computing Machinery, 2020, pp. 1819–1827. isbn: 9781450379885. doi: 10.1145/3394171.3413875.

[283] Jaeyeon Kim et al. "Minimal Adversarial Examples for Deep Learning on 3D Point Clouds". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2021, pp. 7777–7786. doi: 10.1109/ICCV48922.2021.00770. url: https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00770.

[284] Atrin Arya, Hanieh Naderi, and Shohreh Kasaei. "Adversarial Attack by Limited Point Cloud Surface Modifications". In: *2023 6th International Conference on Pattern Recognition and Image Analysis (IPRIA)*. 2023, pp. 1–8. doi: 10.1109/IPRIA59240.2023.10147168.

[285] Nicolas Papernot et al. "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.

[286] Nicholas Carlini and David Wagner. "Towards Evaluating the Robustness of Neural Networks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2017, pp. 39–57. doi: 10.1109/SP.2017.49.

[287] Xiaokang Zhou et al. "Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System". In: *IEEE Internet of Things Journal* 9.12 (2022), pp. 9310–9319. doi: 10.1109/JIOT.2021.3130434.

[288] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. *DeepFool: a simple and accurate method to fool deep neural networks*. 2016. arXiv: 1511.04599 [cs.LG]. url: https://arxiv.org/abs/1511.04599.

[289] Hanxiao Tan and Helena Kotthaus. "Explainability-Aware One Point Attack for Point Cloud Neural Networks". In: Jan. 2023, pp. 4570–4579. doi: 10.1109/WACV56688.2023.00456.

[290] Moustapha Cisse et al. "Houdini: fooling deep structured visual and speech recognition models with adversarial examples". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6980–6990. isbn: 9781510860964.

[291] Pin-Yu Chen et al. "ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. AISec '17. Dallas, Texas, USA: Association for Computing Machinery, 2017, pp. 15–26. isbn: 9781450352024. doi: 10.1145/3128572.3140448.

[292] Ishai Rosenberg et al. "Generic Black-Box End-to-End Attack Against State of the Art API Call Based Malware Classifiers". In: *Research in Attacks, Intrusions, and Defenses*. Ed. by Michael Bailey et al. Cham: Springer International Publishing, 2018, pp. 490–510. isbn: 978-3-030-00470-5.

[293] Minhao Cheng et al. *Query-Efficient Hard-label Black-box Attack:An Optimization-based Approach*. 2018. arXiv: 1807.04457 [cs.LG]. url: https://arxiv.org/abs/1807.04457.

[294] Jianbo Chen, Michael I. Jordan, and Martin 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.

[295] Wieland Brendel, Jonas Rauber, and Matthias Bethge. *Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models*. 2018. arXiv: 1712.04248 [stat.ML]. url: https://arxiv.org/abs/1712.04248.

[296] Andrew Ilyas et al. "Black-box Adversarial Attacks with Limited Queries and Information". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 2137–2146. url: https://proceedings.mlr.press/v80/ilyas18a.html.

[297] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One Pixel Attack for Fooling Deep Neural Networks". In: *IEEE Transactions on Evolutionary Computation* 23.5 (Oct. 2019), pp. 828–841. issn: 1941-0026. doi: 10.1109/tevc.2019.2890858.

[298] Kaidi Xu et al. *Structured Adversarial Attack: Towards General Implementation and Better Interpretability*. 2019. arXiv: 1808.01664 [cs.LG]. url: https://arxiv.org/abs/1808.01664.

[299] Abdullah Hamdi et al. "AdvPC: Transferable Adversarial Perturbations on 3D Point Clouds". In: *Computer Vision – ECCV 2020*. Springer International Publishing, 2020, pp. 241–257. isbn: 9783030586102. doi: 10.1007/978-3-030-58610-2_15.

[300] Sicong Zhang, Xiaoyao Xie, and Yang Xu. "A Brute-Force Black-Box Method to Attack Machine Learning-Based Systems in Cybersecurity". In: *IEEE Access* 8 (2020), pp. 128250–128263. issn: 2169-3536. doi: 10.1109/access.2020.3008433.

[301] Chuan Guo et al. *Simple Black-box Adversarial Attacks*. 2019. arXiv: 1905.07121 [cs.LG]. url: https://arxiv.org/abs/1905.07121.

[302] Jiancheng Yang et al. *Adversarial Attack and Defense on Point Sets*. 2021. arXiv: 1902.10899 [cs.CV]. url: https://arxiv.org/abs/1902.10899.

[303] Maksym Andriushchenko et al. *Square Attack: a query-efficient black-box adversarial attack via random search*. 2020. arXiv: 1912.00049 [cs.LG]. url: https://arxiv.org/abs/1912.00049.

[304] João Vitorino, Nuno Oliveira, and Isabel Praça. "Adaptative Perturbation Patterns: Realistic Adversarial Learning for Robust Intrusion Detection". In: *Future Internet* 14.4 (Mar. 2022), p. 108. issn: 1999-5903. doi: 10.3390/fi14040108.

[305] Xiao Peng, Weiqing Huang, and Zhixin 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.

[306] Jinlai Zhang et al. "3D adversarial attacks beyond point cloud". In: *Information Sciences* 633 (2023), pp. 491–503. issn: 0020-0255. doi: https://doi.org/10.1016/j.ins.2023.03.084.

[307] Yuxin Wen et al. "Geometry-Aware Generation of Adversarial Point Clouds". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.6 (2022), pp. 2984–2999. doi: 10.1109/TPAMI.2020.3044712.

[308] Daizong Liu and Wei Hu. "Imperceptible Transfer Attack and Defense on 3D Point Cloud Classification". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.4 (2023), pp. 4727–4746. doi: 10.1109/TPAMI.2022.3193449.

[309] Keke Tang et al. "NormalAttack: Curvature-Aware Shape Deformation along Normals for Imperceptible Point Cloud Attack". In: *Security and Communication Networks* 2022 (Aug. 2022), pp. 1–11. doi: 10.1155/2022/1186633.

[310] Zhenbo Shi et al. "Shape Prior Guided Attack: Sparser Perturbations on 3D Point Clouds". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.8 (June 2022), pp. 8277–8285. doi: 10.1609/aaai.v36i8.20802.

[311] Kibok Lee et al. *ShapeAdv: Generating Shape-Aware Adversarial 3D Point Clouds*. May 2020. doi: 10.48550/arXiv.2005.11626.

[312] Keke Tang et al. "Deep manifold attack on point clouds via parameter plane stretching". In: *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. isbn: 978-1-57735-880-0. doi: 10.1609/aaai.v37i2.25338.

[313] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. "Hidden Trigger Backdoor Attacks". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07 (Apr. 2020), pp. 11957–11965. issn: 2159-5399. doi: 10.1609/aaai.v34i07.6871.

[314] Sayantan Sarkar et al. "UPSET and ANGRI : Breaking High Performance Image Classifiers". In: (July 2017). doi: 10.48550/arXiv.1707.01159.

[315] Pedro Sandoval-Segura et al. "Autoregressive Perturbations for Data Poisoning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 27374–27386. url: https://proceedings.neurips.cc/paper_files/paper/2022/file/af66ac99716a64476c07ae8b089d59f8-Paper-Conference.pdf.

[316] Seyed-Mohsen Moosavi-Dezfooli et al. *Universal adversarial perturbations*. 2017. arXiv: 1610.08401 [cs.CV]. url: https://arxiv.org/abs/1610.08401.

[317] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. url: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[318] Zongwei Wang et al. "Gray-Box Shilling Attack: An Adversarial Learning Approach". In: *ACM Trans. Intell. Syst. Technol.* 13.5 (Oct. 2022). issn: 2157-6904. doi: 10.1145/3512352.

[319] Zilong Lin, Yong Shi, and Zhi Xue. "IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by João Gama et al. Cham: Springer International Publishing, 2022, pp. 79–91. isbn: 978-3-031-05981-0.

[320] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. 2014. arXiv: 1411.1784 [cs.LG]. url: https://arxiv.org/abs/1411.1784.

[321] Bingyin Zhao and Yingjie Lao. "CLPA: Clean-Label Poisoning Availability Attacks Using Generative Adversarial Nets". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.8 (June 2022), pp. 9162–9170. issn: 2159-5399. doi: 10.1609/aaai.v36i8.20902.

[322] Ravi Chauhan et al. "Polymorphic Adversarial Cyberattacks Using WGAN". In: *Journal of Cybersecurity and Privacy* 1.4 (Dec. 2021), pp. 767–792. issn: 2624-800X. doi: 10.3390/jcp1040037.

[323] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. "Learning Structured Output Representation using Deep Conditional Generative Models". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. url: https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

[324] Hanjun Dai et al. "Adversarial Attack on Graph Structured Data". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 1115–1124. url: https://proceedings.mlr.press/v80/dai18b.html.

[325] Hua Ma et al. "TransCAB: Transferable Clean-Annotation Backdoor to Object Detection with Natural Trigger in Real-World". In: *2023 42nd International Symposium on Reliable Distributed Systems (SRDS)*. 2023, pp. 82–92. doi: 10.1109/SRDS60354.2023.00018.

[326] Guohong Wang et al. "One-to-Multiple Clean-Label Image Camouflage (OmClic) based backdoor attack on deep learning". In: *Knowledge-Based Systems* 288 (Mar. 2024), p. 111456. issn: 0950-7051. doi: 10.1016/j.knosys.2024.111456.

[327] Gilad Baruch, Moran Baruch, and Yoav Goldberg. "A Little Is Enough: Circumventing Defenses For Distributed Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. url: https://proceedings.neurips.cc/paper_files/paper/2019/file/ec1c59141046cd1866bbbcdfb6ae31d4-Paper.pdf.

[328] Virat Shejwalkar and Amir Houmansadr. "Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning". In: *Proceedings 2021 Network and Distributed System Security Symposium*. NDSS 2021. Internet Society, 2021. doi: 10.14722/ndss.2021.24498.

[329] Xiaoyu Cao and Neil Gong. "MPAF: Model Poisoning Attacks to Federated Learning based on Fake Clients". In: June 2022, pp. 3395–3403. doi: 10.1109/CVPRW56347.2022.00383.

[330] Nicolas Papernot et al. *Technical Report on the CleverHans v2.1.0 Adversarial Examples Library*. 2018. arXiv: 1610.00768 [cs.LG]. url: https://arxiv.org/abs/1610.00768.

[331] Jonas Rauber, Wieland Brendel, and Matthias Bethge. *Foolbox: A Python toolbox to benchmark the robustness of machine learning models*. 2018. arXiv: 1707.04131 [cs.LG]. url: https://arxiv.org/abs/1707.04131.

[332] Dou Goodman et al. *Advbox: a toolbox to generate adversarial examples that fool neural networks*. 2020. arXiv: 2001.05574 [cs.LG]. url: https://arxiv.org/abs/2001.05574.

[333] Maria-Irina Nicolae et al. *Adversarial Robustness Toolbox v1.0.0*. 2019. arXiv: 1807.01069 [cs.LG]. url: https://arxiv.org/abs/1807.01069.

[334] Tai Dinh et al. "Categorical data clustering: 25 years beyond K-modes". In: *Expert Systems with Applications* 272 (2025), p. 126608. issn: 0957-4174. doi: https://doi.org/10.1016/j.eswa.2025.126608.

[335] Sawsan Alodibat and Mouhammd Alkasassbeh. "An Enhanced Model of DDoS Attacks Detection Using One-Hot Encoding of Feature's Categories". In: *2025 International Conference on New Trends in Computing Sciences (ICTCS)*. 2025, pp. 133–140. doi: 10.1109/ICTCS65341.2025.10989369.

[336] *NVD - CVE-2024-34997 — nvd.nist.gov*. https://nvd.nist.gov/vuln/detail/CVE-2024-34997. [Accessed 08-06-2025].

[337] *CWE - CWE-502: Deserialization of Untrusted Data (4.17) — cwe.mitre.org*. https://cwe.mitre.org/data/definitions/502.html. [Accessed 08-06-2025].

Appendix A

Level 3 Sequential View

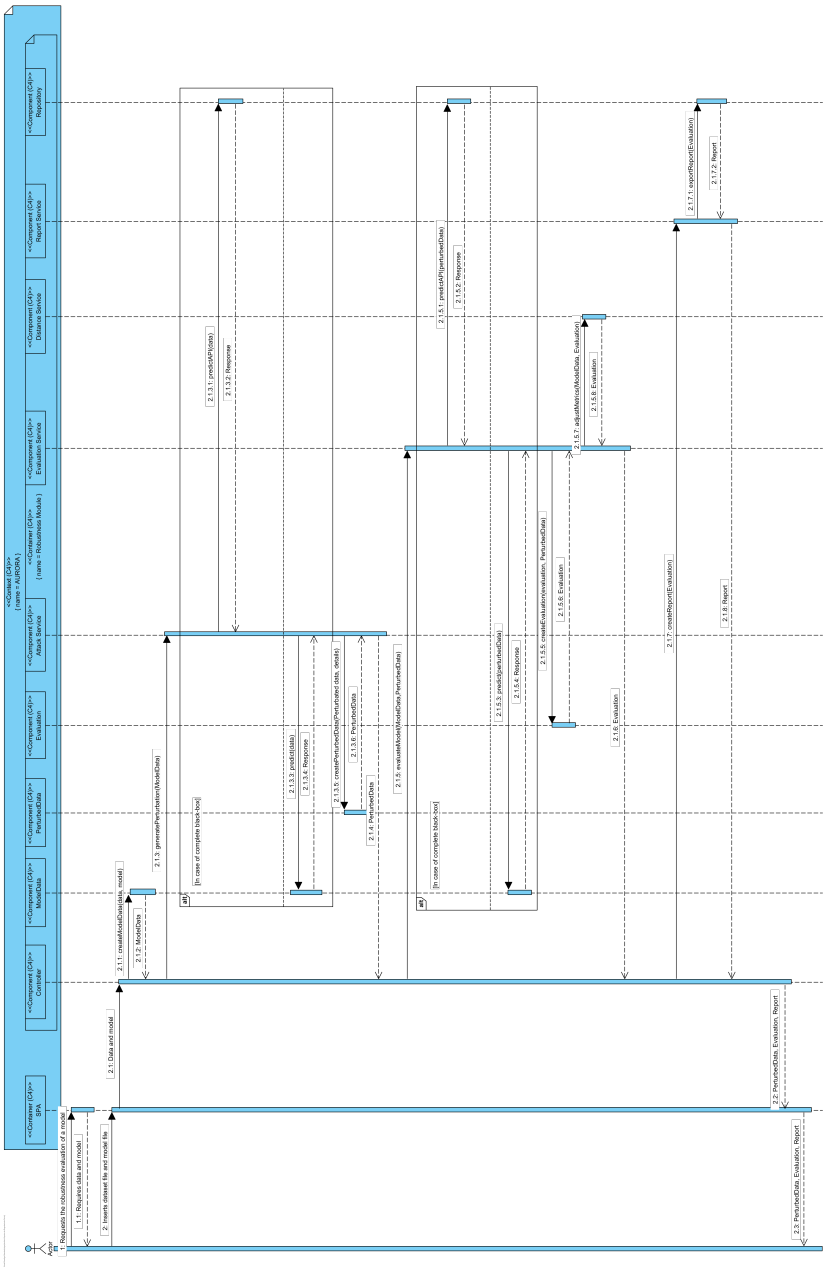

Figure A.1: Level 3 Sequential View.