Smart, Automated, and Reliable Security Service Platform for 6G

Deliverable D3.1
Threat Assessment and Prevention Report

Co-funded by the European Union

ROBUST-6G project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101139068.

Date of delivery:	30/11/2024	Version:	1.0
Project reference:	101139068	Call:	HORIZON-JU-SNS-2023
Start date of project:	01/01/2024	Duration:	30 months

Document properties:

Document Number:	D3.1
Document Title:	Threat Assessment and Prevention Report
Editor(s):	Pedro M. Sánchez Sánchez (UMU), Manuel Gil Pérez (UMU)
Authors:	Pedro M. Sánchez Sánchez (UMU), Fernando Torres Vega, Enrique Tomás Martínez Beltrán (UMU), Manuel Gil Pérez (UMU), Leyli Karaçay (EBY), Betül Güvenç Paltun (EBY), Omer Tuna (EBY), Bartlomiej Siniarski (UCD), Chamara Sandeepa (UCD), Thulitha Senevirathna (UCD), Giovanni Perin (UNIPD), Nikolaos Pappas (LIU), Marios Kountouris (EUR), Ioannis Pitsiorlas (EUR).
Contractual Date of Delivery:	30/11/2024
Dissemination level:	PU
Status:	Final
Version:	1.0
File Name:	ROBUST-6G D3.1_v1.0

Revision History

Revision	**Date**	**Issued by**	**Description**
0.1	22.07.2024	ROBUST-6G WP3	Initial draft with ToC
0.2	10.10.2024	ROBUST-6G WP3	First draft with threat assessment and prevention
0.3	08.11.2024	ROBUST-6G WP3	Second draft with 6G key technical enablers and selected cases
0.4	13.11.2024	ROBUST-6G WP3	First complete draft
0.5	22.11.2024	ROBUST-6G WP3	Second draft after internal review
0.6	29.11.2024	ROBUST-6G WP3	Final complete draft after internal review
1.0	30.11.2024	ROBUST-6G WP3	Final version

Abstract

This deliverable deals with the importance of the trustworthiness of Artificial Intelligence (AI) and Machine Learning (ML) systems by assessing the security threats in 6G networks presented in ROBUST-6G deliverable D2.1. The assessment conducted is specific to AI/ML security threats, and the associated prevention mechanisms for mitigating their risks. To this end, various privacy and security threats are explored that may compromise sensitive information, taking special attention to the 6G key technical enablers and selected cases that were analysed in more detail in D2.1.

Overall, this report presents a comprehensive approach to developing robust and reliable AI/ML systems that protect privacy and data security in 6G networks.

Keywords

Adversarial threats, Privacy threats, Explainability threats, Threat assessment, Threat prevention, Distributed Learning, AI/ML

Disclaimer

Funded by the European Union. The views and opinions expressed are however those of the author(s) only and do not necessarily reflect the views of ROBUST-6G Consortium nor those of the European Union or Horizon Europe SNS JU. Neither the European Union nor the granting authority can be held responsible for them.

Executive Summary

This deliverable presents a comprehensive examination of the security threats affecting Artificial Intelligence (AI) and Machine Learning (ML) systems in the context of 6G networks. As the deployment of advanced AI/ML technologies becomes an integral part of 6G operation, understanding and mitigating the associated vulnerabilities is paramount to ensuring the confidentiality, integrity, and availability (CIA triad) of these systems. This deliverable builds on the fundamental insights provided in the 6G Threat Analysis Report (ROBUST-6G deliverable D2.1) and gives a more detailed analysis of adversarial threats, privacy issues, and the robustness required for AI/ML models, among a number of other threats that would hinder the proper functioning of the AI/ML techniques. Each recognised threat is systematically categorised and assessed, highlighting the potential impact on AI/ML systems deployed in 6G environments.

Along with the assessment of AI security threats, the prevention and mitigation mechanisms associated with the assessed threats are also examined in detail. Both assessment and prevention processes are further reviewed on the selected key 6G technical enablers and selected cases featured in D2.1, extending this selection to an AI-as-a-Service (AIaaS) framework that enables access to AI and ML capabilities through cloud platforms, facilitating their deployment and management.

This deliverable mainly employs a structured methodology based on the STRIDE threat model together with the CIA triad. Other methodologies such as Threat Assessment and Remediation Analysis (TARA) have also been considered, especially in the mitigation processes of the identified threats where cyber risk remediation analysis is also incorporated. Consequently, analyses on the assessment and prevention of AI security threats have been carried out using the STRIDE methodology, for systematic identification and mitigation of these threat impacting AI/ML systems.

In conclusion, this deliverable serves as a critical resource for advancing the understanding of security threats to AI/ML models in 6G networks, taking into consideration various analyses from a threat assessment and mitigation approach. By identifying vulnerabilities and proposing targeted solutions, the ROBUST-6G project aims to facilitate the development of secure, resilient, and future-proof communication systems in the context of the AI/ML techniques to be used. The insights and recommendations provided in this report are essential for fostering innovation and ensuring that AI/ML technologies meet the stringent security requirements of next-generation digital ecosystems.

Table of Contents

List of Tables

List of Figures

Acronyms and abbreviations

Term	Description
ADMM	Alternating Direction Method of Multipliers
AI	Artificial Intelligence
AIaaS	AI-as-a-Service
AUROC	Receiver Operating Characteristic Curve
BIM	Basic Iterative Method
BS	Base Station
CIA	Confidentiality, Integrity and Availability
CRRA	Cyber Risk Remediation Assessment
CTSA	Cyber Threat Susceptibility Analysis
CW	Carlini & Wagner
DFL	Decentralized / Distributed Federated Learning
DL	Deep Learning
D-MIMO	Distributed-Multiple Input Multiple Output
DML	Distributed Machine Learning
DNN	Deep Neural Network
DoS	Denial of Service

DP	Differential Privacy
DP-SGD	Differentially Private Stochastic Gradient Descent
ETSI	European Telecommunications Standards Institute
FGSM	Fast Gradient Sign Method
GAN	Generative Adversarial Network
GDPR	General Data Protection Regulation
GMM	Gaussian Mixture Models
GPT	Generative Pre-trained Transformer
HEAVENS	Healing Vulnerabilities to Enhance Software
IDS	Intrusion Detection System
IoT	Internet of Things
KD	Knowledge Distillation
KM	Knowledge Management
LDP	Local Differential Privacy
LIME	Local Interpretable Model-Agnostic Explanations
LLM	Large Language Model
ME	Model Extraction
MIA	Membership Inference Attack
MIMO	Multiple Input Multiple Output
ML	Machine Learning
NLP	Natural Language Processing
NWDAF	Network Data Analytics Function
O&M	Orchestration and Management
P2P	Peer-to-Peer
PGD	Projected Gradient Descent
PLS	Physical Layer Security
PSNR	Peak Signal-to-Noise Ratio
RF	Radio Frequency
RIS	Reconfigurable Intelligent Surfaces
RISC	RIS Controller
RU	Resource Unit
SGD	Stochastic Gradient Descent
SHAP	SHapley Additive exPlanations

SKG	Secure Key Generation
SSE	Systems Security Engineering
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
SVM	Support Vector Machines
TARA	Threat Assessment and Remediation Analysis
TVRA	Threat, Vulnerability and Risk Assessment
UAP	Universal Adversarial Perturbation
UE	User Equipment
XAI	eXplainable AI

1 Introduction

The goal of Artificial Intelligence (AI) development is to help humans by solving complex tasks and advancing societal good. However, recent studies indicate that AI may inadvertently cause harm, such as making inaccurate decisions in safety-critical contexts or undermining fairness through unintended discrimination against certain groups. As a result, trustworthy AI, which refers to AI systems that are reliable, transparent, fair, secure, and respect privacy, has lately received wide attention from the research community due to the need to minimize the negative effects that AI may have on people.
The topic of trustworthy AI is broad and intricate, and studies related to Trustworthy AI can be categorized into 6 categories as depicted in Figure 1-1: Classification of studies related to Trustworthy AIFigure 1-1.

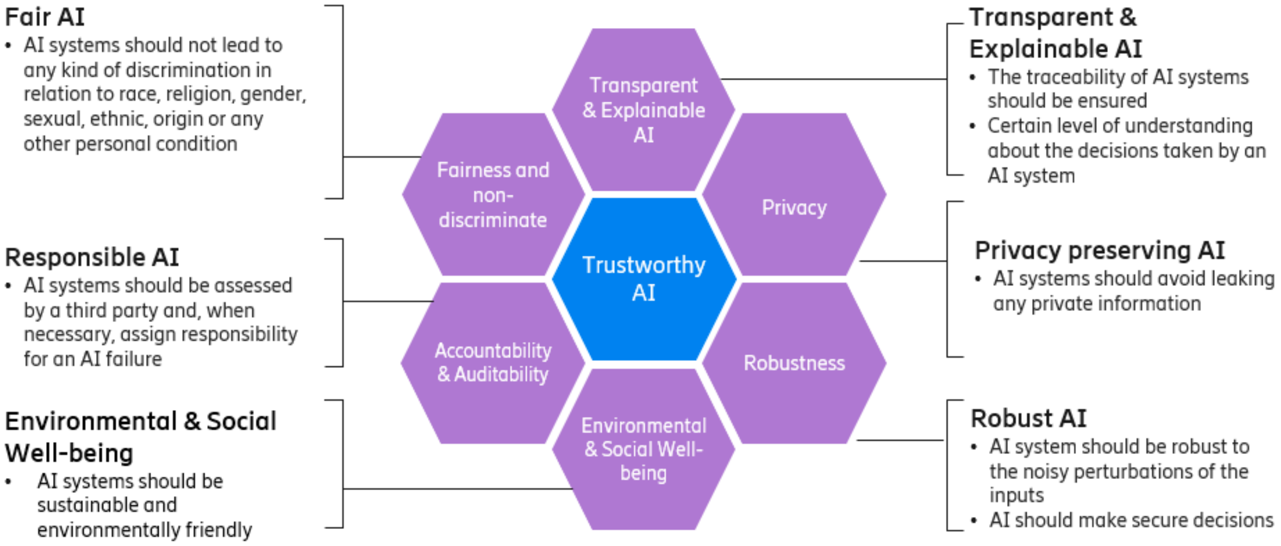

Fair AI
• AI systems should not lead to any kind of discrimination in relation to race, religion, gender, sexual, ethnic, origin or any other personal condition
Responsible AI
• AI systems should be assessed by a third party and, when necessary, assign responsibility for an AI failure
Environmental & Social Well-being
• AI systems should be sustainable and environmentally friendly
Transparent & Explainable AI
Fairness and non-discriminate
Trustworthy AI
Privacy
Accountability & Auditability
Robustness
Environmental & Social Well-being
Transparent & Explainable AI
• The traceability of AI systems should be ensured
• Certain level of understanding about the decisions taken by an AI system
Privacy preserving AI
• AI systems should avoid leaking any private information
Robust AI
• AI system should be robust to the noisy perturbations of the inputs
• AI should make secure decisions

Figure 1-1: Classification of studies related to Trustworthy AI

To begin with, Trustworthy AI is anticipated to possess the qualities of robustness and explainability. In robustness terms, we should guarantee that the decisions of the AI model should not change in the case of tiny modifications to input data. From the explainability perspective, trustworthy AI must enable human-understandable explanations to reduce risks and potential harm. The decisions of the AI models cannot be taken for granted unless the underlying mechanisms behind their predictions are explained. As a result, developing a trustworthy AI system necessitates an insight into how specific decisions are made.

In addition to robustness and explainability, trustworthy AI must also ensure privacy, availability, accountability, and safety. In terms of privacy, AI is expected to safeguard the privacy of all users and prevent any leakage of private information. Besides privacy, it is expected that people should have access to AI systems whenever they need them, and these AI-driven systems should be simple to use for diverse users. Furthermore, no AI system is ever intended to harm anyone, under any circumstances, and it is always expected that user safety is a top priority.

Finally, trustworthy AI should be fair, ethical, and environmentally sustainable. In particular, AI-driven systems should ensure fairness for all users and should not discriminate towards certain groups. It also should function in complete accordance with all applicable rules and regulations as well as the ethical standards of human civilization. From sustainability perspective, AI-driven systems should be environmentally friendly to ensure sustainable development. They should, for instance, consume less energy and produce less pollution.

1.1 Motivation, objectives, and scope

We should expect future communication networks to comply with all the requirements expressed above, as AI is expected to be one of the key enablers of 6G networks. Deep Learning (DL) models are already emerging as a helpful tool for assisting with various communication tasks in different network layers. For example, in wireless networks, AI/Machine Learning (AI/ML) has been utilized to solve a variety of challenges such as encoding/decoding operations, power-allocation/RU selection/beamforming for Multiple Input Multiple Output (MIMO) systems, spectrum sensing, Radio Frequency (RF) signal classification, signal authentication, and anti-jamming. In core network, AI/ML is used in the Network Data Analytics Function (NWDAF) to enhance network data analytics. In the Orchestration and Management (O&M) layer, AI/ML is being used for resource management, fault management, etc. Moreover, we anticipate AI/ML to be a powerful catalyst for security-related functions in 6G networks.

To ensure the trustworthiness of future 6G networks, ROBUST-6G should identify all possible AI-driven communication tasks and use cases. We then need to assess the potential weaknesses of these AI systems by taking into consideration the main pillars of Trustworthy AI, which is one of the most important objectives and scope of this deliverable. And we should focus our efforts on ensuring the required degree of quality for each crucial dimension of Trustworthy AI, taking all requirements into account.

Trying to cope with security threats in telecommunication systems is becoming increasingly difficult as the complexity and volume of these threats increase. The expectations from 6G use case requirements and adoption of new networking concepts, such as cloud computing, fog/edge computing and Internet of Things (IoT) will prevent conventional algorithms previously used to detect attacks in the cybersecurity domain from processing the massive data volumes. Thus, it is expected that DL-based AI architectures will play a significant role for the security related functions within the 6G network. However, the applications of AI in the field of cybersecurity are also encountering difficulties because of the shortcomings of the AI-based methodologies.

This document also serves as a basis for the other ROBUST-6G technical tasks related to the use of AI/ML techniques, which require a high level of *robustness*, *explainability*, *fairness* and *accountability* for their AI-based security solutions.

1.2 Document structure

The present document is structured as follows. In Section 2, several models and methodologies for threat assessment and prevention are discussed, including the TARA and STRIDE models as the most promising ones to follow. Section 3 delves into specific threats related to AI and ML systems, such as adversarial attacks and privacy concerns, highlighting the importance of having robust security measures in place and including the need for explainable AI to ensure transparency in decision-making processes. In Section 4, we emphasise various mitigation processes to prevent attacks on AI and ML lifecycles, with the aim of enhancing the security of AI and ML systems against identified threats. In Section 5, we present examples of potential threats to various 6G technologies, particularly in the physical layer and AI/ML modules. We also discuss in Section 5 the assessment and prevention of the previously identified threats, emphasising their impact on key enablers. Finally, we summarise our conclusions in Section 6.

2 Methodology for threat assessment and prevention

In the current state-of-the-art, there are different methodologies that can be used for threat identification in any kind of scenario. Methodologies that were described and used in ROBUST-6G deliverable D2.1 such as the well-known CIA model (*Confidentiality*, *Integrity* and *Availability*), Microsoft STRIDE [Pot09], ETSI TVRA [ETSI TS 102 165-1], or HEAVENS [HEA16-D2], among others. However, there is currently a gap in the need for methodologies for threat assessment and prevention beyond threat identification and detection.

Of the above enumeration, only the Microsoft STRIDE methodology also addresses the assessment and prevention phases in addition to identification. Therefore, the STRIDE methodology has been used to carry out the presented comprehensive analysis on the assessment and prevention of threats, which affect the security of AI/ML systems.

The principal methodologies that could be used for threat assessment and prevention are listed and described below.

2.1 Threat Assessment and Remediation Analysis

The Threat Analysis and Remediation (TARA) methodology is part of a MITRE portfolio of Systems Security Engineering (SSE) practices [Wyn14]. It presents a systematic approach to assessing and mitigating cyber vulnerabilities of systems during their acquisition cycle, which is composed of several key stages, including the Cyber Threat Susceptibility Analysis (CTSA), the Cyber Risk Remediation Assessment (CRRA) and Knowledge Management (KM). They are illustrated in Figure 2.1.

The CTSA develops a cyber model of the system to identify plausible attack vectors, while the CRRA selects countermeasures from the Catalog to mitigate identified vulnerabilities. The TARA Catalog, as part of the KM, stores information on attack vectors, countermeasures and mitigation maps, enabling a many-to-many relationship between vectors and countermeasures. This facilitates the evaluation of the effectiveness of countermeasures in different attack scenarios. This TARA Catalog should be kept up to date with relevant data to prevent new types of threats and attack vectors from going unnoticed.

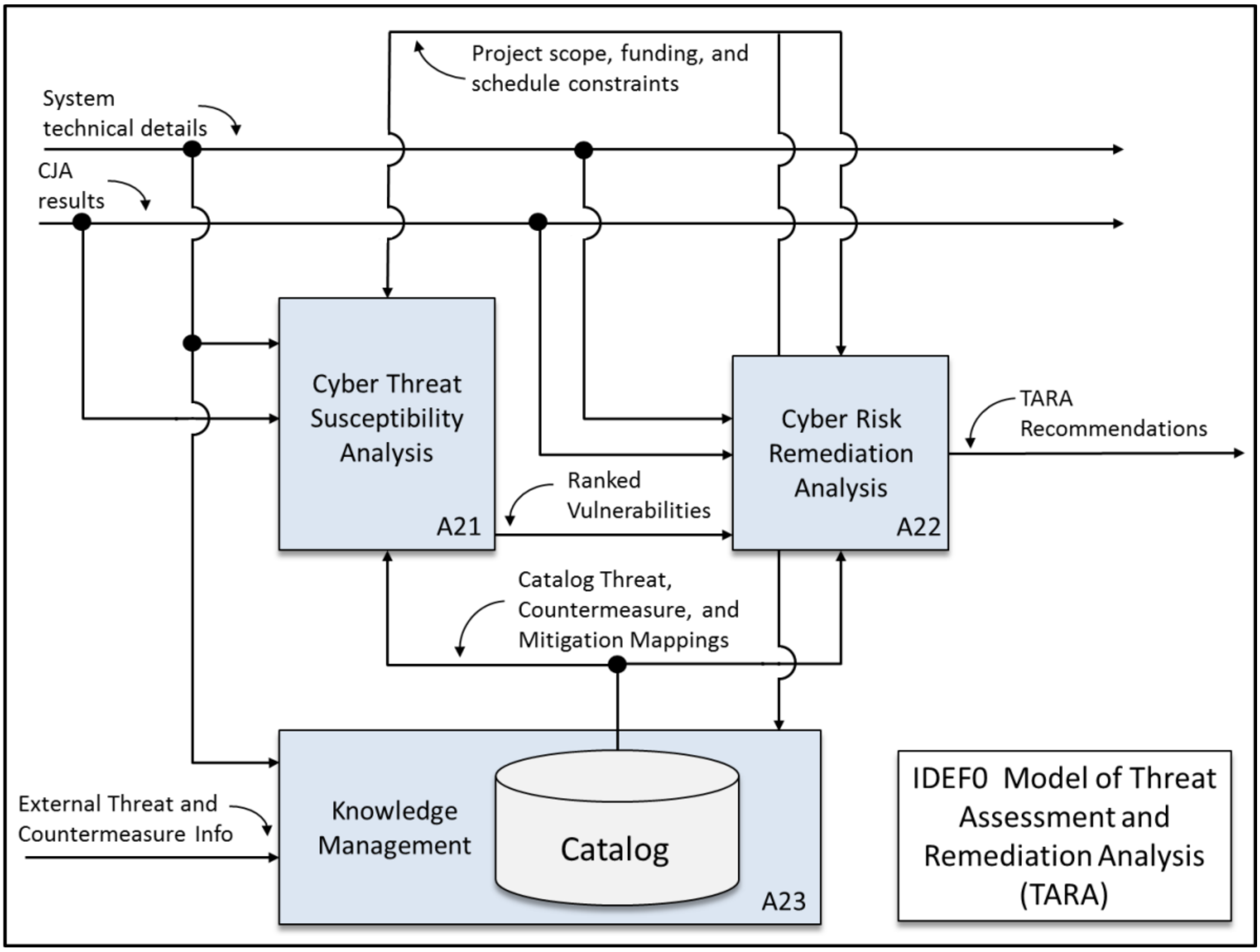

Project scope, funding, and schedule constraints
System technical details
CJA results
Cyber Threat Susceptibility Analysis
A21
Ranked Vulnerabilities
Cyber Risk Remediation Analysis
A22
TARA Recommendations
Catalog Threat, Countermeasure, and Mitigation Mappings
External Threat and Countermeasure Info
Knowledge Management
Catalog
A23
IDEF0 Model of Threat Assessment and Remediation Analysis (TARA)

Figure 2-1: The TARA process (source: [Wyn14])

Although there is currently no related work in the literature that directly applies the TARA methodology to AI/ML systems, this methodology is flexible and can be adapted to assess and mitigate threats in a variety of contexts, including AI/ML as one of the key technologies for the 6G networks. In this context, Section 3 of this report tackles the identification of potential threats that could affect the output of AI/ML models by analysing specific attacks such as manipulation of training data, adversarial attacks or exploitation of vulnerabilities in the AI/ML model. These capabilities correspond to the first CTSA process of the TARA methodology (see Figure 2.1). And in Section 4, similar capabilities to those of the CRRA process are also approached, analysing different mechanisms and techniques such as label smoothing or gradient regularisation to smooth decisions and reduce robustness to adversarial attacks. The prevention techniques of Section 4 will improve the security of AI/ML models and make them more robust against threats.

2.2 Diamond Model of Intrusion Analysis

The methodology proposed by the Diamond Model of Intrusion Analysis [CPB13] sets out a formal framework for documenting and analysing malicious activity. This model is based on four fundamental characteristics: adversary, infrastructure, capability and victim, which are tightly interrelated to provide a comprehensive understanding of intrusion events. The methodology emphasizes the importance of hypothesis generation and documentation, differentiating between facts and assumptions, which improves the accuracy of the analysis.

In addition to the above, a scientific approach is applied. This includes principles of measurement, testability and repeatability, allowing the formulation of analytical hypotheses and automated correlation of events. The structure of the model facilitates the identification of knowledge gaps and the planning of mitigation strategies, promoting effective communication and a coordinated response to cyber threats.

The Diamond model provides a systematic and adaptable approach to intrusion analysis and assessment, which could be used in Section 3 of this report to identify potential threats that could affect AI/ML performance. But it underestimates prevention mechanisms for mitigating the identified threats.

2.3 The STRIDE threat model

The STRIDE methodology, developed by Microsoft, is a powerful framework for threat modelling and security assessment. It identifies six primary threat categories: *Spoofing*, *Tampering*, *Repudiation*, *Information Disclosure*, *Denial of Service*, and *Elevation of Privilege*. By systematically addressing each threat type, STRIDE helps anticipate potential security issues, ensuring a more comprehensive risk and threat management approach. Its structured process simplifies complex cybersecurity challenges, making it easier to understand vulnerabilities, potential threats or targets that may present security flaws early in the development cycle. It also supports a proactive defence strategy, reducing the likelihood of successful attacks and improving overall system resilience.

Table 2-1 displays the desired property of each of the threats categorised by the STRIDE methodology, with a brief definition of each of them. The desired property column also highlights those determined by the CIA model.

Threat	Desired property	Definition
Spoofing	Authenticity	Identity impersonation to gain unauthorized access to systems
Tampering	Integrity *(CIA)*	Unauthorized modification of data or components within a system
Repudiation	Non-repudiability	Denial of actions taken by a user without traceability
Information Disclosure	Confidentiality *(CIA)*	Exposure or leakage of confidential information to unauthorized parties
Denial of Service	Availability *(CIA)*	Disruption of legitimate access to services through saturation or attacks
Elevation of Privilege	Authorization	Gaining higher permissions to access restricted resources

Table 2-1: Desirable property and definition of each threat supported by STRIDE

Using STRIDE, each architectural component in the ROBUST-6G platform could be systematically analysed to detect specific threats or specific vulnerabilities, assess their potential impact on the platform and outline countermeasures to mitigate these security risks.

3 Threat assessment to AI/ML models

This section examines the security threats that could have a major impact on the performance of AI/ML systems and disrupt models in the collaborative and distributed settings characteristic of 6G networks. This analysis details the need for robustness of AI/ML models, as well as protection against privacy-related attacks.

3.1 Adversarial threats

The following are the main threats that make AI/ML models vulnerable, mainly affecting security and privacy by manipulating training data and predictions. Among them, we focus on poisoning and evasion attacks, which manipulate training data to degrade model performance or create backdoors impacting privacy and model accuracy in AI/ML, and alter inputs to cause misclassification, respectively. In addition, this section also explores the transferability and data supply chain attacks, which aim to compromise security and performance by manipulating data and creating transferable attacks between models.

3.1.1 Poisoning attacks

ML systems are vulnerable to a range of adversarial attacks that can compromise their security, privacy and utility. Among these threats, poisoning attacks are considered highly likely, especially in distributed ML scenarios, due to their ability to subtly and maliciously manipulate the training process and propagate the attack towards many benign models in a collaborative ML environment. Poisoning attacks occur when an adversary intentionally injects misleading or corrupted data into the training dataset or alters the learning process itself, with the aim of changing the model's behaviour towards specific, often harmful, objectives. These attacks can be deployed in various contexts, targeting different aspects of the ML pipeline, and their impacts can be severe, ranging from degraded model performance to privacy breaches and the insertion of backdoors.

Poisoning attacks can be launched with the intention of degrading the model performance, either in a targeted or an untargeted manner [SSW+24a]. If the attacker wants to degrade the overall performance like model accuracy for all classes without any particular interest in a target class or objective, they may use untargeted attacks. These attacks are often used by randomly perturbing the data labels via techniques like label flipping, or by injecting random noise into the model parameters. However, considering the targeted attacks, they consider a target property or a class label that gets poisoned, where the adversary aims to modify the model's decision boundary primarily in the targeted label or property. This makes the models to incorrectly classify the target class or data with the targeted property. Considering the implementation of the attack, there are two types of poisoning [CSS+22]: 1) dirty label poisoning and 2) clean label poisoning. In dirty label poisoning, the attacker poisons the model by manipulating the labels in the dataset used to train the model. In clean label poisoning, the attacker does not change the labels, instead, they inject malicious triggers or perturbations in the data that are difficult to detect but lead to manipulations in the decision boundary and thus heavily impact on the model's decision-making process.

The attackers may also use poisoning attacks to artificially alter the decision boundary of a model towards a specific targeted direction [WHS+22]. This can enhance the success rate of other attacks like inference attacks [ZZC+20, WHS+22]. Additionally, backdoor and trigger attacks stemming from poisoning can degrade model performance for specific classes. Such triggers can also lead to privacy breaches, where the trigger activates when a specific property or data in a private dataset is encountered [WHS+22, NLT+23]. Figure 3-1 shows the shift in decision boundary that is triggered via the poisoning attacks.

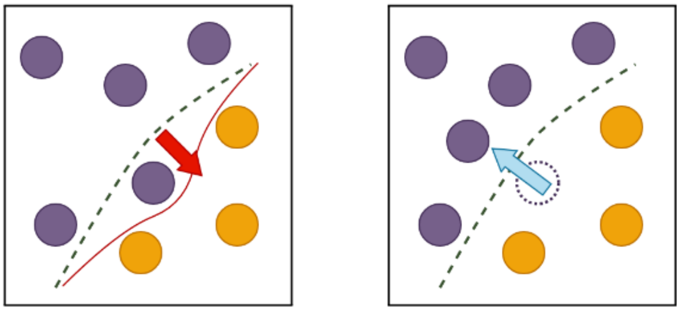


Figure 3-1: Displacement at the decision boundary caused by poisoning attacks

As depicted in Figure 3-1, when an adversary poisons the ML model, the decision boundary tends to deviate from its original position. In a collaborative ML system, a benign client may correct this boundary back to its original form, as the benign client consist of correct data and during local training, the model will optimize towards the real boundary. If this decision boundary correction is related with a sensitive attribute in the target data, the adversary can identify the benign client consists of the sensitive attribute, which makes the correction. Such poisoning triggers leads to attacks like inference, described further in Section 3.1.2.

3.1.2 Evasion attacks

Within the last decade, researchers found out that Deep Neural Networks (DNNs) models are vulnerable to well-crafted malicious perturbations. Szegedy *et al.* [SZS+14] were the first to recognize the prevalence of adversarial cases in the context of image classification. Researchers have shown that a slight alteration in an image can influence the prediction of a DNN model. It is demonstrated that even the most advanced classifiers can be fooled by a very small and practically undetectable change in input, resulting in inaccurate classification as in Figure 3-2. Since then, a lot of research studies were performed in this new field and these studies were not limited just to image classification task. There are several studies available in literature in Natural Language Processing (NLP) and audio domain as well.

Attacks that take advantage of DNN's weakness can substantially compromise the security of the ML-based systems, often with disastrous results. Adversarial evasion attacks mainly work by altering the input samples to increase the likelihood of making wrong predictions. These attacks can cause the model's prediction performance to deteriorate since the model cannot correctly predict the actual output for the input instances. In the context of medical applications, a malicious attack could result in an inaccurate disease diagnosis. As a result, it has the potential to impact the patient's health, as well as the healthcare industry [FBI+19]. Similarly, self-driving cars employ ML to navigate traffic without the need for human involvement. A wrong decision for the autonomous vehicle based on an adversarial attack could result in a tragic accident [GHM23]. Hence, defending against malicious evasion attacks and boosting the robustness of ML models without sacrificing clean accuracy is critical. Presuming that these ML models are to be utilized in crucial areas, we should pay utmost attention to ML models' performance and the security problems of these architectures.

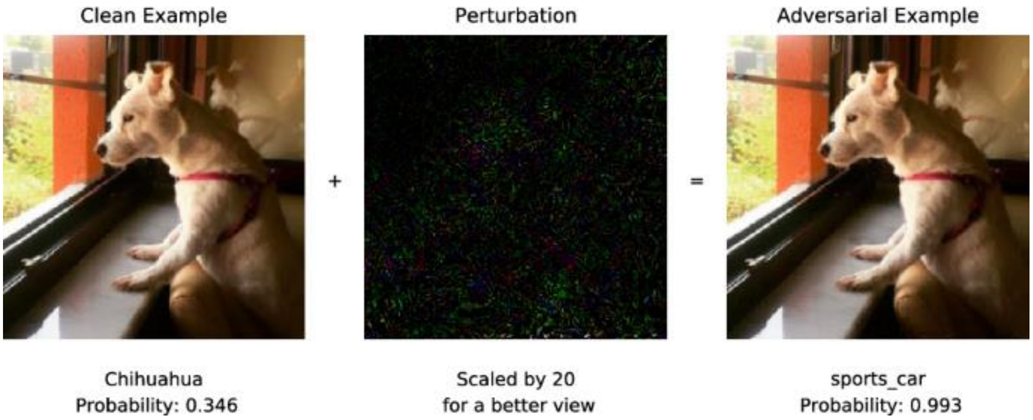
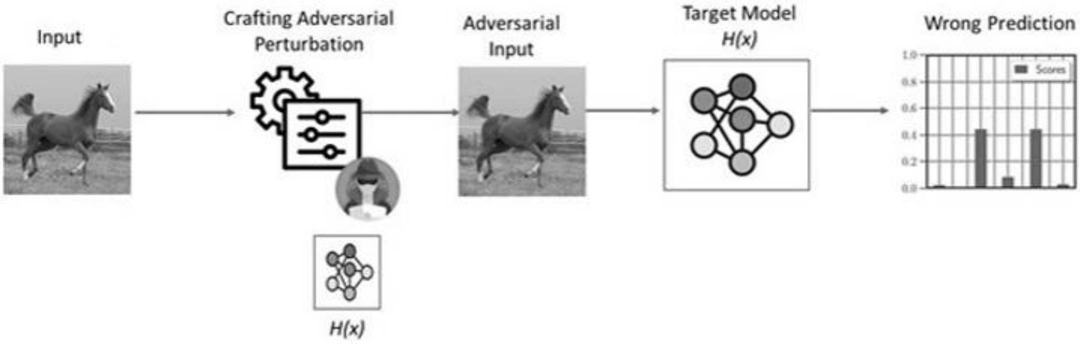


Figure 3-2: Example of adversarial sample

In principle, adversarial strategies in evasion attacks can be classified based on multiple criteria. Based on the attacker's goal, attacks can be classified as untargeted and targeted attacks. In the former, the attacker perturbs the input image, causing the model to predict a class other than the actual class. Whereas in the latter, the attacker perturbs the input image so that a particular target class is predicted by the model. Researchers commonly use 3 types of distance metrics, L_2, L_∞, and L_0. The L_2 distance is the standard Euclidean distance. $L\infty$ distance is the maximum change to any of the pixels. And L_0 distance is the number of distinct pixels. The last criterion for grouping the adversarial attacks is the threat model. Attacks can also be grouped based on the level of knowledge that the attacker has. If the attacker has complete knowledge of the model like architecture, weights, hyper-parameters, etc., we call this kind of setting White-box Settings, which is depicted in Figure 3-3. In this example, the aim of the attacker is to use the target model (H) and find a perturbation which maximize the loss with respect to the actual class.



$$\max_{\delta \in \Delta} Loss(x + \delta, y; \theta_H)$$

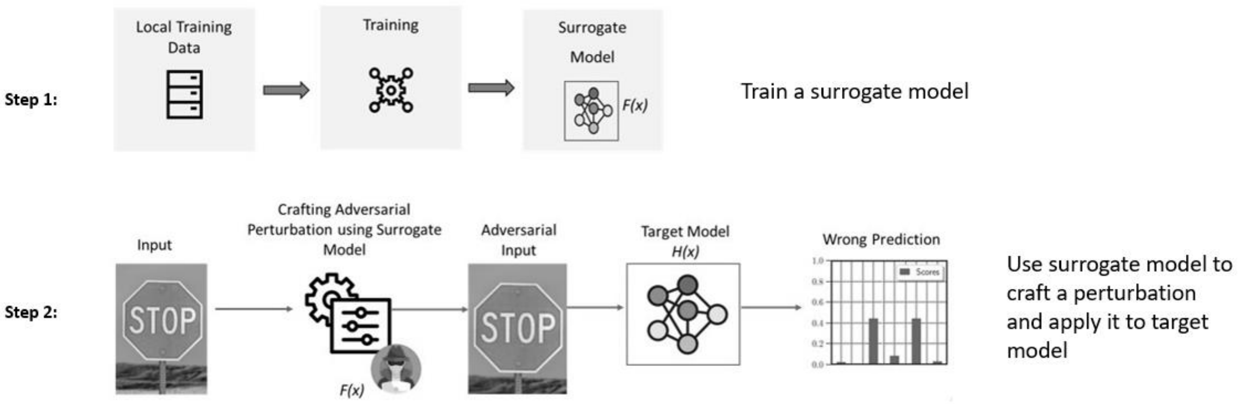
Figure 3-3: Illustration of white-box adversarial attack

However, if the attacker has no information about the deployed model and defence strategy, we call this kind of setting Black-box Settings. This kind of setting is displayed in Figure 3-4. This type of attack scenario is based on the concept of attack transferability in adversarial ML. It has been observed that an adversarial sample crafted using a surrogate model can fool the target AI model.

In the previous context, transferability refers to the capability of a malicious attack to be successful against another, presumably unknowable model. Since the malicious actor does not have a copy of the target model and instead uses his/her own surrogate model, this scenario is also considered as Black-box attack scenario.

Crafting Adversarial Samples

Most attack ideas rely on perturbing the input sample to maximize the model's loss. In recent years, many different adversarial attack techniques have been suggested in the literature. The most known attack algorithms for White-Box Attacks are FGSM, BIM, PGD, Deepfool, and CW, and the most known attack algorithms for inference query based Black-Box Attacks are Boundary, HopSkipJumpAttack, and Square Attack.



$$\max_{\delta \in \Delta} Loss(x + \delta, y;\ \theta_F)$$

Figure 3-4: Illustration of attack transferability based black-box adversarial attack

White-Box Attack Algorithms

Several of the main white-box attack algorithms are discussed below. These attacks attempt to compromise the robustness of AI/ML models by generating adversarial samples that mislead systems, exposing critical vulnerabilities in tasks such as classification or recognition. This underscores the need to develop more resilient models through techniques such as adversarial training.

Fast Gradient Sign Method (FGSM): FGSM uses the idea to employ a gradient based approach and the derivative of the model's loss function with respect to the input sample to identify which direction the input sample's feature values should be altered to minimize the model's loss function. Once this direction is extracted, it alters all features in the opposite direction simultaneously to maximize the loss. We may craft adversarial samples for a model H with a classification loss function represented as J(θ,x,y) by utilizing the formula below, where θ denotes the parameters of the model, x is the benign input, and y_{true} is the real label of our input.

$$x^{adv} = x + \ \varepsilon . sign(\nabla_x J(\theta, x, y_{true}))$$

Basic Iterative Method (BIM) and Projected Gradient Descent (PGD): Kurakin *et al.* proposed a minor but significant enhancement to the FGSM. Instead of taking one large step ε in the direction of the gradient sign, we take numerous smaller steps α and utilize the supplied value ε to clip the output in this method. This method is also known as the Basic Iterative Method (BIM), and it is simply FGSM applied to an input sample iteratively. The equation below describes how to generate perturbed images under the l_{inf} norm for a BIM attack.

$$x_i^* = x$$

$$x_{i+1}^* = \ clip_{x,\varepsilon}\{x_i + \alpha . sign(\nabla_x J(\theta, x_i^*, y_{true}))\}$$

where x is the clean sample input to the model, x_i* is the output adversarial sample at ith iteration, J is the loss function of the model, θ denotes model parameters, y_{true} is the true label for the input, ε is a configurable parameter that limits maximum perturbation amount in given l_{inf} norm, and α is the step size.

As in the case of a PGD attack, it perturbs an input sample x for a number of "i" iterations in the direction of the model's loss function gradient with a tiny step size. Then, it projects the generated adversarial sample back onto the ε -ball of the input after each perturbation step depending on the chosen distance norm. In addition, rather than starting from the original point (ε = 0, in all the dimensions), PGD employs a random start, which can be defined as:

$$x_0 = x + U(-\varepsilon, +\varepsilon)$$

where $U(-\varepsilon, +\varepsilon)$ is the uniform distribution between $(-\varepsilon, +\varepsilon)$.

Deepfool Attack

The Deepfool attack is formulated on the idea that neural network models act like linear classifiers with classes separated by a hyperplane. Starting with the initial input point x_t, the algorithm determines the closest hyperplane and the smallest perturbation amount, which is the orthogonal projection to the hyperplane, at each iteration. The algorithm then computes x_{t+1} by adding the smallest perturbation to the x_t and checks for misclassification. This attack can break defensive distillation method and achieve higher success rates than previously mentioned iterative attack approaches. But the downside is, the produced adversarial sample generally lies close to the decision boundary of the model.

Carlini & Wagner (CW) Attack

Proposed by Carlini and Wagner, it is one of the strongest attack algorithms so far. As a result, it is commonly used as a benchmark for adversarial defence research groups, which try to develop more robust DNN architectures that can withstand adversarial attacks. It is shown that, for the most well-known datasets, the CW attack has a greater success rate than the other attack types on normally trained models. It can also fool defensively distilled models, which other attack algorithms find difficult to generate adversarial samples for, just as Deepfool attack method.

To generate more effective and strong adversarial samples under multiple l_p norms (distance metrics used to limit the perturbation budget), the authors reformulate the attack as an optimization problem which may be solved using gradient descent. A confidence parameter in the algorithm can used to change the level of prediction score for the created adversarial sample. For a normally trained model, the application of a CW attack with a default setting (confidence set to 0) would generally yield to adversarial samples close to the decision boundary. And high confident adversaries are generally located further away from the decision boundary.

Inference query based Black-Box Attacks

In the Black-Box attack scenario, as opposed to the White-Box setup, the adversary only has access to the outputs of the target model (either only the decisions or all the probability scores). For the cases where the adversary has access to all the probability scores of a given input, Ilyas *et al.* [IEM19] proposed score-based methods using zeroth-order gradient estimation for crafting adversarial perturbations. An attacker has just access to decisions in a more practical and realistic scenario that applies to the majority of AI-as-a-Service implementations. Brendel *et al.* [BRB18] introduced Boundary Attack, which generates adversarial examples via rejection sampling and achieves comparable performance with state-of-the-art White-Box attacks. Nevertheless, their approach requires a relatively large number of model queries, rendering it impractical for real-world applications. Later, Chen *et al.* [CJW20] introduced HopSkipJumpAttack which is a decision-based attack method that relies on an estimation of the model's gradient direction and binary-search procedure for approaching the decision boundary. The powerful part of this attack is that it requires significantly fewer model queries than previously proposed decision-based Black-Box attack algorithms yet achieves competitive performance. Finally, Andriushchenko *et. al.* [ACF+20] proposed Square Attack which is again a query efficient Black-Box attack that is not based on the model's gradient and can break defences that utilize gradient masking.

When incorporating AI models into the field of security, we should pay closer attention to an important drawback of AI-driven systems which are adversarial attacks. AI/ML models have recently been discovered to be vulnerable to malicious attacks. In fact, very small and often undetectable changes in data samples are enough to fool state-of-the-art classifiers in inference time and lead to incorrect predictions. In the past few years, we have witnessed extensive research which show the vulnerability of AI-driven systems in different domains like image, text and audio.

The adversaries who have access to data but no direct access to the model parameters may also use adversarial attacks like data poisoning (described in Section 3.1.1) to improve the success rate of black-box inference. In the case where collaborative learning like Federated Learning is done, the attackers can inject backdoored model updates even without access to the target model's data and they can improve the black-box inference attack success rate [WHS+22]. They may also use a set of shadow models, enriched with artificial data generated via models like Generative Adversarial Networks (GANs), to create a similar model to the target model, such that they can estimate the model behaviour of the target model more accurately, enhancing the success rate of black-box inference. These types of inference attacks include GAN-induced membership inference [SSS+17, ZZC+20], which attempts to infer whether a particular data point was part of the training dataset, property inference [WHS+22], that aims to detect specific properties or features within the dataset that are unrelated to the main task. These attacks are often passive, relying on the evaluation of received model updates to carry out the attack. The effectiveness of inference attacks can depend on the attacker's ability to accurately identify decision boundary changes in the models with each update [WHS+22, TLG+19].

And despite the distributed nature of the communication domain and the heterogeneity of the network, we still have the risk of adversarial attacks in a telco environment. However, from the adversary's perspective, there are several important constraints which limits the success of the adversarial attack in a telecom network. Firstly, the adversary mostly does not have access to the details (architecture and weights) of the original AI model, therefore cannot use it in a White-Box setting for crafting adversarial samples. Secondly, the adversary may not have complete knowledge of the input features of the AI model. Lastly, in a practical scenario, the adversary does not have the capability to introduce perturbations to all features of the input sample. However, despite all these limitations, there are proven ways in literature which increase the success of the attacker. Regarding the first limitation, it has been shown that a surrogate AI model might be sufficient to launch an effective attack due to the transferability nature of the adversarial samples. Regarding the second limitation, the universal adversarial perturbation (UAP) method is proposed for cases where complete input knowledge is not available. For instance, recent research studies such as [TKK+23] indicate that the performance of an AI-driven D-MIMO system can be degraded by malicious UE's or RU's which provide adversarial perturbations to the pilot signals. The results indicate that adversarial attacks with optimized perturbations can degrade the performance of the network in terms of both spectral and energy efficiency. Thus, smart defence techniques such as [TK24] are required to overcome the effects of such attack threats.

3.1.3 Transferability and data supply chain attacks

Transferability attacks exploit the phenomenon where adversarial examples which are crafted to deceive one machine learning model can also succeed in misleading other models, even if they have different architectures or are trained on different datasets. This property poses significant security risks, as it allows attackers to generate adversarial inputs without direct access to the target model.

The concept of adversarial transferability was first explored in [SZS14], where the authors demonstrated that adversarial examples generated for one neural network could deceive another network with a different architecture. This discovery highlighted the systemic vulnerability of machine learning models to adversarial attacks and raised concerns about the robustness of deployed AI systems.

Countermeasures against transferability attacks include employing adversarial training [MMS+18], where models are trained on adversarial examples to enhance their robustness. Another approach is to introduce randomness into the model's predictions or preprocessing steps, making it harder for attackers to generate universally transferable adversarial examples [XWZ+17]. Defensive distillation [PMW+16] has also been proposed to reduce model sensitivity to adversarial perturbations, although its effectiveness against transferability attacks is limited.

Data supply chain attacks target the integrity of the data used in the training and operation of machine learning models. By compromising the data collection, preprocessing, or storage processes, attackers can introduce malicious inputs that degrade model performance or induce specific, undesired behaviours. These attacks pose significant risks, especially in systems that rely on continuous data updates or automated data pipelines.

In literature work, [GLG17] explored backdoor attacks, a type of data poisoning where the model behaves normally on standard inputs but produces attacker-specified outputs when presented with inputs containing a specific trigger pattern. This subtle manipulation allows attackers to implant hidden functionalities into the model without affecting its overall performance, making detection challenging.

Defences against data supply chain attacks include robust data validation and sanitation techniques to detect and remove anomalous data points [SKL17]. Implementing secure data collection and storage practices reduces the risk of data tampering. Additionally, techniques like differential privacy (DP, see Section 4.2) [ACG+16] can help protect against data poisoning by adding noise to the training process, although this may impact model accuracy.

3.2 Privacy threats

As Artificial Intelligence becomes increasingly integrated into various aspects of society, concerns about privacy threats have correspondingly intensified. AI systems often rely on large datasets that may contain sensitive or personal information, making them potential targets for malicious activities. Adversaries can exploit vulnerabilities in these models to extract confidential data, infer private attributes, or reconstruct original inputs. Such threats not only compromise individual privacy but also undermine trust in AI technologies.

Additionally, efforts to make AI models more transparent and interpretable can inadvertently introduce new privacy risks. Techniques designed to explain model decisions might reveal sensitive information or be manipulated to conceal malicious behaviour. Understanding and addressing these privacy challenges are crucial for developing robust AI systems that protect user data while maintaining transparency and accountability.

3.2.1 Model inversion attacks

Model inversion attacks harm users' privacy as they aim to infer information about the input data by observing the outputs and the model parameters. The goal is to reconstruct the dataset from knowledge of the predicted labels [MLW+23]. These attacks are easier to perform in a white-box context, i.e., when the attacker can access the trained model.

On the other hand, [FLJ+14] is the seminal paper on model inversion attacks. It studies privacy in pharmacogenetics, where, given the model and some demographic information on the patient, an attacker can predict the patient's genetic markers. DP is suggested as a countermeasure. However, the authors show that, for an effective DP mechanism preserving privacy, the model accuracy is decreased at the point that patients are exposed to higher mortality risks. In [FJR15], the authors develop a class of model inversion attacks exploiting confidence values revealed along with predictions that are effective in several settings. Two models are studied in detail: decision trees and neural networks for face recognition. If the attacker has access to the model, thanks to confidence values it can recover recognizable images of people's faces starting from the label (their name). This work also starts the study on basic countermeasures. Although the attack does not target specifically distributed learning, it can be applied to any setting where confidence intervals are transmitted with the predicted labels at inference time.

Gradients, which represent the direction and rate of change of a function, are fundamental in training machine learning models. They guide the optimization process by indicating how to adjust model parameters to minimize errors. However, gradients can inadvertently expose sensitive information. In [ZLH19], researchers demonstrated that by analyzing gradients shared during federated learning, an adversary could reconstruct input data, such as images or text, that closely resemble the original inputs. This vulnerability is particularly concerning in collaborative learning environments, where participants might unintentionally reveal confidential data through shared updates.

The distributed learning setting is especially targeted in [SS15], where the authors propose that clients train the model locally with their dataset and share only a subset of the updated parameters and gradients with the other participants. This prevents attackers from gaining knowledge of a client's full local model. However, the global model accuracy is affected by this procedure, as lowering the fraction of shared parameters makes learning slower and more difficult to converge. Furthermore, the work in [HAP17] shows that, for the distributed/federated learning setting, even partial parameter sharing and DP countermeasures can be broken. Due to the real-time nature of model updates, the authors show that a GAN can be trained concurrently to generate prototypical examples of a target training set. With regards to this, the authors of [WSZ+19] argue that GANs are effective in reconstructing data samples from the global dataset distribution but attacking specific clients is a more challenging task, although it may be a stronger privacy threat.

To enhance the ability of a GAN to generate samples from a user-specific data distribution, this work couples the GAN model with a multitask discriminator distinguishing i) category, ii) reality, and iii) client identity of input samples. The proposed framework can be installed on the parameter server (PS), working invisibly from clients. In [GBD+20] optimization strategies for the attacker are designed to increase the ability of the attacker to reconstruct the input image from the knowledge of its parameter gradients. The authors prove the attack formally and show that it is effective even when the gradient is averaged across large mini-batches and through multiple stochastic gradient descent (SGD) steps, as it is common practice for FL. However, the paper in [HGS+21] reviews recent developed attacks, including the last mentioned. The authors show that a combination of known defences can significantly weaken the attack. Specifically, 1) the use of batch normalization without sharing normalization statistics; 2) the use of large batch sizes (smaller than 32 is not safe); 3) the perturbation of gradients (e.g., pruning or addition of noise); 4) encoding inputs. As an alternative, a secure defence is to encrypt the gradients with homomorphic encryption, but the computational cost is high.

The authors of [XHH+22] propose AGIC, an approximate gradient inversion attack that reconstructs the input from model/gradient updates across multiple epochs of learning. In this work, the objective function optimized by the attacker is modified using the cosine distance between the dummy gradient (generated by dummy samples) and the received real gradients. AGIC is 5x faster than the benchmarks in attacking FedAvg and has a 50% better peak-signal-to-noise-ratio (PSNR). In [WCG+23], the authors develop an attack resistant to DP and gradient perturbation, where a model trained to minimize the reconstruction error on auxiliary data can invert gradients despite the countermeasures used.

3.2.2 Membership inference attacks

Membership inference attacks (MIAs) are a sort of privacy attack that leaks information about a data record that is included in the model training. In other words, given a data record and some auxiliary information, an adversary can decide whether the specific record is used in the trained dataset or not [SSS+17]. Membership inference attacks can be categorized based on the attacker's knowledge and capabilities to black-box and white-box attacks. In black-box attacks, the prediction outputs or confidence scores of machine learning model which are considered as auxiliary information, are useful in quantifying membership information leakage. For example, a model may show better confidence or lower prediction errors on training data than on unknown data. By studying these patterns, the attacker can determine if a certain data point was included in the training set. In addition, the adversary may train one or more shadow models that imitate the behaviours of the target model using data from a similar distribution. The shadow models' outputs are used to train a secondary model (attack model) that predicts if a data item was included in the target model's training set. In white-box attacks, the attacker has complete control over the model, including its architecture, parameters, and maybe the training process. This access opens more opportunities to infer membership.

Several factors can make a model more vulnerable to membership inference attacks. Those models which perform well on training data but have poor performance on unseen data (overfitted models) are vulnerable to membership inference attacks due to the considerable disparity in model behaviour. This type of attack gain attention in scenarios where sensitive data, such as medical records or financial information, is involved in the training set.

In contrast to traditional membership inference attacks which often rely on static information from models such as prediction confidence scores, a dynamic approach is introduced in [LZB+22] that leverages the trajectory of the model's loss over multiple training epochs to infer whether a particular data point was included in the training set. By analysing how the loss for a specific data point changes over time, the method can distinguish between training data points (which typically have a decreasing loss trajectory) and non-training data points (which may have a different trajectory).

3.2.3 Property inference and FL setting attacks

Collaborative machine learning and federated learning enable several participants, each with their own training dataset, to create a joint model by training locally and communicating model updates. The protocol design of federated learning may exhibit vulnerabilities which can be exploited by (malicious) server and participants. The server can observe individual updates, tamper with the training process, and control the participants' view of global parameters. Participants can also control parameter uploads. Malicious participant can intentionally change inputs or introduce backdoors into the global model. Poisoning attacks and inference attacks are two major types of attacks against federated learning [LYY20]. In [MSC+19], it is demonstrated that the model updates which are shared during the federated learning process can disclose unintended information about the training data of each participant. Potential attacks such as membership inference attacks and property inference attacks can be applied in the FL setting to reveal information about the training data of each participant.

While membership inference attacks focus on the privacy of individual records in the training dataset, in property inference attacks, the adversary focuses on the inference of sensitive global properties of the training dataset and tries to derive dataset attributes that were not explicitly stored as features or connected with the learning task [GWY+18]. An example of property inference is the extraction of information about the ratio of women and men from a patient dataset when this information was not stored as an attribute or label in the dataset. To achieve property inference, the adversary needs a classifier, called a meta classifier, to recognize patterns within the target model. The meta-classifier is trained using the “shadow training” [AMS+15] technique, which involves the adversary training several proxy classifiers to create the training set for the meta-classifier. Given the target model, the trained meta-classifier predicts whether the target model has a specific property or not.

3.2.4 Extraction attacks

In this section, model and data extraction attacks in AI/ML are presented, which could infer private parameters and replicate the functionality. Techniques like data extraction, reconstruction, and functionality extraction could exploit model outputs to uncover sensitive information, which are analysed below.

3.2.4.1 Model extraction attacks

Model extraction (ME) refers to the inference of the parameters of a (private) model in a black-box fashion to generate a surrogate model that has the same functionality as the original model. It was first proposed in [TZJ+16], where the authors show near-perfect performance of the attack for logistic regression, neural networks, and decision trees. However, this attack only works when the adversary has full access to the predictions, i.e., the probabilities for each class in a classification task.

In follow-up works, researchers also succeeded in stealing the hyperparameters used during training, which can be of high commercial value, as they can highly affect the final model performance [WG18]. Countermeasures are also studied in this work, and the authors show that rounding the model parameters before sharing them can reduce the effectiveness of the attack. A further step is done in [OSF19] where the authors, other than stealing the unknown model architecture, can infer other internal information such as the optimizer used for training (e.g., SGD or the ADMM). This information can be used to strengthen adversarial samples and models.

In the recent paper [LPL+24], the authors assess that ML and DL are still highly susceptible to ME attacks. Moreover, the advances of optimizers also enhance the adversaries’ capabilities. Nonetheless, adversarial learning, which is often used as a tool by attackers, is more effective in controlled environments than in real-world scenarios. Current defence mechanisms, i.e., model quantization [WG18] may weaken ME but remain inadequate.

3.2.4.2 Data reconstruction attacks

Data reconstruction attacks are closely related to the previously mentioned MIA attacks (Section 3.2.2) and model inversion attacks (Section 3.2.1). While MIAs can predict whether a *specific* data sample was used to train a given known model and model inversion is devoted to reconstructing statistically the input data distribution so that it is possible to generate *representative* samples, data reconstruction aims to reconstruct *verbatim* training examples by exploiting access to a ML model’s outputs, gradients, or internal parameters.

In [CTW+21], the authors show that it is possible to recover individual training examples by querying a large language model (LLM), specifically, GPT-2. Among the data the authors retrieved, there were verbatim texts such as personal information (names, phone numbers, email addresses) and IRC conversations. The authors proceed by generating text from the attacked model likely to be memorized, then perform an MIA to assess whether the generated data was part of the dataset with standard techniques (perplexity score). By taking the data with low perplexity, one can find a wide variety of memorized text. This first attempt generates a high number of false positives and only detects training data that were seen many times. An improved version of the attack samples the text with a decaying temperature in the Softmax classifier, making the model less confident of the output. This step, together with an improved version of the MIA, improves the performance of the attack.

Another prominent example is the study by Zhao *et al.* [ZSE+24], where the authors introduce a novel attack method targeting federated learning systems. The attack overcomes previous limitations by breaking the anonymity of aggregation, as the leaked data is identifiable and directly tied back to the clients from which it originates. By sending clients customized convolutional parameters, this attack ensures that the weight gradients of data points between clients remain separate even through aggregation. This approach enables the literal reconstruction of a portion of the original training data, recovering exact data points rather than mere approximations, thereby escalating privacy concerns in federated learning scenarios.

Countermeasures against data reconstruction attacks include techniques like DP [ACG+16], which adds noise to gradients or outputs to prevent precise recovery of the original data. Additionally, implementing strict access controls, minimizing the amount of information exposed through model outputs, and using secure aggregation protocols in federated learning can reduce the risk of data reconstruction.

3.2.4.3 Model functionality extraction attacks

Model functionality extraction attacks aim to replicate the behaviour of a target model without direct access to its internal parameters or architecture. By strategically querying the target model and observing its outputs, an adversary can train a substitute model that approximates the target model's functionality. This type of attack raises significant intellectual property concerns, as it allows for the unauthorized reproduction of proprietary models.

In [TZJ+16], the authors demonstrate that attackers can use prediction APIs to train a local model that mimics the target model's behaviour. By systematically querying the target model with inputs sampled from a chosen distribution and recording the outputs, the adversary creates a dataset to train the substitute model. This method has been shown effective against various model types, including logistic regression, neural networks, and decision trees.

Defences against model functionality extraction attacks include limiting the number of queries allowed, implementing query rate limiting, adding noise to the outputs, or employing techniques like model watermarking to detect and trace unauthorized use of extracted models [ABC+18]. Additionally, using robust API access policies and monitoring for abnormal querying patterns can help mitigate the risk of such attacks.

3.3 Explainability threats

Regarding the recent advancement in AI/ML, there is almost zero human involvement for some of these decision-making systems while learning to solve increasingly complex computational tasks. In addition, ML models that attain high performance on these tasks are naturally complex black boxes that are hard to understand. It is possible to say that a trade-off exists between a model's performance and transparency. This has urged the demand for transparency and led to a question about how the AI/ML systems make decisions. Consequently, an active field of research on eXplainable AI (XAI) has emerged with the motivation to make the behaviour and predictions of AI/ML systems understandable to humans and provide transparent decision-making processes of complex AI systems [LFM99, GA19].

However, despite the considerable progress ML has made across various domains, it has also been applied in cyberattacks. Adversaries have been leveraging ML technology to attack the solutions, aiming to steal sensitive information or disrupt normal functioning. These new attacks are different from well-known traditional attacks because they do not exploit software vulnerabilities or breach systems. One way of performing an attack is that attackers use an exposed API the same way as authorized users do and send artificially designed queries over the exposed API to leak sensitive information, instead of directly attacking the systems.

The fields of adversarial AI and XAI were previously assumed to be unrelated. However, recent discoveries suggest a connection between these two fields. The insight continues to pave the way for future work in both domains, highlighting the potential of adversarial robust and interpretable models to coexist. Given the above advancements, the robustness of prediction explanations becomes a significant and challenging issue, especially since users in many applications value the interpretation as much as the prediction itself. Thus, it is crucial to understand how much explanations might be affected by minor systematic perturbations to the input data, which could be produced by adversaries or measurement biases, for both scientific robustness and security concerns. Consequently, this situation calls for a careful balance between them and, consequently, further advancements in the development of security frameworks.

There are several attack methodologies specifically that are designed for model explanations in the literature. They provide several strategies for fooling interpretability and explainability of ML algorithms. The attack methodologies can be basically divided into three common approaches to fool or steal XAI outcomes. The first approach is to fool explainability by using security related attacks such as data poisoning. The second one is to consider an explanation as a function of model and data, so there is a possibility to attack by changing one of these variables, this category can be considered under model-based attacks. The third one is privacy related attacks that aims stealing such as data reconstruction attacks. Based on this categorization, some of the recent adversarial XAI methods are briefly introduced below.

Security-related attacks

One of the first adversarial assaults on interpretability approaches is described in [GAZ19]. The researchers evaluate the trustworthiness of neural network interpretations produced by several popular feature importance methods. They generate perturbations into images, causing unstructured changes in the explanation maps while keeping the predicted label unchanged. According to their results, explanation maps from several commonly used approaches may be arbitrarily manipulated. This vulnerability results from the high complexity and nonlinearity of the models. While [GAZ19] works on unstructured change, later [DAA+19] focuses on structured manipulations to reproduce a given target map on a pixel-by-pixel basis. Resilience against manipulations is increased by only keeping the explanation process smooth and the model unchanged. Another focus of this research is to keep the output of the network constant (approximately). A white-box attack configuration is suggested in [ZWS+20] to simultaneously fool deep neural networks and their interpreters. In the white box setting, the adversary has complete access to the classifier and the interpreter, including their architectures and parameters. The reason for targeting both the model and the interpreter is to show that attacks are successful because of the prediction-interpretation gap. The idea behind that interpreter is often misaligned with the classifier and interpretability can only explain the behaviour of the classifier partially.

Model-based attacks

Unlike security related methods, [HJT19] adjusts the complete network to provide manipulated examples without hurting the accuracy of the model. Thus, instead of perturbing the input data or changing the explanation to a specific target explanation map, modify the parameters of the network. Interpretation results are directly incorporated in the penalty term of the objective function for fine-tuning. [DBJ+20] follows the same way as [HJT19] and modify the model to manipulate the explanations of common saliency methods to hide fairness. For this reason, the method adds an explanation loss term to the original loss in the form of the gradient of the original loss during training with respect to a chosen target feature. This work can be considered one of the earliest that raises concerns about using explanation methods to check model fairness and investigates how these attacks against explanation can mask a model’s discriminatory use of a sensitive feature.

Privacy-related attacks

The authors of [SSZ21] were pioneers in exploring the balance between explaining ML models and preserving privacy. They analysed the risks of divulging model explanations, where an adversary could trace if a query was part of the model’s training set. Their investigation highlighted how certain explanations leak membership information, particularly affecting minorities in the training data with uncertain predictions and varied explanations. [HBP20] ponder whether an explainable model can maintain data privacy. They introduce an approach using locally linear maps (LLM) on simpler models to ensure privacy, revealing only the LLM instead of the entire model. Their experiments highlight the trade-offs between privacy, explainability, and accuracy. For instance, enhancing explainability often involves reducing privacy, while private training benefits from increasing the dimensionality of random projections, illustrating the intricate balance among these aspects. Explainability also poses risks in Machine Learning as a Service “MLaaS”, where predictions and their explanations are provided for each query. Adversaries can exploit XAI to identify critical features and manipulate the training dataset to alter predictions during inference. To counter such attacks, XRAND [NLP+23] introduces the concept of achieving Local Differential Privacy (LDP) in explanations.

4 Prevention of AI/ML threats

The widespread adoption of artificial intelligence and machine learning has brought about transformative benefits across 5G and future 6G technologies. However, this proliferation also exposes AI/ML systems to a variety of security threats that can undermine their reliability and trustworthiness. This section explores the landscape of AI/ML threat prevention techniques. By examining these areas, it aims to provide a comprehensive understanding of how to protect AI/ML systems from malicious exploitation and ensure their secure deployment.

4.1 Adversarial training

Adversarial training is recognized as an intuitive way of defensive strategy in which the robustness of the deep learner is strengthened by training it with adversarial samples. This strategy can be represented mathematically as a Minimax game, as below:

$$\underset{\theta}{min} \quad \underset{|\delta|| \leq \epsilon}{max} \; J(h_\theta(x+\delta), y)$$

where h denotes the model, J denotes the model's loss function, θ represents the model's weights and y is the actual label. δ is the amount of perturbation amount added to input x and it is constrained by the given ε value. The inner objective is maximized by employing the most powerful attack possible, which is often approximated by various adversarial attack types. To reduce the loss resulting from the inner maximization step, the outside minimization objective is used to train the model. This whole process produces a model that is expected to be resistant to adversarial attacks used during the training of the model. For adversarial training, Goodfellow *et al.* [GSS15] used adversarial samples crafted by the FGSM attack. And Madry *et al.* used the PGD attack to build more robust models, but at the expense of consuming more computational resources. Despite the fact that adversarial training is often regarded as one of the most effective defences against adversarial attacks, adversarially trained models are still vulnerable to attacks like CW. And it is known that although adversarially trained models are somewhat resistant to adversarial samples to some extent, these models generally suffer from severe overfitting issue which is known as robustness-accuracy trade-off.

Ensemble adversarial training

As explained in the above section, adversarially trained models can be made robust to white-box attacks (i.e., with knowledge of the model parameters) to a certain extent if the perturbations computed during training closely maximize the model's loss. However, it is shown that adversarially trained models are still vulnerable to Black-box adversarial attacks. As a solution to mitigate Black-box adversarial attacks, Ensemble Adversarial Training is proposed [TKP+18].

This method augments a model's training data with adversarial examples crafted on other static pre-trained models. In this way, it decouples adversarial example generation from the parameters of the trained model and increases the diversity of perturbations seen during training. Intuitively, as adversarial examples transfer between models, perturbations crafted on an external model are good approximations for the inner maximization problem.

4.2 Differential privacy

The concept of Differential Privacy is rooted in providing plausible deniability to participants, typically by introducing random noise to their inputs [Beb19]. In ML, DP techniques can be employed to ensure user privacy, even if model updates are intercepted by an adversary. Models trained using methods like Differentially Private Stochastic Gradient Descent (DP-SGD) [ACG+16] can significantly reduce the risk of data leakage through DP-induced noise.

The $\varepsilon - \delta$ DP framework is formally defined by the following inequality.

$$\Pr[M(x) \in S] \leq e^{\varepsilon} \Pr[M(y) \in S] + \delta$$

where a randomized algorithm M satisfies ε-DP if datasets x and y differ by at most one element ∀ S ⊂ Range(M). Here, ε represents the privacy budget, while δ denotes the probability of privacy leakage, treated as a constant. However, research in [BPS19] indicates that adding noise through DP can reduce model accuracy. The work in [TLC+20] explores the application of LDP to secure model parameter updates, introducing a collaborative training approach that applies utility-aware perturbations to control noise levels. They also highlight that LDP mechanisms can protect against inference attacks. Importantly, DP does not add communication overhead, as the structure and architecture of the model parameters remain unchanged. This approach could be widely adopted across various devices in future networks, given that DP algorithms and noise bounds are well-defined and implemented through DP-based wrappers [YSS+21]. Nevertheless, the reduction in model accuracy is a significant trade-off associated with DP, potentially limiting the utility of ML models [HGL+20].

4.3 Distributed training

Distributed machine learning (DML) is a method of training machine learning models by sharing data and computations over numerous machines or devices. This strategy is especially beneficial when dealing with enormous datasets or complicated models that would take too long or require a lot of resources to run on a single system. Another benefit of DML is its tolerance against machine failures, the system can continue working, often with minimal disruption.

This fault tolerance is essential for long-running training jobs and in environments where hardware reliability cannot be guaranteed. Various architectures and strategies exist within distributed machine learning, including centralized and decentralized approaches.

Centralized Federated Learning is a type of distributed machine learning in which the model is trained across several devices (or clients), yet the data remains local to each device. The devices transmit local model updates, not raw data, to a central server for aggregating.

In Decentralized Federated Learning (DFL), there is no server to aggregate the model updates and coordinate the training process. Instead, it leverages a peer-to-peer (P2P) network of devices (or clients) that collaboratively train a machine learning model while keeping their data local [MQS+23], as shown in Figure 4-1. Each device shares its local model update with a subset of other devices in the network, and the aggregation of these updates is performed collectively without central coordination. By eliminating the central server in the federation, DFL reduces bottlenecks due to server limitations or network congestion. It also avoids the need to trust a central entity responsible for the global model creation, as there is no single authority controlling the process. Additionally, DFL systems are inherently more robust to system failures since the decentralized structure avoids single points of failure. This decentralized approach aligns with distributed ledger technologies and can incorporate blockchain mechanisms to ensure the integrity and verifiability of model updates.

Implementing distributed training involves several technical challenges. Synchronization of model updates among participating devices is a primary concern. In synchronous training, all devices must wait for each other to complete their computations before proceeding, which can lead to inefficiencies if there is significant variability in computation times or network delays. Asynchronous training allows devices to operate independently but introduces complexities in ensuring convergence and maintaining model consistency.

Distributed training can improve scalability by leveraging multiple devices to handle larger datasets and more complex models. However, it also raises concerns about resource management and coordination. As the number of participating devices increases, managing them and their asynchronous communications becomes more complex. Efficient scheduling, load balancing, and resource allocation strategies are required to optimize performance and resource utilization.

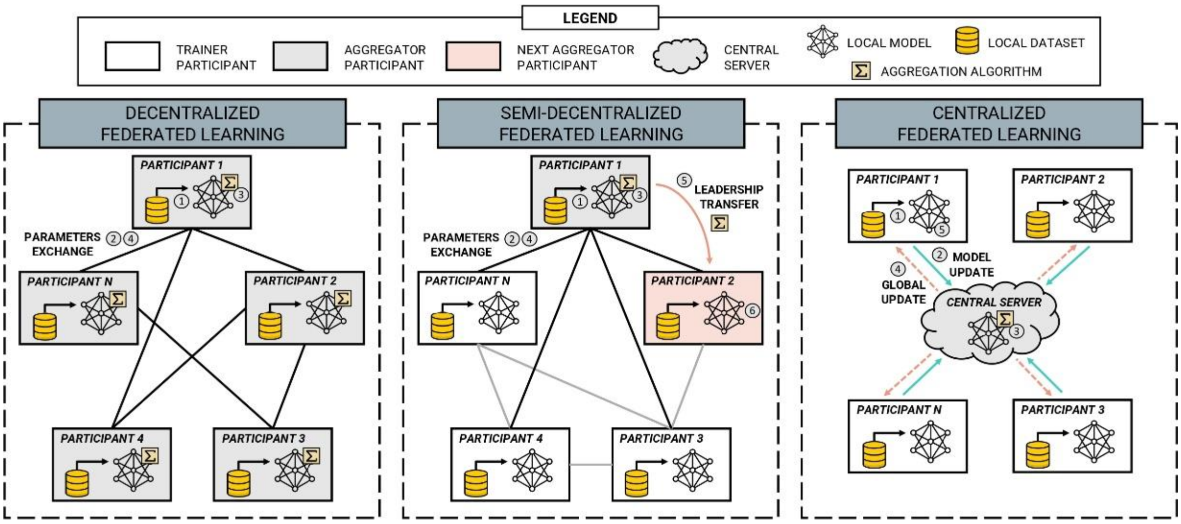


Figure 4-1: Centralized and decentralized federated learning

In addition to scalability and efficiency, robustness improvement is a critical aspect of distributed machine learning. Ensuring that the system can withstand various types of failures and adversarial behaviors is essential for reliable operation. Robustness in distributed settings encompasses resilience to both unintentional faults, such as network issues and hardware failures, and intentional attacks aimed at disrupting the training process or compromising the model.

One of the primary concerns in distributed machine learning is the presence of Byzantine failures, where nodes may behave arbitrarily due to errors or malicious intent. To address this, Byzantine fault-tolerant algorithms have been developed to ensure that the distributed system can reach consensus despite a fraction of faulty or malicious nodes.

Robust aggregation methods play a crucial role in improving robustness. Traditional aggregation techniques like averaging are susceptible to outliers or poisoned updates from malicious clients. To mitigate this risk, advanced methods such as the geometric median, trimmed mean, and Krum have been proposed. These techniques are designed to be robust against a certain number of corrupted updates by minimizing the influence of outliers on the aggregated model.

Addressing adversarial attacks is another critical aspect of enhancing robustness in distributed machine learning. Attackers may attempt to perform model poisoning by injecting carefully crafted updates that degrade model performance or introduce hidden vulnerabilities. Implementing anomaly detection mechanisms at the aggregation server can help identify and discard suspicious updates.

Handling data heterogeneity is essential for robustness, especially in federated learning scenarios where clients possess non-identically distributed data. Variations in data distributions can lead to models that perform poorly on certain subsets or are biased toward specific client data. Algorithms can be adapted to account for this heterogeneity by weighing updates based on local dataset sizes or quality, ensuring a more balanced and robust global model.

4.4 Input anomaly detection

Input anomaly detection has emerged as a critical technique for improving the robustness of deep learning models. Input anomalies refer to data instances that deviate from the training data distribution. They include adversarial examples intentionally crafted to deceive models, out-of-distribution inputs that differ significantly from training data, and noisy or corrupted data [GSS15]. Detecting such anomalies is crucial for maintaining model performance and preventing erroneous outputs or system failures.

Various techniques have been developed for anomaly detection in deep learning [CC19]. Statistical methods for anomaly detection rely on the assumption that normal data points occur in high-probability regions of a statistical model, while anomalies occur in low-probability regions. Techniques such as Gaussian Mixture Models (GMMs) fit a mixture of Gaussian distributions to the data. In GMMs, the probability density function is defined as a weighted sum of multiple Gaussian components, each characterized by a mean and covariance matrix. The mixture weights represent the proportion of each component in the overall distribution. Anomalies are detected based on the likelihood of data points under the model; a low likelihood indicates a potential anomaly.

Reconstruction-based methods involve training a model to reconstruct input data and then measuring the reconstruction error. High reconstruction error suggests that the input is anomalous. Autoencoders are commonly used for this purpose. They consist of an encoder that maps the input to a latent space representation and a decoder that reconstructs the input from this representation. The model is trained to minimize the difference between the input and its reconstruction over the training data. Variations of autoencoders include denoising autoencoders, which are trained to reconstruct clean inputs from corrupted versions by introducing noise into the inputs during training. This encourages the model to learn robust representations that capture the underlying structure of the data. Sparse autoencoders include a sparsity penalty in the loss function to encourage the latent representation to be sparse, promoting feature extraction.

Deep learning models can be used to estimate the probability density of the data. Normalizing flows are a class of models that use a sequence of invertible and differentiable transformations to map a simple base distribution to the data distribution. By applying the change of variables formula, they can compute exact likelihoods for the data. Anomalies are detected by computing the log-likelihood of inputs; low likelihood indicates an anomaly.

Ensemble methods combine multiple models or techniques to improve anomaly detection accuracy. They can involve heterogeneous ensembles, which combine different types of models such as autoencoders, One-Class Support Vector Machines (SVMs), and GANs, or homogeneous ensembles, which use multiple instances of the same model with different initializations or subsets of data. The anomaly scores from different models are aggregated, often by averaging or voting mechanisms, to make a final decision. Calibration of anomaly scores from different models is important for effective aggregation. While ensembles can enhance detection performance, they require more computational resources.

Unsupervised feature learning methods leverage self-supervised learning techniques to learn representations without labelled data. Pretext tasks, such as predicting rotations or context in images, are used to train models. Anomalous inputs typically result in poor performance on these tasks, allowing for anomaly detection. GANs have been adapted for anomaly detection. Training GANs for anomaly detection involves challenges such as instability in the adversarial training process and the risk of mode collapse, where the generator fails to capture the full diversity of the data distribution.

Detecting adversarial attacks is critical for the robustness of deep learning models. Input processing techniques apply transformations to inputs and observe inconsistencies in model predictions. For example, feature squeezing reduces input precision or applies smoothing filters to detect anomalies in the model's response. Statistical tests can be performed in the activation space of hidden layers, estimating densities to identify outliers. Bayesian neural networks use uncertainty estimates to detect anomalies, as adversarial examples often lead to increased predictive uncertainty.

Implementing effective input anomaly detection faces several challenges. High-dimensional data common in deep learning can make density estimation and anomaly detection computationally challenging due to the curse of dimensionality. Complex models may inadvertently learn to reconstruct or classify anomalies as normal, especially if anomalies are not sufficiently different from normal data. Defining appropriate evaluation metrics is non-trivial due to the imbalanced nature of anomaly detection tasks; metrics like the Area Under the Receiver Operating Characteristic Curve (AUROC) and precision-recall curves are commonly used.

4.5 Knowledge distillation

Knowledge Distillation (KD) is a model compression technique that transfers knowledge from a large, complex model (known as the teacher model) to a smaller, more efficient model (called the student model). Initially proposed by Hinton *et al.* [HGD15], KD aims to reduce the computational cost and memory footprint of deep learning models while maintaining a high level of performance. The core idea behind KD is to have the student model mimic the behaviour of the teacher model by learning from its predictions, specifically its soft label outputs, rather than just learning from the ground-truth labels.

The teacher model is typically a large, pre-trained neural network with high accuracy, whereas the student model is a smaller, simpler model that tries to approximate the teacher's behaviour. The distillation process involves training the student model using not just the hard labels (i.e., class labels), but also the soft labels generated by the teacher model, which include the probability distribution over all classes.

As depicted in Figure 4-2, the central concept in Knowledge Distillation is to use the teacher model's output distribution, particularly its logits, to provide additional information to the student model during training [GYM+21]. In typical classification tasks, a neural network output for a sample x is a vector of logits z, which are unnormalized scores representing the model confidence in each class. These logits are passed through a Softmax function to produce probabilities.

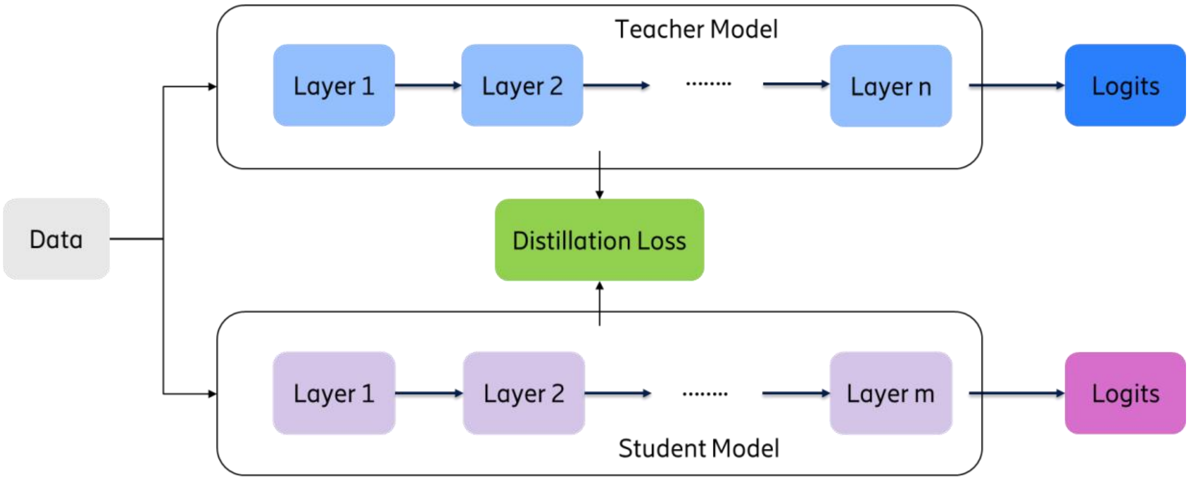


Figure 4-2: Feature-based knowledge distillation

Knowledge Distillation has found numerous applications in compressing deep learning models for deployment on resource-constrained devices such as mobile phones and embedded systems. Beyond the standard teacher-student paradigm, various extensions of KD have emerged. For instance, self-distillation involves training a single model by progressively distilling knowledge within its own layers, while multi-teacher distillation leverages multiple teacher models to improve the accuracy of the student model.

Knowledge distillation for robustness improvement

In traditional ML and DL systems, Knowledge Distillation helps improve model security by smoothing decision boundaries [MOF22]. One of the primary strategies of adversarial attacks is to exploit the sharp decision boundaries in overfitted models, where small perturbations to the input data can cause significant changes in predictions. Through the distillation process, student models tend to have smoother decision boundaries, making them less sensitive to adversarial perturbations. This results in a more robust model that is harder for adversarial examples to exploit.

Additionally, the distillation process enhances the model's ability to generalize by training the student model on soft labels produced by the teacher model. These soft labels provide more nuanced information about the relationship between classes compared to hard binary labels. As a result, the student model is better equipped to handle unseen data, including adversarial inputs that have not been encountered during training. This improved generalization directly contributes to making the model less vulnerable to adversarial attacks.

Reducing model complexity is another benefit of Knowledge Distillation. Larger, more complex models are more prone to overfitting and memorizing the data, which makes them susceptible to adversarial attacks. By distilling the knowledge of the teacher into a smaller student model, KD reduces the complexity of the model, making it less likely to overfit and, consequently, less prone to attacks. This reduced complexity also limits the attack surface that adversaries can exploit.

Furthermore, adversarial examples often rely on the transferability of attacks from one model to another. By training a student model using Knowledge Distillation, the generalized knowledge makes it harder for adversarial examples designed to attack one model to succeed in attacking another. This cross-model transferability is a key factor in many adversarial attacks, and KD helps mitigate this risk.

One specific application of KD to improve robustness is through defensive distillation. In this variant, temperature scaling is applied to the logits of the teacher model, making the student model more resilient to adversarial examples. The softer probability distributions provided by the teacher at a higher temperature allow the student to learn more distributed representations, reducing its vulnerability to adversarial perturbations. This defensive technique has been shown to increase the resistance of models to a variety of adversarial attacks.

Knowledge distillation in federated learning

In Federated Learning, where models are trained across multiple clients without centralized data sharing, Knowledge Distillation plays an important role in addressing several unique security challenges [QZZ+24]. One of the primary issues in FL is dealing with non-IID data, where different clients may have varying data distributions. This heterogeneity makes it difficult to build a robust global model. KD mitigates this by training a global student model that learns from the local teacher models of various clients, instead of doing an averaging or other mathematical model normalization. This process ensures that the global model is robust to the different client data distributions. This process reduces the impact of adversarial perturbations that arise due to the non-IID nature of the data.

Model poisoning, where adversarial clients send corrupted updates to the central server, is a major concern in FL. Knowledge Distillation helps mitigate model poisoning attacks by transferring only distilled knowledge from clients to the central server. Instead of directly sending gradient or model weight updates, KD focuses on transmitting the distilled outputs from the local models, limiting the ability of adversaries to inject malicious updates. This process makes it harder for malicious clients to compromise the integrity of the global model.

Model poisoning, where adversarial clients send corrupted updates to the central server, is a major concern in FL. Knowledge Distillation helps mitigate model poisoning attacks by transferring only distilled knowledge from clients to the central server. Instead of directly sending gradient or model weight updates, KD focuses on transmitting the distilled outputs from the local models, limiting the ability of adversaries to inject malicious updates. This process makes it harder for malicious clients to compromise the integrity of the global model.

Another significant benefit of KD in Federated Learning is the reduction of model size and communication overhead. In FL, clients frequently communicate updates with the central server, and this communication can be a vector for attacks, such as model inversion. By utilizing Knowledge Distillation, a smaller student model can be trained on each client, resulting in smaller updates being transmitted to the server. This reduction in communication overhead not only improves system efficiency but also minimizes the information available to adversaries, reducing the risk of privacy and inference attacks.

KD can also enhance personalization in FL, especially in decentralized settings [JK23]. In this context, clients may not prioritize the global model (i.e., the aggregation of all client models) in its entirety; instead, KD can be used to selectively aggregate specific models or portions of models from other clients. By measuring the similarity between intermediate logit outputs—using statistical distances like the Wasserstein distance between local models—and determining the optimal combination of model copies from neighbouring clients, each client can individually update parameters from a personalized perspective, enhancing performance without sharing local data. Co-distillation, based on local validation datasets, allows each client to assess similarity, enabling effective collaboration across heterogeneous models (e.g., different layer structures). Distillation thus enables interoperability between models with varied architectures, as long as they share at least one common layer of matching dimensionality.

Integrating KD with DP techniques in Federated Learning also enhances security. Differential privacy ensures that individual client data remains protected by limiting the information shared between clients and the server. When combined with Knowledge Distillation, differential privacy helps ensure that only soft labels or logits are transmitted, reducing the risk of model inversion or privacy attacks.

In sum, Knowledge Distillation offers significant advantages for enhancing the robustness and security of ML and DL models, both in traditional centralized settings and in FL environments. In traditional ML/DL systems, KD smooths decision boundaries improve generalization, and reduces model complexity, making models more resistant to adversarial attacks. In FL, KD addresses challenges posed by non-IID data, adversarial clients, and communication overhead. Techniques like defensive distillation and differential privacy, when combined with KD, further enhance security in federated systems by reducing vulnerabilities to model poisoning, data poisoning, and gradient-based attacks. As a result, Knowledge Distillation emerges as a powerful tool for improving the robustness of both centralized and decentralized machine learning systems.

4.6 Model smoothing techniques

Apart from Knowledge Distillation, several other model smoothing techniques are widely used in ML and DL to enhance robustness, particularly against adversarial attacks. These techniques focus on ensuring the model's decision boundaries are smooth and less sensitive to small perturbations in the input data, which are commonly exploited in adversarial scenarios.

Label smoothing

One common technique is Label Smoothing, where the ground-truth labels used for training are softened [XXQ+20]. Instead of assigning a probability of 1 to the correct class and 0 to all others, label smoothing distributes a small portion of the probability mass across incorrect classes. This prevents the model from becoming overly confident in its predictions and encourages smoother decision boundaries. Mathematically, label smoothing adjusts the true label y for a given class by combining it with a small constant ϵ, resulting in smoothed labels. This technique improves robustness by reducing overconfidence in the model's predictions, which can help prevent adversarial examples from causing dramatic shifts in the output.

Adversarial training

As detailed before, Adversarial Training is another highly effective method for improving robustness [BLZ+21]. By continually exposing the model to adversarial examples during training, it becomes better equipped to handle such inputs during inference. In adversarial training, the model is trained to minimize the loss not only on regular inputs but also on perturbed versions. This process softens decision boundaries and makes the model robust to both clean and adversarial perturbed inputs.

Dataset Mixup

Mixup is a data augmentation technique that generates synthetic training examples by interpolating between two samples [ZDK+21]. This technique encourages the model to learn smoother decision boundaries by training it on examples that lie between classes. By training the model on these interpolated examples, Mixup helps the model learn linear relationships between samples, making it more resistant to adversarial attacks that exploit sharp boundaries between classes.

Dropout

Dropout is a regularization technique that randomly disables a proportion of neurons during training, effectively forcing the model to learn with different subsets of neurons in each training iteration [PPS+18]. This introduces randomness and prevents over-reliance on any specific set of neurons, promoting smoother decision-making processes. During training, each neuron's output is multiplied by a binary mask sampled from a Bernoulli distribution, and during inference, all neurons are used, but their weights are scaled appropriately. Dropout encourages the model to be robust against minor disruptions in neuron activity, indirectly improving resistance to adversarial noise.

Gradient regularization

Gradient Regularization, or Input Gradient Penalty, directly addresses the sensitivity of the model's output with respect to its input by penalizing large gradients [RD18]. By minimizing the sensitivity of the model's predictions to changes in the input, this technique enforces smoother decision boundaries. The model is trained by adding a penalty term to the loss function, which minimizes the magnitude of the gradient of the output with respect to the input. This technique directly reduces the model's sensitivity to small input perturbations, making it more robust to adversarial attacks that rely on exploiting these gradients.

Randomized smoothing

Randomized Smoothing is a technique that offers certified robustness against adversarial attacks [CRK19]. In randomized smoothing, Gaussian noise is added to the input, and the model's prediction is averaged over several noisy input versions. This technique smooths the decision boundaries by making predictions less dependent on small perturbations in the input.

4.7 Gradient masking or obfuscation

Gradient masking and obfuscation methods aim to protect models by making the gradients—used by attackers to craft adversarial examples—less informative or inaccessible. By obscuring gradient information, these defences attempt to prevent attackers from successfully optimizing perturbations that can mislead the model into making incorrect predictions.

Randomization methods introduce stochastic elements into the model's architecture or preprocessing steps, causing the gradients to become noisy or "shattered". For instance, adding random noise to the inputs or applying random transformations can make the gradient information unreliable [XWZ+17]. Stochastic activation functions, such as randomly dropping neurons during inference (like dropout during training), can also mask gradients [DAL+18]. This unpredictability in neuron activation patterns makes gradient estimation challenging for attackers.

Utilizing saturating non-linearities is another strategy, where activation functions that saturate at extreme input values (like sigmoid or hyperbolic tangent functions) cause gradients to vanish [CDL+17]. When activation functions saturate, small changes in input lead to negligible changes in output, resulting in near-zero gradients that impede attackers from identifying effective perturbation directions. Additionally, some defences focus on detecting adversarial examples rather than preventing them. By adding a detection mechanism that flags inputs exhibiting abnormal characteristics or statistical deviations from the training data, models can reject or scrutinize potentially malicious inputs [MGF+17]. While not directly masking gradients, this approach adds an additional layer that can interfere with gradient-based attacks.

Despite the variety of gradient masking and obfuscation techniques, research has shown that they often provide a false sense of security. In [ACW18], the authors demonstrate that many defences relying on these techniques can be circumvented. Attackers can exploit alternative methods, such as approximating non-differentiable components with differentiable ones during the backward pass, accounting for randomness by taking the expectation over random transformations applied by the defence or using gradient-free optimization methods like the Square Attack [ACF+20], explained in the Evasion Attacks section, which employs random search strategies to find adversarial examples without relying on gradient information.

Gradient masking does not fundamentally resolve the issue of adversarial vulnerability; it merely obscures the pathways that attackers might use. As attackers develop more sophisticated methods, relying solely on gradient masking becomes insufficient for robust protection. It is recommended to adopt more effective defence strategies, such as adversarial training. Additionally, certified defences [RSL18] offer mathematical guarantees on a model’s robustness within certain perturbation bounds, providing stronger assurance against adversarial attacks.

4.8 Robust XAI

As we have discussed in Section 3.3, XAI methods aim to make machine learning models more interpretable and trustworthy, and they have currently become a necessary component in ML systems. Therefore, it is inevitable that the XAI models will be shipped out with the ML products in the near future more frequently. Thus, the robustness of the XAI methods is becoming equally important in the arena of cybersecurity as much as the security of the ML models. AI approaches are typically divided into *pre-hoc*, which integrate interpretability during development/data processing, *in-model* methods, which use models that are inherently interpretable (like linear models or decision trees), and *post-hoc* methods, which generate explanations for complex models after training using tools like Shapley Additive Explanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME).

Post-hoc methods for XAI are new components added to ML-based systems. This new component can complement the prediction of ML models, weighing heavily on the actions of systems and humans that depend on the ML model. However, they can be also converted to a new attack vector by malicious agents. In some cases, the explanation itself is more important than the prediction. Thus, the removal of the XAI component is also not an option. This is the case for AI used in applications having a societal impact, where predictions must be fair and unbiased. This is also the case for security applications like detection and response (D&R), where an explanation is used to counter and recover from detected attacks using appropriate measures.

A central challenge posed by post-hoc methods is that their explanations can sometimes diverge from the actual predictions of the ML models they interpret, creating a potential vulnerability for attack. Leveraging explainability through transparency could address this issue, as explanations that derive directly from the ML model’s decision process are typically more aligned with its predictions. For an adversarial attack to succeed in these cases, both the ML model and the explanation process must be compromised, a more complex task [KL20]. Moreover, using transparency-based XAI methods provides a degree of protection for explanations, as they can be partially shielded by existing adversarial defences that secure the ML model. Currently, security measures against adversarial ML attacks are more advanced than those for protecting XAI methods. However, when transparency-based methods are not feasible, choosing robust post-hoc explanation techniques can enhance resilience against attacks. For example, empirical studies [SHJ+20] indicate that SHAP demonstrates greater robustness than LIME in concealing biased or unfair outcomes.

Another way to enhance the robustness of explainable methods is to incorporate additional techniques that mitigate any anticipated attack types. One notable attack type is known as “scaffolding attacks” [SHJ+20], which are designed to deceive even well-established security measures, often going unnoticed by auditors and other stakeholders who rely on model explanations. These attacks enable model creators to embed subtle biases, introduce backdoors for unauthorized access, or intentionally compromise system integrity, all while avoiding detection during audits. Once the attack method is brought to light, any party (e.g.: security auditors) that is anticipating such an attack can be equipped with specialized tools to identify them [3]. By detecting and addressing scaffolded models early in the security screening process, auditors can limit potential long-term damage, enhancing both the resilience and trustworthiness of explainable systems.

The robustness and trustworthiness of XAI can also be enhanced using prediction confidence metrics, which not only make AI/ML systems explainable and interpretable but also provide actionable insights into the reliability of the model’s predictions. Prediction confidence offers a mechanism to evaluate the trustworthiness of explanations. For instance, an Intrusion Detection System (IDS) employing post-hoc XAI methods like SHAP or LIME could use the confidence metric to refine the explanations generated, ensuring they align closely with the underlying data distribution and model performance. This integration mitigates the risk of misleading explanations, which can arise when models struggle to generalize to unseen attack vectors or when the explanations themselves are targeted by adversarial attacks.

Thus, incorporating confidence metrics into XAI-enhanced frameworks not only provides robustness against adversarial threats but also ensures trust among human operators, enabling more informed and effective decision-making in cybersecurity environments. Thus, confidence is not merely an auxiliary measure; it is a foundational element that strengthens the synergy between trustworthy AI and secure, reliable intrusion detection.

It is also a possibility that XAI can be used as a tool for improving adversarial privacy violation attacks such as membership inference, model extraction and model poisoning. Since XAI inherently provides additional information about a model, an attacker can exploit the added information to improve the effectiveness of those attacks. The robustness of the applied XAI methods can be increased in this case by following the below steps:

- *Controlled Explainability*: Define the minimum explainability level needed to achieve the intended goal, ensuring that the selected XAI method meets these requirements without revealing unnecessary information. Transparency levels should vary across stakeholders, with model creators needing the highest level and endpoint developers requiring lower levels of interpretability.
- *Restricted Access*: Limit access to model explanations to only essential parties, such as auditors or regulatory bodies, and consider sealing or encrypting explanations for secure access when necessary. A restricted-access approach minimizes the risk of attackers gaining sensitive information.
- *Delayed Availability*: Delays the release of model explanations relative to the model's decisions, slowing down potential adversarial attacks that rely on iterative access to explanations. This approach does not hinder model utility in most cases, though exceptions may exist for stakeholders in real-time monitoring environments.

5 6G key technical enablers and selected cases

This section presents several examples of possible threats that could compromise the proper functioning of the different 6G enablers presented in ROBUST-6G deliverable D2.1 [ROB24-D21]. Specifically, examples of threats in the physical layer, in the AI/ML modules and in federated learning that could cause a major impact on the key 6G technologies. The assessment and prevention of these threats in the various selected cases have been described in Sections 3 and 4 above.

5.1 Reconfigurable Intelligent Surfaces

Reconfigurable Intelligent Surfaces (RIS), also referred to as Intelligent Reflecting Surfaces (IRS) or Large Intelligent Surfaces (LIS), represent a transformative technology poised to revolutionize the wireless communication landscape, particularly in the context of 6G. RIS consists of an array of reflecting elements capable of reconfiguring incident signals, allowing proactive modification of the wireless environment. This capability addresses numerous challenges in modern wireless networks, making RIS a focal point for advancing wireless communication systems. At its core, an RIS is a two-dimensional material structure with programmable macroscopic physical properties. Its most defining feature lies in its reconfigurable electromagnetic (EM) wave response, enabling unprecedented control over wireless channels. Unlike traditional wireless communication networks, where channel behavior is predominantly dictated by the environment, RIS-aided networks empower dynamic control over the channels between transmitters and receivers. This results in enhanced signal strength at terminal devices and provides a new degree of freedom in system design. [LLM+21]

Research has demonstrated the ability of RIS to significantly enhance wireless network performance, including optimized channel gains, improved Quality of Service (QoS), extended coverage range, and reduced energy consumption. These benefits align closely with the overarching goals of 6G, which include achieving ultra-reliable low-latency communication (URLLC), massive machine-type communication (mMTC), and energy-efficient operation. By embedding RIS within 6G architectures, operators can create adaptable, intelligent networks capable of meeting the stringent demands of diverse applications such as immersive experiences, industrial automation, and AI-driven ecosystems.

The use of AI/ML in RIS could be leveraged to achieve key optimization in its operation, in addition to being able to adapt to changing environmental conditions in an intelligent way. Therefore, RIS can be envisaged as a promising avenue for enhancing security through Physical Layer Security (PLS) schemes [KRC+24]. Despite this, as we discussed in ROBUST-6G deliverable D2.1 [ROB24-D21], RIS integration also introduces several potential threats and attacks that require thorough evaluation and implementation of effective mitigation strategies. This subsection describes how the AI/ML techniques that could be used in RIS can be affected by the different threats developed in Section 3, as well as the application of countermeasures (mitigation mechanisms) of Section 4 to address the aforementioned threats.

Among the various threats shown in the threat matrix on RIS, see Table 5-2 in [ROB24-D21], *spoofing* poses a major threat to the authenticity of the communication, as the attacker can manipulate the channel to deceive the receiver. To detect this type of anomaly, AI/ML algorithms can be used during the training phase to generate models based on historical channel data, thus being able to identify unusual patterns that may indicate identity theft attempts [ZLJ+21]. As a possible mitigation, real-time adjustments could be made to the RIS settings to improve the integrity of the attacked signal.

In addition to spoofing attacks, the integration of AI/ML techniques into the RIS security framework delivers a proactive approach to threat prevention, including the other attack types discussed in [ROB24-D21], such as tampering, repudiation, information disclosure and Denial of Service (DoS) threats. AI/ML techniques are crucial to strengthening the security of RIS-enabled networks, ensuring robust and reliable communication in an increasingly complex threat landscape [LLY+23]. *Tampering* can be covered by AI/ML techniques to boost RIS modulation by dynamically adjusting parameters based on detected anomalies. Reinforcement learning, for example, can strengthen eavesdropping encryption and Secure Key Generation (SKG) processes, thereby increasing resilience against tampering [JSL+21].

Eavesdropping is a critical threat in wireless systems due to the broadcast nature of signals. Illegal Reconfigurable Intelligent Surfaces (IRIS), deployed by attackers, increase this risk by passively enhancing signal leakage and enabling interference attacks. IRIS can optimize phase shifts to increase intercepted signal power, degrading the secrecy rate, or amplify interference to disrupt legitimate users, severely impacting signal quality [WLZ+22]. AI/ML techniques can counter these threats by generating adaptive artificial noise and detecting subtle signal deviations through unsupervised learning. These approaches are essential for enhancing the security of RIS-assisted wireless systems and ensuring robust protection of 6G networks.

In RIS technologies, different approaches can be used for the assessment of the above-mentioned threats, which are able to address both attack simulation and system resilience to potential vulnerabilities [SKM+24]. Attack simulation assessment is one of the most used approaches to test how RIS can respond in a non-operational controlled environment. In this way, weaknesses in the security architecture can be identified. Among the different types of attacks examined in Section 3, the three variants of adversarial threats (Section 3.1), model inversion attacks (Section 3.2.1) and extraction attacks (Section 3.2.4) are the threats that could pose the greatest impact and criticality on RIS.

In the case of poisoning attacks, for example, malicious data could be introduced into the RIS Controller (RISC) and then observe how it affects the wireless signal configuration. In this way, it could be evaluated whether the system is able to identify and isolate malicious data. And in the case of model extraction attacks, the simulation for evaluation would have to focus on trying to extract or replicate the RIS control model, observing whether the RIS system has any mechanisms to protect the internal structure of the model or limit access [KKA+24].

Finally, several prevention mechanisms could be implemented to mitigate the threats identified in RIS, mainly to guarantee data integrity and confidentiality, as well as the robustness and operational security of the RIS technology. All these prevention mechanisms also depend on the type of threat identified [PL23]. In poisoning attacks, data filtering and real-time validation techniques could be enforced to detect and remove anomalous or manipulated data before it influences the RISC model. And approaches to evasion attacks can follow an adversarial training approach (Section 4.1) in which threat detection models are exposed to simulated attacks during training. This will increase robustness and the ability to identify attack patterns in real time.

Other types of prevention mechanisms that could be used, as in any other critical system to be safeguarded, would be the implementation of technologies such as blockchain (also proposed in Section 4.3) to secure, for example, the data supply chain and verify the authenticity of the data in the RIS system. Or the use of advanced information encryption techniques such as homomorphic encryption or secure computing techniques to ensure that data can be processed without being exposed. This would allow us to be better protected against data reconstruction attacks.

Last but not least, the differential privacy mechanisms outlined in Section 4.2 can also be implemented to reduce the amount of information that can be extracted from RIS models, limiting the risk of sensitive data reconstruction discussed in Section 3.2.4.2. And in the case of RIS in federated learning contexts, the application of these differential and federated privacy techniques can protect the data exchanged within the federation and make it difficult to infer it.

5.2 AI/ML for RF sensing and localization

6G wireless networks are demanded to integrate localization and sensing capabilities for both estimating the position of active devices, using radio channel characteristics, and detecting passive objects in the environment.

These services will improve network resource management, while AI/ML methods will complement signal processing to improve accuracy in detecting people, activities, and objects. These concepts are developed in more detail below.

Sensing and localization with communication networks

Next-generation wireless networks are called to support an increasing number of devices and heterogeneous applications. As part of the growing functionalities of next-generation wireless networks (including 6G), capabilities, including device localization and sensing of surroundings, are being introduced. These services provide additional benefits to users and improve network resource management [WQW+23, SVB+22].

Localization and sensing services differ in their objective and how information is collected [BYK+23]. Localization is the estimation of the position of an active wireless device, i.e., transmitting radio signals, from radio channel features such as the received signal power, the signal time of flight, and the angles of arrival and departure of the signal [BAB+23]. For example, a base station (BS) can estimate the distance and the relative angle of a connected user equipment (UE) from the signal received in the uplink.

On the other hand, sensing refers to obtaining information about passive objects in the environment, e.g., people, furniture, cars, and road signs. In this case, the fixed and moving objects act as reflectors, diffractors, and scatterers for the signals. In turn, information about the range, velocity, and angular position, for example, is obtained by analysing how the environment modifies radio signals exchanged by two wireless devices.

Localization and sensing can be obtained through mono-static, bi-static, or multi-static systems [LCM+22]. In the first case, the system acts similarly to a radar device, where the transmitter and the receiver are co-located, and sensing parameters are extracted by the signals reflected to the device. Bi-static sensing is a more typical setup in communication networks because it relies on the typical communication setup composed of a transmitter and a receiver that are not co-located. In multi-static sensing, multiple receivers collect the multiple signal copies generated by multi-path propagation from a transmitter device and process this data for localization and sensing.

Localization and sensing methodologies

As described above, localization and sensing target two complementary tasks: while the first considers active targets, the second aims to obtain information about passive devices. However, processing methodologies to obtain the sensing parameters are common. The systems use as sensing primitive the channel estimate computed by the communication devices through training fields in the data packets. The time, frequency, and space diversity in the channel estimates allow for obtaining information about the range, velocity, and angular position of the passive or active target. In addition to estimating these quantities, the objective of sensing may be to obtain other information from the surroundings, such as identifying the people present in the environment or recognizing the activity they are performing. Standard signal processing techniques may not suffice to address the sensing task in these cases. Hence, several AI/ML approaches have been proposed in the literature for the different frequency bands [MCC+23, SVB+22, HSD+22].

AI/ML for localization and sensing

AI/ML algorithms have become increasingly used for sensing and localization purposes when signal processing methods cannot address the task or reach good accuracy. An overview of AI/ML's role in integrating sensing functionalities within wireless networks is presented in [DA23]. Signal processing techniques for sensing can be referred to as model-based approaches as they rely on n models of radio. AI/ML approaches are instead model-free approaches as the algorithms are data-driven and learn how to address the task from examples used during the training process [MCC+23]. The use of AI/ML for localization and sensing ranges from low-level feature extraction and pattern discovery to object detection and recognition, location tracking and prediction, environmental mapping, and cooperative localization [DBB+21]. For localization, ML is usually used to implement fingerprinting-based algorithms that obtain an estimate of the location of the target by analysing the characteristics of radio propagation and finding the best match with the fingerprints learned during training. For sensing, ML is extensively used in indoor environments for people monitoring, e.g., for activity recognition and person identification.

Attacks on ML models for sensing

The attacks on learning systems for environmental monitoring and localisation mainly focus on perturbing the input of the learning model, i.e., the channel estimate. This leads the learning algorithm to have a wrong perception of the surroundings and, in turn, provides a wrong output for the sensing task. The strongest damage is achieved when the adversary has access to the learning architecture used for sensing. This attack strategy is referred to as a white-box attack and entails crafting a malicious perturbation for the input that maximises the loss. Another possible approach, called *transfer attack*, considers a network that has been trained on a different configuration of sensing nodes, i.e., different positions for the base station and the terminals. These two approaches require the adversary to also access the real channel estimate used by the sensing system to perform the task. If such information is not available, the adversary can craft some adversarial sequences based on the knowledge of the environment. An overview of these attack approaches applied to the localisation task and their evaluation in cellular networks is presented in [HGA+24]. Other adversarial attacks on learning-based localisation are presented in [MSB+23].

A black-box impersonating attack is presented in [LCY+24]. The attack is based on the transmission of a malicious channel estimate that when processed by the learning model is recognized as associated with another user in the network. Other false data injection attacks targeting gesture and activity recognition are presented in [SQG+23] and in [MZL+23]. On the other hand, in [LXD+24], the authors present a different approach where the adversary modifies the pilot symbols used for channel estimation. This leads the victim device to estimate the wireless channel wrongly. Using such a wrong estimate as input for the learning model generates wrong sensing results.

Several of the techniques reviewed in Section 4 could be applied to address the above threats, which could be implemented together to create a more secure environment for the ML models used. These include the use of adversarial training to improve robustness, by injecting adversarial samples during the training phase, as well as input anomaly detection techniques to identify unusual patterns in model inputs.

5.3 Privacy and security for distributed learning

Distributed learning will be a key enabler in 6G, as it will allow data to be processed and analysed directly at 6G network nodes, reducing latency and improving energy efficiency by minimising the need to transfer large volumes of information to centralised data centres. Furthermore, it will also support privacy and security in 6G environments, where the density of connected devices and data generation will be exponentially higher.

In this context, distributed learning enables scalable, privacy-preserving model training by keeping data on local devices, but it introduces security and privacy challenges. Adversaries can exploit vulnerabilities in training, update transmission, and aggregation, with multiple entry points increasing risk. Limited resources in distributed systems heighten these vulnerabilities and are susceptible to various attack types outlined in [ROB24-D21] Section 3, that compromise both security and privacy [RJL+23]. Poisoning attacks involve adversaries degrading model quality by injecting malicious data in the distributed learning node or altering model parameters during the training phase, leading to biased or incorrect outputs. Considering the approaches like fully decentralised FL, the threat from poisoning is even higher as any entity can participate in the training process, without intervention from a central aggregator [CL24].

Evasion attacks [KSM+23] in distributed learning can cause misclassifications via adversarial noise. Privacy attacks consist of several methods: model inversion or data reconstruction attacks, where attackers attempt to eavesdrop the model parameters to reconstruct input data from model outputs [SSW+24b] by exploiting learned correlations; membership or property inference attacks [HZS+24], determining if specific data points or selected properties were part of the training set by analysing model responses; model extraction, replicating a model by inferring its parameters through access to its predictions; and functionality extraction, creating an imitation model by observing input-output pairs from the target model. Therefore, the adversaries can exploit different vulnerabilities within the distributed learning process. Hence, the need for robust defence mechanisms to protect data integrity and confidentiality exists in distributed learning mechanisms.

To counter the various privacy and security threats in distributed learning, several defence and detection mechanisms in Section 4 can be employed, specifically tailored for distributed learning. Differential privacy introduces controlled noise to local individual client model updates or outputs, protecting individual data contributions while maintaining overall model utility. This technique limits the impact of any single data point on the final model, mitigating risks from membership inference attacks. Knowledge distillation transfers knowledge from a complex original client model to a simpler one, obscuring the relationship between training data and model outputs, without directly sharing the original client model. By not exposing the original model's parameters directly, it reduces the risk of model inversion, inference and extraction attacks.

Gradient masking or obfuscation modifies the gradients shared during training to prevent adversaries from extracting sensitive information. Obfuscating gradients makes it more difficult for attackers to perform model inversion or extract private data from gradient information. Additionally, anomaly detection mechanisms at both input and output levels in distributed learning can be helpful in eliminating potentially adversarial clients from aggregation. Input anomaly detection monitors data for patterns indicative of adversarial manipulation or poisoning attempts in client training, to identify if an adversary attempts to inject any malicious data into the client model training. Output anomaly detection observes outputs from the client models before aggregation for irregularities that may signal an ongoing attack or model compromise [SSW+24b].

Combination of multiple techniques is also supporting in the timely identification of manipulated models or suspicious results due to adversarial influence.

5.4 AI-as-a-Service framework

AI-as-a-Service (AIaaS) is the delivery of artificial intelligence (AI) and machine learning (ML) capabilities through cloud platforms as a service, allowing users to access AI models, tools, and infrastructure without having to develop or manage these skills on their own. In a 6G network, AIaaS would empower operators, developers, and enterprises to harness AI technologies for various applications and services in a seamless, scalable, and efficient way. Therefore, it is vital to ensure that the AIaaS framework is well-protected against potential threats and malicious attacks.

In [GTN+24], a comprehensive threat analysis of the AIaaS framework is conducted using STRIDE as the selected methodology, where potential security threats are identified in relation to the recognized assets, and corresponding mitigation strategies are proposed. The identified assets are data, model, environment and tool, and process where the data is the riskiest one. AIaaS often involves processing sensitive or personal data in the cloud, increasing the risk of data leakage and unauthorized access if proper data protection measures are not in place. The potential impact of a data breach is significant, leading to privacy violation and regulatory non-compliance (e.g. General Data Protection Regulation, GDPR). To mitigate this risk, it is crucial to anonymize or encrypt data both at rest and in transit and implement strong access controls. Another identified threat in [GTN+24] against AIaaS is adversarial attacks involving both evasion and poisoning attacks. In an evasion attack, the attacker can carefully craft inference input data to deceive the deployed models in the cloud into making incorrect predictions. These wrong predictions degrade the overall performance of the AIaaS system and compromise its integrity leading to financial loss, reputation damage, or even physical harm. To mitigate this type of attack, continuous testing of the AI model against adversarial inputs and apply techniques like adversarial training will be useful to strengthen the model's resilience at the cloud.

Complementarily, in [TKG24], a defence mechanism to mitigate inference queries based black-box attacks during the inference phase of the AI/ML model is proposed. The model’s uncertainty estimations are quantified by the model owner during prediction time and this information is used to update the model weights in the highly uncertain cases, to minimize quantified uncertainty value, leading to more accurate predictions.

Another vulnerability in AIaaS system is the exploitation of potential weaknesses in cloud providers. A malicious actor could replace a legitimate model with a poisoned one (inject false data into the training data), causing the model to perform in an unexpected manner leading to a loss of trust and reputation also financial loss. To detect such attacks, statistical methods such as Gaussian Mixture, or reconstruction-based methods such as autoencoders can be used by the cloud provider for input anomaly detection in the training data. In addition, an unexplained drop in model accuracy or performance can be considered as a sign of a poisoning attack and can be detected by comparing the model performance on a trusted validation set and real-world data to identify any anomalies in the predictions.

Another well-known attack in AIaaS framework is the model inversion attack. The attacker aims to reconstruct sensitive information about the training data by submitting multiple queries to the AIaaS system and analysing the outputs predicted by the model deployed in the cloud. To prevent such attacks, differential privacy can be used by the cloud provider to add noise to the outputs generated by the model or the training process to make it harder for attacker to infer specific details about individual data in the training set. In addition, limiting the amount of information that the model exposes in its outputs can prevent attackers from gaining insight into the training data.

Model extraction attack is also an important concern for cloud-deployed ML models, where the attacker aims to replicate the behaviour and functionality of the deployed model by querying the model and using the responses. The attacker can use the replicated model to provide a competing service which can lead to a loss of market share for AIaaS provider. Limiting the number of queries a user can make within a certain period, and adding noise to the outputs of the model can reduce the attacker's ability to train a surrogate model.

6 Conclusions

We have provided in this deliverable a comprehensive analysis of the security threats affecting AI/ML models in the context of 6G networks. This analysis has been performed following the STRIDE methodology, which systematically categorizes threats into spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. This systematic approach is introduced in Section 2 of this report allows identifying and assessing the potential impact of each threat on AI/ML systems. It has highlighted the need to strengthen security against adversarial attacks, which could manipulate training and prediction data, thus compromising the performance and reliability of AI applications, as well as different types of threats against privacy and explainability of AI/ML models. This in-depth analysis has been carried out in Section 3.

Associated with the analysed threats, a complementary study on prevention and mitigation strategies to protect AI/ML systems from malicious exploitation, and thus ensure their secure deployment, has been carried out in Section 4. To this end, various methods to prevent threats to AI/ML models have been analysed, focusing on techniques such as adversarial training and differential privacy to secure AI/ML systems against adversarial attacks and data leakage. Other mitigation methods have been also highlighted, such as distributed training and model smoothing to improve robustness, as a key challenge for secure AI/ML deployments. In addition, knowledge distillation and explainable AI have also been examined as promising methods to improve security and interpretability while addressing challenges in federated environments.

Finally, Section 5 explores several given scenarios and emerging technologies that are particularly vulnerable to the threats reviewed in the previous sections, and which require the implementation of effective mitigation strategies; threats to key 6G enablers like Reconfigurable Intelligent Surfaces (RIS), AI/ML for RF sensing and localization, and distributed learning. This study has also been conducted on a specific AI-as-a-Service (AIaaS) framework, which allows AI and ML capabilities to be accessed through cloud platforms. All these selected cases and key enablers have allowed us to verify how the different threats studied impact AI/ML, and how they could be remediated through the proposed mitigation methods.

References

[ABC+18] Y. Adi, C. Baum, M. Cissé, B. Pinkas and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring". In Proceedings of the 27th USENIX Security Symposium (USENIX Security 2018), pp. 1615-1631, August 2018.

[ACF+20] M. Andriushchenko, F. Croce, N. Flammarion and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search". In Proceedings of the 16th European Conference on Computer Vision, Lecture Notes in Computer Science vol. 12368, pp. 484-501, August 2020.

[ACG+16] M. Abadi, A. Chu, I. J. Goodfellow, *et al.*, "Deep learning with differential privacy". In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308-318, October 2016.

[ACW18] A. Athalye, N. Carlini and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples". In Proceedings of the 35th International Conference on Machine Learning, pp. 274-283, July 2018.

[AMS+15] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers". International Journal of Security and Networks, vol. 10, no. 3, pp. 137-150, September 2015.

[BAB+23] S. Bartoletti, C. S. Álvarez-Merino, R. Barco, *et al.*, "Positioning methods". Positioning and Location-based Analytics in 5G and Beyond, pp. 19-50, September 2023.

[Beb19] B. Bebensee, "Local differential privacy: A tutorial". In arXiv preprint arXiv:1907.11908, July 2019.

[BLZ+21] T. Bai, J. Luo, J. Zhao, B. Wen and Q. Wang, "Recent advances in adversarial training for adversarial robustness". In Proceedings of the 30th International Joint Conference on Artificial Intelligence, pp. 4312-4321, August 2021.

[BPS19] E. Bagdasaryan, O. Poursaeed and V. Shmatikov, "Differential privacy has disparate impact on model accuracy". Advances in Neural Information Processing Systems, vol. 32, pp. 15453-15462, December 2019.

[BRB18] W. Brendel, J. Rauber and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models" (*Poster*). In Proceedings of the 6th International Conference on Learning Representations, April 2018.

[BYK+23] A. Behravan, V. Yajnanarayana, M. F. Keskin, *et al.*, "Positioning and sensing in 6G: Gaps, challenges, and opportunities". IEEE Vehicular Technology Magazine, vol. 18, no. 1, pp. 40-48, March 2023.

[CC19] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey". In arXiv preprint arXiv:1901.03407, January 2019.

[CDL+17] N. Carlini, G. Deng, J. Li, C. Weinberger and D. Song, "Evaluating the robustness of neural networks: An extreme value theory approach". In arXiv preprint arXiv:1705.10718, May 2017.

[CJW20] J. Chen, M. I. Jordan and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack". In Proceedings of the 2020 IEEE Symposium on Security and Privacy, pp. 1277-1294, May 2020.

[CL24] R. Cui and Y. Liu, "Novel poisoning attacks on decentralized federated learning". Master's thesis, University of Zurich, July 2024.

[CPB13] S. Caltagirone, A. Pendergast and C. Betz, "The diamond model of intrusion analysis". Threat Connect, vol. 298, no. 0704, pp. 1-61, July 2013.

[CRK19] J. Cohen, E. Rosenfeld and Z. Kolter, "Certified adversarial robustness via randomized smoothing". In Proceedings of the 36th International Conference on Machine Learning, pp. 1310-1320, June 2019.

[CSS+22] Y. Chen, C. Shen, Y. Shen, C. Wang and Y. Zhang, "Amplifying membership exposure via data poisoning". Advances in Neural Information Processing Systems, vol. 35, pp. 29830-29844, December 2022.

[CTW+21] N. Carlini, F. Tramèr, E. Wallace, *et al.*, "Extracting training data from large language models". In Proceedings of the 30th USENIX Security Symposium (USENIX Security 2021), pp. 2633-2650, August 2021.

[DA23] U. Demirhan and A. Alkhateeb, "Integrated sensing and communication for 6G: Ten key machine learning roles". IEEE Communications Magazine, vo. 61, no. 5, pp. 113-119, May 2023.

[DAA+19] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame". Advances in Neural Information Processing Systems, vol. 32, pp. 13567-13578, December 2019.

[DAL+18] G. S. Dhillon, K. Azizzadenesheli, Z. Lipton, J. Bernstein, F. K. Hsieh and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense". In arXiv preprint arXiv:1803.01442, March 2018.

[DBB+21] C. De Lima, D. Belot, R. Berkvens, *et al.*, "Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges". IEEE Access, vol. 9, pp. 26902-26925, January 2021.

[DBJ+20] B. Dimanov, U. Bhatt, M. Jamnik and A. Weller, "You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods". In Proceedings of the ECAI 2020 – 24th European Conference on Artificial Intelligence, pp. 2473-2480, September 2020.

[ETSI TS 102 165-1] ETSI TS 102 165-1, "CYBER; Methods and protocols; Part 1: Method and pro forma for threat, vulnerability, risk analysis (TVRA) V5.2.5", January 2022.

[FBI+19] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam and I. S. Kohane, "Adversarial attacks on medical machine learning". Science, vol. 363, no. 6433, pp. 1287-1289, March 2019.

[FJR15] M. Fredrikson, S. Jha and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures". In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1322-1333, October 2015.

[FLJ+14] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing". In Proceedings of the 23rd USENIX Security Symposium, pp. 17-32, August 2014.

[GA19] D. Gunning and D. W. Aha, "DARPA's explainable artificial intelligence (XAI) program". AI Magazine, vol. 40, no. 2, pp. 44-58, Summer 2019.

[GAZ19] A. Ghorbani, A. Abid and J. Zou, "Interpretation of neural networks is fragile". In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 3681-3688, July 2019.

[GBD+20] J. Geiping, H. Bauermeister, H. Dröge and M. Moeller, "Inverting gradients - How easy is it to break privacy in federated learning?". Advances in Neural Information Processing Systems, vol. 33, pp. 16937-16947, December 2020.

[GHM23] M. Girdhar, J. Hong and J. Moore, "Cybersecurity of autonomous vehicles: A systematic literature review of adversarial attacks and defense models". IEEE Open Journal of Vehicular Technology, vol. 4, pp. 417-437, April 2023.

[GLG17] T. Gu, K. Liu, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain". In arXiv preprint arXiv:1708.06733, August 2017.

[GSS15] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples" (*Poster*). In Proceedings of the 3rd International Conference on Learning Representations, May 2015.

[GTN+24] U. Gülen, Ö. F. Tuna, B. Nour, Z. Laaroussi, L. Karaçay and F. Karakoç, "Threat modeling of AI-as-a-Service framework". In Proceeding of the 20th International Conference on Wireless and Mobile Computing, Networking and Communications, October 2024.

[GWY+18] K. Ganju, Q. Wang, W. Yang, C. A. Gunter and N. Borisov, “Property inference attacks on fully connected neural networks using permutation invariant representations”. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 619-633, October 2018.

[GYM+21] J. Gou, B. Yu, S. J. Maybank and D. Tao, “Knowledge distillation: A survey”. International Journal of Computer Vision, vol. 129, no. 6, pp. 1789-1819, January 2021.

[HAP17] B. Hitaj, G. Ateniese and F. Pérez-Cruz, “Deep models under the GAN: Information leakage from collaborative deep learning”. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 603-618, October 2017.

[HBP20] F. Harder, M. Bauer and M. Park, “Interpretable and differentially private predictions”. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, pp. 4083-4090, February 2020.

[HEA16-D2] HEAVENS project consortium, “Deliverable D2: Security models”, March 2016.

[HGA+24] P. Huang, E. Gönültaş, M. Arnold, K. P. Srinath, J. Hoydis and C. Studer, “Attacking and defending deep-learning-based off-device wireless positioning systems”. IEEE Transactions on Wireless Communications, vol. 23, no. 8, pp. 8883-8895, August 2024.

[HGD15] G. E. Hinton, O. Vinyals and J. Dean, “Distilling the knowledge in a neural network”. In arXiv preprint arXiv:1503.02531, March 2015.

[HGL+20] R. Hu, Y. Guo, H. Li, Q. Pei and Y. Gong, “Personalized federated learning with differential privacy”. IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9530-9539, January 2020.

[HGS+21] Y. Huang, S. Gupta, Z. Song, K. Li and S. Arora, “Evaluating gradient inversion attacks and defenses in federated learning”. Advances in Neural Information Processing Systems, vol. 34, pp. 7232-7241, December 2021.

[HJT19] J. Heo, S. Joo and T. Moon, “Fooling neural network interpretations via adversarial model manipulation”. Advances in Neural Information Processing Systems, vol. 32, pp. 2921-2932, December 2019.

[HSD+22] S. Helal, H. Sarieddeen, H. Dahrouj, T. Y. Al-Naffouri and M. S. Alouini, “Signal processing and machine learning techniques for terahertz sensing: An overview”. IEEE Signal Processing Magazine, vol. 39, no. 5, pp. 42-62, September 2022.

[HZS+24] H. Hu, X. Zhang, Z. Salcic, L. Sun, K.-K. R. Choo and G. Dobbie, “Source inference attacks: Beyond membership inference attacks in federated learning”. IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 4, pp. 3012-3029, July-Aug. 2024.

[IEM19] A. Ilyas, L. Engstrom and A. Madry, “Prior convictions: Black-box adversarial attacks with bandits and priors” (*Poster*). In Proceedings of the 7th International Conference on Learning Representations, May 2019.

[JK23] E. Jeong and M. Kountouris, “Personalized decentralized federated learning with knowledge distillation”. In Proceedings of the ICC 2023 – IEEE International Conference on Communications, pp. 1982-1987, May 2023.

[JSL+21] L. Jiao, G. Sun, J. Le and K. Zeng, “Machine learning-assisted wireless PHY key generation with reconfigurable intelligent surfaces”. In Proceedings of the 3rd ACM Workshop on Wireless Security and Machine Learning, pp. 61-66, June 2021.

[KKA+24] M. A. Khan, N. Kumar, S. H. Alsamhi, G. Barb, J. Zywiołek and I. Ullah, “Security and privacy issues and solutions for UAVs in B5G networks: A review”. IEEE Transactions on Network and Service Management, Early Access, October 2024.

[KL20] A. Kuppa and N.-A. Le-Khac, “Black box attacks on explainable artificial intelligence (XAI) methods in cyber security”. In Proceedings of the 2020 International Joint Conference on Neural Networks, pp. 1-8, September 2020.

[KRC+24] W. Khalid, M. A. U. Rehman, T. V. Chien, Z. Kaleem, H. Lee and H. Yu, “Reconfigurable intelligent surface for physical layer security in 6G-IoT: Designs, issues, and advances”. IEEE Internet of Things Journal, vol. 11, no. 2, pp. 3599-3613, January 2024.

[KSM+23] T. Kim, S. Singh, N. Madaan and C. Joe-Wong, “Characterizing internal evasion attacks in federated learning”. In Proceedings of the 26th International Conference on Artificial Intelligence and Statistics, pp. 907-921, April 2023.

[LCM+22] F. Liu, Y. Cui, C. Masouros, *et al.*, “Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond”. IEEE Journal on Selected Areas in Communications, vol. 40, no. 6, pp. 1728-1767, June 2022.

[LCY+24] L. Lu, M. Chen, J. Yu, *et al.*, “An imperceptible eavesdropping attack on WiFi sensing systems”. IEEE/ACM Transactions on Networking, vol. 32, no. 5, pp. 4009-4024, October 2024.

[LFM99] M. Van Lent, W. Fisher, and M. Mancuso, “An explainable artificial intelligence system for small-unit tactical behavior”. In Proceedings of the 19th National Conference on Artificial Intelligence, pp. 900-907, July 2004.

[LLM+21] Liu, Y., Liu, X., Mu, X., Hou, T., Xu, J., Di Renzo, M., & Al-Dhahir, N. (2021). Reconfigurable intelligent surfaces: Principles and opportunities. *IEEE communications surveys & tutorials*, *23*(3), 1546-1577.

[LLY+23] H. Liu, Z. Lin, X. Yuan and Y.-J. A. Zhang: “Reconfigurable intelligent surface empowered over-the-air federated edge learning”. IEEE Wireless Communications, vol. 30, no. 6, pp. 111-118, December 2023.

[LPL+24] J. Liang, R. Pang, C. Li and T. Wang, “Model extraction attacks revisited”. In Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, pp. 1231-1245, July 2024.

[LXD+24] C. Li, M. Xu, Y. Du, *et al.*, “Practical adversarial attack on WiFi sensing through unnoticeable communication packet perturbation”. In Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, pp. 373-387, May 2024.

[LYY20] L. Lyu, H. Yu and Q. Yang, “Threats to federated learning: A survey”. In arXiv preprint arXiv:2003.02133, March 2020.

[LZB+22] Y. Liu, Z. Zhao, M. Backes and Y. Zhang, “Membership inference attacks by exploiting loss trajectory”. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 2085-2098, November 2022.

[MCC+23] F. Meneghello, C. Chen, C. Cordeiro and F. Restuccia, “Toward integrated sensing and communications in IEEE 802.11 bf Wi-Fi networks”. IEEE Communications Magazine, vol. 61, no. 7, pp. 128-133, July 2023.

[MGF+17] J. H. Metzen, T. Genewein, V. Fischer and B. Bischoff, “On detecting adversarial perturbations” (*Poster*). In Proceedings of the 5th International Conference on Learning Representations, April 2017.

[MLW+23] C. Ma, J. Li, K. Wei, *et al.*, “Trusted AI in multiagent systems: An overview of privacy and security for distributed learning”. In Proceedings of the IEEE, vol. 111, no. 9, pp. 1097-1132, September 2023.

[MMS+18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks”. In Proceedings of the 6th International Conference on Learning Representations, April 2018.

[MOF22] J. Maroto, G. Ortiz-Jiménez and P. Frossard, “On the benefits of knowledge distillation for adversarial robustness”. In arXiv preprint arXiv:2203.07159, March 2022.

[MQS+23] E. T. Martínez Beltrán, M. Quiles Pérez, P. M. Sánchez Sánchez, *et al.*, “Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges”. IEEE Communications Surveys & Tutorials, vol. 25, no. 4, pp. 2983-3013, Fourthquarter 2023.

[MSB+23] F. Mitchell, P. Smith, A. Bhaskara and S. K. Kasera, “Exploring adversarial attacks on learning-based localization”. In Proceedings of the 2023 ACM Workshop on Wireless Security and Machine Learning, pp. 15-20, June 2023.

[MSC+19] L. Melis, C. Song, E. De Cristofaro, V. Shmatikov, “Exploiting unintended feature leakage in collaborative learning”. In Proceedings of the 2019 IEEE Symposium on Security and Privacy, pp. 691-706, May 2019.

[MZL+23] X. Meng, J. Zhou, X. Liu, X. Tong, W. Qu and J. Wang, “Secur-Fi: A secure wireless sensing system based on commercial Wi-Fi devices”. Proceedings of the IEEE INFOCOM 2023 - IEEE Conference on Computer Communications, pp. 1-10, May 2023.

[NLP+23] T. Nguyen, P. Lai, H. Phan and M. T. Thai, “Xrand: Differentially private defense against explanation-guided attacks”. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, vol. 37, pp. 11873-11881, February 2023.

[NLT+23] T. D. T. Nguyen, P. Lai, K. Tran, N. Phan and M. T. Thai, “Active membership inference attack under local differential privacy in federated learning”. In Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 5714-5730, April 2023.

[OSF19] S. J. Oh, B. Schiele and M. Fritz, “Towards reverse-engineering black-box neural networks”. In Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Lecture Notes in Computer Science vol. 11700, pp. 121-144, July 2019.

[PL23] P. Porambage and M. Liyanage, “Security and privacy vision in 6G: A comprehensive guide”. John Wiley & Sons, July 2023.

[PMW+16] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks”. In Proceedings of the 2016 IEEE Symposium on Security and Privacy, pp. 582-597, May 2016.

[Pot09] B. Potter, “Microsoft SDL threat modelling tool”. Network Security, vol. 2009, no. 1, pp. 15-18, January 2009.

[PPS+18] S. Park, J.-K. Park, S.-J. Shin and I.-C. Moon, “Adversarial dropout for supervised and semi-supervised learning”. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 3917-3924, February 2018.

[QZZ+24] L. Qin, T. Zhu, W. Zhou and P. S. Yu, “Knowledge distillation in federated learning: A survey on long lasting challenges and new solutions”. In arXiv preprint arXiv:2406.10861, June 2024.

[RD18] A. Ross and F. Doshi-Velez, “Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients”. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 1660-1669, February 2018.

[RJL+23] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera and E. Martínez-Cámara, “Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges”. Information Fusion, vol. 90, pp. 148-173, February 2023.

[ROB24-D21] ROBUST-6G project consortium, “Deliverable D2.1: 6G Threat Analysis Report”, June 2024.

[SKM+24] A. S. de Sena, J. Kibilda, N. H. Mahmood, A. Gomes and M. Latva-aho, “Malicious RIS versus massive MIMO: Securing multiple access against RIS-based jamming attacks”. In arXiv preprint arXiv:2401.07149, January 2024.

[SQG+23] J. Song, C. Qian, Y. Guo, K. Hua and W. Yu, “Attack evaluations of deep learning empowered WiFi sensing in IoT systems”. Proceedings of the IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops, pp. 1-6, May 2023.

[RSL18] A. Raghunathan, J. Steinhardt and P. Liang, “Certified defenses against adversarial examples” (*Poster*). In Proceedings of the 6th International Conference on Learning Representations, April 2018.

[SHJ+20] D. Slack, S. Hilgard, E. Jia, S. Singh and H. Lakkaraju, “Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods”. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 180-186, February 2020.

[SS15] R. Shokri and V. Shmatikov, “Privacy-preserving deep learning”. In Proceedings of the 53rd Annual Allerton Conference on Communication, Control, and Computing, pp. 909-910, September 2015.

[SSL+24] T. Senevirathna, B. Siniarski, M. Liyanage and S. Wang, “Deceiving post-hoc explainable AI (XAI) methods in network intrusion detection”. In Proceedings of the 2024 IEEE 21st Consumer Communications & Networking Conference, pp. 107-112, January 2024.

[SSS+17] R. Shokri, M. Stronati, C. Song and V. Shmatikov, “Membership inference attacks against machine learning models”. In Proceedings of the 2017 IEEE Symposium on Security and Privacy, pp. pp. 3-18, May 2017.

[SSW+24a] C. Sandeepa, B. Siniarski, S. Wang and M. Liyanage, “SHERPA: Explainable robust algorithms for privacy-preserved federated learning in future networks to defend against data poisoning attacks”. In Proceedings of the 2024 IEEE Symposium on Security and Privacy, pp. 204-204, May 2024.

[SSW+24b] C. Sandeepa, B. Siniarski, S. Wang and M. Liyanage, “Rec-Def: A recommendation-based defence mechanism for privacy preservation in federated learning systems”. IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 2716-2728, February 2024.

[SSZ21] R. Shokri, M. Strobel and Y. Zick, “On the privacy risks of model explanations”. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pp. 231-241, May 2021.

[SVB+22] A. Shastri, N. Valecha, E. Bashirov, *et al.*, “A review of millimeter wave device-based localization and device-free sensing technologies and applications”. IEEE Communications Surveys & Tutorials, vol. 24, no. 3, pp. 1708-1749, thirdquarter 2022.

[SZS+14] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, “Intriguing properties of neural networks” (*Poster*). In Proceedings of the 2nd International Conference on Learning Representations, April 2014.

[TK24] Ö. F. Tuna and F. E. Kadan, “Security of AI-driven beam selection for distributed MIMO in an adversarial setting”. IEEE Access, vol. 12, pp. 42028-42041, March 2024.

[TKG24] Ö. F. Tuna, L. Karaçay and U. Gülen, “A novel method to mitigate adversarial attacks against AI-as-a-Service functionality”. In Proceeding of the IEEE Middle East Conference on Communications and Networking, November 2024.

[TKK+23] Ö. F. Tuna, F. E. Kadan and L. Karaçay, “Practical adversarial attacks against AI-driven power allocation in a distributed MIMO network”. In Proceedings of the ICC 2023 – IEEE International Conference on Communications, pp. 759-764, May 2023.

[TKP+18] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh and P. D. McDaniel, “Ensemble adversarial training: Attacks and defenses” (*Poster*). In Proceedings of the 6th International Conference on Learning Representations, April 2018.

[TLC+20] S. Truex, L. Liu, K. H. Chow, M. E. Gursoy and W. Wei, “LDP-Fed: Federated learning with local differential privacy”. In Proceedings of the 3rd International Workshop on Edge Systems, Analytics and Networking, pp. 61-66, April 2020.

[TLG+19] S. Truex, L. Liu, M. E. Gursoy, L. Yu and W. Wei, “Demystifying membership inference attacks in machine learning as a service”. IEEE Transactions on Services Computing, vol. 14, no. 6, pp. 2073-2089, February 2019.

[TZJ+16] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter and T. Ristenpart, “Stealing machine learning models via prediction APIS”. In Proceedings of the 25th USENIX Security Symposium, pp. 601-618, August 2016.

[WCG+23] R. Wu, X. Chen, C. Guo and K. Q. Weinberger, “Learning to invert: Simple adaptive attacks for gradient inversion in federated learning”. In Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence, vol. 216, pp. 2293-2303, July 2023.

[WG18] B. Wang and N. Z. Gong, “Stealing hyperparameters in machine learning”. In Proceedings of the 2018 IEEE Symposium on Security and Privacy, pp. 36-52, 2018.

[WHS+22] Z. Wang, Y. Huang, M. Song, L. Wu, F. Xue and K. Ren, “Poisoning-assisted property inference attack against federated learning”. IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 4, pp. 3328-3340, August 2022.

[WLZ+22] Wang, Yazheng, et al. "Wireless communication in the presence of illegal reconfigurable intelligent surface: Signal leakage and interference attack." *IEEE Wireless Communications* 29.3 (2022): 131-138.

[WQW+23] Z. Wei, H. Qu, Y. Wang, *et al.*, “Integrated sensing and communication signals toward 5G-A and 6G: A survey”. IEEE Internet of Things Journal, vol. 10, no. 13, pp. 11068-11092, July 2023.

[WSZ+19] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang and H. Qi, “Beyond inferring class representatives: User-level privacy leakage from federated learning”. In Proceedings of the 2019 IEEE Conference on Computer Communications, pp. 2512-2520, April 2019.

[Wyn14] J. Wynn, “Threat assessment and remediation analysis (TARA)”. The MITRE Corporation, Case Number 14-2359, October 2014.

[XWZ+17] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie and A. Yuille, “Mitigating adversarial effects through randomization”. In arXiv preprint arXiv:1711.01991, November 2017.

[XHH+22] J. Xu, C. Hong, J. Huang, L. Y. Chen and J. Decouchant, “AGIC: Approximate gradient inversion attack on federated learning”. In Proceedings of the 41st International Symposium on Reliable Distributed Systems, pp. 12-22, September 2022.

[XXQ+20] Y. Xu, Y. Xu, Q. Qian, H. Li and R. Jin, “Towards understanding label smoothing”. In arXiv preprint arXiv:2006.11653, June 2020.

[YSS+21] A. Yousefpour, I. Shilov, A. Sablayrolles, *et al.*, “Opacus: User-friendly differential privacy library in PyTorch”. In arXiv preprint arXiv:2109.12298, September 2021.

[ZDK+21] L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani and J. Zou, “How does Mixup help with robustness and generalization?”. In arXiv preprint arXiv:2010.04819, March 2021.

[ZLH19] L. Zhu, Z. Liu and S. Han, “Deep leakage from gradients”. Advances in Neural Information Processing Systems, vol. 32, pp. 14747-14756, December 2019.

[ZLJ+21] S. Zhang, M. Li, M. Jian, Y. Zhao and F. Gao, “AIRIS: Artificial intelligence enhanced signal processing in reconfigurable intelligent surface communications”. China Communications, vol. 18, no. 7, pp. 158-171, July 2021.

[ZSE+24] J. C. Zhao, A. Sharma, A. R. Elkordy, Y. H. Ezzeldin, S. Avestimehr and S. Bagchi, “Loki: Large-scale data reconstruction attack against federated learning through model manipulation”. In Proceedings of the 2024 IEEE Symposium on Security and Privacy, pp. 1287-1305, May 2024.

[ZWS+20] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo and T. Wang, “Interpretable deep learning under fire”. In Proceedings of the 29th USENIX Security Symposium, USENIX Security 2020, pp. 1659-1676, August 2020.

[ZZC+20] J. Zhang, J. Zhang, J. Chen and S. Yu, “GAN enhanced membership inference: A passive local attack in federated learning”. In Proceedings of the ICC 2020 – 2020 IEEE International Conference on Communications, pp. 1-6, June 2020.