LAZARUS

DELIVERABLE

D3.4 AI Techniques Performance, Reliability, Security

Project Acronym:	LAZARUS	
Project title:	pLatform for Analysis of Resilient and secUre Software	
Grant Agreement No.	101070303	
Website:	https://lazarus-he.eu/	
Contact:	info@lazarus-he.eu	
Version:	1.0	
Date:	25/04/2023	
Responsible Partner:	UCM	
Contributing Partners:	UCM, ARC, UNIPD, DC, LIST, BNR, MAG	
Reviewers:	C. Patsakis (ARC) F. Casino (ARC) N. Lykousas (DC)	
Dissemination Level:	Public	X
	Confidential – only consortium members and European Commission Services	

This project has received financial support from the European Union Horizon 2020 Programme under grant agreement no. 101070303.

Revision History

Revision	Date	Author	Organization	Description
0.1	09/01/2023	Luis Alberto Martínez Hernández Sandra Pérez Arteaga Sara Vanesa Orozco Narváez Luis Javier García Villalba	UCM	Table of contents
0.2	20/01/2023	Luis Alberto Martínez Hernández Sandra Pérez Arteaga Sara Vanesa Orozco Narváez Luis Javier García Villalba	UCM	Initial contributions
0.3	17/02/2023	Various authors	ARC, UNIPD, DC, LIST, BNR, MAG	Initial partner contributions
0.4	16/03/2023	Luis Alberto Martínez Hernández Sandra Pérez Arteaga Sara Vanesa Orozco Narváez Luis Javier García Villalba	UCM	First draft
0.5	04/04/2023	Various authors	UCM, ARC, UNIPD, DC, LIST, BNR, MAG	Second draft
0.6	18/04/2023	Luis Alberto Martínez Hernández Sandra Pérez Arteaga Sara Vanesa Orozco Narváez Luis Javier García Villalba	UCM	Version for review
0.7	21/04/2023	Constantinos Patsakis, F. Casino, N. Lykousass	ARC, LIST	Review comments
0.8	28/04/2023	Luis Alberto Martínez Hernández Sandra Pérez Arteaga Sara Vanesa Orozco Narváez Luis Javier García Villalba	UCM	Integration of changes to address review comments
1.0	29/04/2023	Luis Javier García Villalba	UCM	Final version

Table of Contents

List of Tables

List of Figures

1 Executive Summary

Artificial Intelligence (AI) techniques have made great progress in recent years, allowing users and organizations to provide intelligent functions to their developments and automate processes that require great attention from a person. Machine learning (ML) algorithms such as neural networks, decision trees, support vector machines, and recently Transformers have had remarkable success thanks to their wide range in various domains, such as computer vision, natural language processing, and source code analysis.

In terms of performance, AI algorithms can increase the reliability by reducing human risk and improving the accuracy and interpretability of results. They can detect patterns and anomalies that are difficult to detect with traditional methods and provide predictive maintenance to prevent equipment breakdowns.

In terms of security, AI techniques can help organisations detect and prevent cyber threats by analysing large volumes of data and identifying anomalous behaviour patterns. They can also provide continuous monitoring and threat detection in real-time, reducing the risk of data breaches and other security incidents.

However, AI techniques pose several challenges, including concerns about privacy and data security or the models that process this data. Organisations must ensure that their AI systems comply with regulatory standards to protect people's privacy and avoid any breach that allows an attacker to gain confidential user data.

In this deliverable, we study some attack and defence techniques towards Machine Learning algorithms which will be used to develop a tool capable of measuring the resilience of the tools generated in tasks T3.2 and T3.4 of the project. The development of the architecture will be focused on the creation of Generative Adversarial Networks that will allow the generation of adversarial samples by passing them off as benign samples.

In conclusion, AI techniques have the potential to improve performance, reliability, and security in a wide range of applications, but organisations need to be aware of potential security issues and take steps to mitigate these risks.

2 Introduction

AI has revolutionised the operations of organisations by providing powerful tools to improve processes, automate tasks and avoid human errors when processing data. Thanks to its ability to analyse large volumes of data, identify patterns and make predictions, AI has become a game changer in various industries, including finance, healthcare, manufacturing, and cybersecurity.

AI has notably succeeded in various applications, such as voice and image recognition, natural language processing, web threat detection, video processing, etc. Etc. Moreover, thanks to these techniques, it is possible to have services that enable autonomous car driving, virtual assistants, chats with fluid conversations, etc.

Regarding security, AI techniques can detect and prevent cyber threats by analysing large volumes of data and identifying anomalous behaviour patterns. They can also provide continuous monitoring and real-time threat detection, reducing the risk of data breaches and other security incidents. However, as with any powerful technology, AI techniques also pose several challenges, such as model security, the privacy of customer data and the reliability of results. For example, ML models can be attacked by malicious data that alters their operation and produce wrong results. Therefore, organisations must take appropriate security measures to protect their AI systems and the involved data. This includes implementing robust security protocols, identifying and mitigating vulnerabilities, and training staff on security and privacy.

According to T3.4, Design and development of AI models to measure LAZARUS' resilience, deep learning models, such as Generative Adversarial Networks (GANs), will be developed to assess the security, detect failures and determine the level of robustness of the AI techniques implemented in tasks T3.2, T3.3, against "*adversarial attacks*''. The models will simulate attack and defence by creating multiple perturbations for a specific input. The execution of this task will lead to robust AI models and tools that provide reliable results. It will also help to perform synthetic load testing by generating and mutating various abnormal application traffic patterns. In this sense, this deliverable lists the most popular techniques and attack methods that an attacker must employ to cause models to malfunction or compromise user data. In addition, some techniques to defend algorithms against data leakage through cryptography, artificial intelligence models to counter the most popular attacks or hardware-based protection systems will be reviewed.

The rest of the deliverable is organised as follows: in Section 3, the adversarial threat model is shown where the main techniques and attack surface that an adversary must can take over an organisation's data or model are listed. Section 4 shows the most relevant attacks on ML models. Section 5 lists the most popular defences for protecting AI models. Section 6 discusses the generative modelling and Antagonistic Generative Networks, in section 7 a general architecture of the tare 3.4 tool based on the revised documentation is shown, finally, section 8 shows the concludes the deliverable.

3. Adversarial Threat Model

3.1 Attack Surface on Artificial Intelligence

Any system that uses ML models has basic operating characteristics that are executed sequentially. The process starts with data collection for model training based on the purpose to be given to the ML model, then the process of data analysis and feature engineering are performed, then the selection of features that will serve as input for training a ML model is performed and finally an action is taken based on the model output (see Figure 1).

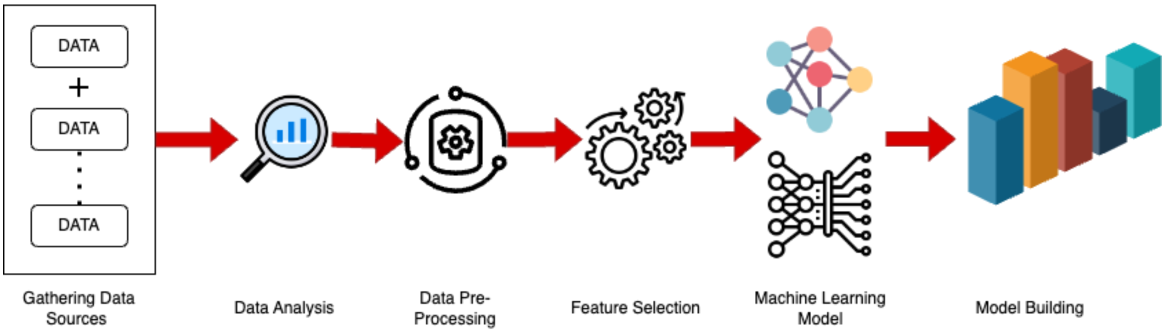

Figure 1: Machine Learning Pipeline

Since the main objective of an attacker is to influence the output of data to corrupt the base target of a model, the attack surface towards these types of environments can be based on the different stages of data processing depending on the actual goal of the attacker. The main attack surfaces are evasion, poisoning, and exploratory attacks.

- **Evasion Attacks** [1]**:** This type of attack is performed during model inference. And it refers to generating an input that contains subtly altered data that at first glance seems normal, while the model classifies it erroneously.
- **Poisoning Attacks** [2]**:** This attack, compared to the evasion attack, is carried out at the time of training, which bases its operation on the injection of designed samples that alter the classifier's decision. The main objective of this attack is to decrease the accuracy of the model [2]. In this attack, an adversary has no access to the model or the initial data set, he can only add new data to the initial data set or modify it.
- **Exploratory Attacks** [1]**:** This attack, tries to obtain the maximum knowledge of the ML model and the pattern of the training data. This attack is usually carried out once the model has been trained and is up and running.
- **Trojans** [2]**:** In the Trojaning technique, unlike the poisoning technique, an attacker does not have access to the initial data set but has access to the model and its parameters and can retrain the model. This type of attack usually happens when organisations open-source pre-trained models where hackers can substitute their own modified versions. The basic idea of Trojaning is to discover new ways to change the behaviour of the model in some specific circumstances so that its base behaviour remains unchanged.

[1] Data driven exploratory attacks on black box classifiers in adversarial domains - https://doi.org/10.1016/j.neucom.2018.02.007

[22] Threats on Machine Learning Technique by Data Poisoning Attack: A Survey - https://link.springer.com/chapter/10.1007/978-981-16-8059-5_36

- **Backdoors[3] :** The main goal of this type of attack is not just to inject some additional behaviour, but to do it in such a way that the backdoor works after retraining the system, allowing a persistent attack.

3.2 Adversarial Capabilities

Currently, attackers have different amounts of information available about the system, including the attack vector used in the threat surface.

3.2.1 Training Phase

Attacks in the training phase, a perpetrator tries to modify a model through partial or total manipulation of the training data [3], and the attacker's capabilities for this type of attack fall into the following categories:

- **Data injection:** In this category, the attacker does not have access to the training data or the ML algorithms. However, he has the possibility of poisoning the training data, and adding new data to the dataset. The main objective of the adversary is for an organisation to train the wrong model.
- **Data modification:** Unlike data injection, the attacker has full access to the training data but not to the ML algorithm. The attacker seeks to enter bad samples into the dataset before it is used to train another model.
- **Logical corruption:** This category has more privileges than the previous two and is one of the most difficult attack categories to stop. The adversary has access to the target model, and it modifies the algorithm influencing the inference.
- **Label Modification:** This attack only allows modification of data set labels for arbitrary data points.

3.2.2 Testing Phase

Runtime attacks do not directly interfere with the model, but modifications are made to produce erroneous results [4]. An important point to consider is the amount of knowledge an attacker has about the target model; this will determine how easy it is to affect its decisions. These attacks are divided into three groups (White-Box, Grey-Box and Black-Box). Table 1 shows a summary of the different investigations related to the abilities that attackers have to violate ML models, taking into account the knowledge they have about the infrastructure, model, inputs, etc.

3.2.2.1 White-Box

In this type of attack, an attacker has full knowledge of the target model, including training data, architectures, and parameters. To achieve a model malfunction, an attacker must study the behaviour of the model and understand the vulnerabilities of the model, e.g., with which input data an erroneous output is produced.

This type of attack has been widely used. For example, in [5] an attack framework based on Universal Adversarial Disturbances (UAP) is proposed, specifically UAPs based on DeepFool[4] and UAPs based on Total Loss Minimization for physical rehabilitation applications, which were applied in the EMG gesture recognition network based on convolutional neural networks (CNN). In [6], a white-box adversarial attack is proposed, this proposal allows to attack ML models with a slight perturbation, in addition to breaking the TRADES[5]

[3] Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses - https://ieeexplore.ieee.org/abstract/document/9743317

[4] DeepFool: a simple and accurate method to fool deep neural networks - https://arxiv.org/abs/1511.04599

[5] Theoretically Principled Trade-off between Robustness and Accuracy - https://arxiv.org/abs/1901.08573

adversarial training models with the highest success rate. Likewise, the generated approach can reduce the accuracy of black box models between 16% and 31% in the black box transfer attack.

Due to the extensive use of adversarial attacks threatening ML models by adding imperceptible perturbations, Wang *et al.* [7] proposes a method to reduce the amount of unnecessary information for the generation of integrated gradient-based adverse examples (IWA): Integrated Finite Point Attack Algorithm (IFPA) and Integrated Universe Attack Algorithm (IUA). IFPA is suitable when there is a predetermined number of disturbance points and IUA is used when there is no predetermined number of disturbance points to obtain more adversarial examples. In the experiments it can be validated that it is possible to generate adversarial examples in all types of data sets with fewer perturbations and a better generation rate.

Also in [8], it is taken into account that they are vulnerable to small perturbations in their input data. Therefore, a method is proposed for the data of samples close to the trained classifier's decision limits, thus identifying possible adverse samples using support vectors of an SVM model trained on the characteristics of DCNN.

3.2.2.2 Grey-Box

Compared to white-box attacks, the attacker knows specific information about the system such as the architecture. Considering that most organisations use public architectures such as Google to support their infrastructure, collecting information from black-box systems by sending inputs and analysing the outputs to learn about the model and train a surrogate model and send attacks to it is possible. However, in [9] it is shown that it is possible to collect information from black box systems, sending inputs and analysing the outputs to know the model and train a surrogate model and send attacks to it to know what the security holes are or where the accuracy of the model falls and send an attack.

3.2.2.3 Black-Box

Unlike the two attacks seen above, black box attacks do not assume any knowledge about the model, its configuration or the technologies that support it [10]. To explore it, they make use of the configuration information and inputs available to explore the model. The effectiveness of this type of attack is based on the attacker's ability to intuit the internal structure of the target model by using the inputs and analysing the outputs.

A critical study presented in [11] demonstrates in a practical way that an attacker can take control of a DNN located on a remote server without knowing the inner workings. The approach is to train a local model that replaces the target network only by having the ability to observe the labels in the network inputs. The proposal shows that the target network performs 84.24% misclassification when entering adverse examples previously generated by the local surrogate. The attack strategy may be able to evade defence strategies to hinder the creation of adverse examples.

Today, some solutions allow critical tasks such as malware detection in an organisation to be carried out by using AI algorithms. These types of solutions are the most attractive to be attacked so that modified malware samples are considered as valid samples. It should be noted that most of the existing malware detection proposals focus on detection algorithms using fixed-dimensional features and are gradually adopting recurrent neural networks (RNN). In [12], an algorithm for generating sequential adversarial examples to attack RNN-based systems is proposed, the generative network is based on the sequence-by-sequence model and a surrogate RNN that matches the victim's RNN is trained, using Gumbel-Softmax to approximate the generated samples. As a result, algorithms based on RNNs fail to detect most of the malicious samples generated by the proposal.

In [13], a proposal is made that uses white-box attack methods (C&W and Deepfool) to generate samples for queries for target network information gathering and proposes a diversity criterion to avoid sampling bias. Experimental results on MNIST and CIFAR-10 show that the proposed method reduces more than 90% of the queries needed to obtain target information significantly increasing the success rate.

In addition, Zhao *et al.* [14] propose performing black-box attacks on suspicious network flow detection algorithms based on ML. This proposal like the previous ones uses an objective learning model substitution approach using the KDD99 and CSE-CIC-IDS2018 dataset by extending the BIM algorithm, based on model understanding in image generation and IDS. The samples generated by the tools emit elude the detection of the target model.

In addition to performing attacks on ML algorithms, it has also been considered to mitigate black box attacks on these solutions; for example, in [15], a method to defend against adversarial perturbations using GANs is proposed. Experiments were performed using classification algorithms such as random forest, and principal component analysis (PCA) and recursive feature elimination to reduce the dimensionality of the dataset, improving the performance of the model and it was shown that adversarial training based on GANs improves the resilience of the model and was able to mitigate black-box attacks. In [16], a study on the security of black-box attack detectors is conducted using a realistic threat model by multiple approximation of samples collected from the attacker's side for query reduction and threshold understanding for anomaly detection.

Articles	Type	Attacks/Defences	Algorithm Target	Application
Xue *et al.* [5]	White	CNN-based myoelectric control system.	CNN	Attacks against classifiers
Wang *et al.* [6]	White	Adversarial samples crafting	DNN	Attacks against classifiers
Wang *et al.* [7]	White	Adversarial samples crafting	DNN	Attacks against classifiers
Nazemi *et al.* [8]	White	Adversarial samples crafting	DCNN	Attacks against classifiers
Papernot *et al.* [9]	Gray	Adversarial samples crafting, adversarial sample transferability	DNN	Digit recognition, black-box attacks against classifiers
Papernot *et al.* [11]	Black	Training a local model to substitute for the target, Adversarial samples crafting	DNN	Control a remotely hosted DNN without knowledge
Hu *et al.* [12]	Black	Adversarial samples crafting	RNN	Attack a RNN based malware detection system
Li *et al.* [13]	Black	Train a substitute model based on the information queried from the target	DNN	Attacks against classifiers
Zhao *et al.* [14]	Black	Learning model substitution	IDS	Suspicious network flow detection
Guo *et al*. [15]	Black	Train a substitute model based on the information queried from the target, Adversarial samples crafting	CNN, SVM, KNN MLP, ResNet	Attack image classification and Network traffic classification
Alahmed *et al.* [16]	Black	Defence strategy to offer better protection for the system against adversarial perturbations	Random Forest, GANs	Defensive Model against adversarial perturbations

Table 1: Summary of Research on the Attack Capabilities of Machine Learning Algorithms

4 Attacks

ML models are exposed to many unique attack vectors that are low risk for other software. For most known attacks on ML models, there are taxonomies of failure modes that expose their commonalities. These taxonomies show that many ML failure modes overlap with the failure modes of additional software. Some of the most popular attack methods are listed below.

4.1 Model Inversion Attack

Model Inversion (MI) attacks threaten models' privacy as an attacker attempts to recreate training examples by accessing the target classifier, revealing the target user attribute values and model output. This attack was first proposed by [17], in which a classifier extracts features from the training data into low-capacity models (logistic regression and a shallow MLP network). This results in the adversary being able to expose the privacy of sensitive records used for training the target model. A basic schematic of the model inversion attack is shown in Figure 2.

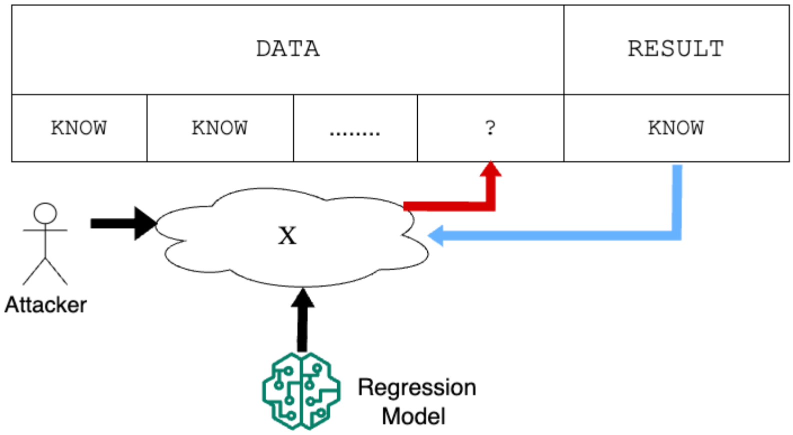

Figure 2: Basic Model Inversion Attack

Because the first study by Fredrikson focuses on drug dose prediction and it is not clear whether the proposed inversion attacks apply to other environments, in [18] an approach is developed to exploit the revealed values along with the predictions. This proposal focuses on two algorithms: decision trees and neural networks for face recognition.

In the proposal by Zhang *et al.* [19], an attack method exploits partial public information of the model to learn a distribution through generative adversarial networks and use it to guide the inversion process. As discussed, one of the main goals is to compromise the privacy of the target models, however most of them focus on privacy during the training process. In [20], describes a set of attacks that seek to compromise the privacy of inference data in collaborative deep learning systems. The idea starts from the fact that when a deep neural network and the corresponding inference task are divided and distributed among different collaborators, a malicious collaborator in which the input is retrieved even if it does not have access to the data or computations of other participants. In the work of He *et al.* the evaluation of different environments, models and datasets is carried out to demonstrate their generalisability.

In [21], a study is carried out in which use is made of architectures based on transposed multimodal CNNs for the generation of investment attacks. This proposal considers the privacy risks that can be brought about by the algorithms used by explainable Artificial Intelligence (XAI), which seek to provide more information to users to understand the decisions of the models, this method can be used to generate model inversion attacks. To understand which explanations, present a higher risk to privacy, analysis of the influence of different types of explanations and factors on investment performance is performed. Even though some models do not provide explanations, a higher return on investment is demonstrated even for non-explainable target models by exploiting surrogate model explanations through attention transfer. This method first invests an explanation from the target prediction and then reconstructs the target image. With this work they highlight the urgency of using new privacy-preserving techniques that balance the dual requirements of explainability and privacy in AI.

One of the most important points for the model inversion attack is that the adversary must query the entire data sets. However, this method is inefficient due to the transfer of large datasets to obtain the prediction values of the inference models. To solve this problem in Mo *et al.* [22], a proposal is made to reduce queries on auxiliary datasets by using Latent information as high dimensional features. This model was evaluated on CNNs using LFW, pubFig, MNIST datasets.

One way to add intelligence to inversion attacks is by using Deep Learning algorithms to better understand the different characteristics of the process and thus perform an optimisation that allows resembling the characteristics of the original model to generate an attack. In [23], an implementation of a white-box attack on facial recognition classification algorithms is presented that allows the distillation of useful knowledge to perform attacks on private models from public data using GAN networks to recover an input image.

In [24], an inversion model attack scenario is proposed under a semi-white-box scenario that has information on the structure and parameters of the target model, but user data is not available. In addition, a Deep MIA-based alert state is proposed which is the integration of generative models into MIA and α-GAN integrated MIA-initiated by a face-based seed (α-GAN-MIA-FS). Furthermore, an MIA search strategy is proposed, using a deep generative model to generate a face image from a random feature vector to reduce the image search space to the feature vector space, this allows the MIA process to efficiently search for a low-dimensional feature vector whose corresponding face image maximises the confidence score. In the proposed attack methodology, the VGGFace2 DataSet is used to train the target system and the PubFig dataset is used for evaluation. To make the training faster, the images are converted to greyscale. For the subjective evaluation 3 parameters were chosen: Naturalness, Similarity and Recognisability. The α-GAN with the Face seed vector showed the best performance in all metrics. Table 2 shows a summary of the following jobs related to Model Inversion Attack.

Articles	Attack Type	Proposed Model	Target	Dataset
Zhang *et al.* [19]	White-Box	- GMI (Generative Model-Inversion)	DNN	MNIST, ChestX-Ray8, CalebA
He *et al.* [20]	White-Box	- Inverse-Network - Shadow Model Reconstruction	DNN	MNIST, CIFAR10
Zhao *et al.* [21]	Black-Box-White-Box	- XAI-Aware	CNN	iCV-MEFED, CelebFaces, MNIST
Mo *et al.* [22]	White-Box	- Latent Information	CNN	LFW, pubFig, MNIST
Chen *et al.* [23]	White-Box	- GAN to the inversion task (GMI)	DNN	CelebA, FFHQ, FaceScrub
Khosravy *et al.* [24]	Semi-white box	- α-GAN-MIA-RS, DCGAN-MIA, Conventional MIA	CNN	VGGFace2, CalebA PubFig

Table 2: Summary of Research on Model Inversion Attack

4.1.1 Gradient Inversion Attack

The so-called gradient-based inversion attack is one of the most popular methods used to perform inversion attacks. The basic idea of this model is that if an adversary has access to the output of a ML model, it can use the gradients of the model concerning its inputs to infer what the input data was. The central idea of this type of attack is to invert the gradient of the loss function for the input. First, an independent model is trained to predict the input from the output. The loss function of this inverse model is then used to calculate the gradient for the input, which is then used to generate a perturbation that is added to the original input sample. The perturbation is chosen to maximise the original model's loss function for a given target outcome. By adding this perturbation to the original input sample, the resulting sample will produce the desired result when run through the original model (see Figure 3).

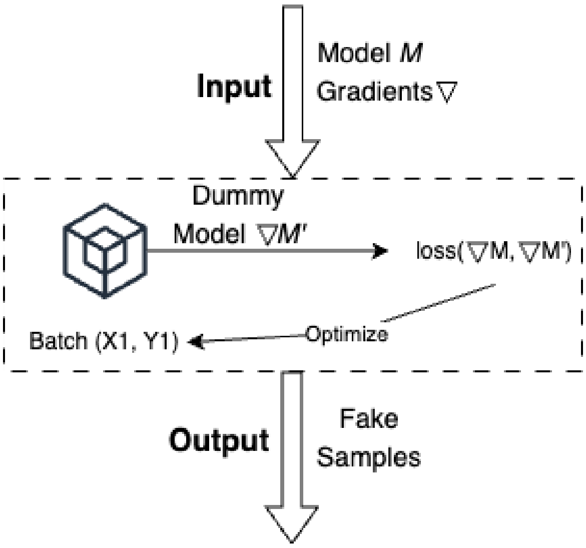

Figure 3: Gradient Inversion Attack

The gradient inversion attack can be used for a variety of purposes, such as generating adverse examples, which are input samples modified in such a way that the model misclassifies them, or for reverse engineering the model to obtain sensitive information about its parameters or training data.

One of the best technologies to train models on multiple devices or servers without centrally shared data and thus improve data privacy is federated learning[6] however, in [25], it is shown that an attacker can reconstruct the private training data from the shared gradient data by designing input regularisation for the fully connected layer, and introducing Total Variation (TV) regularisation for the convolution features, which effectively improve the fidelity of the reconstructed sample. Experiments show that the proposed method breaks the limit on the number of reconstructed samples of previous attacks and achieves higher sample fidelity. Likewise, in Kariyappa *et al.* [26], a label leakage attack towards Vertical Federated Learning (VFL) models is proposed that allows an attacker to extract private labels during the split learning process called ExPloit. The proposal frames the attack as a supervised learning problem using a new loss function that combines gradient fitting and regularisation terms developed using properties of the dataset and models. This attack allows the discovery of private labels with an accuracy of 99.96%.

A general gradient inversion attack framework that can attack FedSGD and FedAVG with improved label inference and sample restoration is shown in [27]. The proposed approach outperforms SOTA attacks by

[6] Federated learning on Google Cloud - https://cloud.google.com/architecture/federated-learning-google-cloud

attacking batches of ImageNet. Likewise, in [28], Approximate Gradient Inversion Attack (AGIC) is proposed as a scheme that efficiently and effectively reconstructs samples from both model updates and over multiple epochs. AGIC approximates the gradient updates of the training samples used from model updates, then takes advantage of the gradient/model updates collected at different epochs, and finally assigns increasing weights to the layers with respect to the neural network structure to increase the quality of the reconstruction.

Li *et al.* [29], proposes E2EGI an end-to-end gradient inversion method that can create reconstructed samples with higher similarity, and the Distributed Gradient Inversion algorithm can implement GIA with batch sizes from 8 to 256 on ML models (such as ResNet-50) and ImageNet datasets. It also creates an 81% label reconstruction algorithm on a batch sample with a label repetition rate of 96%.

Gradient-based inversion attacks are relatively common, that is why several defence schemes have been proposed for this type of attacks, as in [30], AAIA is proposed in which a secure aggregation scheme against inversion attacks for federated learning is proposed. In AAIA, gradients are encrypted before being shared to the central system, making it impossible for attackers to launch attacks. In addition, a new way of constructing shared keys is proposed by making the construction of keys shared by other clients, but not with all the clients in the system. Furthermore, [31] proposes using homomorphic encryption to encrypt gradients before sending them to the central server.

Another defence that can be implemented against gradient attacks is Gradient Perturbation which is divided into several categories:

- **Gradient pruning**: In [32], a defence is suggested by setting small magnitude gradients to zero (i.e. gradient pruning). According to their attack, they show that pruning more than 70% of the gradients would make the recovered images no longer visually recognisable. However, the suggested prune ratio is determined based on the weakest attacks and may not remain safe against the latest generation attack.
- **Adding noise to gradient:** Motivated by DPSGD that adds noise to gradients to achieve differential privacy [33], also suggests defending by adding Gaussian or Laplacian noise to the gradient. They show that successful defence requires adding a high level of noise, so that their accuracy is reduced by more than 30 % with CIFAR-10 tasks. Recent work such as [34], suggests using better pre-training techniques and a large batch size (e.g., 4.096) to achieve better accuracy for DPSGD training.

However, there are some proposals in which an evaluation of defences is performed based on existing optimisation-based attacks such as in [35], where an attack called Learning to Invert is proposed which is based on learning. In the proposal, a model is trained to learn to invert the gradient update to retrieve customer samples. Table 3 summarizes the different works listed in this section related to Gradient Inversion Attacks.

Articles	- Method	Attack/Defence	Scope
Luo *et al.* [25]	- Cosine similarity - Total variation denoising strategy	Attack	- Image
Kariyappa *et al.* [26]	- Label-leakage attack	Attack	- Federated Learning
Geng *et al.* [27]	- Gradient inversion attack framework	Attack/Defense	- Image - Federated Learning
Xu *et al.* [28]	- AGIC - Approximate Gradient Inversion Attack	Attack	- Image - Federated Learning
Li *et al.* [29]	- E2EGI - End-to-End Gradient Inversion - Minimum Loss Combinatorial Optimization	Attack	- Image - Federated Learning
Yang *et al.* [30]	- Key share phase - Aggregation phase	Defence	- Federated Learning
Zhu *et al.* [31]	- Deep Leakage from Gradient	Attack	- Computer Vision - Natural Language Processing
Abadi *et al.* [32]	- Differentially private SGD	Attack	- Computer Vision
Wei *et al.* [33]	- Gradient Leakage Attack	Attack	- Federated Learning
Tramer *et al.* [34]	- Adding noise to gradient	Defence	- Computer Vision
Wu *et al.* [35]	- Learning to Invert	Attack	- Federated Learning

Table 3: Summary of Research on Gradient Inversion Attacks

4.1.2 Reconstruction Attack

Reconstruction attacks are a type of attack on ML models that aim to deduce the inputs given to the model based on its outputs. The idea is to use the model to reconstruct sensitive or private information that was used to train or test the model, such as images, text, audio, etc.

Several techniques can be used to perform reconstruction attacks, depending on the type of model and input data:

- **Generative models:** Generative models, such as generative adversarial networks (GANs) or variational autoencoders (VAEs), can generate synthetic data similar to that used to train the model. By analysing the synthetic data and comparing it with the model output, an attacker can deduce the original input data [36] [37].
- **Optimisation-based attacks:** Optimisation-based attacks use evolutionary or gradient-based algorithms to find the input that produces the output that most closely resembles the output generated by the model. By iteratively adjusting the input based on model feedback, an attacker can find the input that best matches the model's output.
- **Inverse problem formulation:** The inverse problem formulation is a mathematical technique that models the reconstruction problem as an inverse problem, where the input is the unknown variable

and the output is the known variable. The attacker can then use optimisation or numerical techniques to solve the inverse problem and recover the input data.

- **Model-based attacks:** Model-based attacks attempt to reverse engineer the model's internal representations or parameters to deduce the input data. For example, an attacker may attempt to learn the filters or weights used by a convolutional neural network (CNN) to deduce the image used as input.

Table 4 summarizes the different works listed in this section, highlighting the attack method used and the scope of the attack.

Articles	Method	Attack / Defence	Scope
Gambs *et al.* [38]	Classifier Analysis	Attack	Decision tree classifier
Salem *et al.* [39]	hybrid generative model (CBM-GAN)	Attack	Online Learning
Chen *et al.* [40]	Attribute Reconstruction Attack	Attack	Federated Learning
Milli *et al.* [41]	Learning from input gradients	Attack	Computer Vision
Song *et al.* [42]	multi-task GAN – Auxiliary Identification (mGAN-AI)	Attack	Federated Learning
Grubbs *et al.* [43]	Query leakage	Attack	Encrypted databases

Table 4: Summary of Research on Reconstruction Attacks

In [38] the reconstruction attack is introduced whose basic objective is to reconstruct a probabilistic version of the original dataset on which a classifier has been learned from the description of this classifier and possibly some auxiliary information. Furthermore, a general attack framework is proposed that can be used to evaluate the success of a reconstruction attack in terms of a new distance between the reconstructed and original datasets.

Salem et al [39], investigate the use of the output of an ML model to filter information from the dataset used to perform the update. The paper proposes four attacks an encoder-decoder formulation, which allows inferring diverse information from the update set. A hybrid generative model (CBM-GAN) is proposed that is based on generative adversarial networks (GANs) but includes a reconstructive loss that allows reconstructing accurate samples.

In [40] a study of the attribute reconstruction attack (ARA) launched by a malicious server on the federated learning system is carried out. The paper shows that the gradients of the shared local model, after the first local training epoch, can still reveal sensitive attributes. For this work, a method based on gradient matching, cos-matching, is developed to reconstruct the sensitive attributes of the training data of any victim participant. It is also shown that an attacker can reconstruct the sensitive attributes of any record that is not included in the training data of any participant, thus opening a new attack surface in federated learning. It is also shown in [41] that gradient-based explanations of a model quickly reveal the model itself. Furthermore, an algorithm is provided that demonstrably learns a two-layer ReLU network in an environment where the algorithm can query the gradient of the model with respect to the chosen inputs. The number of queries is dimension-independent and near-optimal in its dependence on the size of the model.

In [42], a proposal is made using a framework that incorporates a GAN as a multi-task discriminator to explore user-level privacy leakage by a malicious server attack. This method is called multi-task GAN-Auxiliary

Identification (mGAN-AI) that simultaneously discriminates the category, reality, and client identity of the input samples. The discrimination of client identity allows the generator to retrieve the private data specified by the user. The method proposed by Song *et al.* works "invisibly" on the server side. In [43] a proposal is developed to construct and analyse approximate reconstruction attacks by addressing the problem of ε-approximate database reconstruction (ε-ADR) from range query filtering by providing attacks whose query cost only scales with the relative error ε, and is independent of the size of the database or the number N of possible values of the data items.

4.2 Model Extraction Attack

A model extraction attack (also known as a model stealing attack or model reverse engineering attack) is a type of attack against ML models that and aims to extract or replicate the model by querying it with carefully designed inputs. The attacker may have limited access to the model, e.g. through a black box API or by observing the model's output, but not its internal parameters.

The basic idea of a model extraction attack is to train a surrogate model that mimics the behaviour of the target model. The attacker generates a set of inputs, feeds them into the target model and collects the corresponding outputs. Using this data set, the attacker trains a surrogate model, such as a decision tree or a neural network, that approximates the behaviour of the target model.

Model mining attacks can be used for various purposes, such as intellectual property theft, algorithmic bias analysis or malicious use of stolen models. There are several methods to carry out model mining attacks, including:

- **Query-based attacks:** Query-based attacks involve querying the target model with many inputs and collecting the corresponding outputs. This data is then used to train an alternative model that approximates the behaviour of the target model.
- **Decision-based attacks:** Decision-based attacks use the decision boundaries of the target model to infer its behaviour. The attacker can use this information to train a surrogate model that approximates the decisions of the target model**.**
- **Reconstruction-based attacks:** Reconstruction-based attacks aim to reconstruct the training dataset of the target model from its results. The attacker can use this dataset to train an alternative model approximating the target model**.**

Several works propose attack methods for model extraction. In [44], a model extraction attack on a DNN accelerator implemented in an FPGA is presented. The method obtains DNN model parameters (MLP architecture and weight parameters) using electromagnetic (EM) leakage from the accelerator. This is a typical and powerful method to steal a cryptographic key using EM leakage from a working cryptographic circuit. The proposal demonstrates that an adversary can extract the model parameters even if they are protected with data encryption by identifying 19 of the 20 weight parameters from 60,000 traces measured at a single probe position.

In [45], MEGEX a data-less model extraction attack against a gradient-based explainable AI is proposed, in this method an adversary uses the explanations to train the generative model and reduces the number of queries to steal the model. Experiments show that the proposed method reconstructs high accuracy models: 0.97 and 0.98 the accuracy of the victim model on SVHN and CIFAR-10 datasets with 2M and 20M queries, respectively. Furthermore, an adaptive query parameter duplication (QDP) attack [46] allows the adversary to infer model information through black-box attacks and without prior knowledge of any model parameters or training data.

Wang *et al.* [47] propose a dual far query strategy (DualCF) which mitigates the problems of shifting decision boundaries in surrogate model training by means of the query strategy and the counterfactual far query explanation (CCF). Also, in [48], a generative network-based extraction attack called Generative-Based Adaptive Model Extraction (GAME) is proposed, which adaptively augments the query data in a sample-limited scenario using GANs Auxiliary Classifiers. This attack allows adaptive data generation without original data sets, high fidelity, accuracy, and stability under different data distributions.

Seek [49] is a general extraction method for secure hybrid inference services that only emit class labels, specifically designed from models based on homomorphic encryption (HE) and/or multiparty computation (MPC). This method can extract each layer of the target model independently and is not affected by the depth of the model. For ResNet-18, SEEK can extract a parameter with less than 50 queries on average, with an average error of less than 0.03%.

In [50] an analysis of the model extraction attack towards electronic design automation (EDA) is performed in two real approaches, the first one introduces extraction attacks on Eda models and the second one proposes two attack methods against the unbounded and bounded query budget scenarios, the first is based on a trust-based data selection method to prevent the attacker's model from being misguided by the unreliable pseudo-labels, the second is an iterative information-based data selection method to progressively choose the most informative pseudo-labels for the attacking model and to improve the generalisability of the model. Model extraction attacks are shown to threaten the privacy of EDA models. In [51], different methods for attacking black-box DNN models are studied using two different methods. The first one develops a surrogate model with similar performance to the target model using the results of the target model as training data for the surrogate model, and the second one focuses on obtaining structural information of the target using a timing side-channel attack.

LibSteal[52], is a framework that allows leaking DNN architecture information by inverting the binary library generated from the DL compiler, which is similar or even equivalent to the original one. This method allows to efficiently steal architecture information from the victim's DNN models. To realise a model comparable in accuracy to the original, training of the fake model is performed with the same hyperparameters as the original. Karmakar *et al.*[53] describe an attack method that allows an adversary to extract information about a ML model used by an online service. The attack is based on querying the service's public API and observing the model's responses. The proposed method is able to obtain information about the structure of the model, the training data used, and the decisions made by the model in response to specific inputs.

A model extraction attack method called ActiveBoostThief is described in [54]. Instead of using random queries to extract information from the target model, ActiveBoostThief uses an active learning approach that carefully selects inputs to maximise the information obtained from each query. In addition, the attack uses reinforcement learning techniques to improve the effectiveness of the selected queries.

In addition, in the area of natural language processing (NLP), they also present security problems, in [55] a model extraction attack method based on adversarial transfer to extract information from the target model is presented, the attack has two phases, the model extraction attack and the Adversarial Example Transfer. In the attack it is assumed that a victim model is commercially available as a prediction API for the target task, an adversary tries to reconstruct the victim model by making multiple queries to it obtaining its predictions.

In the area of defence against extraction attacks, SEAT [56] presents a black-box model extraction attack detector. The proposal has a similarity encoder trained by adversarial training which detects accounts that make queries indicating a model extraction attack in progress and cancels these accounts, showing that even against adaptive attackers, SEAT increases the cost of model extraction attacks between 3.8 and 16 times. In

[46] a defence strategy against query attacks is shown by using monitoring-based differential privacy (MDP) by dynamically adjusting the amount of noise added in the model response according to the result of Monitor and effectively defending the QPD attack. Table 5 summarizes the work related to model extraction attacks mentioned in this section.

Articles	Method	Attack / Defence	Scope
Yoshida *et al.* [44]	- Electromagnetic leakage	Attack	- DNN accelerator on FPGA - Multi-layer Perceptron
Miura *et al.* [45]	- Data-Free Model Extraction	Attack	- Gradient-Based Explainable AI
Yan *et al.* [46]	- Monitoring-Based Differential Privacy Mechanism	Defence	- Query-flooding
Wang *et al.* [47]	- CF - Counterfactual explanation of CF	Attack	- Counterfactual Explanations
Xie *et al.* [48]	- Generative-Based Adaptive Model Extraction - Auxiliary classifier GANs	Attack	- MLaaS platform
Chen *et al.* [49]	- Piecewise-linear property of the ReLU activation	Attack	- HE-MPC hybrid inference service
Chang *et al.* [50]	- Model Extraction Attacks	Attack	- Electronic Design Automation models
Lkhagvadorj *et al.* [51]	- Accuracy Extraction Attack - Fidelity Extraction Attack - GANS	Attack	- Computer Vision Models - DNN
Zhang *et al.* [52]	- reversing the binary library generated from the DL Compiler	Attack	- DNN
Karmakar *et al.* [53]	- Black-box Model Stealing Attack - Online and Adaptive Algorithm	Attack	- Natural Language Processing - Computer Vision
Nam *et al.* [54]	- Random Queries	Attack	- APIs
He *et al.* [55]	- Model Extraction Attack (MEA) - Adversarial Example Transfer (AET)	Attack	- Natural Language Processing - BERT
Zhang *et al.* [56]	- Similarity Encoder by Adversarial Training	Defence	- Computer Vision

Table 5: Summary of Research on Model Extraction Attacks

4.3 Inference Attack

An inference attack is a technique attackers use to obtain sensitive information from a database by inferring data from public or semi-public information. This attack is commonly used in ML applications, where the ML model can be used to infer information about the training data or the input data. If a ML model is trained on a dataset that includes sensitive information, an attacker could use the model to infer sensitive information about the individuals in the dataset. In addition, an attacker can also use a ML model to infer information about input data, such as a user's preferences or behaviour [57][58][59][60][61].

To prevent inference attacks, it is important to consider data privacy and design systems that limit access to sensitive information. In addition, it is important to consider the potential impact of data inference and assess the risks associated with the use of ML models. Figure 4 shows a general scheme of the inference attack.

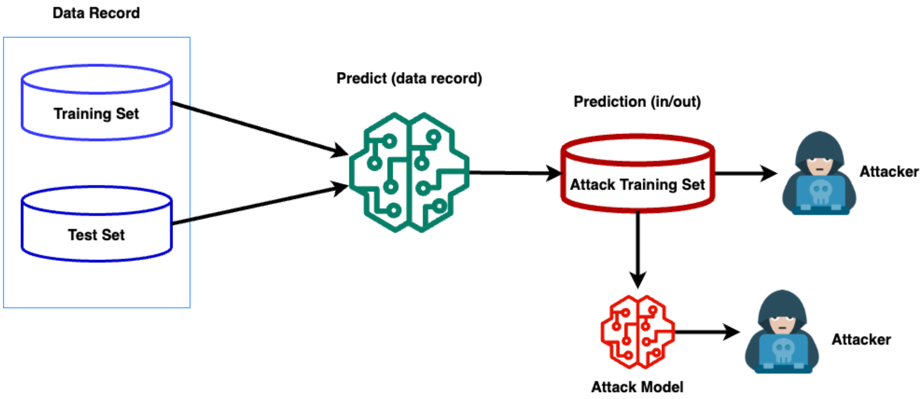

Figure 4: Inference Attack

Choquette-Choo *et al.* [62] perform an attack on ML models that does not require access to the model's predicted confidence score but instead evaluates the robustness of the model's predicted labels under input perturbations to infer membership and shows that a class of defences against membership inference, "confidence masking" because they obfuscate confidence scores to thwart attacks, are insufficient to prevent leakage of private information. The work shows that differential privacy training or regulation are the only current defences that significantly decrease private information leakage.

4.4 Evasion and Poisoning Attack

Evasion attacks are the most common attacks against ML systems. Malicious inputs are cleverly modified to force the model to make a false prediction and evade detection. Evasion attacks focus on the deep learning prediction process, where the attacker generates an adverse example, and the classifier can perform a misclassification on it (see Figure 5).

The poisoning attack differs because the inputs are modified during training, and the model is trained with contaminated inputs to obtain the desired output. In this type of attack, malicious samples are injected into the training dataset, reducing the prediction accuracy of the model.

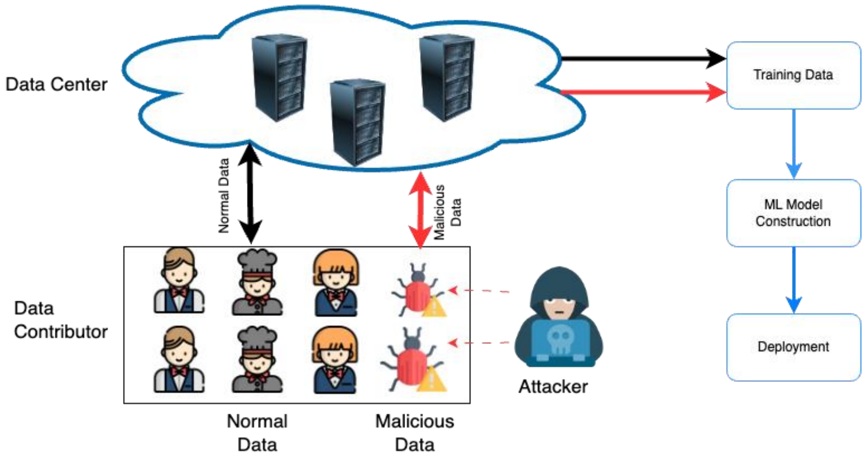

Figure 5: Poisoning Attack

4.5 Exploratory Attack

In this type of attack, the methodology is simple; the attacker seeks to resemble his model to approximate the target model by creating a copy of the original in which he performs tests looking for security replicate it in the original model later the original model, causing the accuracy of the target to be reduced over time as more and more adversaries will get to know the weak points of the model. This method of attack directly compromises the privacy of the model and the company's intellectual property.

5 Defences

There are various techniques and defence measures that can be used to protect ML models against security threats. Some of the most common defences are described below:

- **Differential privacy:** Differential privacy is a technique that adds random noise to data to hide sensitive information. This technique can be used to protect training data and ML model inferences.
- **Masking mechanisms:** Masking mechanisms can hide sensitive information in training data or model inference results. These mechanisms may include removing specific attributes from the data or modifying the values of these attributes**.**
- **Anonymisation:** Anonymisation is a technique to protect personally identifiable data by removing information that can identify a specific individual. This can be achieved by removing names, addresses and other personal data from data sets.
- **Access control:** Access control can restrict access to ML models and their data. This can be achieved through passwords, user authentication and other security techniques.
- **Continuous monitoring:** Continuous monitoring of ML models can help detect and prevent inference attacks and other security threats. This can be achieved by monitoring training data and model inference to detect unusual or anomalous patterns.
- **Homomorphic encryption:** is an emerging field that aims to develop tools to enable the computation of complex models from encrypted data. Preliminary research has focused on computer vision and speech technologies.
- **Decentralisation:** refers to the distribution of data, computation, and decision-making across multiple devices, nodes, or organizations instead of relying on a centralized server or authority.

Adversarial examples demonstrate that many modern ML algorithms can easily break in surprising ways. Adversarial examples are difficult to defend for the following reasons:

- **It is difficult to build a theoretical model of the adversarial example creation process.** The generation of adversarial examples is a complex optimisation process due to its non-linear and non-convex properties for most ML models. The lack of adequate theoretical tools to describe the solution to these complex optimisation problems makes it even more difficult to argue theoretically that a particular defence will rule out a set of adversarial examples.
- **ML models must provide adequate results for every possible input.** Considerable modification of the model to incorporate robustness to adversarial examples may change the elementary objective of the model.

Most current defence strategies are not adapted to all types of adversarial attacks, as one method may block one type of attack but leave another vulnerability open to an attacker who knows the underlying defence mechanism. In addition, the application of these defence strategies can lead to performance overhead and degrade the prediction accuracy of the actual model.

5.1 Differential Privacy

Differential privacy (DP) is a set of methods that facilitate the collection and analysis of data without compromising the right to privacy of the data subjects by eliminating the possibility of knowing whether a particular individual's data is included in the analysis. It is necessary to enable the publication of data in the day-to-day management of companies or research institutes. Differential privacy, as a promising mathematical model, has several attractive properties that can help solve these problems, making it a very valuable tool. For this reason, differential privacy has been widely applied in AI but, to date, no study has documented which differential privacy mechanisms can or have been exploited to overcome its problems or the properties that make it possible.

Users demand more transparency and control over how data is collected, stored, used and shared. In many places, regulators have introduced data privacy rules that will be a benchmark: in the European Union, the General Data Protection Regulation (GDPR) and California, the Consumer Privacy Act have brought concepts such as transparency, "user control" and "privacy at source" to the forefront for companies wishing to develop data-driven products.

With users and regulators now stressing the importance of data protection in business, data professionals are mobilising to create privacy protection tools for AI systems that will be used soon.

There are tools widely used in data protection, such as those that remove identifying values (names, IP addresses, etc.). These mechanisms have, however, certain limitations. There is even sufficient evidence that, by submitting processed data, they can be linked to other databases and lose privacy.

Starting from the same definition of differential privacy, three key features of this mathematical idea can be extracted:

- **Measurement of data privacy loss**. It facilitates, therefore, the control and balancing of privacy and data accuracy.
- **Composition**. PD is characterised by differential and parallel composition. The former helps the execution of multiple analyses separately within a single data set. The latter, in turn, allows a dataset to be split into several unconnected fragments to execute, in each fragment, the techniques encompassed by PD.
- **Post-processing**. It is completely safe to perform any computation or post-processing on the differentially private data. This is because there is no probability of reversing the process.

Figure 6 illustrates a basic differential privacy framework. Considering two datasets that are almost identical but differ in only one record and that access to the datasets is provided by a query function. If we can find a mechanism that can query both datasets and obtain the same results, we can claim that differential privacy is satisfied. In that scenario, an adversary cannot associate the query results with either of the two neighbouring datasets, so the one different record is safe.

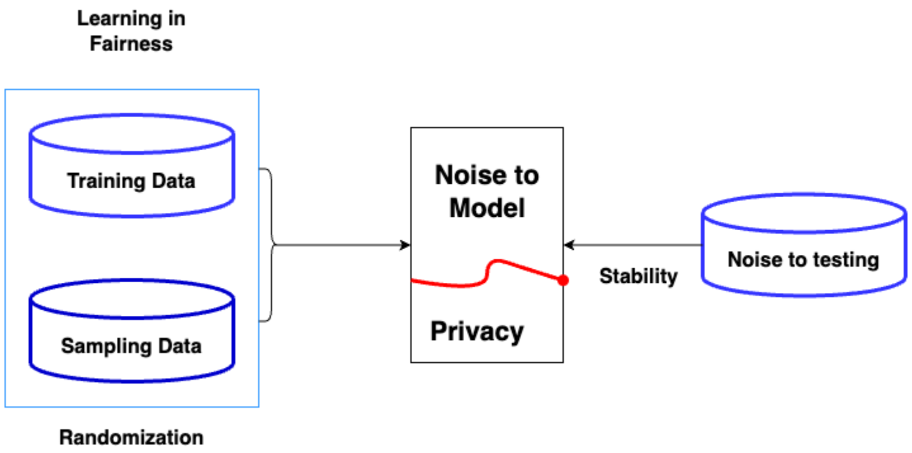

Figure 6: Differential Privacy

At the user level, local differential privacy (LDP) using random response techniques is applied to user data before the data is uploaded to clients at the perimeter. The key idea here is that LDP can provide robust data privacy protection while preserving data statistics to preserve its high utility. At the edge plane, the edge servers reconstruct the perturbed data, and then conventional federated cross-silo learning can be performed between the edge servers and the cloud. Doing so can protect raw user data even if malicious attackers infer the reconstructed plausible data.

Differential privacy has been widely applied to ML processes for privacy protection by adding noise through a Laplace or exponential mechanism to the training data set or loss functions.

5.1.1 Local Differential Privacy

Local differential privacy has recently emerged as a strong privacy measure in contexts where personal information remains private even to data analysts. Working in an environment where data providers and analysts wish to maximise the utility of statistical analyses performed on published data, we study the fundamental trade-off between local differential privacy and utility.

The need for data privacy arises in two different contexts: the local privacy context, such as when individuals disclose their personal information (e.g., voluntarily on social networking sites), and the global privacy context, such as when institutions publish databases of information on multiple individuals or answer queries on such databases (e.g., the US government publishes census data, companies like Netflix publish proprietary data for others to test state-of-the-art ML algorithms). In both contexts, privacy is achieved by randomising the data before publishing it.

5.1.2 Gradient Perturbation

The goal of gradient descent is to minimise the error function by gradually adjusting the learning parameters represented by the different weights. Gradient descent corresponds to this metaphor and is carried out by adjusting the different weights of the neural network until convergence, i.e., a minimum of errors is obtained. This adjustment is done in short steps, using a hyperparameter called learning rate [63].

Gradient perturbation, widely used for differential private optimisation, injects noise into each iterative update to guarantee differential privacy. It has the advantages of:

- does not require a strong assumption on the target because it only needs to constrain the sensitivity of each gradient update rather than the entire learning process.

- can release the noisy gradient at each iteration without damaging the privacy guarantee since differential privacy is immune to post-processing. Therefore, it is a favourable choice in distributed optimisation or federated learning environment.

Gradient perturbation achieves much better empirical utility than output/target perturbations for DP-ERM. However, the existing theoretical utility guarantee for gradient perturbation is the same or strictly inferior to that of output/target perturbation approaches [64].

5.1.3 Objective Perturbation

The target perturbation (i.e., the empirical loss) is the sensitivity of the target perturbation is the maximum change in the minimiser that a log can produce. it uses the largest and smallest eigenvalue (i.e., the smooth and strongly convex coefficient) of the target Hessian matrix to constrain that change. The gradient perturbation is more flexible than the output/target perturbations. For the gradient perturbation, the sensitivity is only determined by the Lipschitz coefficient, which is easy to obtain using the gradient clipping technique. More importantly, we always have ν=μ when the gradient update does not involve perturbation noise. Therefore, gradient perturbation is the only method to exploit such a noise effect among the three existing perturbation times[64].

The standard objective perturbation technique consists of "perturbing" the objective function by adding a random linear term and releasing the minima of the perturbed objective. However, objective perturbation provides privacy. It guarantees only whether the mechanism's output is the exact minimum of the perturbed objective. Practical algorithms for convex optimisation often involve the use of first-order iterative methods, such as gradient descent or stochastic gradient descent (SGD), due to their scalability. However, these methods often offer convergence rates that depend on the number of iterations performed by the method, so they are not guaranteed to reach exact minima in a finite time. As a result, it is unclear whether objective perturbation in its current form can be applied in a practical setting, where one is often limited by resources such as computing power, and reaching minima may not be feasible.

The private SGD algorithm provides optimal risk bounds for a later version of private SGD. It proposes an output perturbation variant that requires the use of permutation-based SGD and reduces sensitivity using the properties of that algorithm. Several papers deal with DP convex ERM in the framework of high-dimensional sparse regression, but the algorithms also require the minima to be obtained[65].

5.1.4 Label Perturbation

Label Perturbation is a defence technique that protects ML models against inference attacks and other malicious attacks. It involves modifying the labels of the training data so that the model's inferences are more difficult to infer or deduce. In the technique, random noise is added to the training data labels to distort the relationships between the input data and the output labels. The amount of noise added depends on the magnitude of the desired impact on model accuracy and the amount of confidential information to be protected.

It is important to note that the label perturbation technique can have an impact on the accuracy of the model. If too much noise is added to the labels, the model may lose its ability to make accurate and useful inferences. Therefore, it is necessary to balance the amount of noise added to the labels with the accuracy of the model.

5.2 Homomorphic Encryption

Homomorphic encryption is an advanced technology that allows users to perform calculations on encrypted data without decrypting it. This technique is used in data privacy protection, allowing users to perform computations on sensitive data without revealing its contents. In traditional cryptography, data is encrypted to protect it during transmission or storage and must be decrypted to perform calculations. In homomorphic encryption, data is encrypted so that calculations can be performed on the encrypted data without decryption (see Figure 7). This means the data remains encrypted throughout the calculation process and is only decrypted once the calculations have been completed.

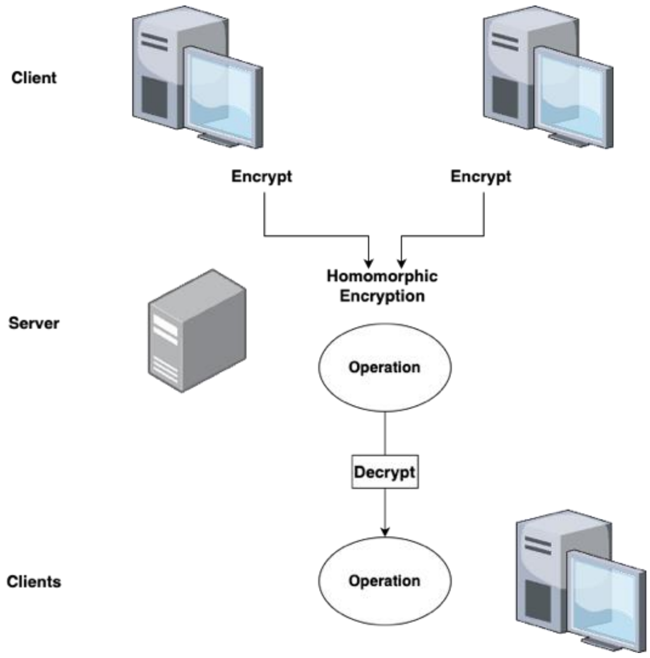

Figure 7: Homomorphic Encryption

There are different types of homomorphic encryption, including Fully Homomorphic Encryption (FHE) and Partially Homomorphic Encryption (PHE). In FHE, arbitrary computations can be performed on the encrypted data, while in PHE, only certain types of computations can be performed.

5.2.1 Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE) is an advanced encryption technology that allows any calculation to be performed on encrypted data without the need to decrypt it first. In other words, FHE allows operations to be performed on encrypted data securely and privately without revealing the original content of the data. The main advantage of FHE is that it allows users to perform advanced calculations on sensitive data without the need to decrypt it, which reduces the risk of exposing private information.

FHE is a complex technique and requires high computational power, which makes it slower and more expensive compared to conventional encryption. In addition, the FHE technique is still under development and is less mature than other encryption techniques.

5.2.2 Partially Homomorphic Encryption

Partially Homomorphic Encryption (PHE) is an encryption technique that allows certain arithmetic operations to be performed on encrypted data without first decrypting it. Unlike FHE that allows any operation to be performed on encrypted data, PHE only allows certain types of operations to be performed, such as addition or multiplication. PHE is less complex and expensive than FHE, making it more suitable for applications where only certain operations need to be performed on the encrypted data.

5.3 Secure Multi-Party Computation

Secure Multi-Party Computation (SMC) is a branch of cryptography that focuses on protecting data privacy in situations where multiple parties collaborate to perform a joint computation. Figure 8 shows a conceptual diagram of SMC. In SMC, multiple parties rely on a third party to perform computations, ensuring that the results are accurate, and protect data privacy. It is used in situations where data privacy is critical, such as in performing calculations where multiple parties have sensitive information that they do not wish to share with other parties [66].

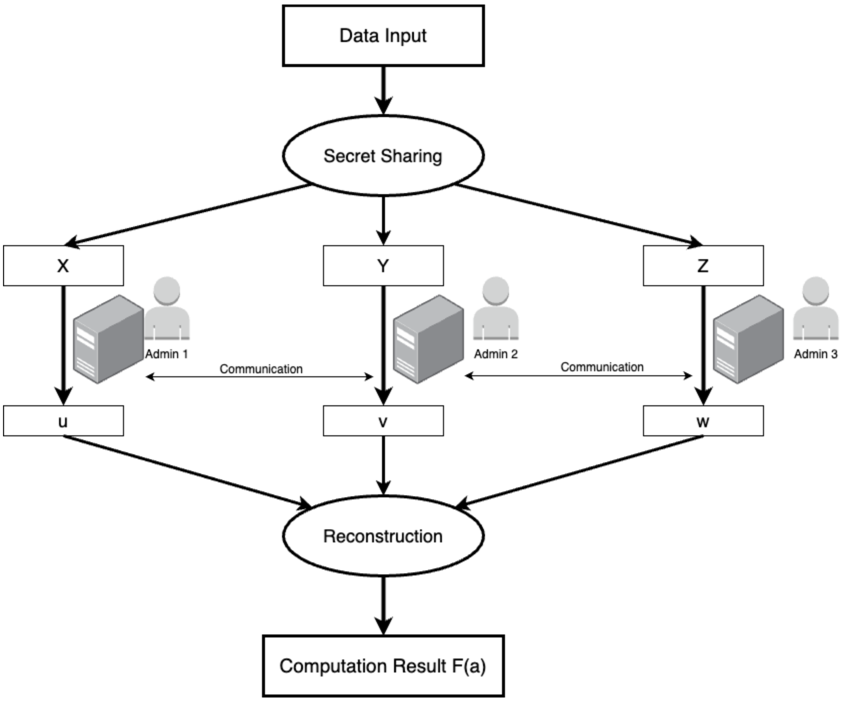

Figure 8: Conceptual Diagram - Secure Multi-Party Computation (SMPC)

In SMC, each party involved in the computation keeps its data private and secure, so other parties cannot see or manipulate it directly. Instead, the parties collaborate to perform the desired computation without revealing their information to the other parties involved. Several protocols and techniques for implementing SMC include public key cryptography, elliptic curve cryptography and cryptography based on threshold functions. These techniques protect data privacy and ensure that the results are accurate and valid [67].

SMC also has applications in the field of ML. In this context, SMC allows multiple parties to collaborate in building ML models without sharing their training data with other parties. The goal of SMC in ML is to allow multiple parties to train a joint ML model without revealing their training data, which can be especially important in situations where the data is sensitive or private. To achieve this, SMC techniques, such as public key cryptography and threshold function-based cryptography, are used to ensure that each party has access only to the information needed to train the model, without compromising the privacy of the other parties' data.

5.4 Trusted Execution Environment

Trusted Execution Environment (TEE) use special hardware and software to create a secure and isolated environment within the host system. This secure environment uses advanced cryptographic techniques and access control mechanisms to protect the software and data within it from unauthorised access. TEEs also guarantee integrity and authenticity by protecting the software and data from external tampering or alteration[68]. In the context of Machine Learning, a TEE can protect the ML model and training data while running on a device or hardware platform that cannot be completely trusted[69][70].

In addition, a TEE can also help prevent data poisoning attacks, which consist of manipulating the training data so that the resulting model behaves in an undesirable way.

6 Generative Modeling

Generative Modeling is a ML technique that generates new data similar to an initial dataset in a synthetic way. This modelling aims to learn patterns and structures from a given dataset and use this knowledge to generate new samples similar to the original ones.

There are different types of generative models, among them:

- **Variational Automatic Encoder (VAE):** a type of neural network learns from a compressed representation of the data and generates new ones based on this data.
- **Generative Adversarial Network (GAN):** a type of neural network consisting of a generator and a discriminator. The generator learns to generate new data that is similar to the training data, while the discriminator tries to distinguish between the real data and the generated data.
- **Autoregressive models:** a type of generative model that models the conditional probability of a sequence of data, given the previous data points in the sequence.

For the development of the tool for task 3.4 of the LAZARUS project, it will focus on the use of GANs as they are successful in creating adversarial samples posing as real ones and generating defences, which will allow for effectively measure the resilience of the LAZARUS tools T3.2 and 3.3, evaluating their security by detecting flaws in the implemented techniques.

6.1 Generative Adversarial Networks

GANs introduced in [71] are a type of AI algorithm designed to solve the problem of generative modelling using deep learning (DL) methods such as Convolutional Neural Networks (CNN). Generative modelling is a neural network that estimates the probability of each observation and creates new samples from the underlying distribution [72]. That is, the model tries to discover and learn regularities or patterns in the input data so that the model can be used to generate new examples that could have been drawn from the original dataset.

GANs consist of two interacting neural networks, a generator, and a discriminator (see Figure 9). The generator network takes random noise as input and tries to generate data that is indistinguishable from the training data. The discriminative network takes real data and generated data as input and tries to distinguish between them. The generator model takes a random vector of fixed length as input and generates a sample in the domain. This vector is randomly drawn from a Gaussian distribution and is used as a noise source for the generative process.

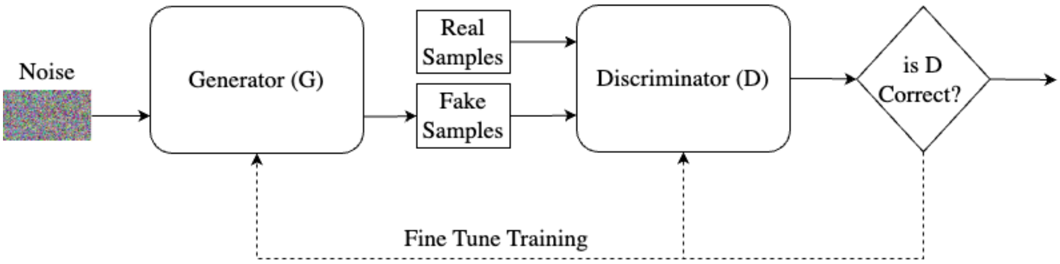

Figure 9: GAN Architecture

The training of these networks is done iteratively in which the discriminator tries to distinguish the real data from the fake data previously created by the generator. The Generator converts the random noise in the input to samples that fool the discriminator. In the training process, the weights and biases of the discriminator network are updated to increase its classification accuracy, to predict which data is really false and which data is really true. And the generator biases and weights are also updated to increase the probability that the discriminator will misclassify, i.e., a false example will be classified as real. It should be noted that the generator network does not have direct access to the real data and bases its learning on its interaction with the discriminator, which has access to both the synthetic samples and those extracted from the real data stack.

To allow the generator to match a synthetic sample as closely as possible to a real sample, an error signal is used in the discriminator to tell whether the data comes from the real stack or from the generator. This error signal, via the discriminator, is used to train the generator to produce better quality fakes. The models use a zero-sum mini-max approach where the Generator tries to maximise the probability and the discriminator tries to minimise it. The loss function used by the GANs is as follows [73]:

$$\mathrm{min_G max_D V(D, G)} = \mathbb{E}_{\mathrm{x \sim p_{data(x)}}}[\log \mathrm{D(x)}] + \mathbb{E}_{\mathrm{x \sim p_z(z)}}[\log\left(1 - \mathrm{D(G(z))}\right)]$$

where x is the real data sample, z is the random noise vector, E represents the expectation, G(z) is the data given by the generator, D(x) indicates the probability of D in the real data x and D(G(z)), the probability that D yields in the generated data G. The goal of D is to bring D(G(z)) close to 0, and the goal of G is to bring it close to 1. The objective of D is to bring D(G(z)) close to 0, and the objective of G is to bring it close to 1. If D gives a probability of 0.5, it means that D cannot decide whether the sample is real or fake.

Today, GANs have been used in a variety of applications, most notably in the area of image processing. GANs have been used for image texture synthesis [74], image resolution enhancement [75] and image generation [76][77]. In the area of security, the use of GANs presents an advantage to the organisation as it allows for the discovery of new security holes and timely mitigations as the use of GANs provides a solid defence mechanism to prepare an organisation's systems for new attacks.

Today, GANs have become very popular for security tasks, for generating adverse examples of malware samples and thus bypassing the anti-malware protections in place.

In [79] they make use of GANs to generate adversarial malware samples called MalGAN. This network uses a surrogate detector to adapt to the black-box malware detection system. A generative network is trained to minimise the malicious probabilities of the generated adversarial samples predicted by the surrogate detector. This approach outperforms traditional algorithms that are based on gradients as it produces a detection rate that tends to zero and makes it difficult to defend based on retraining against adversarial examples. Furthermore, in [80], a proposal is made to generate poisoning attacks using GANs. These networks use three components: generator, discriminator and the target classifier. This approach allows to naturally model detectability constraints and identify regions of the underlying data distribution that may be more vulnerable to data poisoning.

Also in [81], DoS-WGAN, a common architecture that uses Wasserstein Generative Adversarial Networks (WGAN) with gradient penalty technology to evade network traffic classifiers to perform denial-of-service (DoS) attacks, is proposed. This approach allows camouflaging offensive DoS attack traffic as normal network traffic, DoS-WGAN automatically synthesises attack traces that can defeat an existing NIDS/network security defence for DoS cases. It makes use of information entropy to measure the dispersion performance of the generated attack traffic. The proposal achieves a 47.6% drop in IDS detection.

For speech recognition systems, there are proposals such as [82] which proposes an attack using GANs on speech emotion recognition (SER) systems. Experimental evaluations suggest several interesting aspects of effective use of adversarial examples useful for achieving robustness of SER systems, which opens up opportunities for researchers to further innovate in this area.

Lin *et al.* [83] propose a generative adversarial network framework, called IDSGAN, to generate malicious adversarial traffic logs to attack intrusion detection systems by deceiving and evading detection. IDSGAN uses a generator to transform the original malicious traffic logs into malicious adversarial logs. A discriminator classifies the traffic samples and dynamically learns the black box detection system in real-time.

Also, adversarial attacks allow attacks on wireless networks. In [84], a method of spoofing wireless signals by using an adversarial network to generate and transmit synthetic signals that cannot be reliably distinguished from the intended signals is presented. The attack requires capturing various waveform, channel and radio hardware characteristics that are inherent in the original wireless signals. The attack architecture consists of an adversary pair consisting of a transmitter and a receiver that assume the roles of generator and discriminator in the GAN and make use of minimax to generate the best spoofing signals intended to fool the best-trained defence mechanism.

7 LAZARUS Tool

After analysing the state of the art and considering the goals and objectives of LAZARUS, an overview of the base architecture of the tool is provided to measure resilience, assess security, detect flaws and determine the level of robustness of the AI techniques implemented in tasks T3.2, T3.3, against "adversarial attacks".

The tool shall have the capability to perform black and white box testing against the AI techniques implemented in tasks 3.2 and 3.3, in that sense the tool by means of a random input shall have the capability to generate an adversarial sample in the generator module. This sample will be sent to a classification module to perform a relabelling of the real sample and the generated sample, these two samples will be sent to the discrimination module which will be in charge of verifying how much the fake sample resembles the real one, if it finds a significant difference based on the predefined thresholds, this sample will be sent to the generator to readjust the parameters and an additional sample will be generated taking into account the patterns of the sample rejected by the generated one. When the discriminator indicates that both samples are similar, the adversarial sample will be ready to be entered into the target to verify that it is successful in fooling the implemented model. This process is detailed in Figure 10.

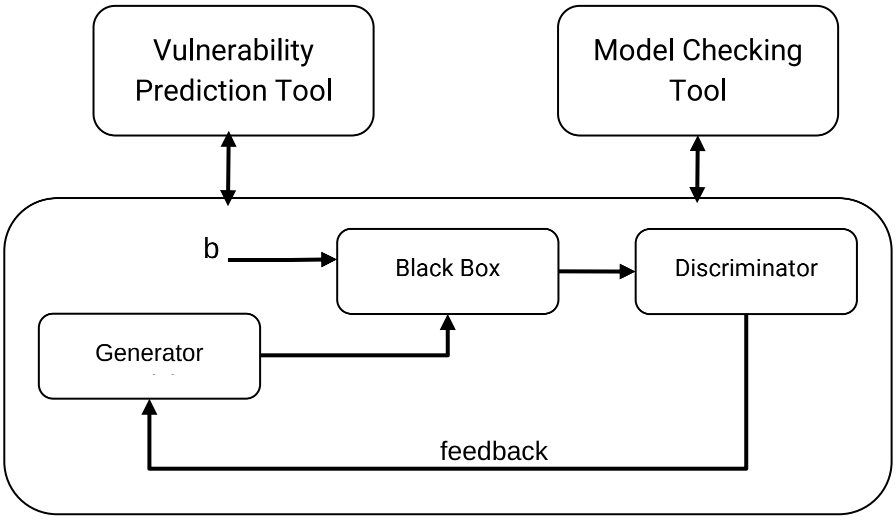

Figure 10: Base Architecture LAZARUS Tool

8 Conclusion

This deliverable list popular attack techniques that an adversary can use to influence the decisions made by an AI algorithm. It should be noted that security in AI algorithms is crucial to ensure the confidentiality, integrity and availability of the data and systems that depend on them. AI algorithms can be vulnerable to adversarial attacks, either by manipulating input data, exploiting weaknesses in models, or by other methods. These attacks can have serious consequences on data privacy, security and reliability of AI systems.

A critical component in the field of cybersecurity and ML to consider is the adversarial threat model. These attacks can target system and model vulnerabilities, and understanding the motives and capabilities of potential attackers is essential to developing robust defences.

To ensure security in AI algorithms, it is necessary to design and develop systems that are robust to adversarial attacks. This involves a thorough understanding of potential threats, implementation of appropriate security measures, and thorough testing to detect and mitigate any vulnerabilities. In addition, it is important to follow good security and data privacy practices at all stages of the AI development process.

The deliverable also mentions some AI-based techniques that allow defending the models implemented by organisations and thereby protecting training data that may contain sensitive customer data. In addition, general defences such as encryption techniques and obfuscation that can be implemented from the base model are shown.

9 References

[1] Recognition, P.; s.r.l., A.L..P.O. Evasion Attacks against Machine Learning https://secml.readthedocs.io/en/stable/tutorials/03-Evasion.html. accessed: 2023-01-06.

[2] Recognition, P.; s.r.l., A.L..P.O. Poisoning Attacks against Machine Learning models https://secml.readthedocs.io/en/stable/tutorials/05-Poisoning.html. accessed: 2023-01-06.

[3] Qiu, S.; Liu, Q.; Zhou, S.; Wu, C. Review of Artificial Intelligence Adversarial Attack and Defense Technologies. Appl. Sci. 2019, 9, 909. https://doi.org/10.3390/app9050909

[4] A. I. Newaz, N. I. Haque, A. K. Sikder, M. A. Rahman and A. S. Uluagac, "Adversarial Attacks to Machine Learning-Based Smart Healthcare Systems," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9322472.

[5] Xue, B., Wu, L., Liu, A., Zhang, X., Chen, X. (2021). White-Box Attacks on the CNN-Based Myoelectric Control System. In: Fang, L., Chen, Y., Zhai, G., Wang, J., Wang, R., Dong, W. (eds) Artificial Intelligence. CICAI 2021. Lecture Notes in Computer Science(), vol 13069. Springer, Cham. https://doi.org/10.1007/978-3-030-93046-2_13

[6] Wang, Y.; Liu, J.; Chang, X.; Wang, J.; Rodríguez, R.J. DI-AA: An Interpretable White-box Attack for Fooling Deep Neural Networks, 2021. doi:10.48550/ARXIV.2110.07305.

[7] Wang, Y.; Liu, J.; Chang, X.; Mišíc, J.; Mišíc, V.B. IWA: Integrated gradient-based white-box attacks for fooling deep neural networks. International Journal of Intelligent Systems 2022, 37, 4253–4276. doi:10.1002/int.22720.

[8] Nazemi, A., & Fieguth, P. (2019). Potential adversarial samples for white-box attacks. arXiv preprint arXiv:1912.06409.

[9] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017, April). Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security (pp. 506-519).

[10] X. Yuan, P. He, Q. Zhu and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 9, pp. 2805-2824, Sept. 2019, doi: 10.1109/TNNLS.2018.2886017.

[11] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017, April). Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security (pp. 506-519).

[12] Hu, Weiwei & Tan, Ying. (2017). Black-Box Attacks against RNN based Malware Detection Algorithms.

[13] Pengcheng, L., Yi, J., & Zhang, L. (2018, November). Query-efficient black-box attack by active learning. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 1200-1205). IEEE.

[14] Guo, S., Zhao, J., Li, X., Duan, J., Mu, D., & Jing, X. (2021). A black-box attack method against machine-learning-based anomaly network flow detection models. Security and Communication Networks, 2021, 1-13.

[15] Alahmed, S., Alasad, Q., Hammood, M. M., Yuan, J. S., & Alawad, M. (2022). Mitigation of Black-Box Attacks on Intrusion Detection Systems-Based ML. Computers, 11(7), 115.

[16] He K, Kim D and Asghar M. Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey. IEEE Communications Surveys & Tutorials. 10.1109/COMST.2022.3233793. 25:1. (538-566).

[17] Fredrikson, M., et al.: Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing.

[18] Fredrikson, M., Jha, S., Ristenpart, T.: 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. ACM (2015).

[19] Zhang, Y.; Jia, R.; Pei, H.; Wang, W.; Li, B.; Song, D. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[20] Zecheng He, Tianwei Zhang, and Ruby B. Lee. 2019. Model inversion attacks against collaborative inference. In Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC '19). Association for Computing Machinery, New York, NY, USA, 148–162.

[21] Zhao, X., Zhang, W., Xiao, X., & Lim, B. (2021). Exploiting explanations for model inversion attacks. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 682-692).

[22] Mo, K., Huang, T., Xiang, X. (2020). Querying Little Is Enough: Model Inversion Attack via Latent Information. In: Chen, X., Yan, H., Yan, Q., Zhang, X. (eds) Machine Learning for Cyber Security. ML4CS 2020. Lecture Notes in Computer Science, vol 12487. Springer, Cham.

[23] Chen, S., Kahla, M., Jia, R., & Qi, G. J. (2021). Knowledge-enriched distributional model inversion attacks. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 16178-16187).

[24] M. Khosravy, K. Nakamura, Y. Hirose, N. Nitta and N. Babaguchi, "Model Inversion Attack by Integration of Deep Generative Models: Privacy-Sensitive Face Generation From a Face Recognition System," in IEEE Transactions on Information Forensics and Security, vol. 17, pp. 357-372, 2022.

[25] Luo, Z, Zhu, C, Fang, L, Kou, G, Hou, R, Wang, X. An effective and practical gradient inversion attack. Int J Intell Syst. 2022; 37: 9373- 9389.

[26] Kariyappa, S., & Qureshi, M. K. (2021). ExPLoit: Extracting Private Labels in Split Learning. In First IEEE Conference on Secure and Trustworthy Machine Learning.

[27] J. Geng et al., "Improved Gradient Inversion Attacks and Defenses in Federated Learning," in IEEE Transactions on Big Data.

[28] Xu, J. (2022). Approximate Gradient Inversion Attack on Federated Learning.

[29] Z. Li, L. Wang, G. Chen, Z. Zhang, M. Shafiq and Z. Gu, "E2EGI: End-to-End Gradient Inversion in Federated Learning," in IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 2, pp. 756-767, Feb. 2023.

[30] Yang, Z., Yang, S., Huang, Y. et al. AAIA: an efficient aggregation scheme against inverting attack for federated learning. Int. J. Inf. Secur. (2023).

[31] Zhu, L., Liu, Z., & Han, S. (2019). Deep leakage from gradients. Advances in neural information processing systems, 32.

[32] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 308-318).

[33] Wei, W., Liu, L., Loper, M., Chow, K. H., Gursoy, M. E., Truex, S., & Wu, Y. (2020). A framework for evaluating gradient leakage attacks in federated learning. arXiv preprint arXiv:2004.10397.

[34] Tramer, F., & Boneh, D. (2020). Differentially private learning needs better features (or much more data). arXiv preprint arXiv:2011.11660.

[35] Wu, R., Chen, X., Guo, C., & Weinberger, K. Q. (2022). Learning to Invert: Simple Adaptive Attacks for Gradient Inversion in Federated Learning.

[36] Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). Deep models under the GAN: information leakage from collaborative deep learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 603-618.

[37] Wang, Y., Shan, S., & Chen, X. (2019). Generative adversarial networks-based data reconstruction attacks against machine learning models. Frontiers of Computer Science, 13(1), 1-14.

[38] Gambs, S., Gmati, A., & Hurfin, M. (2012, July). Reconstruction Attack through Classifier Analysis. In DBSec (pp. 274-281).

[39] Salem, A. M. G., Bhattacharyya, A., Backes, M., Fritz, M., & Zhang, Y. (2020). Updates-leak: Data set inference and reconstruction attacks in online learning. In 29th USENIX Security Symposium (pp. 1291-1308). USENIX.

[40] C. Chen, L. Lyu, H. Yu and G. Chen, "Practical Attribute Reconstruction Attack Against Federated Learning," in IEEE Transactions on Big Data, doi: 10.1109/TBDATA.2022.3159236.

[41] Milli, S., Schmidt, L., Dragan, A. D., & Hardt, M. (2019, January). Model reconstruction from model explanations. In Proceedings of the Conference on Fairness, Accountability, and Transparency (pp. 1-9).

[42] M. Song et al., "Analyzing User-Level Privacy Attack Against Federated Learning," in IEEE Journal on Selected Areas in Communications, vol. 38, no. 10, pp. 2430-2444, Oct. 2020, doi: 10.1109/JSAC.2020.3000372.

[43] P. Grubbs, M. -S. Lacharité, B. Minaud and K. G. Paterson, "Learning to Reconstruct: Statistical Learning Theory and Encrypted Database Attacks," 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019, pp. 1067-1083, doi: 10.1109/SP.2019.00030.

[44] K. Yoshida, T. Kubota, M. Shiozaki and T. Fujino, "Model-Extraction Attack Against FPGA-DNN Accelerator Utilizing Correlation Electromagnetic Analysis," 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), San Diego, CA, USA, 2019, pp. 318-318.

[45] Miura, T., Hasegawa, S., & Shibahara, T. (2021). MEGEX: Data-free model extraction attack against gradient-based explainable AI.

[46] H. Yan, X. Li, H. Li, J. Li, W. Sun and F. Li, "Monitoring-Based Differential Privacy Mechanism Against Query Flooding-Based Model Extraction Attack," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 4, pp. 2680-2694, 1 July-Aug. 2022.

[47] Wang, Y., Qian, H., & Miao, C. (2022, June). DualCF: Efficient Model Extraction Attack from Counterfactual Explanations. In 2022 ACM Conference on Fairness, Accountability, and Transparency (pp. 1318-1329).

[48] Xie, Y., Huang, M., Zhang, X., Dong, C., Susilo, W., & Chen, X. (2022, September). GAME: Generative-Based Adaptive Model Extraction Attack. In Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I (pp. 570-588). Cham: Springer International Publishing.

[49] Chen, S., & Fan, J. (2022). SEEK: model extraction attack against hybrid secure inference protocols. arXiv preprint arXiv:2209.06373.

[50] Chang, C. C., Pan, J., Xie, Z., Hu, J., & Chen, Y. (2023). Rethink before Releasing your Model: ML Model Extraction Attack in EDA.

[51] Lkhagvadorj, D. (2022). Model extraction attack on Deep Neural Networks.

[52] Zhang, J., Wang, P., & Wu, D. LibSteal: Model Extraction Attack towards Deep Learning Compilers by Reversing DNN Binary Library.

[53] Karmakar, P., & Basu, D. Marich: A Query-efficient & Online Model Extraction Attack using Public Data.

[54] Nam, Y., Kang, J., & Lee, J. G. (2021). ActiveBoostThief: Model Extraction Attack Using Reliable Active Learning, 594-596.

[55] He, X., Lyu, L., Xu, Q., & Sun, L. (2021). Model extraction and adversarial transferability, your BERT is vulnerable! arXiv preprint arXiv:2103.10013.

[56] Zhang, Z., Chen, Y., & Wagner, D. (2021, November). Seat: similarity encoder by adversarial training for detecting model extraction attack queries. In Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security (pp. 37-48).

[57] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 3-14).

[58] Fredrikson, M., Jha, S., & Ristenpart, T. (2015). 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).

[66] Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). 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).

[60] Salem, M. B., Zhang, Y., & Yu, H. (2020). An empirical study of privacy risks in machine learning models. IEEE Transactions on Dependable and Secure Computing, 1-1.

[61] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. In Proceedings of the 25th USENIX Security Symposium (pp. 601-618).

[62] Choquette-Choo, C. A., Tramer, F., Carlini, N., & Papernot, N. (2021, July). Label-only membership inference attacks. In International conference on machine learning (pp. 1964-1974). PMLR.

[63] Bonaccorso, G. (2017). Mastering Machine Learning Algorithms (1.ª ed., pp. 103–107). Birmingham, United Kingdom: Packt Publishing. Birmingham, United Kingdom: Packt Publishing.

[64] Yu, D., Zhang, H., Chen, W., Liu, T. Y., & Yin, J. (2019). Gradient perturbation is underrated for differentially private convex optimization. arXiv preprint arXiv:1911.11363.

[65] Iyengar, R., Near, J. P., Song, D., Thakkar, O., Thakurta, A., & Wang, L. (2019, May). Towards practical differentially private convex optimization. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 299-316). IEEE.

[66] Goodfellow, I., Papernot, N., Huang, S., Duan, Y., Abbeel, P., & Clark, J. (2017). Attacking machine learning with adversarial examples. OpenAI Blog, 24.

[67] Zapechnikov, S. (2022). Secure multi-party computations for privacy-preserving machine learning. Procedia Computer Science, 213, 523-527.

[68] https://csrc.nist.gov/Projects/threshold-cryptography.

[69] M. Sabt, M. Achemlal and A. Bouabdallah, "Trusted Execution Environment: What It is, and What It is Not," 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 2015, pp. 57-64, doi: 10.1109/Trustcom.2015.357.

[70] Geppert, T., Deml, S., Sturzenegger, D., & Ebert, N. (2022). Trusted execution environments: applications and organizational challenges. Frontiers in Computer Science, 4(930741).

[71] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Networks,” NIPS’14: Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. Volume 2, no. December 2014, pp. 2672–2680, 2014.

[72] L. Ruthotto y E. Haber, "An introduction to deep generative modeling", GAMM-Mitteilungen, vol. 44, n.º 2, mayo de 2021. Avialable: https://doi.org/10.1002/gamm.202100008

[73] I. K. Dutta, B. Ghosh, A. Carlson, M. Totaro and M. Bayoumi, "Generative Adversarial Networks in Security: A Survey," 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2020, pp. 0399-0405, doi: 10.1109/UEMCON51285.2020.9298135.

[74] Das, D.; Biswas, S.; Sinha, S.; Bhowmick, B. Speech-driven facial animation using cascaded gans for learning of motion and texture. Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16. Springer, 2020, pp. 408–424.

[75] Takano, N.; Alaghband, G. SRGAN: Training Dataset Matters, 2019. doi:10.48550/ARXIV.1903.09922.

[76] Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., & Aila, T. (2021). Alias-free generative adversarial networks. Advances in Neural Information Processing Systems, 34, 852-863.

[77] Phukan, S., Singh, J., Gogoi, R., Dhar, S., Jana, N.D. (2022). COVID-19 Chest X-ray Image Generation Using ResNet-DCGAN Model. In: Mohanty, M.N., Das, S. (eds) Advances in Intelligent Computing and Communication. Lecture Notes in Networks and Systems, vol 430. Springer, Singapore. https://doi.org/10.1007/978-981-19-0825-5_24.

[78] Hu, W., Tan, Y. (2022). Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. In: Tan, Y., Shi, Y. (eds) Data Mining and Big Data. DMBD 2022. Communications in Computer and Information Science, vol 1745. Springer, Singapore.

[79] Hu, W., & Tan, Y. (2023, January). Generating adversarial malware examples for black-box attacks based on GAN. In Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II (pp. 409-423). Singapore: Springer Nature Singapore.

[80] Muñoz-González, L., Pfitzner, B., Russo, M., Carnerero-Cano, J., & Lupu, E. C. (2019). Poisoning attacks with generative adversarial.

[81] Yan, Q., Wang, M., Huang, W. et al. Automatically synthesizing DoS attack traces using generative adversarial networks. Int. J. Mach. Learn. & Cyber. 10, 3387–3396 (2019).

[82] Latif, S., Rana, R., & Qadir, J. (2018). Adversarial machine learning and speech emotion recognition: Utilizing generative adversarial networks for robustness.

[83] Lin, Z., Shi, Y., Xue, Z. (2022). IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection. In: Gama, J., Li, T., Yu, Y., Chen, E., Zheng, Y., Teng, F. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2022. Lecture Notes in Computer Science, vol 13282. Springer, Cham.

[84] Shi, Y., Davaslioglu, K., & Sagduyu, Y. E. (2019, May). Generative adversarial network for wireless signal spoofing. In Proceedings of the ACM Workshop on Wireless Security and Machine Learning (pp. 55-60).