Security and Privacy of AI Knowledge Guide
Issue 1.0.0

Lorenzo Cavallaro | University College London

Emiliano De Cristofaro | University College London

EDITOR
Steve Schneider | University of Surrey

REVIEWERS
James Muir | BAE Systems Digital Intelligence
Jose Such | King's College London
Yang Zhang | CISPA

COPYRIGHT

The CyBOK project would like to understand how the CyBOK is being used and its uptake. The project would like organisations using, or intending to use, CyBOK for the purposes of education, training, course development, professional development etc. to contact it at **contact@cybok.org** to let the project know how they are using CyBOK.

1 INTRODUCTION

Machine Learning (ML) has rapidly become a fundamental technology that underpins countless applications, from natural language processing [7, 15, 18, 86, 97] and computer vision [46, 83] to fraud detection [106] and personalized recommendations [105]. In recent years, there has been a growing understanding of how to use ML in security contexts, leading to the development of advanced tools and techniques for detecting and preventing malicious activities [23, 24, 43]. Readers interested in knowing more about AI for security can refer to the corresponding CyBOK Topic Guide [76]. However, the security and privacy aspects of ML itself remain less understood, posing new challenges and opportunities for researchers and practitioners.

This Cybersecurity Body of Knowledge (CyBoK) Knowledge Guide (KG) aims to describe attacks and defenses that undermine the security and privacy of ML models [1], which define the field of adversarial machine learning. Our focus is on the key challenges, open problems, and promising solutions that have emerged in the context of securing and preserving the privacy of ML models in this rapidly evolving field.

Traditionally, the primary focus of ML research has been on improving performance, often measured in terms of accuracy, precision, recall, and other traditional performance-related metrics. However, as ML systems are increasingly deployed in real-world settings with significant security and privacy implications, there is a growing need to consider additional objectives. These objectives may include robustness against adversarial attacks [44] and distributions shifts [9, 70], fairness in decision-making processes [55], and ensuring the explainability and interpretability of models [58]. Balancing these often competing objectives is a central challenge in developing trustworthy ML systems.

Moreover, privacy concerns are becoming increasingly crucial as ML models are trained on massive amounts of data [15], which may include sensitive or confidential information [16]. There is an inherent tension between the need to release or provide access to data and models for the sake of transparency and collaboration and the requirement to protect data confidentiality as well as the privacy of individuals and organisations. This tension is further exacerbated by the growing sophistication of techniques for inferring sensitive information from ML models, even when the models themselves are considered “black boxes”.

Research in the area of trustworthy ML is advancing rapidly, with new techniques and methodologies being proposed and evaluated regularly. As a result, this CyBoK KG is not intended to provide an exhaustive survey of the field but rather to offer an overview of the current trends, open problems, and promising solutions that are shaping the landscape of adversarial machine learning with a specific focus on the Security and Privacy of AI systems. By exploring these topics, we provide a valuable resource for computer scientists, practitioners, and researchers seeking to better understand and address the security and privacy challenges associated with ML systems.

In the rest of the KG, we will delve into various aspects of adversarial machine learning, covering topics such as evasive attacks, poisoning and backdoor attacks, realisable and problem-space attacks, inference attacks, as well as defenses against adversarial attacks and privacy-preserving techniques. We hope that future extensions of this KG will cover in more detail the broader context of trustworthy ML, including issues related to fairness,

[1] Although this Knowledge Guide focuses on data-driven learning-based algorithms (ML), we may use the broader term AI interchangeably throughout the text.

interpretability, and explainability, as well as related supply chain security, emerging trends and future directions for research in this exciting and rapidly evolving field [22, 77].

2 THREAT MODELS

Before delving into the security and privacy aspects of machine learning, we first introduce relevant adversarial models [10].

Knowledge. One important aspect is what kind of *knowledge* the attacker might have:

- *Perfect-knowledge:* they have some information about the model or its original training data, e.g., ML algorithm, model parameters, network structure, and (some) training data.
- *Zero-Knowledge:* they have no knowledge about the model. Rather, they might explore a model by providing a series of carefully crafted inputs and observing outputs.
- *Partial-Knowledge:* attackers' knowledge sits in between perfect and zero knowledge.

Training vs Inference. Another variable is *where* the attack might take place:

- *Training Phase:* the adversary may alter the training dataset to influence the underlying learning process. They may also attempt to learn the model, e.g., accessing a summary, partial or all of the training data. In the process and depending on the context, the adversary might create a substitute model (also known as auxiliary or surrogate model) to use to mount attacks on the victim system.
- *Inference Phase:* the adversary may generate carefully-crafted test-time examples to affect the model's predictions. The adversary may also collect evidence about the model characteristics by observing inferences made by it.

Passive vs Active. Finally, one can also distinguish between passive and active attacks, roughly mirroring the traditional distinction in security literature between honest-but-curious and fully malicious adversaries:

- *Passive attack:* the adversary passively observes the updates and performs inference, e.g., without changing anything in the training procedure;
- *Active attack:* the adversary actively changes the way they operate.

These knowledge settings represent different realistic threat models and influence security as well as privacy aspects ML systems [10, 91].

3 ADVERSARIAL ML ATTACKS

Adversarial machine learning attacks pose a significant threat to the security and privacy of machine learning systems. In the following section, we outline *evasion*, *poisoning*, and *backdoor* attacks, the main categories of adversarial threats that undermine the security properties of ML systems. Although such attacks expose inherent weaknesses of learning-based algorithms, it is important to consider their implications in the context of realisable (or problem-space) attacks, discussion that we provide at the end of the section.

3.1 Evasion Attacks

Evasion attacks, also known as test-time attacks, occur when an adversary manipulates (perturbs) input data to mislead a machine learning model during its inference phase. In general, the main working mechanism of adversarial attacks is to solve an optimisation problem that aims to find perturbations to the input data, which, when added, cause the target machine learning model to produce incorrect or misleading outputs [10, 17, 87]. These perturbations are typically small and often imperceptible to humans, yet effective enough to fool the model.

The optimisation problem can be formally defined as follows [44]:

$$\begin{aligned} \underset{\delta}{\text{minimize}} \quad & \|\delta\|_p \\ \text{subject to} \quad & \mathcal{C}(x+\delta) \neq \mathcal{C}(x), \qquad \text{(a)} \\ & x+\delta \in [0,1]^n, \qquad \text{(b)} \end{aligned} \tag{1}$$

where x is the original input, δ is the perturbation, $\|\delta\|_p$ is the perturbation's l_p norm (a measure of the size of the perturbation), $\mathcal{C}$ is the classifier, and n is the dimensionality of the input space. The objective function seeks to minimize the l_p-norm of the perturbation δ, subject to the constraint (a) that the classifier produces a different class for the perturbed input $(x+\delta)$ compared to the original input x. The additional constraint (b) ensures that the perturbed input remains within the valid input domain, which is $[0,1]^n$ for normalized inputs.

Adversarial attack algorithms, such as Fast Gradient Sign Method (FGSM) [35], Projected Gradient Descent (PGD) [47], Carlini and Wagner (C&W) [17], and the recently-proposed Fast-minimum Norm (FMN) [75], differ in the specifics of how they solve this optimisation problem and take care of hyperparameter choices, adversarial starting points, and convergence computational complexity. Some attacks use gradient information of the model's loss function to efficiently compute adversarial perturbations, while others employ more advanced optimisation techniques. Despite these differences, the general goal of these attacks is to find an optimal perturbation that effectively fools the target model while minimising the size of the perturbation according to some chosen norm.

Adversarial ML attacks have been originally studied in the context of computer vision tasks. Therefore, the above considerations on minimal l_p-norm perturbations are, in general, a requirement to guarantee adversarial input is perceived as similar to the original input. Biggio and Roli [10] extend this observation and suggest to reason about high vs low confidence attacks, instead, with the former non-necessarily being bounded by minimal perturbation objectives. This reasoning is further exacerbated when we consider domains for which the notion of visual perception has no meaning, such as in software, network, or natural language

processing tasks. Here, in fact, it might not be important to minimize input perturbations but rather to satisfy other constraints, such as semantics [48, 73], which would affect the way in which input objects are manipulated. We discuss so-called problem space or realizable attacks in Section 3.5.

3.2 Poisoning Attacks

Poisoning attacks, in general, are a class of adversarial attacks that target the training phase of a machine learning model [19]. The main working mechanism of poisoning attacks involves injecting carefully crafted points that differ from the training distribution or mislabelled data into the training dataset. These poisoned samples are designed to manipulate the learning process, resulting in a compromised model that misbehaves or performs poorly when deployed.

In general, poisoning attacks can be formulated as an optimisation problem. The attacker aims to find the optimal poisoned samples that maximise the model's loss on a specific target or set of targets, subject to certain constraints. The optimisation problem can be expressed as follows:

$$\begin{aligned} \underset{\{x_i^*, y_i^*\}_{i=1}^N}{\text{maximize}} \quad & L(\theta^*, \{x_i^*, y_i^*\}_{i=1}^N) \\ \text{subject to} \quad & \theta^* = \underset{\theta}{\text{argmin}}\, L(\theta, \{x_i, y_i\}_{i=1}^M \cup \{x_i^*, y_i^*\}_{i=1}^N), \quad \text{(a)} \\ & \text{constraints on } \{x_i^*, y_i^*\}_{i=1}^N, \quad \text{(b)} \end{aligned} \tag{2}$$

where L is the model's loss function, θ represents the model's parameters, $\{x_i, y_i\}_{i=1}^M$ is the original training dataset, $\{x_i^*, y_i^*\}_{i=1}^N$ are the poisoned samples, and M and N are the sizes of the original and poisoned datasets, respectively. The first constraint (a) ensures that the poisoned model, represented by θ^*, is trained on the combined dataset of original and poisoned samples, and the second constraint (b) imposes limitations on the poisoned samples, such as adherence to certain data distributions or limitations on the degree of perturbation allowed. In the context of poisoning attacks, the loss function's intuition is as follows. High values of the loss function indicate that the attacker is successful in compromising the model's performance on a specific target or set of targets. The attacker aims to maximise the model's loss by injecting poisoned samples that mislead the model during the learning process. Conversely, low values of the loss function suggest that the attacker is not successful in affecting the model's performance on the target or set of targets. This means that the model is more robust against the poisoning attack and can still make correct predictions despite the presence of poisoned samples in the training data.

The exact methodology for solving this optimisation problem varies depending on the attack strategy and the constraints imposed on the poisoned samples. Some poisoning attacks use gradient-based methods or bilevel optimization techniques [19], while others employ more heuristic approaches. The overall goal is to craft poisoned samples that effectively compromise the target model while remaining inconspicuous and adhering to any imposed constraints.

3.3 Backdoor Attacks

Backdoor attacks, also known as backdoor or trojan attacks, are a type of poisoning attack that aims to embed a hidden malicious functionality (i.e., the backdoor) into a machine learning model during the training process. The main working mechanism of backdoor attacks involves injecting specially crafted samples, called triggering or trojaned samples, into the training dataset [2]. These samples contain a specific pattern (i.e., the trigger) that, when present in the input data, causes the compromised model to produce a predefined, incorrect output chosen by the attacker.

Similar to the broader poisoning attacks, backdoor attacks can in general be formulated as an optimisation problem, where the attacker's goal is to find the optimal set of triggering samples that maximise the success of the backdoor attack while minimising its detectability [102]. The optimisation problem can be expressed as follows:

$$\begin{aligned} \underset{\{x_i^*, y_i^*\}_{i=1}^N}{\text{maximize}} \quad & P_{\text{success}}(\theta^*, \{x_i^*, y_i^*\}_{i=1}^N) \\ \text{subject to} \quad & \theta^* = \underset{\theta}{\text{argmin}}\, L(\theta, \{x_i, y_i\}_{i=1}^M \cup \{x_i^*, y_i^*\}_{i=1}^N), \quad \text{(a)} \\ & \text{constraints on } \{x_i^*, y_i^*\}_{i=1}^N, \quad \text{(b)} \end{aligned} \tag{3}$$

where P_{success} is the probability of a successful backdoor attack, L is the model's loss function, θ represents the model's parameters, $\{x_i, y_i\}_{i=1}^M$ is the original training dataset, $\{x_i^*, y_i^*\}_{i=1}^N$ are the triggering samples, and M and N are the sizes of the original and triggering datasets, respectively. The first constraint (a) ensures that the backdoored model, represented by θ^*, is trained on the combined dataset of original and triggered samples, and the second constraint (b) imposes limitations on the triggering samples, such as restrictions on the trigger pattern or the degree of perturbation allowed.

The exact methodology for solving this optimisation problem varies depending on the attack strategy and the constraints imposed on the triggering samples. Some backdoor attacks use gradient-based methods or optimisation techniques [80], while others employ more heuristic approaches. The overall goal is to craft triggering samples that effectively implant the backdoor into the target model while remaining inconspicuous and satisfying to any imposed constraints.

3.4 Poisoning vs Backdoor Attacks

From a high-level perspective, both poisoning and backdoor attacks target the training phase of a machine learning model, but their objectives and attack mechanisms are different.

Poisoning attacks aim to degrade the overall performance of a machine learning model by injecting maliciously crafted samples into the training dataset. The attacker's goal is to cause the model to make incorrect predictions or perform poorly on a specific target or set of targets. In poisoning attacks, the attacker typically tries to maximise the model's loss on the targets by solving an optimization problem, which may involve modifying the input data or their labels.

On the other hand, backdoor attacks aim to insert a hidden vulnerability or "backdoor" into the machine learning model during training, which can be exploited later during inference. The attacker trains the model on a dataset containing carefully crafted samples with specific

[2] Supply chain attacks against ML model codebase are also theoretically possible, but not discussed here.

triggers or patterns, which are associated with an incorrect target label. The model learns to recognise these triggers and associate them with the target label. During inference, when the attacker provides an input containing the trigger, the model produces the incorrect target output, allowing the attacker to control the model's behaviour. In backdoor attacks, the attacker's goal is to maximise the probability of success P_{success} when the trigger is present while maintaining the model's performance on clean data [102].

In summary, poisoning attacks focus on degrading the overall performance of the model, while backdoor attacks aim to insert a hidden vulnerability that can be exploited later by the attacker. The formalisation of these attacks may be similar, but their objectives and attack mechanisms are different.

Recent comprehensive surveys on poisoning and backdoor attacks (and defenses) are provided by Tian et al. [89] and Ciná et al. [19].

3.5 Realisable Attacks

So far, we have discussed adversarial attacks that involve reasoning about perturbations in the feature space, the geometric space where input data is mapped into. This happens particularly when the input and feature spaces are closely related, as in computer vision tasks. In these tasks, the feature space is in fact derived from the input data (e.g., pixel values of an image), and a deep learning model learns hierarchical representations of these features at different layers.[3] When input and feature spaces are not closely related, it is necessary to reason about the input space on its own (the problem space) and different ways in which we can abstract and represent it. For instance, in the context of malware classification, one can define a feature mapping function that abstracts programs in a binary feature space or learns a suitable representation of programs [4]. Here, adversarial perturbations as outlined earlier affect the feature space and can thus influence a classifier's prediction. However, the underlying attacks would represent unrealistic threats as they would only affect the feature space and not the problem space with related constraints.

Realisable attacks are a category of adversarial attacks that focus on creating real adversarial objects. For instance, in malware classification tasks, realisable adversarial attacks are not just concerned at creating digital adversarial examples (an adversarial example that exists in the feature space and breaks the classifier), but also focused at generating an adversarial object that exists in the physical world (e.g., an "adversarial" malware that still behaves as such while being classified as goodware.) As such, these attacks may not require to minimise l_p-norm perturbations but might need to satisfy specific problem-space constraints. These attacks take into account the practical limitations and properties of the targeted domain, such

[3] Adversarial attacks and the training of deep learning models both involve backpropagation [78], but they have different goals and targets. (Although the focus here is on deep learning models, shallow learning-based algorithms are equally vulnerable to adversarial perturbations.) As mentioned earlier, the goal of adversarial attacks is to find input perturbations that cause the model to produce incorrect predictions while minimizing the perceptibility of the perturbations. In this case, the attacker backpropagates the gradient of the loss function with respect to the input data. By computing the gradient, the attacker can identify the direction in which the input should be perturbed to maximize the model's output error. Conversely, the goal of training a (deep) learning model is to learn its optimal parameters (weights and biases) that minimize the loss function over the training dataset. Here, the gradient of the loss function with respect to the model parameters is computed using backpropagation. This gradient information is then used to update the model parameters iteratively, typically via optimization algorithms like Stochastic Gradient Descent (SGD).

[4] The CyBoK AI for Security Topic Guide [76] and the Malware & Attack Technology Knowledge Area [50] offer additional information on the topic.

as the physical constraints of an adversarial image patch [14], the syntax and semantics of code transformations and [73], or natural language [48].

An adversarial image patch represents realisable attacks in computer vision tasks. These attacks require one to place a carefully designed patch on an object (e.g., a grayscale rectangle on a road sign) to cause a learning-based computer vision system to misclassify the object. These attacks differ from traditional adversarial examples as they do not require imperceptible perturbations; instead, they create a visible patch that blends into the scene, exploiting the model's vulnerabilities in a real-world context.

Adversarial patch attacks can be formalised as an optimisation problem, where the goal is to find the optimal patch that maximises the model's classification or detection error. The optimisation problem can be expressed as follows[5]:

$$\begin{aligned} \underset{P}{\text{maximize}} \quad & L(P, \theta) \\ \text{subject to} \quad & P \in \mathcal{P}, \end{aligned} \tag{4}$$

where, P represents the adversarial patch, θ represents the model parameters, $L(P, \theta)$ is a function measuring the loss of the classifier induced by the patch, and $\mathcal{P}$ denotes the set of valid patches, which may include constraints on the patch size, shape, and other properties.

To solve this optimization problem, attackers typically use gradient-based methods, such as gradient ascent or evolutionary algorithms, to iteratively update the patch. By solving this optimization problem, attackers can generate adversarial patches that effectively deceive deep learning models in real-world scenarios. These patches can be printed and placed on objects, causing the model to misclassify the object or fail to detect it, even though the patch is visible and not imperceptible like traditional adversarial examples.

Although adversarial attacks were initially explored in computer vision tasks, recent work by Pierazzi et al. [73] reformulated adversarial attacks to provide principled reasoning across domains where the input and feature spaces are not closely related. In this context, the feature mapping function is often neither invertible nor differentiable, necessitating an approach that deals with the inverse feature-mapping problem: it is insufficient to identify adversarial examples solely in the feature space; one must also comprehend how to project those points onto real objects and reason about the underlying implications.

Projecting adversarial points from the feature space back to the problem space introduces side-effect features as a byproduct of satisfying problem space constraints (e.g., preserving semantics and ensuring plausibility). Side-effect features exist to make the attack realistic, as they facilitate adherence to the inherent constraints of the problem space. This perspective broadens the understanding of adversarial attacks, encompassing not only computer vision but also other application domains where the relationship between input and feature spaces is less straightforward.

[5]This basic formulation can be extended to include additional challenges, such as environment conditions, spatial constraints, physical limits on imperceptibility, and fabrication errors as outlined in Eykholt et al. [30].

4 DEFENDING AGAINST ADVERSARIAL ML ATTACKS

The main categories of defenses against adversarial attacks can be broadly classified into the following main groups. Adversarial training, out-of-distribution detection and selective classification (also known as classification with rejection), and certified defenses; defenses against poisoning and backdoor attacks include data sanitisation, robust learning algorithms, and forensics analysis to identify and remove poisoned samples or backdoor triggers.

Without any loss of generality, it is worth noting that defending against adversarial attacks is a vibrant and open research field and it always requires reasoning about adaptive attackers [91].

4.1 Adversarial Training

Adversarial training (AT) is a widely used defense technique primarily focused against test-time adversarial attacks [35, 53]. The main idea behind AT is to augment the training dataset with adversarial examples, enabling the model to learn robust features and become more resistant to adversarial perturbations. The relation between AT, attack success rate (ASR), and the model's performance on clean data is crucial to understanding its effectiveness.

Formally, AT can be represented as a min-max optimisation problem:

$$\min_{\theta} \mathbb{E}(x, y) \sim D \left[\max \delta \in S, L(f_\theta(x + \delta), y) \right] \tag{5}$$

where θ denotes the model parameters, D represents the training data distribution, f_θ is the model, L is the loss function, and S is the set of allowed perturbations. In this equation, the objective is to find the optimal model parameters, represented by θ, that minimise the expected loss over the data distribution D, where each data point is a tuple (x, y) representing the input and its corresponding label. The loss function L measures the difference between the model's prediction, given by $f_\theta(x + \delta)$, and the true label y. The inner maximisation problem searches for the adversarial perturbation δ within a set S that maximises the loss. This ensures that the model is trained on adversarial examples, making it more robust to adversarial attacks. The outer minimisation problem aims to find the model parameters θ that provide the best performance against these adversarial examples.

In practice, AT can be applied with both feature-space and problem-space adversarial examples. Feature-space AT focuses on modifying the input data in the feature domain, while problem-space AT explores adversarial perturbations within the specific problem constraints. The choice between feature-space and problem-space AT depends on the application and the extent to which it is feasible to generate adversarial examples within the problem domain. Dyrmishi et al. have recently shown how reasoning on the benefits of feature-space as opposed to problem-space AT and vice versa is still an open problem [29].

Although promising, AT comes with challenges too. AT provides empirical robustness rather than theoretical guarantees, as certified models do (cfr. Section 4.3). Since only partial regions of the feature space can be explored exhaustively, AT robustness is generally tied to the underlying adversarial l_p-norm perturbations and attacks. It has also been observed that increasing robustness to adversarial attacks often leads to a decrease in accuracy on clean data as often clean and adversarial points lie close to each other, challenging the models' generalization [96].

Finally, AT is computationally expensive. Generating adversarial examples during training requires solving an inner optimisation problem, which can significantly increase the training time. Several approximations and techniques have been proposed to accelerate AT, such as using fast gradient methods [53] or leveraging transferability between models [93].

4.2 Out-of-Distribution (OOD) Detection and Adversarial Examples

Adversarial attacks and out-of-distribution (OOD) samples [6] are closely related, as both involve inputs that are different from the distribution of the training data. In the context of adversarial attacks, adversaries craft adversarial examples by applying intentional perturbations to the input data, aiming to cause misclassification [35, 87]. Depending on the domain, adversarial perturbations might not be minimal and attacks may need to deal with problem-space constraints [73]. On the other hand, OOD samples are instances that come from a different distribution than the one used to train the model, and they may naturally occur during the testing phase [9, 39, 59].

The connection between adversarial attacks and OOD samples is particularly evident when considering the challenges faced by machine learning models in detecting and handling both types of inputs. For instance, deep neural networks have been shown to be overconfident in their predictions for both adversarial examples and OOD samples, assigning high confidence scores to incorrect predictions [61]. Moreover, several studies have demonstrated that adversarial examples can be considered as extreme cases of OOD samples, lying near the decision boundaries of the model [96].

Recent research has explored the possibility of leveraging the similarities between adversarial attacks and OOD samples to design more effective defenses against both types of inputs. For example, adversarial training, which is commonly used as a defense against adversarial attacks, has been shown to improve the model's robustness to OOD samples [104].

In summary, the connection between adversarial attacks and OOD samples is an active area of research, with potential implications for the design of more robust and secure machine learning systems. By understanding and exploiting the similarities between these two phenomena, it is possible to develop more effective defenses and improve the model's performance on a wide range of challenging inputs.

4.3 Certified Models

Certified models aim to provide guarantees on the model's predictions within an l_p-norm hypersphere around the input samples. These models are designed to be robust against adversarial attacks within a predefined perturbation budget. One popular approach for constructing certified models is randomised smoothing [20]. The main idea of this technique is to add random noise to the input before passing it through the classifier and then averaging the model's predictions over multiple noisy samples. This process effectively smooths the decision boundary of the classifier, making it more robust against small perturbations, such as those introduced by adversarial attacks.

In more detail, given an input sample x, randomised smoothing generates multiple noisy samples by adding Gaussian noise with zero mean and a predefined variance. These noisy

[6] That is, data points that differ significantly from the training distribution.

samples are then passed through the classifier, and the final prediction (i.e., the certification) is obtained by averaging the predictions or taking a majority vote.

Although randomised smoothing is a test-time defense, it can be combined with other training-time defenses, such as adversarial training, to further improve the model's robustness against a wide range of adversarial attacks.

More formally, given a classifier f and an input x, the smoothed classifier g is defined as:

$$g(x) = \arg\max_{c \in C} \mathbb{P}(f(x + \epsilon) = c), \tag{6}$$

where C is the set of classes and ϵ is a random noise vector sampled from a Gaussian distribution. The certification radius r for an input x and a class c is then defined as the maximum l_p-norm distance such that with high probability, any perturbation within this distance will not change the model's prediction:

$$r(x, c) = \sup r \geq 0 : \mathbb{P}(f(x + \delta) \neq c) \leq \alpha, \forall \delta \in B_p(x, r), \tag{7}$$

where $B_p(x, r)$ denotes the l_p-norm ball centered at x with radius r, and α is a user-defined confidence level.

Certified models offer strong guarantees against adversarial attacks but have some limitations. For instance, they typically suffer from reduced performance on clean data and increased computational complexity [49]. Additionally, certification methods often require stronger assumptions about the threat model (e.g., the requirement to reason within a l_p-norm ball of perturbation), limiting their applicability to specific types of adversarial attacks. Despite these drawbacks, certified models represent an important step towards achieving robust and secure machine learning systems.

4.4 Defenses against Poisoning and Backdoor Attacks

Defending against poisoning and backdooring attacks involves detecting and mitigating the effects of maliciously manipulated training data. One approach to address these attacks is to employ forensics techniques to identify poisoned datasets [85]. Such techniques typically involve analysing the statistical properties of the training data, and looking for anomalies or patterns that indicate the presence of malicious samples.

For example, some forensic methods leverage unsupervised learning techniques, such as clustering, to group similar data points and identify potential outliers. Others use supervised learning to train a classifier that can distinguish between clean and poisoned samples based on their feature representations.

Recent comprehensive surveys on poisoning and backdoor attacks (and defenses) are provided by Tian et al. [89] and Ciná et al. [19].

5 PRIVACY IN MACHINE LEARNING

In this section, we consider how privacy issues arise in the context of Machine Learning. We refer the reader to the CyBOK Knowledge Area on Privacy and Online Rights [94] for a broader discussion of Privacy.

5.1 Inference about Members of the Population

One kind of privacy violation may come from an adversary, who, given some kind of access to a trained model, tries to learn *something* about the training data.

Statistical Disclosure. The adversary learns something about the input to the model; in theory, one would like to control statistical disclosure (this is also known as the "Dalenius desideratum" [21]), in that a model should reveal no more about the input to which it is applied than would have been known otherwise. However, as also pointed out in [82], this cannot be achieved by any useful model [27].

Model Inversion. An adversary can use the model's output to infer the values of sensitive attributes used as input to the model. Fredrikson et al. [31, 32] first showed how an attacker could rely on outputs from a classifier to infer sensitive features used as inputs to the model itself. Given the model and some demographic information about a patient whose records are used for training, an attacker might predict sensitive attributes of the patient.

Note that it may not be possible to prevent this if the model is based on statistical facts about the population. E.g., suppose that training the model has uncovered a high correlation between a person's observable features and their genetic predisposition to a certain disease; this correlation is now a publicly known fact that allows anyone to infer information about a person's genome [82].

Inferring Class Representatives. Model inversion can be generalised to an adversary, who, given some access to the model, infers features that characterise each class, making it possible to construct representatives of these classes [40, 72].

5.2 Inference about members of the training dataset

Next, we focus on the privacy of individuals whose data was used to train the model. Of course, members of the training dataset are members of the population, too. Therefore, one should focus on what the model reveals about them beyond what it reveals about an arbitrary member of the population.

5.2.1 Membership Inference Attacks (MIA)

The attack involves an adversary who, given a model and an exact data point, tries to infer whether or not that point was used to train the model.

MIA can directly violate privacy if inclusion in a training set is itself sensitive based on the nature of the task at hand. For example, if health-related records are used to train a classifier, discovering that a specific record was used for training inherently leaks information about the individual's health. Similarly, if images from a database of criminals are used to train a model predicting the probability that one will re-offend, successful membership inference exposes an individual's criminal history.

Overall, when a record is fully known to the adversary, learning that it was used to train a particular model is an indication of information leakage through the model. On the other hand, MIA can also be used by regulators to support the suspicion that a model was trained on personal data without an adequate legal basis or for a purpose not compatible with the data collection. For instance, DeepMind was found to have used personal medical records provided by the UK's National Health Service for purposes beyond direct patient care, the basis on which the data was collected [99].

Inference via overfitting. MIA against zero-knowledge machine learning models was first studied by Shokri et al. [82], in the context of supervised learning. They focus on classification models trained by commercial Machine Learning as a Service (MLaaS) providers, such as Google and Amazon, whereby a user has API access to a trained model. Their approach exploits differences in the model's response to inputs seen vs not seen during training. For each class of the targeted model, they train a *shadow model*, with the same machine learning technique; the intuition is that the model ends up "overfitting" on data used for training [82]. Overfitting is a modeling error that occurs when a function is too closely fit to a limited set of data points and thus performs better on the training inputs than on the inputs drawn from the same population but not used during the training. Therefore, the attacker can exploit the confidence values on inputs belonging to the same classes and learn to infer membership.

Follow-up work studying and/or going beyond overfitting includes [79, 95, 103]. Other researchers have also studied MIA in other contexts, such as generative models [38] and federated learning [56, 60].

5.2.2 Property Inference

As mentioned, model inversion aims to infer properties that characterise an entire class: for example, given a face recognition model where one of the classes is Bob, infer what Bob looks like (e.g., Bob wears glasses). By contrast, property inference focuses on the adversarial goal of inferring properties that are *true of a subset of the training inputs but not of the class as a whole*. For instance, when Bob's photos are used to train a classifier, can the attacker infer that Alice appears in some of the photos?

Melis et al. [56] focus on the properties that are *independent* of the class's characteristic features in the context of federated learning. In contrast to the face recognition example, where "Bob wears glasses" is a characteristic feature of an entire class, in their gender classifier study, they infer whether people in Bob's photos wear glasses—even though wearing glasses has no correlation with gender. There is no "legitimate" reason for a model to leak this information; it is purely an artifact of the learning process.

5.3 Inferring Model Parameters

Model owners often charge others for queries to their commercially valuable models. However, while the query interface of an ML model may be widely accessible, the model itself and the data on which it was trained may be proprietary and confidential. Moreover, for security applications such as spam or fraud detection, an ML model's secrecy is critical to its utility; an adversary that can learn the model can also often evade detection [5].

Model Extraction. The concept of model extraction (or stealing) is first explored by Tramer et al. [92]. In this kind of attack, an adversary with zero-knowledge access, but no prior knowledge of an ML model's parameters or training data, aims to steal the model parameters. The intuition behind their attack is to exploit the information-rich outputs returned by the ML prediction APIs, e.g., high-precision confidence values in addition to class labels.

Overall, Tramer et al. [92]'s work is focused on inferring model parameters. In follow-up work, other researchers have gone beyond inferring model parameters and perform hyperparameter stealing [100], architecture extraction [63], etc.

Functionality Stealing. Here, the goal here is to create "knock-offs" of the model (also known as surrogate models) solely based on input-output pairs observed from queries. In [65], Orekondy et al. do so solely based on input-output pairs observed from MLaaS queries. More specifically, the adversary interacts with a zero-knowledge "victim" by providing it with input images and obtaining respective predictions. The resulting image-prediction pairs are used to train a knock-off model, e.g., to compete with the victim model at the victim's task.

Additional work in this category includes [67], whereby the adversary trains a local model to substitute for a victim deep neural network (DNN), using inputs synthetically generated by an adversary and labeled by the target DNN.

5.4 Take Aways

Sensitive Training Data. There has been a very significant amount of research work on membership inference attacks against ML. Arguably, this is motivated by 1) the seriousness of the privacy risks stemming from such attacks, 2) the fact that MIA is often just a signal of leakage and can serve as a canary for broad privacy issues, and 3) the interesting challenges in making attacks more effective, less reliant on strong assumptions, etc.

Overall, several attacks have been proposed in the context of a wide variety of datasets (images, text, etc.), models (discriminative, generative, federated), as well as threat models (API access, perfect-knowledge, zero-knowledge, active, passive, etc.). Such attacks are realistic, but their effectiveness depends on the actual settings, e.g., the adversary's knowledge of records, model parameters, etc., and are likely to affect certain users more than others. Nonetheless, we are confident in arguing that practitioners and researchers must think hard about whether deploying ML models in the wild is a good idea whenever training data is sensitive, with regards to privacy concerns. Further work is needed to provide clear guidelines and usable tools for practitioners willing to provide access to trained models. More precisely, they should be enabled to fully understand the privacy risks for the users whose data is used for training in specific data/specific learning tasks.

Limitations of model inversion. Although research roughly falling in the "model inversion" category is important, we believe there are some limitations in what this means for privacy. Class members produced by model inversion and GANs are similar to the training inputs only

if all members of the class are similar, as is the case for MNIST (the dataset of handwritten digits used in [40]) and facial recognition. This does not violate the privacy of the training data; it simply shows that machine learning works as it should. A trained classifier reveals the input features characteristic of each class, thus enabling the adversary to sample from the class population.

Therefore, the informal property violated by such attacks is, roughly speaking: "a classifier should prevent users from generating an input that belongs to a particular class or even learning what such an input looks like." However, it is not clear why this property is desirable or whether it is even achievable.

Property inference needs further work. Overall, property inference attacks are not to be ignored, even though their effectiveness depends on the context. As mentioned earlier, inferring sensitive attributes is really a privacy breach when the attacker can confidently assess those attributes related to records in the training set, and even more so if they do not leak simply because the class which the model is learning to classify is strictly correlated.

The only "attack" we are aware of in this sense is that of Melis et al. [56], which has only been studied in the context of collaborative learning. Even in that case, the authors essentially show that the accuracy of the attack quickly degrades with an increasing number of participants. In fact, if this is large enough, then differentially private defenses based on the moments accountant method [2] (discussed in Section 6) can be used to thwart such attacks.

It remains, however, an open research question to investigate whether property inference attacks: 1) are possible, as per our definition, in non-collaborative learning settings and at scale and 2) can be thwarted in collaborative settings involving a small number of participants.

Overall, we can categorise privacy defenses against attacks discussed in this document based on the main tools they rely on. These include advanced *privacy-enhancing technologies* like cryptography and differential privacy as well approaches used as part of the learning process (mainly training) to reduce the information available to the adversary.

6 PRIVACY DEFENSES

6.1 Cryptography

Cryptography, and more precisely encryption, can be used to protect data confidentiality. There are two main primitives that are relevant in the context of ML and in general data analysis/processing: 1) *secure multi-party computation* (SMC), and 2) *fully homomorphic encryption* (FHE). SMC allows two or more parties to jointly compute a function over their inputs, while keeping those inputs hidden from each other. Typically, SMC protocols build on tools like garbled circuits, secret sharing, oblivious transfer (for a detailed overview of such tools, we refer the reader to **securecomputation.org**). Whereas FHE is an encryption scheme that allows processing of the underlying cleartext data while it still remains in encrypted form, without giving away the secret key. Additional details on Cryptography can be found in the Cryptography Knowledge Area [84].

Cryptography in ML can support confidential computing scenarios where, for instance, a server has a model trained on its private data and wishes to provide inferences (e.g., classification) on clients' private data. In this context, there are a number of research proposals and prototypes in the literature that allow the client to obtain the inference result without revealing their input

to the server while at the same time preserving the confidentiality of the server's model. For instance, privacy-enhancing tools based on secure multi-party computation (SMC) and fully homomorphic encryption (FHE) have been proposed to securely train supervised machine learning models, such as matrix factorisation [62], linear classifiers [12, 36], decision trees [13, 51], linear regressors [25], and neural networks [11, 34, 52, 57].

SMC has also been used to build privacy-preserving neural networks in a distributed fashion. For instance, SecureML [57] starts with the data owners (clients) distributing their private training inputs among two non-colluding servers during the setup phase; the two servers then use MPC to train a global model on the clients' encrypted joint data. Then, Bonawitz et al. [11] use secure multi-party aggregation techniques, tailored for federated learning, to let participants encrypt their updates so that the central parameter server only recovers the sum of the updates.

Confidentiality vs Privacy. Overall, cryptography in ML is really aimed at protecting *confidentiality*, rather than *privacy*, which constitutes the main focus of our report. The two terms are often confused, both in the context of ML and in general, but they actually refer to different properties. Confidentiality is an explicit design property whereby one party wants to keep information (e.g., training data, testing data, model parameters, etc.) hidden from both the public and other parties (e.g., clients with respect to servers or vice-versa). Whereas, for the sake of this guide, privacy is about protecting against *unintended* information leakage, whereby an adversary aims to infer sensitive information through some (intended) interaction with the victim. In other words, cryptographically-enforced confidential computing does not provide any guarantees about what the output of the computation reveals. Therefore, we will focus on privacy rather than confidentiality defenses.

6.2 Differential Privacy (DP)

What is Differential Privacy (DP)? DP addresses the paradox of learning nothing about an individual while learning useful information about a population [28]. Generally speaking, it provides rigorous, statistical guarantees against what an adversary can infer from learning the result of some randomised algorithm.

Typically, differentially private techniques protect the privacy of individual data subjects by adding random noise when producing statistics. In other words, DP guarantees that an individual will be exposed to the same privacy risk whether or not her data is included in a differentially private analysis.

Formally, for two non-negative numbers ϵ, δ, a randomised algorithm $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy if and only if, for any neighbouring datasets D and D' (i.e., differing at most one record), and for the possible output $S \subseteq Range(\mathcal{A})$, the following formula holds:

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in S] + \delta$$

The ϵ, δ parameters. Differential privacy analysis allows for some information leakage specific to individual data subjects, controlled by the privacy parameter ϵ. This measures the effect on each individual's information on the output of the analysis. With smaller values of ϵ, the dataset is considered to have stronger privacy, but less accuracy, thus reducing its utility. An intuitive description of the ϵ privacy parameter, along with examples, is available in [101].

Sensitivity. The notion of the *sensitivity* of a function is very useful in the design of differentially private algorithms. It is usually defined with respect to a neighbouring relationship. Given a query F on a dataset D, the sensitivity is used to adjust the amount of noise required for $F(D)$.

More formally, if F is a function that maps a dataset (in matrix form) into a fixed-size vector of real numbers, we can define the L_i-sensitivity of F as: $S_i(F) = \max_{D,D'} ||F(D) - F(D')||_i$, where $|| \cdot ||_i$ denotes the L_i norm, $i \in \{1, 2\}$ and D and D' are any two neighboring datasets.

DP in ML. The state-of-the-art method for providing access to information free from inferences is to satisfy differential privacy (DP) [26]. This applies to ML as well, and more precisely to providing access to models that have been trained on (sensitive) datasets [2, 66, 68, 71, 81]. More precisely, there are two main privacy-preserving model-training approaches in literature: 1) using noisy stochastic gradient descent (noisy SGD) [3], and 2) Private Aggregation of Teacher Ensembles (PATE) [66, 68].

Noisy SGD. ML models are often trained using stochastic gradient descent (SGD). The intuition is to add noise to the SGD process; however, the main challenge is to do so while ensuring that the noise is carefully calibrated. The sensitivity of the final value of θ (the parameter vector) to the elements of the training data is generally hard to analyse. On the other hand, since the training data affects θ only via the gradient computations, one may achieve privacy by bounding gradients (by clipping) and by adding noise to those computations. In a nutshell, this is the idea behind the seminal work by Abadi et al. [2], particularly in the accounting of privacy loss, hence it is generally referred to as the *Moments Accountant* algorithm.

PATE. To protect the privacy of training data during learning, PATE transfers knowledge from an ensemble of "teacher" models trained on data partitions to a "student" model. Intuitively, privacy is provided by training teachers on disjoint data, and strong guarantees stem from the noisy aggregation of teachers' answers.

Collaborative Learning. In the collaborative learning setting, Shokri and Shmatikov [81] support distributed training of deep learning networks in a privacy-preserving way. Specifically, their system relies on the input of independent entities which aim to collaboratively build a machine-learning model without sharing their training data. To this end, they selectively share subsets of noisy model parameters during training. Moreover, federated learning proposals tackle the problem of training deep learning models with differential privacy guarantees for the tasks of training language models [54] and digits classification [33].

6.3 Trusted Execution Environments

A different line of work focuses on privacy (as well as integrity) guarantees for ML computations in untrusted environments (i.e., tasks outsourced by a client to a remote server, including MLaaS) by leveraging so-called Trusted Execution Environments (TEEs), such as Intel SGX or ARM TrustZone. TEEs use hardware and software protections to isolate sensitive code from other applications while attesting to its correct execution. The main idea is that TEEs outperform purely cryptographic approaches by multiple orders of magnitude. For more details about Hardware Security, please refer to the corresponding CyBOK Knowledge Area [98].

In this area, there are three main approaches. The first includes work supporting oblivious data access patterns [64] and, in general, training for a range of ML algorithms run inside SGX [41, 42]. The second, by Tramer and Boneh [90], focuses on *high performance* execution of Deep Neural Networks (DNNs) in TEEs, by efficiently partitioning DNN computations between trusted and untrusted devices. The third, by Hanzlik et al. [37], is essentially a guarded offline deployment of MLaaS: models are executed locally on the client's side (therefore, the data never leaves the device).

6.4 ML-Specific Approaches

Finally, a number of ML techniques are used to reduce information available to the adversary to mount their attacks. For instance, *dropout* [8] is a popular technique often used to mitigate overfitting in neural networks; as such, this might reduce the effectiveness of MIAs based on overfitting.

Additional techniques in this space include weight normalisation (re-parameterisation of the weights vectors that decouples the length of those weights from their direction), dimensionality reduction (e.g., only using inputs that occur many times in the training data), selective gradient sharing (in collaborative learning, participants could share only a fraction of their gradients during each update), etc.

7 CONCLUSION

This Cybersecurity Body of Knowledge (CyBoK) Knowledge Guide (KG) presented an overview of security and privacy in Machine Learning (ML). We introduced and reasoned about adversarial ML and privacy challenges in ML, and discussed mitigation techniques. As mentioned, this KG is not meant to provide an exhaustive survey of the problem space but rather to help readers familiarise themselves with some of the most important notions related to security and privacy in ML. While research in this space is making tremendous and fast-paced progress year after year, there are still a number of important open research problems, both inherently in the ML security/privacy context and with respect to adjacent issues.

With respect to security, despite promising directions outlined in Section 4 and open source libraries [1, 74, 88], detecting adversarial and out of distribution (OOD) examples (commonly referred to as *adversarial drift* [45, 48, 69, 73] remains an open problem [9, 59, 104]. This challenge is further exacerbated by the fact that *security is inherently adversarial* and thus it requires reasoning about adaptive attackers [91] and realistic threat models [4]. The role of representations is fundamental, in terms of the way in which we abstract raw data to create or learn embeddings. However, there is a lack of clarity around how different representations impact the entire ML pipeline in terms of performance, explainability, robustness to adversarial drift and more general trustworthiness properties.

Similarly, in the privacy space, while techniques that guarantee Differential Privacy (DP) can be used to minimise privacy concerns, these are neither easy to use for non-experts nor do they provide a one-size-fits-all solution. For instance, their effectiveness strongly depends on the specific learning task at hand, data distributions, etc. It is also unclear how to reliably quantify the inherent utility-privacy tradeoffs, what exactly the ϵ parameter means in practice, or how to set it.

The implication of the notions covered in this KG vis-à-vis ethical, societal, and legal aspects is also under-explored. For instance, while data protection regulations like the GDPR heavily focus on the concept of identification, what it means for a person to be “identified, directly or indirectly” is unclear. Clearly, more work is needed linking up privacy attacks (and defenses) with regulation and data protection efforts. There are also, at times, conflicting objectives stemming from defenses – e.g., using DP techniques to protect privacy can yield disparate impact on accuracy for underrepresented classes such as minorities [6].

Overall, the community of researchers and practitioners in ML/AI security and privacy is growing and working at an unprecedented pace; we are looking forward to progress and new

results in the broader space over the next few years.

REFERENCES

[1] Cleverhans. **https://github.com/cleverhans-lab/cleverhans**.

[2] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[3] M. Abadi, U. Erlingsson, I. Goodfellow, H. B. McMahan, I. Mironov, N. Papernot, K. Talwar, and L. Zhang. On the protection of private information in machine learning systems: Two recent approches. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 1–6. IEEE, 2017.

[4] G. Apruzzese, H. S. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. A. Roundy. Position: "real attackers don't compute gradients": Bridging the gap between adversarial ML research and practice. In *First IEEE Conference on Secure and Trustworthy Machine Learning*, pages 339–364, 2023.

[5] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015.

[6] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

[7] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[8] P. Baldi and P. J. Sadowski. Understanding dropout. *Advances in neural information processing systems*, 26, 2013.

[9] F. Barbero, F. Pendlebury, F. Pierazzi, and L. Cavallaro. Transcending transcend: Revisiting malware classification in the presence of concept drift. In *IEEE Symposium on Security and Privacy*, 2022.

[10] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognit.*, 84:317–331, 2018.

[11] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[12] J. W. Bos, K. Lauter, and M. Naehrig. Private Predictive Analysis on Encrypted Medical Data. *Journal of Biomedical Informatics*, 50:234–243, 2014.

[13] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine learning classification over encrypted data. Technical report, Cryptology ePrint Archive Report 2014/331, 2014.

[14] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017.

[15] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[16] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[17] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[18] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[19] A. E. Cinà , K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B. A. Moser, A. Oprea, B. Biggio, M. Pelillo, and F. Roli. Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Computing Surveys*, mar 2023.

[20] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019.

[21] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15(429-444), 1977.

[22] H. Debar. *The Cyber Security Body of Knowledge v1.1.0, 2021*, chapter Security Operations & Incident Management. University of Bristol, 2021. KA Version 1.0.2.

[23] S. H. H. Ding, B. C. M. Fung, and P. Charland. Asm2vec: Boosting static representation robustness for binary clone search against code obfuscation and compiler optimization. In *IEEE S& P*, 2019.

[24] M. Du, F. Li, G. Zheng, and V. Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[25] W. Du, Y. S. Han, and S. Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 222–233. SIAM, 2004.

[26] C. Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[27] C. Dwork and M. Naor. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *Journal of Privacy and Confidentiality*, 2(1), 2010.

[28] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[29] S. Dyrmishi, S. Ghamizi, T. Simonetto, Y. L. Traon, and M. Cordy. On the empirical effectiveness of unrealistic adversarial hardening against realistic adversarial attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1384–1400, Los Alamitos, CA, USA, may 2023. IEEE Computer Society.

[30] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust Physical-World Attacks on Deep Learning Visual Classification. In *Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[32] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart. Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In *23rd USENIX security symposium (USENIX Security 14)*, pages 17–32, 2014.

[33] R. C. Geyer, T. Klein, and M. Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint 1712.07557*, 2017.

[34] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*, pages 201–210. PMLR, 2016.

[35] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[36] T. Graepel, K. Lauter, and M. Naehrig. Ml confidential: Machine learning on encrypted data. In *International conference on information security and cryptology*, pages 1–21. Springer, 2012.

[37] L. Hanzlik, Y. Zhang, K. Grosse, A. Salem, M. Augustin, M. Backes, and M. Fritz. MLCapsule: Guarded offline deployment of machine learning as a service, 2021.

[38] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. LOGAN: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):33–152, 2019.

[39] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[40] B. Hitaj, G. Ateniese, and F. Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

[41] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel. Chiron: Privacy-preserving machine learning as a service. *arXiv:1803.05961*, 2018.

[42] N. Hynes, R. Cheng, and D. Song. Efficient deep learning on multi-source private data. *arXiv:1807.06689*, 2018.

[43] J. Jang, D. Brumley, and S. Venkataraman. Bitshred: feature hashing malware for scalable triage and semantic analysis. In *Proceedings of the 2011 ACM SIGSAC Conference on Computer and Communications Security*, 2011.

[44] A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar. *Adversarial Machine Learning*. Cambridge University Press, 2019.

[45] A. Kantchelian, S. Afroz, L. Huang, A. C. Islam, B. Miller, M. C. Tschantz, R. Greenstadt, A. D. Joseph, and J. D. Tygar. Approaches to adversarial drift. In *Proceedings of the*

2013 ACM Workshop on Artificial Intelligence and Security, AISec '13, page 99–110, New York, NY, USA, 2013. Association for Computing Machinery.

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet: Classification with deep convolutional neural networks. In *Advances in Neural Information Proccessing Systems (NIPS)*, 2012.

[47] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017.

[48] E. LaMalfa and M. Kwiatkowska. The king is naked: on the notion of robustness for natural language processing. In *Association for the Advancement of Artificial Intelligence*, 2021.

[49] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[50] W. Lee. *The Cyber Security Body of Knowledge v1.0, 2019*, chapter Malware & Attack Technology. University of Bristol, 2019. KA Version 1.0.

[51] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Annual International Cryptology Conference*, pages 36–54. Springer, 2000.

[52] J. Liu, M. Juuti, Y. Lu, and N. Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 619–631, 2017.

[53] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[54] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private language models without losing accuracy. *CoRR*, abs/1710.06963, 2017.

[55] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021.

[56] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy*, pages 691–706, 2019.

[57] P. Mohassel and Y. Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE, 2017.

[58] C. Molnar. *Interpretable Machine Learning*. 2nd edition, 2022.

[59] J. G. Moreno-Torres, T. Raeder, R. Alaiz-RodríGuez, N. V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern Recogn.*, 45(1):521–530, jan 2012.

[60] M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*, pages 739–753, 2019.

[61] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High

confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015.

[62] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft. Privacy-preserving ridge regression on hundreds of millions of records. In *2013 IEEE symposium on security and privacy*, pages 334–348. IEEE, 2013.

[63] S. J. Oh, B. Schiele, and M. Fritz. Towards reverse-engineering black-box neural networks. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 121–144, 2019.

[64] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa. Oblivious multi-party machine learning on trusted processors. In *USENIX Security*, pages 619–636, 2016.

[65] T. Orekondy, B. Schiele, and M. Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4954–4963, 2019.

[66] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net, 2017.

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

[68] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson. Scalable private learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net, 2018.

[69] F. Pendlebury. *Machine Learning for Security in Hostile Environments*. PhD thesis, Royal Holloway, University of London, 2021.

[70] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time. In *28th USENIX Security Symposium*, Santa Clara, CA, 2019. USENIX Association. USENIX Sec.

[71] N. Phan, X. Wu, and D. Dou. Preserving differential privacy in convolutional deep belief networks. *Machine learning*, 106(9-10):1681–1704, 2017.

[72] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. Privacy-preserving deep learning: Revisited and enhanced. In *Applications and Techniques in Information Security: 8th International Conference, ATIS 2017, Auckland, New Zealand, July 6–7, 2017, Proceedings*, pages 100–110. Springer, 2017.

[73] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro. Intriguing properties of adversarial ml attacks in the problem space. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1308–1325. IEEE Computer Society, 2020.

[74] M. Pintor, L. Demetrio, A. Sotgiu, M. Melis, A. Demontis, and B. Biggio. secml: Secure and explainable machine learning in python. *SoftwareX*, 18:101095, 2022.

[75] M. Pintor, F. Roli, W. Brendel, and B. Biggio. Fast minimum-norm adversarial attacks through adaptive norm constraints. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang,

and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20052–20062. Curran Associates, Inc., 2021.

[76] M. Rhode. AI for Security Topic Guide. In A. Rashid, Y. Cherdantseva, A. Martin, and S. Schneider, editors, *CyBOK Knowledge Guides and Topic Guides*. University of Bristol, 2023. TG Version 1.0.0.

[77] C. Rossow and S. Jha. *The Cyber Security Body of Knowledge v1.1.0, 2021*, chapter Network Security. University of Bristol, 2021. KA Version 2.0.0.

[78] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[79] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[80] G. Severi, J. Meyer, S. Coull, and A. Oprea. Explanation-Guided backdoor poisoning attacks against malware classifiers. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1487–1504. USENIX Association, Aug. 2021.

[81] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[82] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[83] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[84] N. Smart. *The Cyber Security Body of Knowledge v1.1.0, 2021*, chapter Cryptography. University of Bristol, 2021. KA Version 1.0.1.

[85] J. Steinhardt, P. W. Koh, and P. Liang. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3520–3532, Red Hook, NY, USA, 2017. Curran Associates Inc.

[86] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.

[87] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[88] The Linux Foundation®. Adversarial robustness toolbox: A python library for ml security. **https://adversarial-robustness-toolbox.org/**.

[89] Z. Tian, L. Cui, J. Liang, and S. Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Comput. Surv.*, 55(8), dec 2022.

[90] F. Tramèr and D. Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net, 2019.

[91] F. Tramer, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1633–1645. Curran Associates, Inc., 2020.

[92] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Stealing machine learning models via prediction APIs. In *USENIX Security*, 2016.

[93] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

[94] C. Troncoso. *The Cyber Security Body of Knowledge v1.1.0, 2021*, chapter Privacy & Online Rights. University of Bristol, 2021. KA Version 1.0.2.

[95] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. Towards demystifying membership inference attacks. *arXiv:1807.09173*, 2018.

[96] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[97] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[98] I. Verbauwhede. *The Cyber Security Body of Knowledge v1.1.0, 2021*, chapter Hardware Security. University of Bristol, 2021. KA Version 1.0.1.

[99] J. Vincent. Google DeepMind will use machine learning to spot eye diseases early. **https://www.theverge.com/2016/7/5/12095830/google-deepmind-nhs-eye-disease-detection**, 2016.

[100] B. Wang and N. Z. Gong. Stealing hyperparameters in machine learning. In *IEEE Symposium on Security and Privacy*, pages 36–52, 2018.

[101] A. Wood, M. Altman, A. Bembenek, M. Bun, M. Gaboardi, J. Honaker, K. Nissim, D. R. O'Brien, T. Steinke, and S. Vadhan. Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21:209, 2018.

[102] L. Yang, Z. Chen, J. Cortellazzi, F. Pendlebury, K. Tu, F. Pierazzi, L. Cavallaro, and G. Wang. Jigsaw puzzle: Selective backdoor attack to subvert malware classifiers. In *IEEE Symposium on Security and Privacy*, volume abs/2202.05470, 2023.

[103] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[104] M. Yi, L. Hou, J. Sun, L. Shang, X. Jiang, Q. Liu, and Z. Ma. Improved ood generalization via adversarial training and pretraing. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11987–11997. PMLR, 18–24 Jul 2021.

[105] S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38, 2019.

[106] Y. Zhu, D. Xi, B. Song, F. Zhuang, S. Chen, X. Gu, and Q. He. Modeling users' behavior sequences with hierarchical explainable network for cross-domain fraud detection. In *WWW '20: Proceedings of The Web Conference 2020*, 2020.