



AI SECURITY: Defending Machine Learning Against Adversarial Attacks

*Ashitha^a, Prof. Riya Prakash^b, Dr. L. C. Manikandan^c **

^aB.Tech. Student, Department of CSE, Thrissur, Kerala, India. E-mail: ashitha.code@gmail.com

^bAssistant Professor, Department of CSE, Universal Engineering College, Thrissur, Kerala, India. E-mail: riyajothish13@gmail.com

^cProfessor, Department of CSE, Universal Engineering College, Thrissur, Kerala, India. E-mail: lcmanikandan@gmail.com

* Corresponding author: Dr. L.C.Manikandan

E-mail: lcmanikandan@gmail.com

ABSTRACT:

Artificial Intelligence (AI) and Machine Learning (ML) have become important technologies in applications such as healthcare, autonomous systems, cybersecurity, image recognition and natural language processing. However, the increasing dependence on ML models has also introduced new security vulnerabilities. Adversarial attacks are deliberate manipulations designed to cause machine-learning systems to produce incorrect or unintended outputs. These attacks can target different stages of the machine-learning lifecycle, including training data, model development and inference. This paper presents an overview of adversarial attacks against machine-learning systems and examines major defense mechanisms used to improve model robustness. The study discusses adversarial examples, evasion attacks, poisoning attacks, black-box attacks, adversarial attacks against reinforcement learning and other emerging threats. Various defense techniques, including adversarial training, robust optimization, input preprocessing, anomaly detection and model-level protection, are also examined. The paper further discusses challenges such as attack transferability, computational cost, robustness-accuracy trade-offs and the limitations of existing defenses. The review highlights the importance of adopting a layered AI-security approach in which machine-learning models are continuously evaluated, monitored and protected throughout their lifecycle.

Keywords: Artificial Intelligence, Machine Learning, AI Security, Adversarial Attacks, Adversarial Examples, Adversarial Defense, Machine Learning Security

INTRODUCTION

Artificial Intelligence (AI) and Machine Learning (ML) have rapidly developed into important technologies for solving complex problems across different domains. Machine-learning models are now used in image recognition, cybersecurity, healthcare, autonomous vehicles, financial systems and many other applications. Although these systems can achieve high levels of accuracy, their dependence on learned patterns makes them vulnerable to deliberate manipulation. Attackers can construct inputs or manipulate data in ways that cause an ML model to produce incorrect results while the input may appear normal to a human observer [2].

The security weaknesses of machine-learning models became particularly evident with the discovery of adversarial examples. Szegedy et al. demonstrated that carefully designed, small perturbations to input samples could cause neural networks to make incorrect predictions [9]. Goodfellow et al. later investigated the phenomenon and showed that adversarial examples could be generated efficiently using the gradient information of neural networks [10]. These findings established adversarial machine learning as an important area of AI security.

Adversarial attacks can occur during different stages of the machine-learning lifecycle. During deployment, attackers may manipulate inputs to evade detection or cause incorrect classification. During training, attackers may manipulate training data to influence the behavior of the resulting model. Other attacks attempt to exploit the model through black-box queries, transfer adversarial examples between models, or compromise AI-driven systems through broader cybersecurity vulnerabilities [1], [6].

Therefore, AI security cannot be limited to protecting the infrastructure on which an ML model runs. The model itself, its training data, inputs, outputs and learning process must also be considered from a security.

2. BACKGROUND OF ADVERSARIAL MACHINE LEARNING

Adversarial Machine Learning refers to the study of attacks and defenses involving machine-learning systems. Unlike conventional software attacks, adversarial ML attacks exploit characteristics of the learning process itself. An attacker may intentionally modify data or inputs to influence the predictions, decisions or behavior of an ML model [1]. The attacker's objective can vary depending on the application. In an image-classification system, an attacker may attempt to make a model classify a stop sign as another object. In cybersecurity, malicious software may be modified so that an ML-

based detection system classifies it as benign. In autonomous systems, manipulated sensor information could potentially affect decisions made by an intelligent agent. Adversarial attacks can be categorized according to the stage at which they occur, the attacker's knowledge of the target model and the attacker's objective. Common categories include evasion attacks, poisoning attacks, black-box attacks and attacks against reinforcement-learning systems [1], [3], [6]. Table 1 shows the Conventional ML Vs Adversarial ML.

Table 1. Conventional ML Vs Adversarial ML

Aspects	Conventional ML	Adversarial ML
Input	Generally trusted	Maybe manipulated
Training data	Assumed clean	Maybe poisoned
Attacker	Often ignored	Explicitly considered
Goal	High prediction accuracy	Accuracy + Robustness
Security Focus	Limited	Central concern

2.1 Adversarial Examples

An adversarial example is an input that has been intentionally modified to cause a machine-learning model to make an incorrect prediction. The modification can be extremely small and may not be easily noticeable to humans. Szegedy et al. initially demonstrated this unexpected behaviour in neural networks [9]. Goodfellow et al. explained adversarial examples using the linear behaviour of high-dimensional models and proposed an efficient method for generating them known as the Fast Gradient Sign Method (FGSM) [10]. These findings demonstrated that high classification accuracy on normal test data does not necessarily guarantee robustness against maliciously constructed inputs.

The existence of adversarial examples presents a major challenge for security-sensitive AI applications because an attacker does not necessarily need to modify the model itself. Manipulating the input may be sufficient to influence the final decision.

3. TYPES OF ADVERSARIAL ATTACKS

Adversarial attacks against machine learning systems can be classified according to the stage of the machine learning lifecycle that is targeted, the attacker's objective, and the level of access available to the attacker. The two major stages targeted by adversaries are the training phase and the inference phase. During training, attackers may manipulate the data used to construct the model, whereas during inference they may modify input samples to influence the model's predictions [1], [2].

Common adversarial attacks include evasion attacks, data poisoning attacks, backdoor attacks, model extraction attacks, and inference attacks. Each attack exploits a different weakness in an ML system and therefore requires appropriate security mechanisms. Understanding these attack categories is essential for designing effective defense strategies. Classification of major adversarial attacks shown in Table 2.

Table 2. Classification of major adversarial attacks

Attack type	Target Stage	Main Objective	Typical Examples
Evasion attack	Inference	Cause incorrect predictions	PGSM, PGD
Data poisoning	Training	Manipulate model learning	Label/Feature poisoning
Backdoor attack	Training	Implant hidden behavior	Trigger-based attack
Model extraction	Inference/API	Reproduce model behavior	Query-based extraction
Inference attack	Inference	Obtain sensitive information	Membership inference

3.1 Evasion Attacks

Evasion attacks are performed during the inference stage, after an ML model has been trained and deployed. The attacker modifies a legitimate input in a way that causes the model to produce an incorrect prediction while attempting to keep the modification small or difficult to detect [1], [3].

A common example is an adversarial example, where a carefully designed perturbation is added to an input such as an image. Although the resulting image may appear almost unchanged to a human observer, the ML model may classify it incorrectly. Gradient-based methods such as the Fast Gradient Sign Method (FGSM) generate adversarial examples by using information about the model's gradients [4]. More powerful iterative approaches, such as Projected Gradient Descent (PGD), repeatedly modify the input while keeping the perturbation within a specified constraint [5].

Evasion attacks are particularly concerning in applications where incorrect predictions can have serious consequences. For example, an adversarial modification to an object recognized by an autonomous vehicle could potentially cause the system to misinterpret its surroundings. Therefore, improving the robustness of models against carefully crafted inputs is an important goal of adversarial ML research.

3.2 Data Poisoning Attacks

Data poisoning attacks target the training phase of a machine learning system. In these attacks, an adversary introduces manipulated or malicious samples into the training dataset with the intention of influencing the behaviour of the trained model [6].

The attacker may modify the features or labels of training samples, add specially constructed data, or manipulate a portion of the training dataset. Depending on the attack objective, poisoning can cause a general reduction in model performance or produce specific incorrect behaviour. Since modern ML systems often depend on large datasets collected from multiple sources, ensuring that training data is trustworthy can be challenging [1], [6].

Poisoning attacks demonstrate that the security of a trained model is closely connected to the integrity of its training data. Even when the model architecture and learning algorithm are secure, malicious modifications to the data can potentially compromise the final model. Data validation, dataset sanitization, anomaly detection, and robust training methods can therefore play an important role in reducing poisoning risks.

3.3 Backdoor Attacks

Backdoor attacks are a specialized form of training-time attack in which an attacker attempts to implant hidden behaviour into a machine learning model. The attacker introduces specially designed training samples containing a particular trigger and associates those samples with a desired target output [7], [8].

After training, the model may behave normally on clean inputs but produce an attacker-controlled prediction when the trigger is present. This makes backdoor attacks particularly difficult to identify because conventional testing on clean datasets may show that the model performs normally.

For example, in an image-classification system, an attacker could introduce a small visual pattern into selected training images and associate them with a particular class. When the same pattern appears in a future input, the compromised model may classify the input as the attacker's chosen class. Backdoor attacks therefore pose a significant threat to ML systems whose training data or pre-trained models originate from untrusted sources [7], [8].

3.4 Model Extraction Attacks

Model extraction attacks attempt to obtain information about a machine learning model by repeatedly interacting with it. These attacks are particularly relevant to ML models provided through online APIs or other query-based services. An attacker can submit carefully selected inputs and observe the model's outputs to construct a substitute model that approximates the behaviour of the target model [9].

The main concern associated with model extraction is the potential loss of intellectual property and the exposure of proprietary model behaviour. A successful extraction attack may also make other attacks easier because the attacker can use the substitute model to develop adversarial examples and then transfer them to the original system [9], [10].

Limiting excessive queries, monitoring unusual query patterns, and applying appropriate access-control mechanisms can help reduce the risk of model extraction. However, designing these protections while maintaining convenient access to legitimate users remains a challenge.

3.5 Inference Attacks

Inference attacks attempt to obtain sensitive information about a machine learning model or its training data by analysing the model's behaviour. One important example is the membership inference attack, in which an adversary attempts to determine whether a particular data record was included in the model's training dataset [11].

Such attacks can reveal information that was intended to remain private. For example, determining whether a person's medical record was part of a healthcare dataset could itself reveal sensitive information. The risk can become greater when models behave differently on training samples compared with previously unseen samples, particularly when the model has overfitted its training data [11].

Inference attacks demonstrate that protecting an ML system requires more than preventing incorrect predictions. Privacy must also be considered when models are trained using sensitive or personally identifiable data. Techniques such as differential privacy, regularization, and limiting information exposed through model outputs can help reduce these risks.

Comparison of Adversarial Attack Characteristics shown in Table 3.

Table 3. Comparison of Adversarial Attack Characteristics

Attack	Attacker action	Primary impact	Detection difficulty
Evasion	Modifies input	Misclassification	High
Poisoning	Manipulates training data	Model degradation/manipulation	High
Backdoor	Inserts trigger samples	Targeted malicious behavior	Very high
Model Extraction	Sends repeated queries	Model/IP leakage	Medium-High
Inference	Analyzes model outputs	Privacy leakage	High

4. ADVERSARIAL ATTACK TECHNIQUES

Adversarial attacks can be implemented using different techniques depending on the type of machine learning model, the attacker's knowledge, and the desired outcome. In the case of deep learning, many attack techniques exploit the gradients, decision boundaries, or training process of the target model. Some techniques generate small perturbations to existing inputs, while others manipulate training data to influence the learned model. The following subsections describe several widely studied adversarial attack techniques. Comparison of major adversarial attack techniques shown in Table 4.

4.1 Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is a gradient-based technique introduced by Goodfellow *et al.* for generating adversarial examples efficiently [4]. The method is based on the observation that neural networks can be vulnerable to small, carefully directed changes in their input.

FGSM uses the gradient of the loss function with respect to the input to determine the direction in which the input should be modified. The perturbation is then added to the original input according to a predefined magnitude. The adversarial example can be represented as:

$$x' = x + \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \quad \dots\dots\dots (1)$$

where x represents the original input, y is the correct label, θ represents the model parameters, J is the loss function, and ϵ controls the magnitude of the perturbation [4].

The major advantage of FGSM is its simplicity and computational efficiency. However, because it generally performs a single update, stronger iterative attacks may be able to overcome defenses that are effective against FGSM.

4.2 Projected Gradient Descent (PGD)

Projected Gradient Descent (PGD) is an iterative extension of gradient-based adversarial attacks and is considered a strong first-order attack [5]. Instead of generating the perturbation in a single step, PGD repeatedly updates the input in the direction that increases the model's loss.

After each update, the modified input is projected back into a predefined region around the original input. This ensures that the perturbation remains within the allowed constraint. A simplified update can be represented as:

$$x^{t+1} = \Pi_{\mathcal{S}} [x^t + \alpha \text{sign}(\nabla_x J(\theta, x^t, y))] \quad \dots\dots\dots (2)$$

where α represents the step size and Π represents the projection operation that keeps the adversarial example within the specified perturbation region [5]. Because PGD explores the input space through multiple iterations, it can generate stronger adversarial examples than single-step approaches. It is therefore frequently used to evaluate the robustness of machine learning models and adversarial defense methods.

4.3 Carlini and Wagner (C&W) Attack

The Carlini and Wagner (C&W) attack is an optimization-based adversarial attack designed to generate highly effective adversarial examples [12]. Unlike simple gradient-sign methods, the C&W approach formulates adversarial example generation as an optimization problem.

The attack attempts to find a perturbation that causes the model to produce an incorrect or attacker-selected prediction while minimizing the size of the perturbation. Different distance measures, including L_0 , L_2 , and L_∞ , can be used to measure the difference between the original and adversarial inputs [12].

C&W attacks have demonstrated the ability to bypass several defensive techniques and can produce relatively small perturbations. However, their optimization process is generally more computationally expensive than simpler methods such as FGSM.

4.4 DeepFool

DeepFool is an iterative technique designed to find a small perturbation capable of changing the prediction of a classifier [13]. The method approximates the decision boundaries of the classifier and determines the direction in which the input should be moved to cross the nearest boundary.

The process begins with the original input and repeatedly calculates a perturbation that moves it closer to a decision boundary. Once the classification changes, the resulting input is considered an adversarial example.

A major objective of DeepFool is to identify perturbations with relatively small magnitude. This makes the technique useful for studying the sensitivity of classification models and evaluating their robustness against subtle input modifications [13].

4.5 Data Poisoning Techniques

Unlike FGSM, PGD, C&W, and DeepFool, which primarily manipulate inputs during inference, data poisoning techniques target the training process. The attacker modifies or inserts malicious samples into the training dataset before the model is trained [6].

Poisoning can be performed by changing the features or labels of legitimate training samples or by injecting specially constructed samples. The objective may be to reduce the overall accuracy of the model, cause specific misclassifications, or influence the model's decision boundary.

The effectiveness of poisoning depends on factors such as the proportion of poisoned samples, their placement within the feature space, and the learning algorithm used by the target model. Since modern ML systems may obtain training data from multiple external sources, ensuring data integrity is an important part of defending against poisoning attacks.

4.6 Backdoor Attack Techniques

Backdoor attacks involve deliberately introducing a hidden association between a particular trigger and a target output during the training process [7], [8]. The attacker typically inserts poisoned samples containing the trigger into the training dataset and assigns them a chosen target label.

Once the model has learned the association, it may behave normally on clean inputs but produce the attacker's desired output whenever the trigger is present. This makes backdoor attacks particularly difficult to identify using conventional accuracy testing.

Backdoor techniques can use different types of triggers, including small image patterns, patches, modifications to text, or other features depending on the application. Effective defense therefore requires examining both the training data and the learned behavior of the model.

Table 4: Comparison of major adversarial attack techniques

Technique	Attack Category	Main principle	Iterative?
FGSM	Evasion	Uses gradient sign to modify input	No
PGD	Evasion	Repeated gradient-based perturbation	Yes
C&W	Evasion	Optimization-based perturbation	Yes
DeepFool	Evasion	Moves input toward decision boundary	Yes
Data Poisoning	Poisoning	Manipulates training data	Depends on attack
Backdoor attack	Backdoor	Associates trigger with target output	Training-based

5. DEFENSE MECHANISMS AGAINST ADVERSARIAL ATTACKS

The increasing number of adversarial attacks has led to the development of various defense mechanisms for improving the robustness and security of machine learning models. No single defense method can provide complete protection against every type of adversarial attack because different attacks exploit different weaknesses in an ML system. Defense mechanisms can therefore be applied at different stages, including data preparation, model training, input processing, and inference. Common approaches include adversarial training, input preprocessing, defensive distillation, robust optimization, attack detection, data sanitization, and certified defenses [1], [14]. Comparison of defense mechanisms shown in Table 5.

5.1 Adversarial Training

Adversarial training is one of the most widely studied approaches for improving the robustness of machine learning models. In this approach, adversarial examples are generated during the training process and included along with legitimate training samples. The model is then trained to correctly classify both clean and adversarial inputs [5], [15].

The main objective of adversarial training is to expose the model to potential attacks before deployment. Instead of learning only from naturally occurring examples, the model learns to perform correctly under deliberately perturbed conditions. A commonly used formulation involves optimizing the model against the worst-case perturbation within a predefined region:

$$\min_{\theta} E(x, y) [\max_{\delta \in S} L(\theta, x + \delta, y)] \quad \dots\dots\dots (3)$$

where θ represents the model parameters, δ represents an adversarial perturbation, S defines the allowed perturbation region, and L represents the loss function [5].

Although adversarial training can significantly improve robustness, it can also increase computational requirements because adversarial examples must be generated during training. It may also result in a trade-off between robustness and performance on clean data.

5.2 Input Preprocessing

Input preprocessing methods attempt to reduce or eliminate adversarial perturbations before the input is provided to the machine learning model. The basic idea is to transform the input into a form that preserves useful information while reducing characteristics introduced by an adversarial attack [14]. Common preprocessing approaches include image resizing, feature squeezing, noise reduction, compression, and other input transformations. For example, feature squeezing reduces the complexity of input representations by limiting the range of features or reducing their precision. This can make certain adversarial perturbations less effective [16].

The main advantage of preprocessing is that it can be implemented without completely retraining the original model. However, attackers may develop adaptive attacks specifically designed to bypass the preprocessing operation. Therefore, preprocessing is generally more effective when combined with other defense mechanisms.

5.3 Defensive Distillation

Defensive distillation is a technique originally proposed to reduce the sensitivity of neural networks to small input changes [17]. The approach involves training a model using softened probability distributions generated by another model or training stage.

By producing smoother output distributions, defensive distillation aims to make the model's decision function less sensitive to small perturbations. Early research demonstrated that distillation could reduce the effectiveness of certain adversarial attacks [17].

However, subsequent studies showed that some attacks could bypass or overcome distillation-based defenses. Therefore, defensive distillation alone cannot be considered a universal solution to adversarial attacks. Its importance lies primarily in demonstrating how modifying the behavior of a model can influence its robustness.

5.4 Robust Optimization

Robust optimization aims to train machine learning models to maintain reliable performance even when their inputs are subjected to worst-case perturbations. Rather than optimizing only for average performance on clean data, the model is optimized to perform well within a specified range of possible adversarial modifications [5], [18].

Robust optimization is closely related to adversarial training, as both approaches consider potentially harmful perturbations during model development. The difference is that robust optimization provides a broader mathematical framework for handling uncertainty and worst-case conditions.

This approach can improve resistance to strong adversarial attacks, particularly when the perturbation constraints are properly defined. However, robust optimization may require considerable computational resources and can become difficult to apply to large and complex models.

5.5 Attack Detection

Attack detection mechanisms attempt to identify whether an input or model behavior is potentially malicious before allowing the system to make a final decision. These techniques can analyze characteristics of inputs, model outputs, or intermediate representations to identify patterns associated with adversarial examples [19].

Detection methods may use statistical analysis, machine learning classifiers, anomaly detection, or specialized indicators of adversarial behavior. If an input is classified as suspicious, the system can reject it, request additional verification, or pass it through another security mechanism.

The main challenge is that an effective detector must distinguish between legitimate unusual inputs and genuinely adversarial ones. An attacker may also modify the attack strategy to avoid detection. Therefore, detection mechanisms should ideally be evaluated against adaptive adversaries.

5.6 Data Sanitization

Data sanitization is particularly important for defending against data poisoning and backdoor attacks. The objective is to identify and remove suspicious, corrupted, or malicious samples from the training dataset before they influence the model [6], [20].

Sanitization techniques can examine statistical properties, labels, feature distributions, and relationships between training samples. Outlier detection and data filtering can help identify samples that differ significantly from legitimate data. However, sophisticated poisoning attacks may be designed to remain statistically similar to legitimate samples. As a result, simple outlier removal may not always be sufficient. Combining data sanitization with robust training and continuous monitoring can provide stronger protection.

5.7 Certified Robustness

Certified robustness attempts to provide a formal guarantee that a model's prediction will remain unchanged for all input perturbations within a predefined region [21]. Instead of testing a model against a limited collection of adversarial examples, certified methods mathematically establish whether certain perturbations can change the model's output.

For example, a model may be certified to maintain the same prediction for any perturbation whose magnitude remains below a specified threshold. This provides stronger assurance than empirical testing because the guarantee covers an entire defined perturbation region.

However, certified robustness can be computationally challenging, particularly for large neural networks. The strength of the guarantee also depends on the perturbation model and certification technique used. Despite these limitations, certified defenses are an important direction for developing trustworthy and formally robust AI systems.

Table 5. Comparison of defense mechanisms

Defense mechanism	Main purpose	Targeted attacks	Major advantages	Limitations
Adversarial training	Train model with adversarial examples	Evasion	Strong empirical robustness	High computational cost
Input Preprocessing	Reduce adversarial perturbations	Evasion	Can work before interference	Can be bypassed
Defensive distillation	Reduce model sensitivity	Evasion	Smooths model behaviour	Vulnerable to adaptive attacks

Robust Optimization	Optimize against worst-case inputs	Evasion	Improves robustness	Computationally expensive
Attack detection	Identify malicious inputs	Evasion	Enables attack rejection	False positives/negatives
Data Sanitization	Remove malicious training data	Poisoning/Backdoor	Protects training process	Sophisticated attacks may evade detection
Certified Robustness	Mathematically guarantee robustness	Evasion	Provides formal guarantees	Difficult for large models

5.8 Comparative Analysis of Defense Strategies

The effectiveness of a defense mechanism depends on the type of attack, the available computational resources, and the security requirements of the application. Adversarial training is generally effective against many inference-time attacks because it directly exposes the model to adversarial inputs during training. Data sanitization is more appropriate for training-time attacks because it focuses on protecting the integrity of the training dataset. Input preprocessing and attack detection can provide additional protection at the inference stage, while certified robustness offers stronger mathematical guarantees under specific assumptions. However, each method has limitations, and attackers can often adapt their strategies to target known defenses [14], [19].

Therefore, practical ML security increasingly favors layered defense, where multiple mechanisms are combined. For example, a system may use sanitized training data, adversarial training, input monitoring, and attack detection simultaneously. Such a multi-layered approach can reduce dependence on any single defense and provide greater resilience against evolving adversarial techniques.

Table 6. Attack-defense relationship

Attack	Primary vulnerability	Suitable defense
FGSM	Sensitivity to input perturbation	Adversarial training, preprocessing
PGD	Gradient-based vulnerability	Adversarial training, robust optimization
C&W	Decision boundaries	Adversarial training, certified defenses
DeepFool	Decision boundary sensitivity	Robust training, detection
Data Poisoning	Training- data integrity	Data sanitization, robust learning
Backdoor Attack	Hidden trigger associations	Data sanitization, backdoor detection
Model Extraction	Excessive model access	Query monitoring, access control
Membership inference	Information leakage	Privacy-preserving learning, regularization

6. COMPARISON OF ADVERSARIAL ATTACKS AND DEFENSE MECHANISMS

Adversarial attacks differ considerably in their objectives, attack stages, required knowledge, and methods of manipulating machine learning systems. Similarly, defense mechanisms have different levels of effectiveness depending on the type of attack being considered. Therefore, comparing attacks and defenses is important for identifying suitable security strategies for different ML applications.

Evasion attacks primarily target the inference stage by modifying input samples, while poisoning and backdoor attacks target the training process. Model extraction and inference attacks mainly exploit the information exposed through model interfaces and outputs. Because these attacks operate at different stages, a defense that is effective against one category may provide little or no protection against another [1], [14].

6.1 Attack and Defense Comparison

Adversarial training is one of the most effective empirical approaches against many gradient-based evasion attacks because the model is explicitly exposed to adversarial examples during training [5]. However, its effectiveness depends on the types of adversarial examples used during training. A model trained against one attack may remain vulnerable to a different or stronger attack. Table 6 shows the Attack-defense relationship.

For training-time attacks, such as data poisoning, protecting the integrity of the training dataset is particularly important. Data sanitization and anomaly detection can help identify suspicious samples before they are incorporated into the training process [6]. Backdoor attacks require additional analysis because poisoned samples may appear similar to legitimate data and the malicious behavior may remain hidden until a specific trigger is presented [20]. Input preprocessing and attack detection can provide an additional layer of protection during inference. These approaches can identify or reduce suspicious perturbations before they reach the main model. However, adaptive attackers may design adversarial examples specifically to bypass such defenses [14]. Certified robustness differs from most empirical defense techniques because it attempts to provide mathematical guarantees within a specified perturbation range. Although this provides stronger assurance, the computational cost and limitations of certification methods make them difficult to apply universally to large-scale models [21]. Attack strength and defense considerations shown in Table 7.

Table 7. Aattack strength and defense considerations

Attack Technique	Computational Requirement	Typical Attack Strength	Defense Consideration
FGSM	Low	Moderate	Adversarial training
PGD	Medium-High	High	Robust/adversarial training
C&W	High	High	Robust optimization/certification
DeepFool	Medium	High	Robust training/detection
Data Poisoning	Variable	Potentially high	Data validation/sanitization
Backdoor Attack	Variable	High for targeted inputs	Backdoor detection

6.2 Trade-off Between Accuracy and Robustness

One of the major challenges in adversarial defense is the trade-off between standard accuracy and adversarial robustness. A model optimized only for clean data may achieve high accuracy under normal conditions but become vulnerable to small adversarial perturbations. On the other hand, increasing robustness through adversarial training or robust optimization can sometimes reduce performance on clean examples [18].

This trade-off is particularly important in real-world applications. For example, a healthcare system may require both high diagnostic accuracy and resistance to manipulated inputs. Similarly, an autonomous system must maintain reliable performance under both normal and adversarial conditions. Therefore, defense strategies should be selected based on the security requirements and acceptable risk level of the application rather than focusing on robustness alone.

6.3 Layered Defense Approach

The comparison of different attacks demonstrates that no single defense mechanism is sufficient to secure an entire ML system. A more practical approach is to use multiple defensive layers throughout the ML lifecycle.

At the training stage, data can be validated and sanitized to reduce poisoning and backdoor risks. During model development, adversarial training and robust optimization can improve resistance to malicious inputs. During deployment, input monitoring and attack detection can identify suspicious activity, while access-control mechanisms can reduce the risk of model extraction.

This layered approach provides protection at multiple points rather than relying on one mechanism. If one defense is bypassed, another layer may still detect or reduce the impact of the attack. Such defense-in-depth strategies are particularly important as attackers continue to develop adaptive techniques against existing defenses.

6.6 Discussion

The comparison shows that adversarial security must address multiple dimensions of an ML system rather than concentrating on a single attack type. Evasion attacks can often be addressed through robust training and input-level defenses, whereas poisoning and backdoor attacks require greater attention to training-data integrity. Model extraction and inference attacks require protection of the model interface and sensitive information.

The choice of defense should therefore consider the attack type, attacker capabilities, model architecture, application requirements, computational resources, and acceptable security risk. Combining complementary defense mechanisms can provide broader protection than relying on an individual technique. This layered perspective is essential for developing secure and trustworthy machine learning systems.

7. APPLICATIONS OF ADVERSARIAL MACHINE LEARNING SECURITY

Machine learning models are increasingly being integrated into systems that influence important decisions in healthcare, transportation, cybersecurity, finance, and Internet of Things (IoT) applications. While these technologies provide significant benefits, their dependence on data and automated decision-making also creates opportunities for adversarial manipulation. An attack that causes only a minor classification error in a conventional application could have serious consequences when the same technology is used in a safety- or security-critical environment [1], [14].

Therefore, adversarial machine learning security is becoming an important consideration in the design and deployment of intelligent systems. The following subsections describe some major application areas in which adversarial attacks can create security risks.

7.1 Healthcare

Machine learning is increasingly used in healthcare for medical image analysis, disease prediction, diagnosis, and clinical decision support. Deep learning models can assist in identifying abnormalities in medical images such as X-rays, CT scans, and MRI scans. However, adversarial manipulation of medical inputs can potentially cause a model to produce an incorrect diagnosis or classification [22].

For example, carefully designed perturbations could alter the output of an image-based diagnostic system while remaining difficult for a human observer to identify. Data poisoning can also affect healthcare models if malicious or incorrect samples are introduced into the training dataset. Since incorrect predictions in healthcare can directly influence clinical decisions, robustness and data integrity are particularly important.

Defense mechanisms such as adversarial training, input validation, anomaly detection, and secure data management can help reduce these risks. However, healthcare systems require particularly careful evaluation because increasing security should not compromise the accuracy and reliability expected in clinical environments.

7.2 Autonomous Vehicles

Autonomous vehicles depend on machine learning models for tasks such as object detection, traffic-sign recognition, lane detection, and environmental perception. These models continuously process information from cameras, sensors, and other sources to make decisions about the vehicle's surroundings. Adversarial attacks can manipulate visual inputs and cause an autonomous system to misclassify objects or traffic signs. Small changes to physical objects, such as road signs, may potentially influence a vision-based model even when the modification appears insignificant to humans [23]. Such attacks are particularly concerning because incorrect perception can affect subsequent driving decisions.

Robust training, adversarial testing, sensor fusion, input validation, and real-time anomaly detection can help improve the resilience of autonomous systems. Because these systems operate in dynamic environments, defenses must also account for physical-world variations rather than relying only on digitally generated adversarial examples.

7.3 Facial Recognition

Facial recognition systems use machine learning to identify or verify individuals based on facial characteristics. They are used in areas such as authentication, access control, surveillance, and identity verification.

Adversarial attacks can manipulate facial images or introduce carefully designed patterns that interfere with recognition models. An attacker may attempt to cause the system to misidentify a person or prevent a legitimate individual from being recognized. Research has demonstrated that adversarial perturbations can affect deep neural network-based face recognition systems [24].

Possible defenses include adversarial training, robust feature extraction, image preprocessing, liveness detection, and multi-factor authentication. Combining facial recognition with additional verification mechanisms can reduce the consequences of successful adversarial manipulation.

7.4 Cybersecurity

Machine learning is increasingly used in cybersecurity applications such as malware detection, spam filtering, intrusion detection, and network traffic classification. These systems analyze large volumes of data to identify suspicious behavior and potential threats.

Attackers can exploit the same ML systems by modifying malicious inputs so that they resemble legitimate data. This is an example of an evasion attack, where the attacker attempts to bypass a security classifier without necessarily modifying the underlying malicious activity [25]. Poisoning attacks can also be used to influence cybersecurity models by introducing manipulated data into their training processes.

Because attackers and defenders continuously adapt to one another, cybersecurity applications require continuous monitoring and model updating. Robust training, adversarial testing, data sanitization, and anomaly detection can help improve the resilience of ML-based security systems.

7.5 Internet of Things (IoT)

IoT environments contain large numbers of connected devices that generate data and increasingly rely on machine learning for monitoring, anomaly detection, and automated decision-making. These systems may be deployed in homes, industrial environments, healthcare facilities, and smart infrastructure.

The distributed nature of IoT creates several potential attack surfaces. Attackers may manipulate sensor readings, compromise devices, or inject malicious data into ML training processes. A poisoned or manipulated model could consequently make incorrect decisions about device behavior or network activity [26].

Resource limitations are another challenge because many IoT devices have restricted computational power, memory, and energy. Consequently, security mechanisms for IoT-based ML systems must provide effective protection without imposing excessive computational overhead.

7.6 Financial Systems

Machine learning is widely used in financial applications such as fraud detection, credit assessment, transaction monitoring, and risk analysis. These systems process large amounts of data to identify unusual patterns and support automated decisions.

Adversaries may attempt to manipulate transaction data or other input characteristics to evade fraud detection systems. Poisoning attacks could also influence a model if malicious samples enter its training data. Successful attacks could result in financial losses, fraudulent transactions, or incorrect risk assessments.

Robust model training, continuous monitoring, anomaly detection, secure data pipelines, and periodic security evaluation can help reduce these risks. Since financial systems operate with sensitive information, privacy-preserving techniques are also important when developing and deploying ML models.

8. CHALLENGES AND LIMITATIONS

Although adversarial machine learning has developed rapidly, achieving reliable security against adversarial attacks remains difficult. Existing attack and defense techniques often depend on assumptions about the attacker, model architecture, perturbation type, or application environment. A defense that performs well against one attack may provide limited protection against another. In addition, improving adversarial robustness can introduce computational overhead and may affect the model's performance on normal inputs. Recent research continues to identify challenges involving robustness–accuracy trade-offs, certification, adaptive attackers, and practical deployment.

8.1 Computational Cost

One of the major limitations of adversarial defense is the computational cost involved in training and evaluation. Techniques such as adversarial training require the generation of adversarial examples during the training process. Iterative attacks such as Projected Gradient Descent (PGD) may require multiple model evaluations for generating a single adversarial sample. As a result, training can become significantly slower than conventional machine learning training.

The additional computational requirements can be particularly problematic for large deep learning models and resource-constrained systems such as IoT devices. Therefore, developing lightweight defense techniques without significantly reducing security remains an important research challenge.

8.2 Accuracy–Robustness Trade-off

A well-known challenge in adversarial machine learning is the trade-off between standard accuracy and adversarial robustness. A model trained to resist adversarial perturbations may experience reduced performance on clean, unmodified inputs. Research has demonstrated that this relationship can depend on the model, data distribution, attack assumptions, and definition of robustness.

This creates a difficult design problem for practical systems. For example, a healthcare or autonomous-vehicle system must maintain high accuracy under normal conditions while also remaining resistant to malicious inputs. Increasing robustness at the cost of excessive clean-data accuracy may therefore not be acceptable in safety-critical applications.

8.3 Adaptive and Evolving Attackers

Attackers can modify their strategies after learning about the defense mechanism used by a system. Such attackers are commonly referred to as adaptive attackers. A defense that is effective against a fixed attack may become ineffective when the attacker understands its operation and develops a method specifically designed to bypass it.

This creates an ongoing interaction between attackers and defenders. Consequently, evaluating a defense only against known attacks may provide an unrealistic assessment of its security. Defense mechanisms must therefore be tested against multiple attack strategies, including adaptive and previously unseen attacks.

8.4 Transferability and Black-Box Attacks

Many adversarial examples can transfer between different machine learning models. An adversarial example generated for one model may sometimes cause incorrect predictions in another model, even when the attacker does not have complete knowledge of the target model.

This transferability increases the threat posed by black-box attacks because an attacker may construct a substitute model and use it to generate adversarial inputs. Therefore, simply hiding the architecture or parameters of a model does not necessarily guarantee security.

8.5 Difficulty of Reliable Attack Detection

Detecting adversarial inputs before they reach the target model is another significant challenge. Some adversarial examples are intentionally designed to remain very similar to legitimate inputs, making them difficult to distinguish using simple statistical or preprocessing techniques.

Detection systems can also produce false positives, where legitimate inputs are incorrectly classified as attacks. Conversely, false negatives may allow malicious inputs to pass through undetected. This makes reliable attack detection particularly important for applications where incorrect decisions can have serious consequences.

8.6 Lack of a Universal Defense

There is currently no single defense mechanism that can protect machine learning models against every possible adversarial attack. Adversarial training may provide strong protection against certain evasion attacks, while data sanitization is more appropriate for poisoning attacks. Similarly, privacy-preserving methods may be required for attacks targeting information about training data.

Therefore, practical security generally requires a layered defense strategy involving secure data collection, training-time protection, robust model development, input monitoring, and continuous security evaluation.

8.7 Real-World Deployment Challenges

Many adversarial machine learning techniques are evaluated using benchmark datasets and controlled experimental environments. However, real-world systems operate under changing data distributions, hardware constraints, sensor noise, network conditions, and evolving user behavior.

For example, an autonomous vehicle may encounter changes in weather, lighting, road conditions, and sensor quality that are not fully represented in a laboratory dataset. Similarly, financial and cybersecurity systems continuously receive new patterns of data. Therefore, maintaining robustness after deployment requires continuous monitoring, testing, and model updating.

Recent research also highlights that even certified robustness should not automatically be interpreted as complete model security, since certification usually applies under specific assumptions and threat models.

9. FUTURE SCOPE

The increasing use of artificial intelligence and machine learning in critical applications has made adversarial security an important area of research. Although existing techniques have improved the ability of models to withstand several types of attacks, attackers continue to develop more sophisticated methods. Future research must therefore focus on creating machine learning systems that are not only accurate but also robust, explainable, privacy-preserving, and capable of adapting to changing threats.

9.1 Automated and Adaptive Defense

Future adversarial defense systems can incorporate automated mechanisms that continuously monitor model behavior and identify changes in attack patterns. Instead of relying entirely on predefined attack signatures, adaptive defenses could modify their detection and protection strategies according to the observed threat environment.

Machine learning itself can be used to identify suspicious patterns and automatically select suitable defensive techniques. Such systems could reduce the need for manually updating security rules whenever a new attack method appears.

9.2 Improved Certified Robustness

Certified robustness is expected to remain an important research direction. Unlike empirical robustness testing, which evaluates a model against selected attacks, certification attempts to provide mathematical guarantees that a model's prediction will remain unchanged within a specified perturbation region. Future research can focus on developing certification methods that are computationally efficient and scalable to larger neural networks. Improving the size of the certified region while maintaining high prediction accuracy could make certified defenses more practical for safety-critical applications.

9.3 Explainable and Trustworthy AI

The integration of explainable artificial intelligence (XAI) with adversarial machine learning can improve the understanding of model behavior under both normal and malicious inputs. Explainability techniques can help identify unusual changes in model decisions and provide information about why a particular prediction was made.

Future systems could combine robustness evaluation with explainability to identify suspicious decision patterns. This would be particularly useful in healthcare, finance, and autonomous systems, where understanding the reasoning behind an AI decision is important for establishing trust.

9.4 Security of Generative and Multimodal AI

The rapid development of generative AI and multimodal models introduces new adversarial security challenges. Modern systems can process combinations of text, images, audio, and other forms of information. Attackers may therefore exploit interactions between different modalities to manipulate model outputs.

Future research should investigate adversarial attacks and defenses specifically designed for multimodal and generative models. Security mechanisms will need to consider not only individual inputs but also interactions between different types of data.

9.5 Real-Time Attack Detection

Many practical applications require decisions to be made within a short period of time. Therefore, future defense systems should detect adversarial inputs efficiently without introducing significant delays.

Research can focus on lightweight detection models, efficient feature analysis, and hardware-assisted security mechanisms. Such approaches could be particularly valuable for autonomous vehicles, IoT devices, network security systems, and other applications where real-time decisions are required.

9.6 Privacy-Preserving Robust Machine Learning

Adversarial security and data privacy are closely related. Techniques such as membership inference demonstrate that attackers may obtain information about the data used to train a model. Future research should therefore consider robustness and privacy together rather than treating them as completely separate problems.

Privacy-preserving approaches such as differential privacy, secure computation, and federated learning can be investigated alongside adversarial defense techniques. The objective is to develop models that can resist malicious inputs while also reducing the risk of sensitive information leakage.

9.7 Secure Machine Learning Throughout the Model Lifecycle

Future security approaches should protect the entire machine learning lifecycle rather than focusing only on the final deployed model. Security mechanisms can be incorporated during data collection, preprocessing, training, model validation, deployment, and continuous monitoring.

This lifecycle-based approach can help identify vulnerabilities at an early stage and reduce the possibility of attackers exploiting weaknesses that remain unnoticed until deployment.

10. CONCLUSION

Adversarial machine learning has emerged as an important area at the intersection of machine learning and cybersecurity because machine learning models can be vulnerable to intentionally crafted attacks. These vulnerabilities can occur at different stages of the machine learning lifecycle, including data collection, model training, deployment, and inference. Adversarial threats include evasion attacks, data poisoning, backdoor attacks, model extraction, and privacy-related attacks, demonstrating that the security of an ML system cannot be evaluated solely on the basis of its accuracy on clean data [1], [2].

This paper examined the major types of adversarial attacks and discussed important attack techniques such as FGSM, PGD, Carlini and Wagner attacks, DeepFool, poisoning, and backdoor attacks. Various defense mechanisms, including adversarial training, input preprocessing, robust optimization, attack detection, data sanitization, and certified robustness, were also analyzed. Existing research shows that these approaches can improve model robustness, but their effectiveness depends on the attack type, threat model, and assumptions under which the defense is applied [3], [4].

The analysis further demonstrated that adversarial machine learning has significant implications for practical applications such as healthcare, autonomous vehicles, cybersecurity, IoT, facial recognition, and financial systems. In these environments, incorrect predictions can lead to consequences beyond ordinary model errors, including safety risks, privacy violations, financial losses, and security failures. Recent research continues to emphasize the importance of robust and privacy-preserving AI, particularly when machine learning is deployed in critical domains [5].

Despite considerable progress, several challenges remain. High computational requirements, the trade-off between robustness and standard accuracy, adaptive attackers, difficulties in detecting sophisticated adversarial inputs, and the absence of a universal defense mechanism continue to limit the practical deployment of secure ML systems. Consequently, adversarial robustness should be treated as an ongoing security requirement rather than as a one-time model evaluation.

Future research should focus on developing adaptive and computationally efficient defenses, improving certified robustness, securing generative and multimodal AI, strengthening privacy-preserving machine learning, and enabling continuous monitoring throughout the ML lifecycle. Recent work also emphasizes the importance of evaluating adversarial threats across the complete lifecycle of AI systems rather than concentrating on individual attack stages [6], [7].

In conclusion, secure machine learning requires a shift from accuracy-focused development toward security-aware, robust, and trustworthy AI design. No single defense mechanism can provide complete protection against the continuously evolving adversarial threat landscape. A combination of secure data management, robust training, attack detection, privacy protection, continuous testing, and lifecycle-aware security is therefore essential for building reliable machine learning systems capable of operating safely in real-world environments.

REFERENCES

- [1] B. Biggio and F. Roli, "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018, doi: 10.1016/j.patcog.2018.07.023.
- [2] Y. Li, B. Wu, H. Wu, and Y. Wang, "The Security of Machine Learning in an Adversarial Setting: A Survey," *Journal of Parallel and Distributed Computing*, 2018.
- [3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *International Conference on Learning Representations (ICLR)*, 2015.

- [4] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial Attacks and Defenses Against Deep Neural Networks: A Survey," *Procedia Computer Science*, vol. 140, pp. 152–161, 2018, doi: 10.1016/j.procs.2018.10.315.
- [5] Y. Yuan, W. Li, and X. Wu, "A Survey of Practical Adversarial Example Attacks," *Cybersecurity*, vol. 1, art. no. 9, 2018, doi: 10.1186/s42400-018-0012-9.
- [6] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 19–35, 2018, doi: 10.1109/SP.2018.00057.
- [7] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain," arXiv preprint arXiv:1708.06733, 2017.
- [8] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning Attack on Neural Networks," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2018.
- [9] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing Machine Learning Models via Prediction APIs," in *25th USENIX Security Symposium (USENIX Security 16)*, pp. 601–618, 2016.
- [10] 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 Asia Conference on Computer and Communications Security (ASIA CCS)*, pp. 506–519, 2017, doi: 10.1145/3052973.3053009.
- [11] 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)*, pp. 3–18, 2017, doi: 10.1109/SP.2017.41.
- [12] N. Carlini and D. A. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017, doi: 10.1109/SP.2017.49.
- [13] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016, doi: 10.1109/CVPR.2016.282.
- [14] 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 (ICLR)*, 2018.
- [15] W. Xu, D. Evans, and Y. Qi, "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks," in *Network and Distributed System Security Symposium (NDSS)*, 2018.
- [16] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2016, doi: 10.1109/SP.2016.41.
- [17] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically Principled Trade-off Between Robustness and Accuracy," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 7472–7482, 2019.
- [18] J. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified Adversarial Robustness via Randomized Smoothing," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 1310–1320, 2019.
- [19] 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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1625–1634, 2018, doi: 10.1109/CVPR.2018.00175.
- [20] F. Tramèr, N. Carlini, W. Brendel, and A. Madry, "On Adaptive Attacks to Adversarial Example Defenses," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [21] Z. Wang, J. Ma, X. Wang, J. Hu, Z. Qin, and K. Ren, "Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems," *ACM Computing Surveys*, vol. 55, no. 7, art. no. 134, 2022, doi: 10.1145/3538707.
- [22] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein, "Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses," arXiv preprint arXiv:2012.10544, 2020.
- [23] A. Raghunathan, S. M. Xie, F. Yang, J. Duchi, and P. Liang, "Understanding and Mitigating the Tradeoff Between Robustness and Accuracy," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 7909–7919, 2020.
- [24] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4312–4321, 2019.
- [25] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into Transferable Adversarial Examples and Black-box Attacks," in *International Conference on Learning Representations (ICLR)*, 2017.
- [26] S. Ma, Y. Liu, G. Tao, W.-C. Lee, and X. Zhang, "NIC: Detecting Adversarial Examples in the Wild via Neural Network Invariant Checking," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.