International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Robustness of Intelligent Security Systems Under Adversarial Machine Learning Attacks: A Critical Survey

Sapna[1], Gagandeep[2]

[1]Assistant Professor, Computer Science and Engineering, SBSSU Gurdaspur
[2]Manager, Cyber Security

Abstract
Deep learning is now central to modern security tools like intrusion detection and malware scanners. However, these intelligent systems are surprisingly fragile when faced with adversarial machine learning. While they perform well in standard tests, they can be easily deceived by small, intentional changes to input data. This survey provides a clear look at how these attacks work and how we can stop them. We categorize attacks by when they happen and what the attacker knows, focusing on key methods like Fast Gradient Sign Method and Carlini and Wagner. We also map current defences such as adversarial training and feature squeezing against specific threats to see what actually works. Our goal is to highlight the gaps in current security and show why we need AI that is built to be resilient from the start.

Keywords: Adversarial Machine Learning, Intelligent Security Systems, Robustness, Evasion Attacks, Poisoning, Intrusion Detection, Fast Gradient Sign Method and Carlini and Wagner.

I. INTRODUCTION

Intelligent security systems, such as Intrusion Detection Systems (IDS) and malware scanners are now relying on deep learning to identify complex malicious behaviours of the systems or devices. Even though these systems are very accurate in normal situations, they are easily tricked. By making tiny, hidden changes to the data, a person would not even notice an attacker can confuse the AI and force it to make a mistake. Unlike traditional software bugs, these attacks target the statistical logic of the model [1]. This paper provides a systematic overview of this threat landscape and strategies for building resilient defense infrastructures.

II. TAXONOMY OF ATTACKS

Adversarial attacks in machine learning can be understood by examining when they occur during the model lifecycle and how much information the attacker possesses about the system. This perspective helps in analyzing both the technical feasibility and practical impact of such attacks.

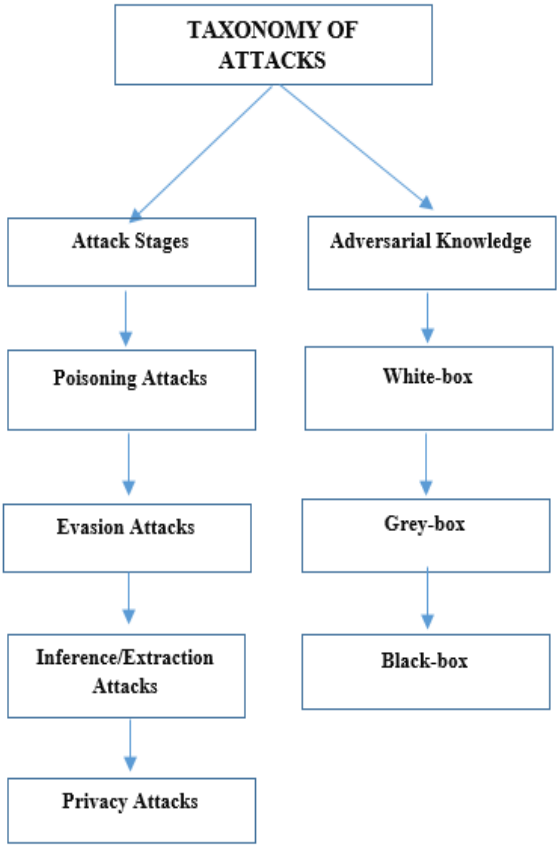

Fig 1. Taxonomy of Attacks

A. Attack Stages

1) Poisoning Attacks

Poisoning attacks occur while the model is still learning from data. The goal of attacker is to corrupt the model's underlying logic by injecting malicious samples into the training set. This can result in a Denial of Service where the model becomes generally inaccurate, or a Backdoor where the model works perfectly except when it sees a specific trigger. [2] For example, in a Spam Filter, an attacker could send thousands of emails containing a specific neutral word like berry alongside spam content. If the filter retrains on this data, it might learn to incorrectly flag any legitimate email containing berry word as spam, effectively blocking valid communication. Another example, when a Facial Recognition System could be poisoned to grant access to any unauthorized person wearing a specific pair of red glasses, while still correctly identifying everyone else. [3]

2) Evasion Attacks

Evasion attacks are the most common threats to live security systems. These occur after the model is trained and fully operational. The attacker attempts to bypass detection by making subtle, often invisible, modifications to a malicious input. [4] In Malware Detection, a hacker might take a known virus and append dead code lines of programming that do nothing but change statistical signature of the file. To a human or a sandbox, the file is still a virus, but to an AI-based antivirus, the fingerprint shifts just enough to be classified as non-malicious. Similarly, in an Intrusion Detection System (IDS), an attacker might slightly alter the timing of data packets to mimic the flow of normal web browsing, allowing a data breach to slip past the monitor unnoticed. [5]

3) Inference/Extraction Attacks

These attacks target the intellectual property of the model or the privacy of the data used to build it. The attacker performs high-volume query probing to map the model's decision boundaries and extract its underlying logic. [6] For Example, In Model Extraction, an attacker might send a massive volume of fake transactions to a Fraud Detection Application Programming Interface(API) of the bank. By recording which transactions are blocked and which are allowed, attacker can train a model that perfectly tricks the AI of the bank, effectively stealing the proprietary logic. Another example, In Membership Inference, an attacker might query a health-security AI with a specific person's medical record. If the model responds with an unusually high confidence score, it confirms that the person's private data was part of the Training Dataset, leading to a significant privacy leak. [7]

4) Privacy Attacks

Privacy attacks focus on extracting sensitive information about the individuals whose data was used to train the model. Techniques like Membership Inference or Model Inversion exploit the model's tendency to memorize specific details from its training set. If a model was trained on private medical records or personal emails, a privacy attack could allow an adversary to reconstruct specific records or confirm a person's presence in a sensitive database, leading to significant legal and ethical breaches. [8]

B. Adversarial Knowledge

In the context of intelligent security systems, Adversarial Knowledge refers to the amount of information an attacker possesses regarding the target machine learning internal logic, training data, and architecture of the model. This level of transparency dictates the strategy and success rate of an attack, as it determines whether the adversary can precisely calculate a weakness or must rely on trial and error.

1) White-box

In white-box attack, the attacker has complete knowledge, including access to the internal architecture of the model, including trained parameters, optimization functions, and the original training data. This allows them to use mathematical gradients to find the perfect perturbation that will trigger a misclassification [9].For example, if an attacker has the full source code and parameters for a Deep Learning based Malware Scanner, attackers can precisely identify which bits of a malicious file to flip so that the scanner's internal logic misclassifies the virus as a non-malicious document. These are the most potent attacks because they are mathematically optimized for the specific target. [10]

2) Grey-box

A Grey-box attack occurs when the attacker has limited information, such as the feature set the system uses or the general type of algorithm (like knowing the system uses a Random Forest but not the exact trees). In a Network Intrusion Detection System (NIDS), a grey box attacker might know that the system monitors packet length and inter-arrival times but does not have the exact thresholds for an alert. The attacker performs feature-level masking, adjusting the malicious samples to reside within the typical feature space of benign traffic. This increases the probability of a False Negative, as the model struggles to differentiate the attack from genuine system activity. [11]

3) Black Box

In a Black Box attack, the adversary has zero knowledge of the internal workings and can only observe the inputs they send and the outputs the system returns. These attacks often rely on a phenomenon called Adversarial Transferability, where an attack that fools one model often fools another. For instance, an attacker targeting a proprietary Cloud-based Fraud Detection API might build their own replica of model at home using public data. They find an evasion trick that works on their home model and then send that

same modified transaction to the Cloud API. Even though they never saw the Cloud API's code, there is a high probability the trick will still work because both models learned similar patterns from the data. [12]

III. ALGORITHMIC LANDSCAPE AND DOMAIN IMPACT

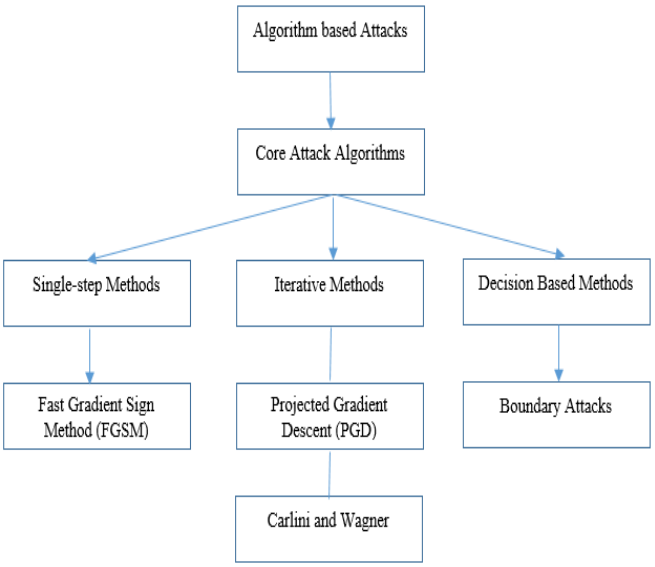

Fig 2. Algorithm based Attacks

The Algorithmic Landscape of adversarial attacks defines the mathematical and logical methods used to generate malicious inputs, categorized primarily by how the attacker calculates the necessary changes. Following are the types of attacks:

A. Core Attack Algorithms

Core attack algorithms represent the specific mathematical and exploitation logic used to generate malicious inputs. These are generally split into two categories based on how many steps they take to find a vulnerability:

- Single-step Methods
- Iterative Methods
- Decision Based Methods

1) Single-step Methods

Single-step methods are the most computationally efficient type of adversarial attack. These are designed to generate a malicious input using only one calculation. These methods work by finding the gradient essentially the direction in which the model's error increases the most and moving the input data one step in that direction. Because they only require a single pass through the model's logic, they are incredibly fast and can be used to launch attacks in real-time environments where speed is critical, such as live network traffic filtering. [13]

Single-step methods include:

a) Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is one of the earliest and simplest gradient-based attack strategies. Instead of making large modifications to an input, FGSM applies a small perturbation in the direction that increases the model's prediction error. Although computationally efficient, the attack often produces detectable patterns when robust training methods are applied. [14] For example, an attacker targeting an AI-powered email filter might use FGSM to quickly identify which few characters in a phishing link need

to be swapped to make the URL appear legitimate to the scanner. While fast, these attacks are often easier to block with basic defensive training. [14]

2) Iterative methods

Iterative methods are more sophisticated adversarial attacks that find vulnerabilities by taking multiple small, calculated steps rather than a single large leap. Iterative methods include:

a) Projected Gradient Descent (PGD)

Projected Gradient Descent (PGD) extends the single-step gradient approach by performing multiple constrained updates to the input. At each iteration, the perturbation is adjusted while ensuring it remains within a predefined boundary. This iterative refinement typically results in stronger adversarial examples compared to single-step methods. [14] For example, PGD in an Intelligent Security System would be an attacker targeting a Deep Learning-based Malware Scanner. A single-step attack (FGSM) might change a few bits of a virus, but the file might still be detected if the change was not precise enough. By employing PGD, the adversary executes over 50 iterations; in each step, they perform a gradient-based bit flip to incrementally maximize the scanner's benign confidence score until the malicious file successfully bypasses detection. If the structure of the file becomes corrupted or the changes become too large, the projection step restores integrity of the file while keeping the malicious logic intact. By the final iteration, the virus has been transformed into a file that the AI identifies as a harmless document, successfully evading the security system. [14]

b) Carlini and Wagner

Carlini and Wagner is one of the most powerful and sophisticated iterative white-box attacks, designed to bypass defensive measures. Carlini and Wagner treats the attack as an optimization problem where the objective is to find the smallest possible perturbation that successfully flips the model's classification. [15] For example, Carlini and Wagner attack could be used to target a Facial Recognition System at a high-security checkpoint. While a simpler attack might add noticeable noise to a photo to cause a mismatch, a Carlini and Wagner attack would iteratively analyze neurons of the system to identify the most sensitive pixels on a face of the person. After hundreds of adjustments, the attacker could produce an image that looks identical to an unauthorized individual to any human guard, but which the AI-driven system identifies as a genuine employee with 99% confidence. Because the attack is so mathematically precise, it can often defeat Defensive Distillation and other common adversarial defenses. [15]

3) Decision-Based Iterative

A Decision-Based Iterative attack is a sophisticated black-box method that generates adversarial examples without any access to the model's internal gradients or confidence scores. A decision-based attack relies entirely on trial and error by observing only the final label (e.g., Cat or Dog) returned by the AI. [16]

a) Boundary Attacks

Boundary Attack is a highly effective black-box iterative method that does not require any knowledge of the model's internal architecture or gradients. An example of a Boundary Attack would be an adversary trying to bypass a Cloud-based Content Filter that blocks explicit images. The attacker starts with an image that is totally benign (like a picture of a cat) which the AI easily clears. They then iteratively blend this cat image with a target explicit image. In each of the thousands of queries, the algorithm checks about the AI can detect image is morphed or genuine and after that it will move the image one step closer to the explicit version. Eventually, through pure trial and error and observing the model's output, the attacker creates an image that looks almost entirely like the forbidden content but is still classified by the logic of AI as a genuine image. [16]

B. Security Domain Impact: IDS Vs. Malware

In the context of the Security Domain Impact, the distinction between Network Intrusion Detection Systems (NIDS) and Malware Detection is primarily defined by the nature of the data and the timing of the defense. NIDS operates on continuous, flow-based data, acting like a monitor for a constant stream of network traffic to ensure protocol compliance. Because network threats exist for live traffic, NIDS must work in real-time, identifying and blocking suspicious patterns as they happen to prevent an active breach. Its primary concern is the behavior of the communication, checking if the traffic follows the established rules of the network.

Malware Detection, conversely, focuses on discrete, binary data, meaning it inspects individual files or blocks of code rather than a continuous stream. Its primary constraint is execution integrity, ensuring that a program is what it claims to be and does not contain hidden or malicious instructions. Unlike the active monitoring of NIDS, malware defense typically occurs in a pre-execution or static window, where the goal is to scan and neutralize a threat before the file is ever allowed to run on the system. While NIDS guards the communication infrastructure of the organization, malware detection guards the packages or applications are being delivered. [17]

Feature	**Intrusion Detection (NIDS)**	**Malware Detection**
Data Type	Continuous/Flow-based	Discrete/Binary
Primary Constraint	Protocol Compliance	Execution Integrity
Attack Window	Real-time (Active)	Pre-execution (Static)

Table I Security Domain Impact: IDS Vs. Malware

IV. DEFENSE MECHANISMS AND EFFECTIVENESS

Defending security systems requires a multi-layered approach to counter evolving threats. Following are defense categories:

1. Strategic Defense Categories

a) Proactive (Training-Phase) Defenses

These methods aim to build inherent robustness into the model before it is deployed. These methods are:

Adversarial Training: The most common defense where the model is trained on a mix of clean data and adversarial examples. This teaches the model to recognize and correctly classify deliberately modified inputs, though it can be computationally expensive and may slightly reduce accuracy on normal data. [18]

Defensive Distillation: Defensive distillation attempts to reduce a model's sensitivity to small input variations by training it on softened probability outputs rather than hard labels. While initially proposed as a method to obscure gradient information, subsequent research has shown that determined attackers can often bypass this mechanism.. [18]

Robust Network Design: In Robust Network Design, Modifying the actual architecture of the neural network to reduce sensitivity, such as adding denoising layers or using parameters pruning to remove vulnerable neurons. [19]

2. Reactive (Inference-Phase) Defenses

These methods focus on detecting or neutralizing attacks in real-time as the model processes live data.

Input Pre-processing and Sanitization: In Input Pre-processing and Sanitization, Data is cleaned or transformed before reaching the model to strip away adversarial noise. [18]

Feature Squeezing: In Feature Squeezing, Reduces the complexity of the input by decreasing color bit-depth or applying spatial smoothing to remove fine-grained perturbations. [18]

Purification: In this method, uses separate models (like Autoencoders or GANs) to reconstruct or sanitise a clean version of the input, removing malicious noise while keeping essential features. [18]

Adversarial Detection: In Adversarial Detection method, uses a separate monitoring system to flag suspicious inputs that deviate from typical statistical distributions. If an input is flagged as adversarial, the system can reject it or trigger a safe-mode response. [18]

3. Formal and Architectural Protections

This category of defense focuses on the structural integrity of the AI itself. Instead of just showing the model more data, these methods change the mathematical rules or the internal design of the network to make it inherently harder to trick. [20] Following are the types of this defense:

Certified Robustness: Certified robustness techniques provide formal guarantees that a model's prediction will remain stable within a defined perturbation range. Although mathematically appealing, these methods often struggle to scale efficiently for complex deep learning architectures used in real-world security systems. [20]

Ensemble Defenses: In Ensemble Defenses, it combines multiple models with different architectures to make a final decision. An attack designed to fool one model is less likely to fool all of them simultaneously. [20]

Differential Privacy: This model adds controlled noise to the data or gradients to ensure that individual training samples cannot be extracted by an attacker, protecting against inference and model extraction attacks. [20]

V. DEFENSE-TO-ATTACK MAPPING

Following mapping outlines the effectiveness of various defensive mechanisms against the four primary categories of AI-specific attacks: Evasion, Poisoning, Extraction, and Privacy [21].

Defense Mechanism	Evasion	Poisoning	Extraction	Privacy
Adversarial Training	High	Medium	None	None
Data Sanitization/Filtering	None	High	None	None
Differential Privacy	Medium	Medium	Medium	High
Rate Limiting / Query Auditing	Medium	None	High	High

Feature Squeezing / Smoothing	High	None	None	None
Defensive Distillation	High	None	None	None
Output Obfuscation (Argmax)	Medium	None	High	Medium
Certified Robustness	High	Medium	None	None

Table II: Defense-To-Attack Mapping

VI. CONCLUSION

Despite significant progress in adversarial defense research, achieving reliable robustness in intelligent security systems remains an unresolved challenge. Many proposed defenses demonstrate effectiveness under controlled experimental settings but degrade under adaptive or real-world attack scenarios. This highlights the need for domain-aware and practically deployable robustness strategies. Moving forward, the focus must shift toward problem-space attacks that respect real-world domain constraints, ensuring that security AI is not merely accurate in vacuum, but truly resilient against sophisticated, human-led adversaries.

REFERENCES

1. N. Mohamed, “Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms,” Knowledge and Information Systems, p. 6969–7055, 2025.
2. J. M. X. W. J. H. Z. Q. a. K. R. Zhibo Wang, “Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems,” ACM Computing Surveys, pp. 1-36, jul 2023.
3. M.-S. H. N. V. Murray, “Foundations of Intelligent Systems,” in 15th International Symposium, ISMIS 2005, Saratoga Springs, NY, USA, 2005.
4. K. n. a. k. K. S. K. L. G. G. Lakkimsetty Nandini1, “AI VIGIL-GUARD: A Real-Time Adversarial Attack Detection and,” International Research Journal of Engineering and Technology (IRJET), pp. 1004-1012, December, 2025.
5. J. H. S. T. N. Z. A. N. A. W. F. U. A. W. Siraj Uddin Qureshi, “Systematic review of deep learning solutions for malware detection and forensic analysis in IoT,” Journal of King Saud University - Computer and Information Sciences, pp. 2-9, 2024.
6. L. L. K. D. N. Z. G. Z. a. Y. D. Kaixiang Zhao, “A Systematic Survey of Model Extraction Attacks and Defenses: State-of-the-Art and Perspectives,” arXiv, pp. 1-55, 2025.
7. P. R. Mulea, “AI-Augmented proactive cyber detection and,” December 2025. [Online]. Available: https://scholar.sun.ac.za/server/api/core/bitstreams/0cca714e-1036-4bd8-9478-122becea6889/content.
8. M. R. a. S. Garcia, “A Survey of Privacy Attacks in Machine Learning,” ACM Computing Surveys, pp. 1-34, Apr 2024.

9. P. T. a. G. R. Anant Thunuguntla, "Defenses Against Adversarial Attacks on Object Detection: Methods and Future Directions," MDPI, 2025.
10. K. L. H. A. H. T. B. a. o. Arash Mahboubi, "Evolving techniques in cyber threat hunting: A systematic review," Journal of Network and Computer Applications, pp. 1-34, 2024.
11. P. E. M. Joao Vitorino, "SoK: Realistic adversarial attacks and defenses for intelligent network," Computers & Security, pp. 1-10, 2023.
12. T. K. A. B. A. B. E. K. C. N. E. S. &. D. A. Sotiris Pelekis, "Adversarial machine learning: a review of methods, tools, and critical industry sectors," Springer Nature Link, p. 58, 2025.
13. Q. H. a. Y. Z. Jun Chen, "A survey of gradient normalization adversarial attack methods," ACM Digital Library, 2023.
14. J.-A. a. S. L.-M. William Villegas, "Evaluating the Robustness of Deep Learning Models against Adversarial Attacks: An Analysis with FGSM, PGD and CW," Big Data and Computing, p. 8, 2024.
15. P. a. o. Marek Pawlicki, "A meta-survey of adversarial attacks against artificial intelligence algorithms, including diffusion models," ELSEVIER Neuro Computing, pp. 1-16, 2025.
16. J. R. M. B. Wieland Brendel, "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models," in ICLR 2018 Conference Blind Submission, 2018.
17. F. D. G. ,. D. H. a. o. SABRINE ENNAJI, "Adversarial Challenges in Network Intrusion Detection Systems: Research Insights," IEEE, pp. 148613-148645, 2025.
18. R. M. a. o. JASMITA MALIK, "A Systematic Review of Adversarial Machine Learning Attacks, Defensive Controls, and," IEEE Access, pp. 99383-99421, 2024.
19. L. e. al, "Improving neural network robustness through neighborhood preserving layers," Image and Vision Computing, p. 123, 2022.
20. Y. Z. a. o. Ruipu Ma, "Advancements in adversarial example defense for deep learning models: a review," Springer Nature Link, p. 9, 2026.
21. L. D. Y. T. Z. W. Z. a. P. S. Y. Shuai Zhou, "Adversarial Attacks and Defenses in Deep Learning: From a Perspective of Cybersecurity," ACM Computing Surveys, pp. 1-39, 2023