**Understanding of Adversarial Attacks and an Analysis on CW attacks Simulation**

**Introduction**

Artificial intelligence (AI) and machine learning (ML) technologies have become integral to critical sectors such as healthcare, finance, and autonomous transportation. However, their widespread adoption has exposed significant vulnerabilities, notably their susceptibility to adversarial attacks. These attacks are deliberate efforts to deceive AI models by introducing minor, often imperceptible changes to input data, resulting in incorrect predictions or decisions. An adversarial attack involves making subtle modifications to input data that cause the model to misclassify or make erroneous decisions. These changes exploit the model's sensitivity to minor perturbations, which, although minimal to human observers, can significantly alter the model's output. For instance, a model trained to recognize handwritten digits might correctly identify an image of the number "7," but with slight, calculated perturbations, the same image can be misclassified as a different number, such as "1."

Among the various adversarial attack techniques, the Carlini & Wagner (C&W) attack is notable for its effectiveness and sophistication. The C&W attack generates adversarial examples by solving an optimization problem that minimizes the perturbation added to the original image while ensuring the altered image is misclassified by the model. This approach strikes a balance between imperceptibility to humans and deception of the AI model, making it a powerful tool for adversarial attacks. The significance of C&W attacks lies in their ability to reveal deep vulnerabilities in AI models, prompting the need for robust defenses and highlighting the importance of ongoing research in this area.

Adversarial attacks can be categorized into different types based on their approach and target stage in the AI system lifecycle. Some common types include poisoning attacks, which occur during the training phase where an attacker introduces malicious data into the training set, compromising the model's performance; evasion attacks, performed at the inference stage where an attacker modifies input data to evade detection or cause misclassification; transfer attacks, which leverage the vulnerability of one model to attack another by exploiting the similarities between models trained on similar data; and model extraction attacks, which involve extracting the model's parameters or architecture by querying the model and analyzing its responses. The C&W attack is primarily an evasion attack, distinguished by its method of crafting adversarial examples. Unlike simpler methods that may apply uniform noise or perturbations, the C&W attack utilizes an optimization-based approach to produce more subtle and effective adversarial examples.

The real-world implications of adversarial attacks are profound, with documented incidents across various domains. These incidents underscore the necessity for heightened security measures and continuous monitoring. Recent advancements in the field have focused on

improving the robustness of AI models and developing sophisticated detection mechanisms. Collaborations between academic institutions, industry leaders, and governmental bodies are crucial for addressing these challenges and fostering innovation in defense strategies.

Comprehensive defense strategies against adversarial attacks include adversarial training, where models are trained with adversarial examples to improve their resilience, as well as data augmentation and the implementation of intrusion detection and anomaly detection systems. Regularly updating and patching AI models to address vulnerabilities, alongside strict access controls and real-time monitoring systems, can also help mitigate the risks associated with these attacks. The impact on AI integration is significant, as securing AI systems is paramount for their continued adoption and trust in critical applications.

Understanding and simulating adversarial attacks, such as the C&W attack, are crucial for developing robust AI systems. By studying these attack techniques and their impact on AI models, researchers can devise effective countermeasures to enhance the security and reliability of AI applications across various domains.

**Literature Survey**

The rapid growth of artificial intelligence (AI) and machine learning (ML) technologies has led to their integration into critical areas such as healthcare, finance, and autonomous transportation. However, this integration has also revealed the vulnerabilities of AI systems, particularly their susceptibility to adversarial attacks.

Adversarial attacks are intentional efforts to deceive AI models by introducing small, often unnoticeable changes to input data. These modifications exploit the sensitivity of ML models to minor alterations, causing them to make incorrect predictions or decisions. The impact of such attacks can be significant, leading to financial losses, security breaches, and even life threatening situations in critical applications.

Understanding the nature and consequences of adversarial attacks is essential for ensuring the reliability and trustworthiness of AI systems. Researchers have identified various types of adversarial attacks, including poisoning attacks, evasion attacks, transfer attacks, and model extraction attacks. These attacks can target different stages of the AI system lifecycle, from the training phase to the deployment phase.

To combat these threats, researchers and practitioners have developed several defensive strategies. One approach is adversarial training, where models are trained with adversarial examples to improve their resilience. Data augmentation and the implementation of intrusion detection and anomaly detection systems are also effective methods. Additionally, regularly updating and patching AI models to address vulnerabilities and implementing strict access controls and monitoring systems can help detect and mitigate adversarial activities in real time.

This literature review aims to provide a comprehensive overview of current research on adversarial attacks and defenses in AI models. By exploring different attack techniques, their real-world implications, and proposed countermeasures, this review seeks to contribute to

ongoing efforts to secure AI systems across various applications. The review also examines the impact of adversarial attacks on AI models, such as data poisoning and privacy leakage. Finally, the review explores the potential of AI-based defenses, such as adversarial training and adversarial testing, to mitigate the risks associated with adversarial attacks.

THEMATIC EXPLANATION OF DIFFERENT TYPES OF ADVERSARIAL ATTACKS

A thematic literature survey comparing various adversarial attacks on AI systems is crucial for understanding their impact and developing more robust models. This survey focuses on several well-known attacks: Projected Gradient Descent (PGD), Fast Gradient Sign Method

(FGSM), Carlini & Wagner (CW), DeepFool, Boundary Attack, Jacobian-based Saliency Map Attack (JSMA), Expectation Over Transformation (EOT), Zeroth-Order Optimization (ZOO), HopSkipJumpAttack, and SparseFool.

Projected Gradient Descent (PGD)

Projected Gradient Descent (PGD) is a widely-used method for generating adversarial examples with strong perturbations. It iteratively perturbs input data to maximize the loss function, constrained within an epsilon ball around the original data point (Madry et al., 2019). PGD systematically explores the perturbation space, aiming to find the most effective adversarial perturbations that can fool a neural network model. This attack method is especially useful for evaluating the robustness of AI systems, as it often uncovers vulnerabilities that simpler methods might miss. PGD's impact on AI systems is significant, as it provides a rigorous benchmark for assessing model robustness against adversarial attacks.

Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) perturbs input data by adding noise proportional to the sign of the gradient of the loss function with respect to the input (Goodfellow et al., 2015). This method is designed for computational efficiency, making it suitable for large scale applications. FGSM leverages the vulnerability of neural networks to small, carefully crafted perturbations, demonstrating how even minor modifications can lead to incorrect predictions. The impact of FGSM on AI systems highlights the need for robust models that can withstand adversarial noise without compromising performance.

Carlini & Wagner (CW)

The Carlini & Wagner (CW) attack is an optimization-based method that finds adversarial examples by solving a specific optimization problem, minimizing a defined loss function subject to constraints that ensure the perturbations remain small (Carlini & Wagner, n.d.). This attack is known for its effectiveness in bypassing various defenses, making it a strong benchmark for evaluating model robustness. The CW attack is particularly adept at generating subtle adversarial examples that can evade detection, posing a significant challenge to current defense mechanisms. Its impact on AI systems is profound, often revealing weaknesses that other attacks fail to exploit.

DeepFool

DeepFool iteratively perturbs an input data point until it crosses the decision boundary of a deep neural network (Moosavi-Dezfooli et al., n.d.). This approach aims to achieve high fooling rates with minimal perturbations by measuring the distance to the decision boundary and adjusting the input accordingly. DeepFool's ability to generate subtle adversarial examples with minimal changes makes it a powerful tool for testing model robustness. The impact of DeepFool on AI systems is notable for its effectiveness in generating adversarial examples that are difficult to distinguish from legitimate inputs.

Boundary Attack

Boundary Attack is a decision-based adversarial attack that does not rely on gradient information (Brendel et al., n.d.). It starts with a large perturbation and reduces it while staying on the decision boundary. This method is particularly effective against black-box models where gradient information is unavailable. Boundary Attack's ability to operate without gradient information makes it a versatile tool for testing the robustness of AI systems in real-world scenarios. The impact of Boundary Attack on AI systems lies in its robustness against models with gradient masking defenses, highlighting the need for comprehensive robustness strategies.

Jacobian-based Saliency Map Attack (JSMA)

JSMA perturbs the input by modifying pixels that have the highest or lowest influence on the classifier’s output (McDaniel et al., 2016). This method targets specific features within the input, making it highly effective in generating adversarial examples with minimal changes. JSMA's targeted approach allows for precise control over the adversarial perturbations, leading to more efficient attacks. The impact of JSMA on AI systems emphasizes the importance of understanding feature sensitivity and robustness at the pixel level.

Expectation Over Transformation (EOT)

EOT synthesizes robust adversarial examples by applying random transformations to the input during the attack process (Athalye et al., n.d.). This method aims to generate adversarial examples that remain effective under various transformations, such as rotations and translations. EOT's approach to creating robust adversarial examples that can withstand multiple transformations highlights the challenges in defending against such attacks. The impact of EOT on AI systems underscores the necessity for defense mechanisms that can protect against adversarial examples robust to different transformations.

Zeroth-Order Optimization (ZOO)

Zeroth-Order Optimization (ZOO) enables adversarial attacks by estimating gradients through zeroth-order optimization techniques, avoiding the need for gradient information (Chen et al., 2017). This method is effective against black-box models where gradients are

inaccessible. ZOO's ability to estimate gradients without direct access to model internals makes it a powerful tool for adversarial attacks in practical scenarios. The impact of ZOO on AI systems demonstrates the feasibility of adversarial attacks in black-box settings, stressing the importance of robustness even when model internals are unknown.

HopSkipJumpAttack

HopSkipJumpAttack generates adversarial examples using minimal queries by iteratively perturbing an initial random perturbation toward misclassification (Chen et al., n.d.). This method contrasts with gradient-based approaches by focusing on query efficiency, making it suitable for scenarios with limited query budgets. HopSkipJumpAttack's efficiency in generating adversarial examples with minimal queries highlights the vulnerabilities of AI systems in query-limited settings. The impact of HopSkipJumpAttack on AI systems reveals the need for robust defenses against decision-based attacks.

SparseFool

SparseFool is designed to generate adversarial examples with minimal perturbations, altering only a few pixels to fool classifiers (Modas et al., n.d.). This method challenges the robustness of models by exploiting their reliance on specific features within the input. SparseFool's ability to create adversarial examples with sparse modifications makes it a significant threat to AI systems, as it can evade detection while causing misclassification. The impact of SparseFool on AI systems highlights the necessity of defenses that can withstand sparse adversarial perturbations, ensuring model reliability even with minimal input changes.

In summary, each adversarial attack method exploits different aspects of AI systems, revealing various vulnerabilities. By understanding these attacks, researchers can develop more robust models capable of withstanding a wide range of adversarial perturbations, enhancing the overall security and reliability of AI systems.

COMPARISION BETWEEN THE ADVESARIAL ATTACKS

We will compare various adversarial attacks on AI systems using the following criteria points:

Attack Methodology: Describes the specific technique or algorithm used to generate adversarial examples, which helps in understanding the nature and complexity of the attack

Perturbation Norm: Indicates the mathematical norm (e.g., L0, L2, L∞) used to measure the perturbation, which is essential for quantifying the extent of changes made to the input data

Perturbation Size: The magnitude of the perturbation is crucial for evaluating the balance between attack effectiveness and imperceptibility

Targeted vs. Non-Targeted: Determines whether the attack aims for a specific incorrect classification (targeted) or any incorrect classification (non-targeted), impacting the attack's complexity and success rate

White-Box vs. Black-Box: Refers to the level of access the attacker has to the model's parameters and structure, affecting the feasibility and approach of the attack

Transferability: The ability of adversarial examples to deceive different models, which is important for assessing the broader impact of an attack

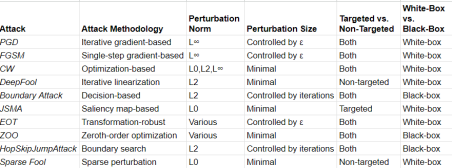
Attack Success Rate: Measures how often the attack successfully causes misclassification, indicating its effectiveness

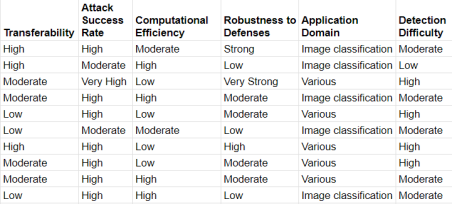
Computational Efficiency: The resources required to perform the attack, which is important for understanding the practicality of deploying the attack

Robustness to Defenses: The ability of the attack to bypass existing defensive measures, highlighting its sophistication and resilience

Application Domain: Specifies the field (e.g., image classification, speech recognition) where the attack is most effective, which helps in assessing its real-world relevance

Detection Difficulty: How challenging it is to detect the adversarial examples, which is crucial for evaluating the stealthiness of the attack

Fig(a):Part 1 of the comparison between the above adversarial attack.

Fig(b):Part 2 of the comparison between the above adversarial attack.

The table compares various adversarial attack methods in machine learning across several criteria. Each attack method has a distinct approach, from gradient-based (FGSM, PGD) to optimization-based (CW) and decision boundary manipulation (DeepFool). Perturbation norms like L0, L2, and L∞ measure the impact of changes, with attacks being either targeted or non-targeted. Methods vary in computational efficiency and robustness against defenses, with some like CW showing high success rates and robustness. The table also highlights the transferability and detection difficulty of each method, with applications primarily in image classification but extending to other domains.

COMPARISION BETWEEN DIFFERENT METHODOLOGIES

In AI security, various attack methods exploit machine learning model vulnerabilities, using specific techniques to compromise system integrity, confidentiality, or availability. Below, we detail 30 distinct methodologies.

1. Adversarial Examples:

- Adversarial examples involve creating inputs with small, carefully crafted perturbations that cause the model to make incorrect predictions. These perturbations are often imperceptible to humans but can significantly impact the model's output. The methodology typically involves using gradient-based optimization techniques, such as the Fast Gradient Sign Method (FGSM) or Projected Gradient Descent (PGD), to determine the minimal changes needed to mislead the model. Researchers often generate these adversarial examples by iteratively adjusting the input to maximize the model's prediction error.

2. Model Inversion:

- Model inversion attacks aim to reconstruct input data from the model's outputs. By leveraging the relationships between inputs and outputs, attackers can infer sensitive information about the training data. The methodology involves querying the model with various inputs and analyzing the corresponding outputs to reverse-engineer the original data. This process may use techniques like gradient descent or other optimization methods to

minimize the difference between the model's outputs and the desired target outputs, effectively revealing private information.

3. Data Poisoning:

- Data poisoning attacks involve injecting malicious data into the training set to corrupt the learning process. The methodology includes crafting poisoned data points that lead the model to learn incorrect patterns. These attacks can be carried out by subtly modifying legitimate training data or introducing entirely new malicious data points. The poisoned data is designed to look similar to regular data, making it difficult to detect. During training, the model incorporates these malicious examples, leading to incorrect or harmful behavior during inference.

4. Evasion Attacks:

- Evasion attacks focus on creating inputs that can bypass the model's defenses and cause it to produce incorrect predictions. The methodology involves generating inputs with slight alterations that are undetectable to humans but can fool the model. This is commonly used in security contexts, such as bypassing spam filters or malware detectors. Techniques like gradient-based optimization or heuristic methods are employed to craft these inputs, ensuring they remain effective against the model while evading detection mechanisms.

5. Model Stealing:

- Model stealing attacks involve replicating a target model by extensively querying it and using the input-output pairs to train a surrogate model. The methodology includes gathering data through the model's API, often using techniques like active learning to efficiently approximate the target model. By making a large number of queries and collecting the corresponding outputs, attackers can train their own model that mimics the behavior of the original. This stolen model can then be used for unauthorized purposes or further attacks.

6. Membership Inference Attacks:

- Membership inference attacks determine whether a specific data point was part of the model's training set by analyzing the model's output confidence scores. The methodology exploits differences in model behavior between training and non-training data to infer sensitive information. Attackers query the model with a particular input and observe the output probabilities, using statistical methods to assess the likelihood that the input was part of the training data. This can reveal private information about individuals whose data was used during training.

7. Backdoor Attacks:

- Backdoor attacks embed hidden triggers in the training data that cause the model to behave maliciously when the trigger is present in the input data. The methodology involves introducing these triggers during training, making the model exhibit unexpected behavior under specific conditions. For example, an attacker might add a particular pattern or

watermark to the training images that, when seen during inference, causes the model to misclassify the input. These triggers are typically designed to be subtle and difficult to detect.

8. Exploratory Attacks:

- Exploratory attacks probe the model to gather information about its structure, decision boundaries, and vulnerabilities. The methodology includes systematic querying and analysis to gain insights that can be used to design more targeted attacks. By analyzing the model's responses to various inputs, attackers can map out the decision boundaries and identify weak points. This information can then be used to craft more effective adversarial examples or other types of attacks that exploit the identified vulnerabilities.

9. Adaptive Attacks:

- Adaptive attacks dynamically adjust their strategies based on feedback from the model. The methodology involves starting with an initial approach and iteratively modifying it in response to the model's defences and outputs to improve the attack's success rate. Attackers monitor the model's responses to their inputs and adapt their techniques accordingly, making their attacks more effective over time. This can include modifying input perturbations, adjusting attack parameters, or changing the attack vector entirely.

10. Transferability Attacks:

- Transferability attacks exploit the fact that adversarial examples created for one model can often be effective against another model trained on similar data. The methodology involves generating adversarial inputs for a surrogate model and using them to attack the target model. This leverages the commonalities between different models' decision boundaries, making it possible to transfer attacks across models. Transferability attacks highlight the challenges in defending against adversarial examples, as securing one model does not necessarily protect against attacks on similar models.

11. Model Parameter Manipulation:

- Model parameter manipulation attacks alter the model's training process by manipulating gradient updates or optimization steps. The methodology includes injecting adversarial gradients or modifying the loss function to bias the learning process. Attackers can subtly influence how the model updates its parameters during training, causing it to learn incorrect patterns or behaviors. This can be achieved by tampering with the training data, altering the training algorithm, or directly manipulating the model's internal parameters.

12. Privacy Attacks:

- Privacy attacks extract sensitive information from the model’s outputs, revealing private data about individuals in the training set. The methodology includes techniques like model inversion, membership inference, and data extraction to infer private information. Attackers leverage the model's predictions and confidence scores to reconstruct input data or identify

specific examples from the training set. These attacks can compromise the confidentiality of the training data and expose sensitive information.

13. Resource Exhaustion:

- Resource exhaustion attacks overwhelm the AI system by generating excessive requests, leading to denial-of-service (DoS). The methodology includes exploiting the system's capacity limits to disrupt normal operations and degrade performance. Attackers flood the system with a high volume of requests, consuming computational resources, memory, or network bandwidth. This can cause the system to slow down, crash, or become unresponsive, impacting its availability and reliability.

14. Fault Injection:

- Fault injection attacks introduce deliberate errors or faults into the system to disrupt model performance. The methodology includes manipulating hardware components or software aspects to cause the model to make incorrect predictions or behave unpredictably. Attackers can inject faults at various levels, such as modifying input data, tampering with intermediate computations, or altering the final outputs. These faults can lead to degraded performance, increased error rates, or unexpected behavior in the model.

15. Social Engineering:

- Social engineering attacks exploit human weaknesses to gain unauthorized access or information. The methodology includes manipulating individuals through psychological tactics, such as phishing, pretexting, or baiting, to obtain confidential data or credentials. Attackers deceive individuals into revealing sensitive information or performing actions that

compromise the security of the AI system. This can involve impersonating trusted entities, creating fake scenarios, or leveraging social manipulation techniques to achieve their goals.

16. Model Subversion:

- Model subversion involves subtle modifications to the model to alter its behavior without being easily detectable. The methodology includes changing weights, introducing biases, or embedding hidden functionality that activates under specific conditions. Attackers can modify the model's internal parameters or structure to cause it to produce incorrect or malicious outputs under certain inputs. These changes are often designed to be inconspicuous, making it difficult to detect the subversion without thorough inspection.

17. Trojan Attacks:

- Trojan attacks embed hidden malicious functionality in the model that activates when specific conditions are met. The methodology involves introducing triggers during training, causing the model to exhibit malicious behavior only when the trigger is present. For example, an attacker might embed a specific pattern in the training data that, when encountered during inference, causes the model to misclassify or produce harmful outputs.

These triggers are designed to be subtle and difficult to detect, ensuring the Trojan remains hidden until activated.

18. Data Extraction Attacks:

- Data extraction attacks aim to extract sensitive data used during the model’s training. The methodology includes techniques like model inversion or side-channel analysis to reconstruct training data and expose private or proprietary information. Attackers analyze the model's outputs and internal states to infer the data it was trained on. This can compromise the confidentiality of the training data and reveal sensitive information about individuals or proprietary datasets.

19. Model Obfuscation:

- Model obfuscation involves obscuring the model's functionality or outputs to make it difficult for attackers to analyze or interpret. The methodology includes adding noise, encrypting weights, or using techniques that conceal the model's internal workings. By making the model's structure and parameters less transparent, obfuscation aims to prevent reverse engineering and protect against model theft or unauthorized use. This can involve techniques like parameter encryption, adding random perturbations to outputs, or using complex architectures that are harder to decipher.

20. Model Manipulation through API:

- Model manipulation through API exploits vulnerabilities in the model's API to alter its behavior. The methodology includes injecting adversarial inputs, altering input-output mappings, or exploiting API vulnerabilities to cause the model to produce incorrect or malicious outputs. Attackers can manipulate the model's responses by crafting inputs that exploit weaknesses in the API's implementation. Securing the API with robust authentication, input validation, and monitoring mechanisms can help prevent unauthorized access and manipulation.

21. Stealth Attacks:

- Stealth attacks aim to remain undetected while compromising the model. The methodology includes making gradual, subtle changes to inputs or model parameters to avoid triggering security mechanisms. By carefully adjusting inputs or modifying the model's internal state, attackers can achieve their malicious goals without raising alarms. These attacks often involve a slow, incremental approach to avoid detection and maintain long-term access to the compromised system.

22. Gradient-Based Attacks:

- Gradient-based attacks use gradient information to generate adversarial examples that maximize prediction errors. The methodology includes calculating perturbations that effectively mislead the model, exploiting its sensitivity to specific features. By analyzing the gradients of the model's loss function with respect to the input, attackers can identify the most

impactful changes to the input data. Techniques like the Fast Gradient Sign Method (FGSM) or Projected Gradient Descent (PGD) are commonly used to craft these adversarial examples.

23. Side-Channel Attacks:

- Side-channel attacks extract information from physical side effects of model computation, such as power consumption or timing information. The methodology includes analyzing these side channels to infer internal states or data, compromising model confidentiality. Attackers can monitor the model's physical characteristics, such as electromagnetic emissions or execution timing, to gain insights into its internal processes. This information can be used to reconstruct sensitive data or understand the model's behavior.

24. Label Flipping:

- Label flipping attacks corrupt the training data by flipping labels, causing the model to learn incorrect associations. The methodology includes changing the labels of benign samples to malicious ones, leading to misclassification during inference. Attackers can introduce mislabeled data into the training set, causing the model to associate incorrect labels with certain inputs. This can lead to systematic misclassification and degraded model performance.

25. Sparse Vector Attacks:

- Sparse vector attacks exploit the sparse nature of data representations to introduce adversarial perturbations. The methodology includes manipulating a small number of key features to cause significant changes in the model's output. By identifying and altering critical features in the input data, attackers can maximize the impact of their perturbations while minimizing the overall changes. This can be particularly effective in models that rely on sparse input representations, such as text or high-dimensional data.

26. Model Drift Exploitation:

- Model drift exploitation takes advantage of changes in model performance over time due to evolving data distributions. The methodology includes introducing data that causes the model to drift, leading to degraded performance and less accurate predictions. Attackers can systematically introduce data that shifts the distribution of the training or input data, causing the model to adapt to these changes in a way that degrades its performance. This can lead to reduced accuracy, increased error rates, or biased predictions.

27. Federated Learning Attacks:

- Federated learning attacks exploit vulnerabilities in federated learning frameworks, where multiple nodes collaboratively train a model without sharing raw data. The methodology includes injecting malicious updates or extracting data from participating nodes. Attackers can introduce poisoned updates during the federated learning process, causing the global model to learn incorrect patterns. Additionally, they can analyze updates from other nodes to infer sensitive information about their local data.

28. Model Watermarking Attacks:

- Model watermarking attacks embed hidden identifiers in the model to trace unauthorized use or distribution. The methodology includes embedding unique markers that can be used to identify stolen or misused models. By introducing specific patterns or modifications during training, the model can be uniquely identified through these watermarks. This can help detect and trace unauthorized copies of the model, protecting intellectual property and preventing misuse.

29. Black-Box Optimization Attacks:

- Black-box optimization attacks optimize adversarial inputs by querying the model without accessing its internals. The methodology includes using techniques like genetic algorithms to refine inputs and cause the model to produce incorrect outputs. Attackers can systematically query the model with different inputs, analyzing the outputs to identify patterns that maximize the prediction error. This iterative process allows them to craft effective adversarial examples without needing access to the model's internal structure.

30. Algorithmic Bias Exploitation:

- Algorithmic bias exploitation involves manipulating inherent biases in AI algorithms to skew results. The methodology includes introducing biased data or manipulating training processes to cause the model to produce systematically biased outputs. Attackers can exploit existing biases in the training data or model architecture to influence the model's predictions in a way that favors certain outcomes. This can lead to biased decision-making, discrimination, or other harmful effects.

| **Adversarial and**  **Evasion Attacks** | **Data and Model**  **Manipulation Attacks** | **Infrastructure and Side Channel Attacks** |
| --- | --- | --- |
| Adversarial Examples | Model Inversion | Exploratory Attacks |
| Evasion Attacks | Data Poisoning | Privacy Attacks |
| Adaptive Attacks | Model Stealing | Resource Exhaustion |
| Transferability Attacks | Membership Inference  Attacks | Fault Injection |
| Gradient-Based Attacks | Backdoor Attacks | Social Engineering |
| Label Flipping | Model Parameter  Manipulation | Stealth Attacks |

| Sparse Vector Attacks | Model Subversion | Side-Channel Attacks |
| --- | --- | --- |
| Model Drift Exploitation | Trojan Attacks |  |
| Black-Box Optimization Attacks | Data Extraction Attacks |  |
|  | Model Obfuscation |  |
|  | Model Manipulation through API |  |
|  | Federated Learning Attacks |  |
|  | Model Watermarking  Attacks |  |
|  | Algorithmic Bias  Exploitation |  |

The table categorizes methodologies for attacking AI systems into three groups: 1. Adversarial and Evasion Attacks:

These attacks trick the AI model into making incorrect predictions by manipulating input data, exploiting the model's vulnerabilities. Examples include adversarial examples and evasion attacks.

2. Data and Model Manipulation Attacks:

These involve tampering with training data or model parameters to alter behavior, degrade performance, or extract information. Examples include data poisoning, model stealing, and backdoor attacks.

3. Infrastructure and Side-Channel Attacks:

These target the AI system's operational environment, exploiting hardware, software, and human factors. Examples include side-channel attacks, resource exhaustion, and social engineering.

In conclusion, the detailed exploration of various adversarial attacks on AI systems, including PGD, FGSM, CW, DeepFool, Boundary Attack, JSMA, EOT, ZOO, HopSkipJumpAttack, and SparseFool, reveals the significant challenges these attacks pose to the security and reliability of AI applications. Each method employs unique techniques to generate adversarial examples, whether through gradient-based approaches, optimization of custom loss functions, or iterative processes to approximate decision boundaries. The study of these adversarial attacks provides crucial insights into the diverse methodologies used to deceive AI systems,

highlighting the importance of understanding the underlying mechanisms that make these attacks successful.

Analysing the perturbation norms and sizes used in these attacks, it becomes evident that the choice of norm and the magnitude of the perturbations play critical roles in the effectiveness of the attack and its perceptibility to human observers. Methods like PGD and CW optimize for specific norms to ensure that perturbations are both minimal and effective, emphasizing

the need for careful consideration of these factors in designing robust defense mechanisms.

The examination of targeting strategies and model access levels reveals the varying degrees of sophistication required for different attacks. While white-box attacks like PGD benefit from full access to model parameters, making them highly effective, black-box attacks such as Boundary Attack and ZOO demonstrate versatility by relying solely on model outputs. This distinction underscores the importance of developing defense strategies that can counteract both white-box and black-box attacks, ensuring comprehensive protection for AI systems.

Transferability is another critical aspect explored in this survey, where the ability of adversarial examples to generalize across different models poses a broader threat to AI applications. Understanding the factors that contribute to transferability helps in devising more resilient models that can withstand attacks targeting not only the primary model but also other related systems.

The success rate and computational efficiency of adversarial attacks are essential metrics that determine their practical feasibility. Single-step methods like FGSM offer quick and efficient attacks but may lack the potency of iterative methods like PGD and CW, which, despite being computationally intensive, achieve higher success rates. Balancing these factors is crucial for developing defenses that can effectively mitigate attacks without imposing excessive computational burdens.

Robustness to defenses is a recurring theme in the analysis of these attacks. Techniques like adversarial training and gradient masking are commonly employed to enhance model resilience, yet attacks such as CW are specifically designed to bypass these defenses. This highlights the ongoing arms race between attack strategies and defense mechanisms, driving the need for continuous innovation in security practices.

The application domains and detection difficulties of adversarial attacks extend beyond image recognition to fields like natural language processing and cybersecurity. The subtlety of perturbations in methods like SparseFool, which creates imperceptible changes, complicates the detection process, necessitating advanced techniques to identify and mitigate such threats effectively.

This literature survey underscores the critical need for robust defense mechanisms that combine multiple approaches, including model robustness, anomaly detection, and secure data handling. The comprehensive comparison of adversarial attacks provided here not only enhances our understanding of their mechanisms and impacts but also serves as a valuable

resource for researchers and practitioners in the field of AI security. By delving into the intricacies of each attack method, we gain the knowledge necessary to develop more resilient AI systems capable of withstanding adversarial threats.

The purpose of this literature survey was to offer a thorough examination of various adversarial attacks, shedding light on the complexities and challenges associated with securing AI systems. This knowledge is instrumental in guiding the development of advanced defense strategies that can protect AI applications from potential adversarial threats. The insights gained from this survey will prove invaluable for those working in AI security, fostering the creation of more robust and reliable AI systems.

By synthesizing the current state of research on adversarial attacks, this survey provides a foundation for future investigations aimed at enhancing AI security. The detailed analysis of each attack method, coupled with the exploration of defense mechanisms and application domains, equips researchers with the information needed to address the evolving landscape of adversarial threats. Ultimately, this literature survey contributes to the broader effort to safeguard AI systems, ensuring their safe and reliable deployment across various domains.

In essence, this survey highlights the importance of continuous vigilance and innovation in the field of AI security. As adversarial attacks evolve and become more sophisticated, so too must our defense strategies. The knowledge compiled in this survey serves as a crucial tool for researchers and practitioners, enabling them to stay ahead of emerging threats and protect AI systems from potential adversarial exploits. By fostering a deeper understanding of adversarial attacks and their implications, this survey paves the way for the development of more resilient and secure AI applications, ultimately contributing to the advancement of the field and the protection of AI-driven technologies.

**Methodology**

The CW attack, introduced by Nicholas Carlini and David Wagner in 2017, is known for its effectiveness in creating adversarial examples that are difficult to detect and defend against. Its use of optimization functions to perturb images, as opposed to heuristic approaches like PGD and FGSM, makes it both computationally efficient and particularly challenging. This sophisticated approach allows the attack to minimize perturbations such that the differences between the original and adversarial images are imperceptible to humans but detectable by AI.

Initially, we aimed to implement the CW attack using the CIFAR-10 dataset. However, we encountered significant challenges, primarily due to the increased complexity of CIFAR-10 compared to MNIST. As a result, we shifted our focus to MNIST, which, with its 28x28 grayscale images, offered a simpler and more manageable dataset for our experiments. MNIST's less complex nature not only facilitated a clearer understanding of the fundamentals but also allowed for faster training speeds, making it ideal for our initial explorations.

Our model for this experiment was a straightforward Convolutional Neural Network (CNN). It consisted of two convolutional layers followed by two fully connected layers. The first convolutional layer applied 32 filters with a 3x3 kernel to grayscale images, followed by ReLU activation and padding to maintain spatial dimensions. The second convolutional layer used 64 filters with similar settings. Each convolutional layer was followed by a 2x2 max pooling operation to down sample the feature maps.

For adversarial attack simulations, we utilized the CW attack, which involves iteratively optimizing perturbed images to achieve minimal perceptual changes while ensuring misclassification. The attack process begins by cloning and detaching the input images, setting them to require gradients to enable optimization. We used an Adam optimizer to adjust pixel values, aiming to minimize a loss function that combines mean squared error (MSE) loss and negative cross-entropy loss. This loss function encourages minimal perturbations while driving the model to misclassify the perturbed images.

After generating adversarial examples, we evaluated their impact by comparing predictions on both original and adversarial images. We also calculated the perturbation as the difference between adversarial and original images. Visualizations of the results, including original images, adversarial images, and perturbations, were produced using matplotlib to illustrate the effects of the CW attack clearly.

By employing the MNIST dataset and a simple CNN model, we were able to effectively study the CW attack's behavior and its implications, providing valuable insights into adversarial machine learning techniques.

**Result**

The results depicted in the visualization indicate the successful execution of the Carlini & Wagner (C&W) attack, demonstrating its effectiveness in deceiving the AI model. The original image, which was accurately predicted as '7' by the model, was subtly altered to create an adversarial example that the model misclassified as '1'. The perturbations introduced are minimal and visually indistinguishable to the human eye, yet they are sufficient to mislead the AI model. The perturbation image highlights the specific modifications made to the original image to generate the adversarial example. These perturbations are subtle but strategically designed to exploit the model's weaknesses. This experiment underscores the vulnerability of AI models to adversarial attacks and emphasizes the necessity for developing robust defense mechanisms to safeguard against such manipulations.

**Conclusion**

The experiment successfully demonstrates the vulnerability of AI models to adversarial attacks, particularly through the Carlini & Wagner (C&W) attack method. The visualization

shows that a seemingly negligible alteration to an input image can drastically mislead the model. Specifically, an image originally and correctly identified as the digit '7' by the model was modified with imperceptible perturbations, resulting in its misclassification as '1'.

This outcome underscores the critical challenge adversarial attacks pose to AI systems, especially as they become integral to sectors such as healthcare, finance, and autonomous transportation. The success of the C&W attack in this experiment highlights the necessity for robust defense mechanisms to ensure the reliability and security of AI systems.

Adversarial attacks can lead to significant real-world consequences, such as financial losses, security breaches, and even life-threatening situations. Therefore, developing and implementing effective defenses is paramount. Techniques such as adversarial training, data augmentation, and advanced intrusion detection systems are essential strategies for enhancing the resilience of AI models against these sophisticated attacks. Regular updates, strict access controls, and real-time monitoring further bolster these defenses.

In summary, this experiment illustrates the pressing need for continued research and development in securing AI systems against adversarial threats. Ensuring the trustworthiness of AI in critical applications requires a comprehensive approach to understanding, detecting, and mitigating adversarial attacks.

REFERENCES:-

1. Madry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A. (2019). Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [cs, stat]. [online] Available at: https://arxiv.org/abs/1706.06083.

2. Goodfellow, I., Shlens, J. and Szegedy, C. (n.d.). Published as a conference paper at ICLR 2015 EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES. [online] Available at: https://arxiv.org/pdf/1412.6572.

3. Carlini, N. and Wagner, D. (n.d.). Towards Evaluating the Robustness of Neural Networks. [online] Available at: https://arxiv.org/pdf/1608.04644 [Accessed 4 Jul. 2024].

4. Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P., Polytechnique, E. and De Lausanne, F. (n.d.). DeepFool: a simple and accurate method to fool deep neural networks. [online] Available at: https://arxiv.org/pdf/1511.04599 [Accessed 19 May 2024].

5. Brendel, W., Rauber, J. and Bethge, M. (n.d.). DECISION-BASED ADVERSARIAL ATTACKS: RELIABLE ATTACKS AGAINST BLACK-BOX MACHINE LEARNING MODELS. [online] Available at: https://arxiv.org/pdf/1712.04248 [Accessed 4 Jul. 2024].

6. McDaniel, P., Papernot, N. and Celik, Z.B. (2016). Machine Learning in Adversarial Settings. IEEE Security & Privacy, [online] 14(3), pp.68–72.

doi:https://doi.org/10.1109/msp.2016.51.

7. Athalye, A., Engstrom, L., Ilyas, A. and Kwok, K. (n.d.). Synthesizing Robust Adversarial Examples. [online] Available at: https://arxiv.org/pdf/1707.07397 [Accessed 4 Jul. 2024].

8. Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J. and Hsieh, C.-J. (2017). ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models. [online] 17. doi:https://doi.org/10.1145/3128572.3140448.

9. Chen, J., Jordan, M. and Wainwright, M. (n.d.). HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. [online] Available at: https://arxiv.org/pdf/1904.02144 [Accessed 4 Jul. 2024].

10. Modas, A., Moosavi-Dezfooli, S.-M., Frossard, P., Polytechnique, E. and De Lausanne, F. (n.d.). SparseFool: a few pixels make

11. Gurina A, Eliseev V. Quality Criteria and Method of Synthesis for Adversarial Attack Resistant Classifiers. *Machine Learning and Knowledge Extraction*. 2022; 4(2):519-541. https://doi.org/10.3390/make4020024

12.Kurniawan, A., Ohsita, Y. and Murata, M. (2022). Experiments on Adversarial Examples for Deep Learning Model Using Multimodal Sensors. *Sensors*, 22(22), p.8642. doi:https://doi.org/10.3390/s22228642.

13.Fredrikson, M., Jha, S. and Ristenpart, T. (2015). Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS ’15*. [online] doi:https://doi.org/10.1145/2810103.2813677.

14.Simms, N. (2023). *Data Poisoning: A New Threat to Artificial Intelligence Data Poisoning: A New Threat to Artificial Intelligence*. [online] Available at: https://digitalcommons.lasalle.edu/cgi/viewcontent.cgi?article=1051&context=mathcompcap stones.

15.Simms, N. (2023). *Data Poisoning: A New Threat to Artificial Intelligence Data Poisoning: A New Threat to Artificial Intelligence*. [online] Available at: https://digitalcommons.lasalle.edu/cgi/viewcontent.cgi?article=1051&context=mathcompcap stones.

16.Daryna Oliynyk, Mayer, R. and Rauber, A. (2023). I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences. doi:https://doi.org/10.1145/3595292

17.Hu, H., Zhang, X., Salcic, Z., Sun, L., Choo, K.-K.R. and Dobbie, G. (2024). Source Inference Attacks: Beyond Membership Inference Attacks in Federated Learning. *IEEE*

*Transactions on Dependable and Secure Computing*, [online] pp.1–18. doi:https://doi.org/10.1109/tdsc.2023.3321565.

18.Saha, A., Subramanya, A. and Pirsiavash, H. (2020). Hidden Trigger Backdoor Attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07), pp.11957– 11965. doi:https://doi.org/10.1609/aaai.v34i07.6871.

19.EXPLOREADV: TOWARDS EXPLORATORY ATTACK FOR NEURAL NETWORKS. (2022). Available at: https://arxiv.org/pdf/2301.01223.

20.Tramèr, F., Carlini, N., Google and Brendel, W. (n.d.). *On Adaptive Attacks to Adversarial Example Defenses*. [online] Available at:

https://proceedings.nips.cc/paper/2020/file/11f38f8ecd71867b42433548d1078e38-Paper.pdf.

21.Liu, Y., Chen, X., Liu, C. and Song, D. (n.d.). *Published as a conference paper at ICLR 2017 DELVING INTO TRANSFERABLE ADVERSARIAL EX- AMPLES AND BLACK-BOX ATTACKS*. [online] Available at: https://arxiv.org/pdf/1611.02770.

Chen, X., Liu, C., Li, B., Lu, K. and Song, D. (n.d.). *Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning*. [online] Available at:

https://arxiv.org/pdf/1712.05526.

22.Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D. and Ristenpart, T. (n.d.). *Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing*. [online] Available at: https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper fredrikson-privacy.pdf.

23.Desnitsky, V., Kotenko, I. and Zakoldaev, D. (2019). Evaluation of Resource Exhaustion Attacks against Wireless Mobile Devices. *Electronics*, 8(5), p.500.

doi:https://doi.org/10.3390/electronics8050500.

24.Kurakin, A., Brain, G., Openai, I. and Bengio, S. (n.d.). *Workshop track -ICLR 2017 ADVERSARIAL EXAMPLES IN THE PHYSICAL WORLD*. [online] Available at: https://arxiv.org/pdf/1607.02533.

25.ResearchGate. (n.d.). *(PDF) SOCIAL ENGINEERING AND CYBER SECURITY*. [online] Available at:

https://www.researchgate.net/publication/315351300\_SOCIAL\_ENGINEERING\_AND\_CY BER\_SECURITY.

26.Nicolae, M.-I., Sinn, M., Tran, M., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I. and Edwards, B. (n.d.). *Adversarial Robustness Toolbox v1.0.0*. [online] Available at: https://arxiv.org/pdf/1807.01069.

27.Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W. and Zhang, X. (2018). Trojaning Attack on Neural Networks. *Proceedings 2018 Network and Distributed System Security Symposium*. doi:https://doi.org/10.14722/ndss.2018.23291.

28.Papernot, N. and Mcdaniel, P. (n.d.). *Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning*. [online] Available at:

https://arxiv.org/pdf/1803.04765 .

29.Uesato, J., O'donoghue, B., Van Den Oord, A. and Kohli, P. (n.d.). *Adversarial Risk and the Dangers of Evaluating Against Weak Attacks*. [online] Available at: https://arxiv.org/pdf/1802.05666.

30.Shafahi, A., Huang, R., Studer, C., Feizi, S. and Goldstein, T. (n.d.). *ARE ADVERSARIAL EXAMPLES INEVITABLE?* [online] Available at: https://wrhuang.com/publication/shafahi 2019-inevitable/shafahi-2019-inevitable.pdf.

31.Hitaj, B., Ateniese, G. and Perez-Cruz, F. (n.d.). *Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning*. [online] Available at: https://arxiv.org/pdf/1702.07464.

32.Mądry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A. (n.d.). *Towards Deep Learning Models Resistant to Adversarial Attacks*. [online] Available at: https://arxiv.org/pdf/1706.06083.

33.Nasr, M., Shokri, R. and Houmansadr, A. (n.d.). *Comprehensive Privacy Analysis of Deep Learning Passive and Active White-box Inference Attacks against Centralized and Federated Learning*. [online] Available at: https://www.comp.nus.edu.sg/~reza/files/Shokri-SP2019.pdf.

35.Goodfellow, I., Shlens, J. and Szegedy, C. (n.d.). *Published as a conference paper at ICLR 2015 EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES*. [online] Available at: https://arxiv.org/pdf/1412.6572.

36.Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J. and Lu, F. (2020). Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, p.107332. doi:https://doi.org/10.1016/j.patcog.2020.107332.

37.Zhu, C., Huang, W., Shafahi, A., Li, H., Taylor, G., Studer, C. and Goldstein, T. (n.d.). *Transferable Clean-Label Poisoning Attacks on Deep Neural Nets*. [online] Available at: https://proceedings.mlr.press/v97/zhu19a/zhu19a.pdf.

38.Melis, L., Song, C., De Cristofaro, E. and Tech, C. (n.d.). *Exploiting Unintended Feature Leakage in Collaborative Learning Vitaly Shmatikov*. [online] Available at: https://www.cs.cornell.edu/~shmat/shmat\_oak19.pdf.

39.Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D. and Shmatikov, V. (n.d.). *How To Backdoor Federated Learning*. [online] Available at:

https://proceedings.mlr.press/v108/bagdasaryan20a/bagdasaryan20a.

40.Uesato, J., O'donoghue, B., Van Den Oord, A. and Kohli, P. (n.d.). *Adversarial Risk and the Dangers of Evaluating Against Weak Attacks*. [online] Available at: https://proceedings.mlr.press/v80/uesato18a/uesato18a.pdf .

41.Yefet, N., Alon, U. and Yahav, E. (2020). Adversarial examples for models of code. *Proceedings of the ACM on Programming Languages*, [online] 4(OOPSLA), pp.1–30. doi:https://doi.org/10.1145/3428230.