International Journal of Safety and Security Engineering
Vol. 15, No. 8, August, 2025, pp. 1673-1679
Journal homepage: http://iieta.org/journals/ijsse

A Review on Adversarial Attacks and Defenses on Image Classification Models

Anuja Jana Naik[1*], Siddesh Vishnu Savant[2], François Maas[1]

[1] Department of Electronics and Computer Engineering, Padre Conceicao College of Engineering, Verna 403401, India
[2] Department of Computer Science and Engineering, Padre Conceicao College of Engineering, Verna 403401, India

Corresponding Author Email: anuja@pccegoa.edu.in

https://doi.org/10.18280/ijsse.150812

Received: 5 July 2025
Revised: 1 August 2025
Accepted: 15 August 2025
Available online: 31 August 2025

Keywords:
deep learning, image classification, adversarial attacks, adversarial examples, defense techniques, model robustness, deep learning security

ABSTRACT

Deep learning has transformed image classification by enabling machines to learn intricate patterns from large datasets. However, these models are increasingly exposed to adversarial attacks, small, carefully crafted changes to input images that can mislead even well-trained classifiers. This review offers a structured examination of such vulnerabilities by classifying attacks into five major types: white-box, black-box, poisoning, inference, and extraction. While prior surveys have broadly discussed these threats, this work distinguishes itself by drawing direct connections between specific attack strategies and practical defense mechanisms. It places particular emphasis on emerging methods like spectral signature analysis and feature squeezing, which are gaining traction for their applicability in real-world systems. The review also critically evaluates the effectiveness of established defenses such as adversarial training and defensive distillation. It highlights the persistent gaps and challenges in safeguarding image classification models. This paper serves as a resource for researchers and practitioners aiming to develop more resilient deep learning systems in security-sensitive domains.

1. INTRODUCTION

Deep learning has emerged as a foundational pillar of modern artificial intelligence, enabling breakthroughs across a range of fields, including image recognition, speech processing, and natural language understanding. Among these, image classification has advanced significantly with the advent of convolutional neural networks (CNNs), which excel at automatically extracting and learning complex visual features from large-scale datasets. These capabilities have facilitated the automation of tasks that once relied heavily on human judgment, leading to widespread adoption in high-stakes domains such as autonomous vehicles, medical diagnostics, and surveillance systems.

However, the increasing dependence on deep learning models has revealed a critical vulnerability: their susceptibility to adversarial attacks. These attacks involve subtly altered inputs, often imperceptible to human observers, that can mislead models into making incorrect predictions. For instance, a stop sign manipulated with minute perturbations might be misclassified by an autonomous vehicle as a speed limit sign, potentially resulting in catastrophic outcomes. The growing awareness of such vulnerabilities has fueled extensive research into understanding, detecting, and defending against adversarial manipulations.

The motivation behind this research is not merely academic, but it reflects the urgent need to ensure the safety, reliability, and trustworthiness of AI systems in real-world deployments. In sectors like finance, healthcare, and law enforcement, incorrect predictions due to adversarial interference can lead to ethical concerns, financial losses, or even human harm. Moreover, adversarial examples often exploit inherent properties of neural networks, such as high-dimensional input spaces and local linearity, making them challenging to eliminate completely. This persistent threat has driven the community to pursue more resilient training algorithms, improved model architectures, and robust evaluation frameworks.

Importantly, the challenge of adversarial robustness is not confined to computer vision alone. Similar vulnerabilities have been identified in speech recognition systems, language models, and even reinforcement learning agents. As AI models are increasingly integrated into interconnected environments like smart cities and edge computing systems, adversarial risks become compounded by exposure to dynamic and unpredictable data streams. This makes it imperative to design defenses that are not only technically sound but also scalable and adaptable to evolving threat landscapes.

While a substantial body of work has focused on designing attack strategies, defense mechanisms have developed in parallel. Initial approaches centered on input preprocessing techniques and defensive distillation aimed at reducing model sensitivity to minor input changes. More recent strategies include adversarial training, where models are exposed to adversarial examples during the learning phase, and detection-based methods that monitor feature space behavior or neuron activations for anomalies. Yet, despite these efforts, existing defenses often fail against adaptive or unseen attacks, highlighting the need for more generalized and robust solutions.

This paper offers a structured review of adversarial attacks and their corresponding defense mechanisms, with a focus on classifying threats by their nature, i.e., white-box, black-box, poisoning, inference, and extraction and aligning them with specific countermeasures. By synthesizing these dimensions, the work aims to elucidate the core challenges in securing image classification systems and to identify gaps that warrant further research. The sections that follow explore the technical underpinnings of adversarial vulnerabilities, provide a taxonomy of attack techniques, and evaluate the strengths and limitations of state-of-the-art defenses.

The upcoming sections provide an in-depth analysis of deep learning models and their vulnerabilities. Section 2 delves into different techniques of adversarial attacks, explaining their mechanisms and notable methods. Section 3 presents defense strategies against adversarial attacks. Finally, section 4 concludes the paper by outlining key insights and highlighting subsequent prospects for enhancing the robustness of deep learning models.

2. ADVERSARIAL ATTACKS

Adversarial attacks exploit vulnerabilities in deep learning models by introducing imperceptible perturbations to inputs, causing misclassification. These attacks can be classified into five major categories: White-box attacks, black-box attacks, poisoning attacks, extraction attacks, and inference attacks. In this section, we will describe these attacks.

2.1 White-box attacks

White-box attacks assume the attacker has full knowledge of the model architecture, parameters, and gradients, allowing for precise crafting of adversarial examples. White-box attacks, due to their reliance on internal model details, are often considered a theoretical concern. However, they are highly relevant in settings where models are deployed on devices accessible to attackers. For instance, in mobile applications using on-device image recognition (e.g., face authentication apps or AR filters), attackers may reverse-engineer model parameters and craft adversarial inputs to bypass security features.

2.1.1 Fast Gradient Sign Method

Fast Gradient Sign Method (FGSM) [1] is the simplest adversarial attack, generating perturbations by using L_∞ metric. Given an input $x1$ with true label $y1$, the adversarial example $\hat{x}1$ is computed as:

$$\hat{x}1 = x1 - \epsilon \cdot \text{sign}(\nabla_{x1} J(\theta, x1, y1)) \quad (1)$$

where, $\nabla_{x1} J(.)$ is the gradient of a cost function of $x1$. The size of the perturbations is defined by the error, ϵ. FGSM takes each pixel of the input image, $x1$, and adds ϵ to obtain the output $\widehat{x1}$. This attack perturbs inputs in the direction of the gradient, misleading the model while keeping perturbations minimal.

2.1.2 Carlini & Wagner attacks

Carlini and Wagner [2] proposed a family of adversarial attacks, commonly referred to as C&W attacks, which leverage continuous optimization to generate minimal perturbations capable of fooling a classifier. Unlike gradient-based methods such as FGSM or PGD, C&W attacks formulate adversarial example generation as a restrained optimization issue. The aim of the attack is to identify the smallest possible perturbation ρ such that the modified input $x1 + \rho$ is misclassified by the target model while ensuring minimal distortion. Mathematically, this is expressed as:

$$\min \|\rho\|_p + c \cdot f(x1 + \rho)\rho \quad (2)$$

where, $\|\rho\|_p$ represents the norm of the perturbation, commonly measured using L_0, L_2, or L_∞ norms. c is a constant that balances perturbation magnitude and misclassification. $f(x1 + \rho)$ is an objective function that ensures misclassification, typically defined such that $f(x1 + \rho) \leq 0$ when $x1 + \rho$ is classified as the target adversarial label. They illustrated that this attack is distinctly effective against defended neural networks, causing imperceptible perturbations while achieving a good rate of success. The attack is principally appreciated for its adaptability to different norm constraints and its ability to bypass defenses such as defensive distillation.

2.2 Black-box attacks

In this attack assumption is that the attacker has no exposure to model parameters or gradients, relying on query-based methods or transferability of adversarial examples. Black-box attacks pose a significant practical threat since they require no access to internal model architecture. This makes them especially dangerous in commercial APIs for image recognition (e.g., Google Vision API). Attackers have demonstrated the ability to query such systems and generate adversarial examples using limited feedback, such as confidence scores or predicted labels. In some cases, researchers successfully fooled a commercial facial recognition system into misidentifying individuals with minimal queries, exposing risks in surveillance and identity verification systems.

2.2.1 Boundary attack

Boundary attack is a powerful attack designed to spawn adversarial examples with minimal perturbations [3]. In this input image is adjusted iteratively to gradually move neighboring to the model's decision boundary, while ensuring that the perturbation remains small and imperceptible. This attack works on decision boundary queries, making it suitable for black-box settings rather than requiring exposure to the model's gradients or its internal parameters. The attack starts by initializing a point far from the decision boundary and progressively reduces the perturbation by exploring points on the decision boundary. This method works efficiently due to fewer queries compared to other black-box attack techniques. These iterations are repeated until they cause the model to misclassify.

Mathematically, the optimization problem can be worked out as:

$$\min \|\delta\|_p \text{ s.t. } f(x1+\delta) \neq f(x1),\ \delta \quad (3)$$

where, δ represents the perturbation applied to the input $x1$. The model's output after applying the perturbation is $f(x1 + \delta)$. The constraint $f(x1 + \delta) \neq f(x1)$ ensures that the perturbed input is classified differently from the original input.

This iterative process allows the Boundary Attack to generate adversarial examples that are difficult to detect, as the perturbations remain small while still achieving high adversarial success rates.

2.2.2 One-pixel attack

One-pixel attack is a minimalist adversarial attack that produces adversarial examples by modifying only a single pixel of the input image [4]. Despite the small perturbation, this attack is efficient and can cause current existing deep learning models to misclassify the image. This attack perturbs the input images' single pixel, such that the rate of misclassification increases. It can be used for testing the strength of models to minimal changes since it can exploit vulnerabilities in the model with a minute change in the input. The attack is typically executed by choosing the pixel to perturb and testing various color values using a random search or optimization method to find the perturbation until a successful misclassification is achieved.

Table 1. Summary table of white-box attacks

Method Used	Advantages	Disadvantages	Suggested Solution
Fast Gradient Sign Method (FGSM) [1]	Fast and computationally efficient	Produces easily detectable perturbations	Use iterative methods (I-FGSM, PGD) for stronger attacks
Carlini & Wagner Attacks [2]	More robust than FGSM, iterative refinement	Computationally expensive	Reduce iterations while maintaining effectiveness
L-BFGS [5]	Produces small perturbations	Computationally expensive, Impractical for real-time attacks	Use more efficient gradient-based methods like PGD
DeepFool Attack [3]	Minimal required perturbations, effective for white box models	Assumes linearity, less effective on highly nonlinear models	Apply methods considering non-linearity
Iterative Fast Gradient Sign Method (I-FGSM) [6]	Improves FGSM with multiple iterations	Slower than FGSM	Adjust iteration count for speed efficiency trade-off
Universal Adversarial Perturbations (UAPs) [7]	Effective across multiple images	Less effective on robust models	Adapt perturbations to a specific model
Jacobian-Based Saliency Map Attack (JSMA) [8]	Targeted perturbations on specific pixels	Computationally expensive, requires detailed model information	Use more efficient computation methods
NewtonFool Attack [9]	Minimal perturbations, effective for nonlinear models	Less effective for linear models	Combine with other methods for better efficiency
Elastic Net Attack (EAD) [10]	Sparse perturbations, effective against L1	Computationally expensive	Optimize implementation to reduce computation time
Targeted Universal Adversarial Perturbations (T-UAPs) [11]	Targeted universal perturbations are effective across multiple images	Less effective on robust models	Adapt perturbations to specific models
Brendel & Bethge Attack [12]	Minimal perturbations, effective for robust models	Computationally expensive	Optimize implementation to reduce computation time
Wasserstein Attack [13]	Realistic perturbations, consider data distribution	Computationally expensive	Use approximations to reduce computation time
Shadow Attack [14]	Effective against distillation-based defences	Computationally expensive	Optimize implementation to reduce computation time

Table 2. Summary table of black-box attacks

Methods Used	Advantages	Disadvantages	Suggested Solution
Boundary Attack [15]	Does not use gradients, effective for black box attacks	May require a large number of queries	Optimize efficiency by reducing the number of required queries
One Pixel Attack [16]	Modifies only one pixel, simple and effective	Limited to specific images, less effective on robust models	Increase the number of modified pixels or combine with other attacks
Zeroth Order Optimization (ZOO) Attack [17]	Does not require surrogate models, effective in black box settings	Computationally expensive, requires numerous model evaluations	Use dimensionality reduction and sampling techniques to improve efficiency
Spatial Transformation Attack [11]	Applies spatial transformations, which are difficult to detect	May be less effective on certain models	Combine with other types of attacks to improve effectiveness
Upset & Angry Attack [18]	Targets binary neural networks, effective for this model type	Limited to binary models, less relevant for standard networks	Adapt techniques for non-binary networks
Houdini Attack [19]	Designed for speech and image recognition tasks, effective in these areas	May not generalize to other tasks	Adapt the attack for other types of data and models
Simple Black Box Adversarial Attack [20]	Simple to implement, does not require knowledge of the model	Less effective than other sophisticated attacks	Improve efficiency by combining with other methods
Few Pixel & Threshold Attacks [21]	Modifies a small number of pixels, effective for certain images	Less effective on robust models or complex images	Increase the number of modified pixels or use optimization techniques
HopSkipJump Attack [22]	Effective in black box settings, requires few queries	May be less effective on highly robust models	Combine with other attacks to improve effectiveness
ColorFool Attack [23]	Modifies image colors, difficult to detect	May be less effective on models insensitive to colors	Combine with other types of perturbations
Square Attack [23]	Effective in black box settings, requires few queries	May be less effective on highly robust models	Optimize attack parameters to improve efficiency

Although the perturbation is one pixel, the attack can be very effective against certain image classification models that are not robust to minute changes. This attack exemplifies the vulnerabilities of deep learning frameworks and affects its robustness in the presence of small, imperceptible perturbations. Tables 1 and 2 summarize white box and black box attacks, streamlining their disadvantages and solutions for the same.

2.3 Poisoning attacks

Poisoning attacks involve the deliberate manipulation of a model's training data to degrade its performance or embed malicious behavior. By injecting crafted examples during the learning phase, attackers can influence how the model generalizes to unseen data. These attacks can significantly reduce a model's accuracy, increase false positives or negatives, and, in more targeted cases, force the model to behave incorrectly when presented with specific inputs.

Poisoning attacks are broadly categorized into two types:

- **Clean-Label Poisoning:** Injected samples appear benign and are labeled correctly but are crafted to shift decision boundaries subtly.
- **Backdoor Attacks:** A specific trigger (such as a pattern or pixel patch) is embedded into training images to cause targeted misclassification when the same trigger appears at test time.

Table 3 gives a summary of various poisoning attacks. The consequences of poisoning attacks can be severe and often go undetected due to their subtlety. For example:

- **Accuracy Degradation:** Even a small percentage of poisoned data (as little as 0.1–1%) can reduce a model's top-1 accuracy by a significant margin.
- **Targeted Misclassification:** Models can be trained to classify any image containing a backdoor trigger as a specific class, despite otherwise correct generalization.
- **Shifted Decision Boundaries:** Clean-label attacks can cause a classifier to misclassify inputs near a particular class without noticeable changes in test performance.

In safety-critical systems, such attacks may lead to catastrophic consequences, for instance, misclassifying X-ray scans in medical diagnostics or misidentifying individuals in facial recognition. Some real-world examples include:

- **Autonomous Vehicles:** Backdoor triggers subtly embedded in traffic signs during data collection can cause misclassification (e.g., treating stop signs as yield signs).
- **Facial Recognition Systems:** Clean-label poisoning can train a system to misidentify certain faces when presented with glasses or hats that were used as triggers during training.
- **Content Moderation:** Poisoned training data in hate speech detection systems can shift boundaries, leading to either over-flagging benign posts or under-detecting harmful content.

Table 3. Summary table of poisoning attacks

Methods Used	Advantages	Disadvantages	Suggested Solution
Adversarial Backdoor Embedding [24]	Allows backdoor injection in machine learning models, hard to detect	Can be detected by advanced defense methods	Develop more sophisticated attack techniques to bypass defences
Bullseye Polytope Attack [25]	Clean label poisoning attack improves success rate and transferability	May require a large number of poisoned samples	Optimize sample generation to reduce the required quantity
Poisoning Attack on SVM [26]	Affects the performance of support vector machines by altering training data	Can be detected by anomaly detection techniques	Use more subtle poisoning techniques to avoid detection
Input Model Co-Optimization Attack [4]	Simultaneously optimizes inputs and models to enhance attack efficiency	Can be computationally expensive	Develop more efficient optimization algorithms
Convex Polytope Attack [4]	Generates poisoned samples by forming a convex polytope around the target in the feature space	May require a large number of poisoned samples	Reduce the number of required samples by improving the attack strategy

Table 4. Summary table of inference attacks

Methods used	Advantages	Disadvantages	Suggested Solution for Disadvantages
Model Inversion Attack [27]	Reconstruct sensitive training data by exploiting model access, raising privacy concerns	May require full model access and significant computational resources	Develop more effective attack techniques and explore black box approaches
Reconstruction Attack [27]	Aims to reconstruct input data from model outputs, highlighting potential data leakage	May be limited by model complexity and the quality of available data	Improve reconstruction algorithms and use deep learning techniques for better accuracy

Table 5. Summary table of extraction attacks

Methods Used	Advantages	Disadvantages	Suggested Solution
Copycat Networks [4]	Can reproduce the behaviour of a target model using unlabelled data, facilitating model extraction	May require a large number of queries to the target model, which can be detected	Optimize the number of queries and use sampling techniques to reduce detection
Functionally Equivalent Extraction [28]	Extracts a functionally equivalent model using extraction attacks, ensuring high fidelity	Limited to neural networks with specific architectures, such as two dense layers with ReLU activation	Extend the method to more complex neural network architectures

2.4 Extraction and inference attacks

In extraction attack's goal is to copy model parameters or replicate the functionality of a trained model. Inference attacks focus on extracting sensitive information from a trained model, such as details about the data it was trained on. Tables 4 and 5 summarize extraction and inference attacks.

3. DEFENSE MECHANISMS

Adversarial defense techniques aim to enhance the robustness of deep learning models against carefully crafted perturbations. These strategies can be grouped into four major categories: architectural modifications, training-based approaches, auxiliary component additions, and poisoning-specific defenses. Each category offers trade-offs between robustness, computational overhead, and generalization.

3.1 Modifications to the ANN

These methods involve structural changes to the neural network to enhance robustness against adversarial perturbations.

Defensive Distillation [6]: It is a technique designed to train neural networks that are less sensitive to adversarial inputs. Instead of training a model with hard labels, it utilizes soft probabilities obtained from a first model trained with a higher temperature in the softmax function:

$$\sigma_i(z) = \frac{e^{z_i/T}}{\sum_{j=0}^{n-1} e^{z_i/T}} \quad (4)$$

where, z_i represents the logits (pre-softmax activations), T is the temperature parameter (typically set $T > 1$ for distillation), $\sigma_i(z)$ is the probability assigned to class i. The first model produces these smoothed probabilities, which are then used as targets to train a second model at the same temperature T. Once trained, the distilled model is deployed with a temperature $T = 1$. This approach reduces the model's sensitivity to small perturbations, making adversarial examples less effective. It is effective against simple gradient-based attacks like FGSM; low additional cost once distilled. It is vulnerable to stronger or adaptive attacks such as Carlini & Wagner (C&W). Also, it may slightly reduce model accuracy on clean data. It is best suited for systems with low resource constraints and relatively static attack surfaces, such as embedded systems or controlled APIs.

3.2 Modifications to the training

This category includes training techniques that improve the model's resilience against adversarial perturbations.

- **Brute-Force Adversarial Training:** It aims to intensify the strength of the model by exposing it to adversarial examples during training. The process involves the following steps:
- **Generation of Adversarial Examples:** Adversarial examples are generated by applying minute perturbations to the original inputs, designed to mislead the model while remaining similar to the original data.
- **Incorporation into Training:** These adversarial examples are then incorporated in the training dataset alongside the original examples.
- **Training Objective:** The model is trained not only to classify original examples correctly but also to recognize and correctly classify the adversarial examples.
- **Reinforcing Decision Boundaries:** By being exposed to these perturbations, the model adjusts its decision boundaries to be more resilient to future attacks.
- **Continuous Improvement:** As the model is exposed to new adversarial examples, it becomes increasingly resistant to future manipulations, reducing the likelihood of misclassification under attack.

This approach strengthens the model's security by ensuring it can handle manipulated data without sacrificing its performance on normal data. It is considered one of the most effective strategies for improving robustness across various attacks. It significantly increases training time and computational cost; often overfits to specific attack types (e.g., FGSM vs. PGD), making it less effective against unseen threats. It is ideal for critical applications like medical diagnostics and autonomous systems, where robustness outweighs cost and training time.

3.3 Additions to the ANN

This category involves adding auxiliary components to enhance adversarial defense without modifying the model itself.

- **Feature Squeezing:** It reduces input complexity to mitigate adversarial perturbations. This approach includes reducing the color depth of images, e.g., converting RGB images to fewer bits per channel, and applying median filtering to remove noise-induced artifacts. It is easy to implement and introduces minimal computational cost. It may reduce input quality and affect legitimate classification accuracy; ineffective against adaptive adversaries. It is useful as a preprocessing filter in lightweight applications or online content moderation systems.
- **Trap Doored Model:** Embedding specific patterns in the model to detect adversarial tampering. It provides a verifiable integrity check during deployment. It has limited research and practical deployment experience and may conflict with privacy regulations. It is mostly used in secure federated learning or IP-sensitive models in commercial settings.
- **Defense against Universal Adversarial Perturbations (UAPs):** It involves detecting adversarial noise that generalizes across multiple inputs.
- **MagNet:** A framework using autoencoders to detect and reform adversarial samples before classification. It is capable of recovering perturbed inputs to their clean form. It is susceptible to white-box attacks that are aware of the autoencoder behaviour, where tuning is required for each dataset. It is suited for non-critical but high-accuracy systems like recommender engines or visual content sorting.

3.4 Defenses against poisoning attacks

These defenses aim to detect and mitigate poisoning attacks that attempt to corrupt training data. Table 6 summarizes findings from empirical studies, highlighting detection performance and practical limitations.

Table 6. Empirical evaluation of poisoning attack defense mechanisms

Defense Method	Dataset Used	Detection Accuracy	Key Observations	Limitations
Spectral Signature Analysis [24]	CIFAR-10, SVHN	> 90%	Detected backdoor samples with high precision using SVD on feature space	Ineffective against clean-label poisoning
Activation Pattern Analysis [29]	GTSRB, Tiny ImageNet	~85%	Identified poisoned neurons via abnormal activation clustering	High resource use; may yield false positives on noisy datasets
Activation Clustering (DBSCAN) [29]	GTSRB	~82-87%	Unsupervised clustering helped separate clean vs. poisoned samples	Less effective if poisoning is highly distributed across layers
SentiNet (Region Attribution) [30]	ImageNet, CIFAR-10	~88%	Uses interpretability tools to detect local backdoor regions	Limited scalability and sensitive to trigger size/position
STRIP (Entropy-Based Detection) [31]	MNIST, CIFAR-10	> 90%	Uses input perturbations and entropy metrics to identify poisoned behavior	Requires access to runtime prediction entropy; ineffective for label flipping

4. CONCLUSION

Adversarial attacks continue to pose serious challenges to the reliability and trustworthiness of deep learning models, particularly in image classification tasks. This review provided a comprehensive synthesis of various attack methodologies ranging from white-box and black-box attacks to more sophisticated poisoning and model extraction techniques, as well as defense mechanisms that attempt to counter these threats. Despite notable progress in developing defensive strategies such as adversarial training, spectral signature analysis, and feature squeezing, no method has proven universally effective across diverse attack types. The dynamic nature of adversarial threats calls for a continuous evolution of both detection and prevention approaches. Looking forward, several promising research directions warrant attention. The development of real-time adversarial input detection systems that are both lightweight and accurate remains a critical need, particularly for deployment in latency-sensitive environments. As clean-label poisoning attacks grow more subtle and complex, advanced anomaly detection techniques and robust self-supervised training paradigms may help mitigate their effects. Additionally, the field must invest in designing defenses that are transferable across modalities and datasets, addressing the challenge of maintaining robustness in dynamic or cross-domain applications. Thus, defending deep learning systems against adversarial attacks remains a dynamic and urgent challenge. A holistic approach that integrates algorithmic robustness, transparent design, and ethical data governance is essential to ensure the safe deployment of AI systems in real-world applications.

REFERENCES

[1] Goodfellow, I.J., Shlens, J., Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572. https://doi.org/10.48550/arXiv.1412.6572

[2] Carlini, N., Wagner, D. (2017). Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, pp. 39-57. https://doi.org/10.1109/SP.2017.49

[3] Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P. (2016). DeepFool: A simple and accurate method to fool deep neural networks. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 2574-2582. https://doi.org/10.1109/CVPR.2016.282

[4] Correia-Silva, J.R., Berriel, R.F., Badue, C., De Souza, A.F., Oliveira-Santos, T. (2018). Copycat CNN: Stealing knowledge by persuading confession with random non-labeled data. In 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, pp. 1-8. https://doi.org/10.1109/IJCNN.2018.8489592

[5] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., et al. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199. https://doi.org/10.48550/arXiv.1312.6199

[6] Papernot, N., McDaniel, P. (2016). On the effectiveness of defensive distillation. arXiv preprint arXiv:1607.05113. https://doi.org/10.48550/arXiv.1607.05113

[7] Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P. (2017). Universal adversarial perturbations. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 86-94. https://doi.org/10.1109/CVPR.2017.17

[8] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., et al. (2016). The limitations of deep learning in adversarial settings. In 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbruecken, Germany, pp. 372-387. https://doi.org/10.1109/EuroSP.2016.36

[9] Cisse, M., Adi, Y., Neverova, N., Keshet, J. (2017). Houdini: Fooling deep structured prediction models. arXiv preprint arXiv:1707.05373. https://doi.org/10.48550/arXiv.1707.05373

[10] Chen, P.Y., Sharma, Y., Zhang, H., Yi, J., Hsieh, C.J. (2018). Ead: Elastic-net attacks to deep neural networks via adversarial examples. In Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11302

[11] Kurakin, A., Goodfellow, I.J., Bengio, S. (2018). Adversarial examples in the physical world. In Artificial Intelligence Safety and Security, pp. 99-112. Chapman and Hall/CRC.

[12] Su, J., Vargas, D.V., Sakurai, K. (2019). One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation, 23(5): 828-841. https://doi.org/10.1109/TEVC.2019.2890858

[13] Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A. (2019). Exploring the landscape of spatial robustness. In Proceedings of the 36th International Conference on Machine Learning, pp. 1802-1811.

https://proceedings.mlr.press/v97/engstrom19a.html?utm_medium=email&utm_source=transaction.
[14] Andriushchenko, M., Croce, F., Flammarion, N., Hein, M. (2020). Square attack: A query-efficient black-box adversarial attack via random search. In Computer Vision – ECCV 2020, pp. 484-501. https://doi.org/10.1007/978-3-030-58592-1_29
[15] Brendel, W., Rauber, J., Bethge, M. (2017). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint arXiv:1712.04248. https://doi.org/10.48550/arXiv.1712.04248
[16] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083. https://doi.org/10.48550/arXiv.1706.06083
[17] Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J. (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15-26. https://doi.org/10.1145/3128572.3140448
[18] Sarkar, S., Bansal, A., Mahbub, U., Chellappa, R. (2017). UPSET and ANGRI: Breaking high performance image classifiers. arXiv preprint arXiv:1707.01159. https://doi.org/10.48550/arXiv.1707.01159
[19] Kotyan, S., Vargas, D.V. (2022). Adversarial robustness assessment: Why in evaluation both L_0 and L_∞ attacks are necessary. PloS One, 17(4): e0265723. https://doi.org/10.1371/journal.pone.0265723
[20] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., et al. (2017). Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506-519. https://doi.org/10.1145/3052973.3053009
[21] Khamaiseh, S.Y., Bagagem, D., Al-Alaj, A., Mancino, M., Alomari, H.W. (2022). Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification. IEEE Access, 10: 102266-102291. https://doi.org/10.1109/ACCESS.2022.3208131
[22] Chen, J., Jordan, M.I., Wainwright, M.J. (2020). HopSkipJumpAttack: A query-efficient decision-based attack. In 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, pp. 1277-1294. https://doi.org/10.1109/SP40000.2020.00045
[23] Sewak, M., Sahay, S.K., Rathore, H. (2020). An overview of deep learning architecture of deep neural networks and autoencoders. Journal of Computational and Theoretical Nanoscience, 17(1): 182-188. https://doi.org/10.1166/jctn.2020.8648
[24] Tran, B., Li, J., Madry, A. (2018). Spectral signatures in backdoor attacks. Advances in Neural Information Processing Systems, 31.
[25] Aghakhani, H., Meng, D., Wang, Y.X., Kruegel, C., Vigna, G. (2021). Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, pp. 159-178. https://doi.org/10.1109/EuroSP51992.2021.00021
[26] Biggio, B., Nelson, B., Laskov, P. (2012). Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389. https://doi.org/10.48550/arXiv.1206.6389
[27] Chopra, B., Singh, S., Gupta, S. (2024). Adversarial attacks and defense strategies on image classification. International Research Journal of Modernization in Engineering Technology and Science, 6(5): 427-436. https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2024/55114/final/fin_irjmets1714996291.pdf.
[28] Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., Papernot, N. (2020). High accuracy and high fidelity extraction of neural networks. In 29th USENIX Security Symposium (USENIX Security 20), pp. 1345-1362. https://www.usenix.org/conference/usenixsecurity20/presentation/jagielski.
[29] Chen, X., Liu, C., Li, B., Lu, K., Song, D. (2017). Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526. https://doi.org/10.48550/arXiv.1712.05526
[30] Chou, E., Tramer, F., Pellegrino, G. (2020). SentiNet: Detecting localized universal attacks against deep learning systems. In 2020 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, pp. 48-54. https://doi.org/10.1109/SPW50608.2020.00025
[31] Gao, Y., Xu, C., Wang, D., Chen, S., et al. (2019). STRIP: A defence against trojan attacks on deep neural networks. In Proceedings of the 35th Annual Computer Security Applications Conference, pp. 113-125. https://doi.org/10.1145/3359789.3359790