MANTECH Publications
Journal of Artificial Intelligence, Machine Learning and Soft Computing
Volume 1, Issue 2, July-December 2016

Robust & Adversarial Machine Learning

Shivam Jain[1], Ashok Pandey[2], Sukesh Baitha[3]
Assistant Professor, Associate Professor
Department of AI & Soft Computing for Energy Systems
St. Aloysius College, Mangalore,India
Email ID: *Shivam62jain@rediffmail.com[1], ashokpandey49@gmail.com[2], baithasss@yahoo.com[3]*

Abstract

Machine learning (ML) has revolutionized various sectors including healthcare, finance, autonomous systems, and cybersecurity. However, real-world deployment exposes models to uncertainties, noisy inputs, and malicious adversarial attacks. ***Robust machine learning*** *focuses on enhancing model stability against noise and perturbations, while* ***adversarial machine learning*** *studies deliberate attacks that exploit model vulnerabilities. This paper provides a comprehensive review of robust and adversarial ML, covering recent advancements, threat models, attack and defense mechanisms, evaluation metrics, and practical applications. Additionally, we explore the challenges of achieving both robustness and accuracy and highlight future research directions. A detailed discussion of defense strategies, including adversarial training, certified robustness, and robust optimization, is presented, along with comparisons of current methodologies.*

Keywords: *Robust Machine Learning, Adversarial Attacks, Adversarial Training, Certified Robustness, Deep Learning Security, Threat Models, Perturbation Analysis*

INTRODUCTION

Machine learning algorithms, particularly deep learning, have achieved remarkable success in tasks like image recognition, natural language processing, and decision-making systems. However, real-world environments are often noisy, incomplete, or maliciously manipulated. Standard ML models, while highly accurate on clean data, often fail in these scenarios. This

raises critical concerns in safety-sensitive domains such as autonomous driving, medical diagnosis, and financial prediction.

Robust machine learning aims to improve model stability and performance in the presence of noise, outliers, or unforeseen distribution shifts. **Adversarial machine learning (AML)** studies methods by which malicious actors intentionally manipulate inputs to deceive ML models. Understanding both robustness and adversarial aspects is crucial for deploying reliable AI systems.

This paper explores:

1. Types of adversarial attacks and threat models.
2. Techniques to enhance model robustness.
3. Evaluation metrics and benchmark datasets.
4. Challenges, trade-offs, and future directions.

2. BACKGROUND

Machine learning models have shown impressive capabilities in diverse tasks. However, their performance can degrade when exposed to unexpected noise, data shifts, or malicious manipulations. Understanding the background of robustness and adversarial attacks is essential to developing reliable ML systems.

2.1 Robust Machine Learning

Robustness in machine learning refers to a model's ability to maintain high predictive performance when inputs deviate from the conditions seen during training. These deviations may arise due to random noise, environmental variations, or shifts in data distribution over time. Robust ML ensures that models are not only accurate on clean, ideal datasets but also resilient to real-world uncertainties.

Types of Perturbations:

1. **Random Noise:**

Random noise refers to unpredictable variations in input data that are not adversarial but can affect performance. Examples include Gaussian noise added to image pixels, or sensor measurement errors in robotics. For instance, a camera sensor in autonomous vehicles may

introduce random pixel-level noise due to low light conditions.

2. **Environmental Changes:**

Real-world conditions such as lighting, weather, or sensor drift can significantly alter input data characteristics. Autonomous vehicles trained on sunny day images may misclassify objects in foggy or rainy conditions unless trained for robustness.

3. **Distribution Shifts:**

Training and deployment data often follow different distributions. This can occur due to demographic changes, evolving user behavior, or differences in hardware sensors. Techniques like domain adaptation and transfer learning aim to mitigate these challenges.

Formal Definition:

Let fθf_\thetafθ denote a model with parameters θ\thetaθ, and (x,y)∼D(x, y) \sim \mathcal{D}(x,y)∼D represent data samples drawn from distribution D\mathcal{D}D. Let δ\deltaδ be a perturbation representing noise, environmental changes, or adversarial influence. Robust ML seeks to minimize the expected loss over perturbed inputs:

minθE(x,y)∼D[ℓ(fθ(x+δ),y)]\min_\theta \mathbb{E}_{(x,y) \sim \mathcal{D}} \big[\ell(f_\theta(x+\delta), y)\big]θminE(x,y)∼D[ℓ(fθ(x+δ),y)]

Here:

- ℓ\ellℓ is the loss function, such as cross-entropy for classification tasks.
- δ\deltaδ can be constrained, e.g., ∥δ∥∞≤ϵ\|\delta\|_\infty \leq \epsilon∥δ∥∞≤ϵ, to ensure perturbations remain realistic.

Examples of Robustness Techniques:

- **Data Augmentation:** Adding noisy or transformed versions of inputs during training.
- **Regularization:** Techniques like weight decay or dropout reduce overfitting and increase robustness.
- **Robust Optimization:** Explicitly optimizing models to reduce sensitivity to perturbations using min-max formulations.

Robust ML is particularly critical in **safety-critical systems** like medical diagnosis,

autonomous navigation, and financial decision-making, where small perturbations can lead to catastrophic failures.

2.2 Adversarial Machine Learning

While robustness addresses unintentional perturbations, **adversarial machine learning (AML)** focuses on deliberate manipulations designed to mislead models. Adversarial attacks exploit vulnerabilities in model decision boundaries, causing models to make incorrect predictions even when perturbations are imperceptible to humans.

Categories of Adversarial Attacks:

1. **Evasion Attacks:**
   - Occur at **test time**.
   - Attackers slightly modify input features to induce misclassification.
   - Example: Slightly altering pixels in an image to make a stop sign appear as a yield sign to an autonomous vehicle model.

2. **Poisoning Attacks:**
   - Occur at **training time**.
   - Malicious data is injected into the training set, causing the model to learn incorrect patterns.
   - Example: Adding mislabeled financial transactions to a fraud detection dataset to reduce detection accuracy.

3. **Model Extraction & Inference Attacks:**
   - Aim to **steal model parameters** or **extract sensitive training data**.
   - Example: Querying a machine learning API repeatedly to reconstruct its model weights or infer private user data.

2.2.1 Threat Models

Adversarial attack strategies are defined based on the attacker's knowledge and access to the model:

- **White-box Attacks:**

The adversary has complete knowledge of the model architecture, parameters, and training data. This allows for highly effective gradient-based attacks.

- **Black-box Attacks:**

The adversary can only observe inputs and outputs of the model, without access to internal parameters. Black-box attacks often rely on transferability of adversarial examples or surrogate models.

- **Gray-box Attacks:**

The adversary has partial knowledge, such as knowing the model architecture but not weights, or having access to a subset of the training data.

2.2.2 Common Adversarial Attack Methods

Attack	Type	Description
FGSM (Fast Gradient Sign Method)	Evasion	Perturbs input in the direction of gradient to maximize loss
PGD (Projected Gradient Descent)	Evasion	Iterative version of FGSM for stronger attacks
DeepFool	Evasion	Minimal perturbation to cross decision boundary
CW (Carlini & Wagner)	Evasion	Optimized to minimize perturbation while misclassifying
Data Poisoning	Training	Malicious data alters model during training

3. ROBUSTNESS TECHNIQUES

Robustness techniques aim to improve machine learning model resilience against perturbations, both unintentional and adversarial. These methods can be broadly categorized into **training-time defenses**, **certified defenses**, and **architecture-based strategies**.

3.1 Adversarial Training

Adversarial training is one of the most widely used and effective methods to defend against adversarial attacks. It involves **augmenting the training data** with adversarial examples so that the model learns to correctly classify both clean and perturbed inputs.

Formally, adversarial training is formulated as a **min-max optimization problem**:

min𝑓0θE(x,y)∼D[max𝑓0δ∈Sℓ(fθ(x+δ),y)]\min_\theta \mathbb{E}_{(x,y) \sim \mathcal{D}} \Big[\max_{\delta \in S} \ell(f_\theta(x+\delta), y) \Big]θminE(x,y)∼D[δ∈Smaxℓ(fθ(x+δ),y)]

Where:

- ℓ\ellℓ is the loss function (e.g., cross-entropy for classification).
- δ\deltaδ is the adversarial perturbation constrained by SSS, often defined as ‖δ‖∞≤ϵ\|\delta\|_\infty \leq \epsilon‖δ‖∞≤ϵ.
- θ\thetaθ represents model parameters.

Key Points:

- The inner maximization simulates the **strongest adversarial attack** within allowable bounds.
- The outer minimization trains the model to **minimize loss on these adversarial examples**.

Pros:

- Provides strong robustness against known attacks.
- Directly trains the model to handle perturbations.

Cons:

- Computationally expensive, especially for iterative attacks like PGD.
- May slightly reduce accuracy on clean (non-adversarial) data.
- Robustness is often attack-specific; unseen attacks may still succeed.

Example: Training a CIFAR-10 image classifier with PGD adversarial training improves adversarial accuracy from ~0% to ~45–50%, though clean accuracy may drop slightly.

3.2 Defensive Distillation

Defensive distillation is inspired by model compression techniques. It involves training a model on the **softened probabilities** (soft labels) produced by a previously trained model instead of hard labels.

Procedure:

1. Train a teacher model fTf_TfT on standard data.
2. Extract the softmax outputs pT(x)p_T(x)pT(x) with temperature T>1T > 1T>1.
3. Train a student model fSf_SfS to match these probabilities.

minθ∑(x,y)∈DKL(fS(x),pT(x))\min_\theta \sum_{(x,y) \in \mathcal{D}} \text{KL}\big(f_S(x), p_T(x)\big)θmin(x,y)∈D∑KL(fS(x),pT(x))

Key Idea: Soft labels reduce sensitivity to input perturbations because the model learns smoother decision boundaries.

Limitations:

- Effective against simple attacks like FGSM, but **fails against strong iterative attacks** (e.g., PGD, CW).
- Mostly considered a **baseline defense** rather than a practical solution today.

3.3 Certified Robustness

Certified robustness methods aim to provide **provable guarantees** that a model's prediction will not change under bounded perturbations. This is essential in safety-critical domains like autonomous driving and medical diagnosis.

Techniques:

1. **Interval Bound Propagation (IBP):**
   - Propagates input intervals through the network to estimate output bounds.
   - Guarantees correctness if all possible perturbations within a bound produce the same output.
2. **Randomized Smoothing:**
   - Adds Gaussian noise to the input and uses majority voting over multiple noisy samples.
   - Produces a **smoothed classifier** with provable L2L_2L2-norm robustness.

Advantages:

- Provides mathematically guaranteed robustness within specified bounds.

Challenges:

- Often conservative, reducing clean accuracy.
- Computationally expensive for large networks.

3.4 Gradient Masking

Gradient masking hides gradient information to prevent gradient-based attacks like FGSM or PGD. Methods include:

- Non-differentiable activations.
- Obfuscated gradients.

Limitations:

- Considered a **weak defense**, as adaptive attacks can circumvent gradient obfuscation.
- May **give a false sense of security** without actually improving robustness.

3.5 Ensemble Methods

Ensemble methods combine multiple models to improve robustness:

Techniques:

- Voting-based ensembles.
- Mixtures of models with different architectures or training strategies.

Benefits:

- Reduces the success rate of adversarial attacks due to diversity.
- Can improve both clean and adversarial accuracy.

Example: Combining ResNet and DenseNet models trained with adversarial training can reduce PGD attack success from 50% to 30%.

4. EVALUATION METRICS

Evaluating robust and adversarial machine learning models requires **going beyond standard accuracy**, because a model that performs well on clean test data may fail catastrophically under perturbations. Metrics must quantify **how models behave under adversarial or noisy conditions** and help compare defense strategies effectively.

4.1 Robust Accuracy

Robust accuracy measures a model's ability to correctly classify **adversarially perturbed inputs**. It is a fundamental metric in adversarial ML research.

Formally:

Robust Accuracy=# correctly classified perturbed samplesTotal perturbed samples\text{Robu st Accuracy} = \frac{\text{\# correctly classified perturbed samples}}{\text{Total perturbed samples}}Robust Accuracy=Total perturbed samples# correctly classified perturbed samples

Key Points:

- Only perturbed (adversarial or noisy) inputs are considered.
- High clean accuracy does not guarantee high robust accuracy; a model may have 95% clean accuracy but 0% robust accuracy if vulnerable to attacks.

Example:

Suppose we evaluate a CIFAR-10 model against PGD attacks with $\epsilon = 8/255$\epsilon = 8/255$\epsilon=8/255$:

- 1000 perturbed images tested.
- 470 correctly classified.
- Robust Accuracy = 470 / 1000 = **47%**.

Interpretation: This shows that the model correctly predicts less than half of the adversarially perturbed examples.

4.2 Certified Accuracy

Certified accuracy is a stricter metric used with **provably robust models**. It represents the fraction of inputs for which a model **guarantees correctness** within a bounded perturbation.

Certified Accuracy=# inputs guaranteed correct under perturbationTotal inputs\text{Certified Accuracy} = \frac{\text{\# inputs guaranteed correct under perturbation}}{\text{Total inputs}}Certified Accuracy=Total inputs# inputs guaranteed correct under perturbation

Key Points:

- Applies to methods like **Randomized Smoothing** or **Interval Bound Propagation**.
- Provides **mathematical guarantees** rather than empirical estimates.
- Especially important for **safety-critical systems**, e.g., autonomous vehicles, medical diagnosis, or aerospace applications.

Example:

A model may achieve 88% clean accuracy, but randomized smoothing may guarantee correct predictions for 38% of inputs under an L2L_2L2 perturbation of 0.5.

4.3 Attack Success Rate

Attack success rate quantifies the **effectiveness of adversarial attacks** against a model. It is complementary to robust accuracy:

Attack Success Rate=1−Robust Accuracy\text{Attack Success Rate} = 1 - \text{Robust Accuracy}Attack Success Rate=1−Robust Accuracy

Key Points:

- High attack success rate indicates poor defense against attacks.

- Used to compare models under the **same attack type and perturbation budget**.

Example:

Continuing the CIFAR-10 example:

- Robust accuracy = 47%
- Attack success rate = 1 − 0.47 = **53%**

This means that the PGD attack successfully fooled the model in 53% of cases.

4.4 Perturbation Norms

Perturbation norms quantify the **magnitude of adversarial changes** applied to inputs. They help evaluate whether attacks are realistic or imperceptible. Common norms include:

1. **L0L_0L0 Norm (Sparsity):**
   - Counts the number of features modified.
   - Useful for evaluating attacks that change only a small subset of pixels or words.
   - Example: Changing 5 pixels in a 28×28 MNIST image.
2. **L2L_2L2 Norm (Euclidean Distance):**
   - Measures the overall magnitude of changes:
     ‖δ‖2=∑iδi2\|\delta\|_2 = \sqrt{\sum_i \delta_i^2}‖δ‖2=i∑δi2
   - Example: A small but distributed perturbation across an image may have L2=1.2L_2 = 1.2L2=1.2.
3. **L∞L_\inftyL∞ Norm (Maximum Change):**
   - Measures the maximum absolute change applied to any feature:
     ‖δ‖∞=maxi|δi|\|\delta\|_\infty = \max_i |\delta_i|‖δ‖∞=imax|δi|
   - Often used in pixel-constrained attacks where each pixel is only allowed to change by a small value ϵ\epsilonϵ.
   - Example: ϵ=8/255\epsilon = 8/255ϵ=8/255 in CIFAR-10 attacks.

Why Norms Matter:

- Norms define **attack constraints**.
- Smaller perturbations are harder to detect and represent realistic adversarial scenarios.
- Evaluation using norms allows **fair comparison** across different defense strategies and attack types.

Table 2: compares metrics across datasets:

Dataset	Clean Accuracy	Robust Accuracy (PGD)	Certified Accuracy
MNIST	99.1%	94.2%	92.5%
CIFAR-10	93.5%	72.3%	70.1%
ImageNet	76.8%	45.5%	42.3%

5. RECENT ADVANCES

5.1 Robust Optimization Techniques

Recent works integrate robust optimization into training objectives:

- Min-max optimization frameworks
- Regularization methods to penalize sensitivity to perturbations
- Domain adaptation for robustness under distribution shift

5.2 Generative Adversarial Defenses

Generative models, including GANs, can be used to denoise inputs or project perturbed examples back to the data manifold.

5.3 Multi-Objective Robustness

Balancing clean accuracy with adversarial robustness remains a key research focus. Techniques like TRADES (TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization) explicitly model this trade-off.

APPLICATIONS

1. **Autonomous Vehicles:** Enhancing robustness to sensor noise, lighting changes, and malicious inputs.
2. **Medical Imaging:** Ensuring diagnostic AI is robust against scanner noise and adversarial attacks.
3. **Finance:** Protecting fraud detection models from adversarial manipulations.
4. **Cybersecurity:** Defending intrusion detection systems and malware classifiers against evasion attacks.

CHALLENGES AND OPEN RESEARCH DIRECTIONS

1. **Robustness vs Accuracy Trade-off:** Enhancing robustness often reduces clean data accuracy.

2. **Scalability:** Certified robustness techniques struggle on large datasets.
3. **Adaptive Attacks:** New attacks continuously emerge, requiring iterative defense development.
4. **Cross-Domain Robustness:** Generalizing robustness across modalities and datasets is challenging.
5. **Evaluation Standardization:** Lack of standardized benchmarks makes comparison difficult.

CONCLUSION

Robust and adversarial machine learning represents a critical research area bridging security and reliability in AI systems. While adversarial attacks reveal vulnerabilities, robust ML and defensive strategies can mitigate risks. This paper provided a comprehensive review of current methodologies, evaluation techniques, challenges, and applications. Future research must focus on scalable, provably robust models capable of maintaining performance across real-world adversarial scenarios while balancing clean accuracy. Achieving this will be essential for deploying AI in high-stakes domains safely and reliably.

REFERENCES

1. Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *ICLR*.
2. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards Deep Learning Models Resistant to Adversarial Attacks. *ICLR*.
3. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z., & Swami, A. (2016). Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples. *ACM CCS*.
4. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2018). Ensemble Adversarial Training: Attacks and Defenses. *ICLR*.
5. Cohen, J., Rosenfeld, E., & Kolter, J. Z. (2019). Certified Adversarial Robustness via Randomized Smoothing. *ICML*.
6. Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L., & Jordan, M. (2019). Theoretically Principled Trade-off between Robustness and Accuracy. *ICML*.
7. Kurakin, A., Goodfellow, I., & Bengio, S. (2017). Adversarial Machine Learning at Scale. *ICLR*.

8. Carlini, N., & Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. *IEEE S&P*.
9. Liu, Y., Chen, X., Liu, C., & Song, D. (2017). Delving into Transferable Adversarial Examples and Black-box Attacks. *ICLR*.
10. Hendrycks, D., & Dietterich, T. (2019). Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *ICLR*.