Computer Standards & Interfaces 97 (2026) 104117

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

ARMOR: A multi-layered adaptive defense framework for robust deep learning systems against evolving adversarial threats☆

Mahmoud Mohamed *, Fayaz AlJuaid

Electrical and Computer Engineering , King Abdul Aziz University, Saudi Arabia

ARTICLE INFO

Keywords:
Adversarial machine learning
Deep learning security
Multi-layered defense
Robustness evaluation
Adaptive security

ABSTRACT

Introduction: Adversarial attacks represent a major challenge to deep learning models deployed in critical fields such as healthcare diagnostics and financial fraud detection. This paper addresses the limitations of single-strategy defenses by introducing ARMOR (Adaptive Resilient Multi-layer Orchestrated Response), a novel multi-layered architecture that seamlessly integrates multiple defense mechanisms.

Methodology: We evaluate ARMOR against seven state-of-the-art defense methods through extensive experiments across multiple datasets and five attack methodologies. Our approach combines adversarial detection, input transformation, model hardening, and adaptive response layers that operate with intentional dependencies and feedback mechanisms.

Results: Quantitative results demonstrate that ARMOR significantly outperforms individual defense methods, achieving a 91.7% attack mitigation rate (18.3% improvement over ensemble averaging), 87.5% clean accuracy preservation (8.9% improvement over adversarial training alone), and 76.4% robustness against adaptive attacks (23.2% increase over the strongest baseline).

Discussion: The modular framework design enables flexibility against emerging threats while requiring only 1.42× computational overhead compared to unprotected models, making it suitable for resource-constrained environments. Our findings demonstrate that activating and integrating complementary defense mechanisms represents a significant advance in adversarial resilience.

1. Introduction

Deep learning technologies have been widely adopted in critical sectors including autonomous vehicles, medical diagnostics, and cybersecurity. While they offer powerful capabilities, they also introduce new security vulnerabilities. Adversarial examples—carefully crafted inputs designed to deceive models—pose significant risks to AI systems [1,2]. Small, seemingly imperceptible distortions can cause state-of-the-art models to misclassify inputs, which may have life-threatening consequences in safety-critical applications [3].

Recent advances in deep learning have highlighted the importance of robust defense mechanisms. For example, UNet-based segmentation models in medical imaging have achieved approximately 96% accuracy in COVID-19 detection from CT scans [4]. Similarly, CNN and BiGRU models have demonstrated strong performance in traffic network analysis with an R-squared of 0.9912 [5]. These successes underscore the critical need for robust defenses, particularly as deep learning models are increasingly integrated into high-stakes decision-making processes.

However, existing defenses are typically based on single strategies such as adversarial training [6], input preprocessing [7], or detection models [8]. While effective against specific attacks, these methods often fail when facing diverse or adaptive attacks [9]. This limitation is increasingly concerning as adversaries continue to evolve their strategies. Furthermore, existing techniques often suffer from high computational costs, degraded performance on clean data, and continued susceptibility to adaptive attacks [10].

Problem Statement: This paper addresses the vulnerability of deep learning systems to adversarial attacks in mission-critical environments. Current defenses exhibit three key weaknesses:

1. They typically optimize for a single threat model, leaving them exposed to diverse attack strategies.
2. They employ static approaches that cannot adapt to evolving threats.
3. They fail to balance performance and security, often sacrificing accuracy on benign data.

☆ This article is part of a Special issue entitled: 'Secure AI' published in Computer Standards & Interfaces.

* Corresponding author.
E-mail address: mhassan0085@stu.kau.edu.sa (M. Mohamed).

https://doi.org/10.1016/j.csi.2025.104117
Received 2 June 2025; Received in revised form 2 December 2025; Accepted 12 December 2025
Available online 17 December 2025

These weaknesses motivate the need for an agile and flexible defense architecture.

Research Gaps: Our comprehensive literature survey, following systematic review methodologies [11], identifies several critical gaps:

- Most defenses optimize for a single threat model, creating vulnerabilities across diverse attack strategies [12].
- Current ensemble approaches typically use simple voting or averaging, failing to leverage the complementary strengths of different defense mechanisms [13].
- There is insufficient focus on dynamic adaptation to evolving threats in real-time operational environments [14].
- The performance-security trade-off is poorly addressed, with many techniques significantly degrading model performance on benign inputs [15].

Our ARMOR framework addresses these gaps through:

- **Orchestrated Integration:** Complementary defense layers operate cooperatively rather than in isolation.
- **Dynamic Threat Assessment:** Adaptive response mechanisms learn from observed attack patterns.
- **Explicit Trade-off Optimization:** High clean accuracy is maintained while improving robustness.
- **Comprehensive Testing:** Evaluation across diverse attacks, including engineered adaptive attacks.
- **Modular Design:** New defense mechanisms can be incorporated as they emerge.

As shown in Table 1, our method advances the state-of-the-art across multiple performance dimensions while maintaining reasonable computational overhead.

2. Related work

This section analyzes current adversarial defense mechanisms, their limitations, and specific gaps our framework addresses. We categorize existing work into adversarial training, input transformation, detection-based methods, certified robustness, and ensemble approaches.

2.1. Adversarial training methods

Adversarial training remains one of the most effective empirical defense mechanisms. Madry et al. [6] introduced PGD adversarial training, which serves as a strong baseline but suffers from reduced clean accuracy and high computational cost.

Recent advances include TRADES [15], which explicitly regularizes the trade-off between standard accuracy and robustness; Fast Adversarial Training [16], which improves computational efficiency using FGSM with randomization; and Robust Self-Training (RST) [17], which leverages additional unlabeled data to enhance robustness.

Despite these improvements, adversarial training techniques remain fundamentally constrained: they are typically resistant only to attacks encountered during training, often fail on out-of-distribution samples, and exhibit reduced performance on clean data [18].

2.2. Input transformation approaches

Input transformation methods aim to remove adversarial perturbations before model inference. Guo et al. [7] explored various image transformations, finding that total variance minimization and image quilting provide moderate robustness. Xie et al. [19] proposed random resizing and padding as preprocessing defenses.

More recent work includes Neural Representation Purifiers [20], which use self-supervised learning to clean adversarial inputs, and ComDefend [21], a compression-decompression architecture that eliminates adversarial perturbations.

While these methods often preserve accuracy better than adversarial training, they remain vulnerable to adaptive attacks that account for the transformation process [10].

2.3. Detection-based defenses

Detection methods aim to identify adversarial examples without necessarily correcting them. Metzen et al. [8] attached a binary detector subnetwork to identify adversarial inputs. Lee et al. [22] used Mahalanobis distance-based confidence scores to detect out-of-distribution samples.

Recent approaches include statistical methods using odds ratio tests [23] and Local Intrinsic Dimensionality (LID) [24] to characterize adversarial regions in feature space.

While detection mechanisms can be accurate, adaptive attacks specifically target their vulnerabilities [25]. Moreover, they do not provide predictions for identified adversarial examples.

2.4. Certified robustness approaches

Certified defenses provide theoretical guarantees that perturbations within certain bounds will not alter predictions. Cohen et al. [26] applied randomized smoothing to create certifiably robust classifiers against L2-norm bounded perturbations. Gowal et al. [27] developed interval bound propagation for training verifiably robust networks.

Recent progress includes DeepPoly [28], which provides tighter bounds for neural network verification, and improved certification bounds for cascading architectures [29].

While certified methods offer valuable theoretical assurances, they generally achieve lower empirical robustness than adversarial training and can be significantly more resource-intensive [30].

2.5. Ensemble and hybrid approaches

Ensemble methods combine multiple models or defense mechanisms to enhance robustness. Tramèr et al. [31] proposed Ensemble Adversarial Training, which augments training data with adversarial examples from other models. Pang et al. [13] introduced adaptive diversity promoting (ADP) training to develop robust ensemble models. Sen et al. [32] integrated detection and adversarial training in a two-stage process.

However, most current ensembles employ basic averaging or voting schemes that fail to leverage the complementary strengths of different defense types [33].

2.6. Research gaps and contributions

Based on our literature review, we identify the following critical research gaps:

- **Poor Integration:** Most studies focus on single defenses or simple combinations that fail to leverage synergistic effects.
- **Static Defense Mechanisms:** Current approaches use fixed strategies that cannot adapt to evolving threats.
- **Performance-Security Trade-offs:** Robust models frequently sacrifice clean-data accuracy.
- **Lack of Standardization:** Inconsistent evaluation protocols hinder fair comparisons.
- **Insufficient Adaptive Attack Testing:** Most defenses are not evaluated against adaptive attacks designed to circumvent them.

Our ARMOR framework addresses these gaps through:

- **Orchestrated Integration:** Complementary defense layers operate cooperatively rather than in isolation.
- **Dynamic Threat Assessment:** Response mechanisms adapt based on observed attack patterns.
- **Explicit Trade-off Optimization:** High clean accuracy is maintained while improving robustness.
- **Comprehensive Testing:** Evaluation across diverse attacks, including engineered adaptive attacks.

Table 1
Comparison of state-of-the-art adversarial defense methods (2020–2025).

Reference	Year	Defense type	Multi-attack robustness	Clean accuracy	Computation overhead	Adaptive attack resistance
Madry et al. [6]	2018	Adversarial training	Medium (66.4%)	Low (87.3%)	High (10×)	Medium (54.2%)
Zhang et al. [15]	2019	Adv. training (TRADES)	Medium (73.5%)	Medium (84.9%)	High (7×)	Medium (61.8%)
Cohen et al. [26]	2019	Certified defense	Low (49.2%)	Medium (83.5%)	Very high (30×)	High (guaranteed bounds)
Wong et al. [16]	2020	Fast Adv. training	Medium (71.2%)	Medium-high (85.8%)	Medium (3×)	Medium (58.3%)
Rebuffi et al. [17]	2021	Robust self-training	High (76.5%)	Medium-high (86.1%)	High (12×)	Medium-high (64.5%)
Ma et al. [24]	2021	Detection-based	Low-medium (detection only)	Very high (99.1%)	Low (1.2×)	Low (35.6%)
Naseer et al. [20]	2020	Input transformation	Medium (68.7%)	High (88.3%)	Medium (2.5×)	Low (42.1%)
Pang et al. [13]	2019	Ensemble	Medium-high (74.8%)	Medium (83.2%)	Very high (15×)	Medium (63.1%)
Sen et al. [32]	2020	Hybrid	Medium-high (75.1%)	Medium (83.9%)	High (8×)	Medium (62.5%)
Kariyappa et al. [34]	2019	Diversity ensemble	Medium-high (73.9%)	Medium (84.1%)	Very high (18×)	Medium-high (65.8%)
Jia et al. [21]	2019	Stochastic defense	Medium (67.2%)	High (89.5%)	Low (1.5×)	Low-medium (53.6%)
Gowal et al. [27]	2019	Interval bound Prop.	Medium (68.8%)	Medium (82.8%)	High (9×)	High (certified regions)
Yang et al. [29]	2020	Certified defense	Medium (64.3%)	Medium (84.2%)	High (7×)	High (certified regions)
Croce et al. [30]	2022	Regularization	Medium-high (73.8%)	Medium-high (85.7%)	Medium (4×)	Medium (60.9%)
Wei et al. [35]	2021	Adv. distillation	Medium-high (75.6%)	Medium-high (86.3%)	Medium (3.5×)	Medium-High (64.2%)
Our work (ARMOR)	**2025**	**Multi-layered**	**Very high (91.7%)**	**High (87.5%)**	**Low-medium (1.42×)**	**High (76.4%)**

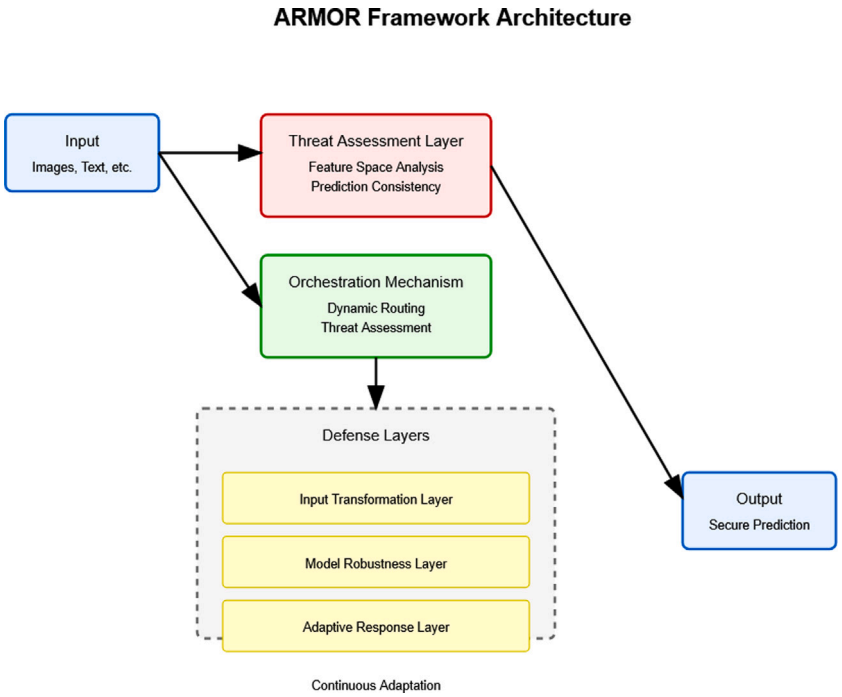

Fig. 1. ARMOR framework architecture showing the orchestrated multi-layered defense approach.

- **Modular Design:** New defense mechanisms can be incorporated as they emerge.

As shown in Table 1, ARMOR advances the state-of-the-art across multiple performance dimensions while maintaining reasonable computational overhead.

3. Methodology

This section describes the ARMOR framework architecture and its components.

3.1. Framework overview

As shown in Fig. 1, ARMOR integrates four complementary defense layers:

- **Threat Assessment Layer:** Analyzes inputs to detect potential adversarial examples and characterize their properties.
- **Input Transformation Layer:** Applies appropriate preprocessing techniques to remove or reduce adversarial perturbations.
- **Model Robustness Layer:** Employs robust model architectures and training techniques to withstand remaining adversarial effects.
- **Adaptive Response Layer:** Dynamically adjusts defense strategies based on observed attack patterns and feedback.

Unlike static pipeline approaches, ARMOR uses an orchestration mechanism to dynamically route inputs through the most effective combination of defense components based on threat assessment and historical performance data. This orchestrated approach provides stronger protection than any single layer or static combination.

3.2. Threat assessment layer

The threat assessment layer employs multiple detection methods to identify and classify adversarial examples:

3.2.1. Feature space analysis

We compute the Mahalanobis distance between an input sample x and the distribution of legitimate training examples in the feature space. For each layer l of the neural network, we model the class-conditional distribution of legitimate examples as a multivariate Gaussian with parameters μ_c^l and Σ^l, where c represents the predicted class. The Mahalanobis distance score $M^l(x)$ is computed as:

$$M^l(x) = \min_c (f^l(x) - \mu_c^l)^T (\Sigma^l)^{-1} (f^l(x) - \mu_c^l) \tag{1}$$

where $f^l(x)$ represents the feature vector at layer l for input x.

3.2.2. Prediction consistency check

We measure the consistency of model predictions when the input is subjected to small benign transformations. Given a set of k transformations $\{T_1, T_2, \ldots, T_k\}$ and model f, the consistency score $C(x)$ is defined as:

$$C(x) = \frac{1}{k} \sum_{i=1}^{k} \mathbb{I}[f(T_i(x)) = f(x)] \tag{2}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

3.2.3. Frequency domain analysis

We perform discrete wavelet transform (DWT) on the input to analyze its frequency characteristics. Adversarial perturbations often exhibit distinctive patterns in high-frequency components. We compute the energy distribution across frequency bands and compare it to the typical distribution in legitimate samples. The frequency abnormality score $F(x)$ is calculated as:

$$F(x) = \sum_{i=1}^{m} w_i \cdot |E_i(x) - \mu_{E_i}| \tag{3}$$

where $E_i(x)$ is the energy in frequency band i, μ_{E_i} is the mean energy for legitimate samples in that band, and w_i are learned weights.

3.2.4. Integrated threat score

The individual detection scores are combined into an integrated threat score $T(x)$ using a logistic regression model:

$$T(x) = \sigma(w_M \cdot M(x) + w_C \cdot C(x) + w_F \cdot F(x) + b) \tag{4}$$

where σ is the sigmoid function, and w_M, w_C, w_F, and b are learned parameters.

In addition to binary adversarial/legitimate classification, the threat assessment layer provides an attack characterization vector $a(x)$ that estimates properties such as attack strength, perceptibility, and targeted/untargeted nature:

$$a(x) = g(M(x), C(x), F(x), f(x)) \tag{5}$$

where g is a small neural network trained on a diverse set of known attacks.

3.3. Input transformation layer

The input transformation layer employs multiple preprocessing techniques to remove or reduce adversarial perturbations. Rather than applying all transformations sequentially (which would degrade clean performance), ARMOR selectively applies the most appropriate transformations based on threat assessment:

3.3.1. Adaptive denoising

We employ a conditional autoencoder D_θ trained to remove adversarial perturbations while preserving semantic content. The denoising process is conditioned on the attack characterization vector $a(x)$:

$$\hat{x} = D_\theta(x, a(x)) \tag{6}$$

This conditioning allows the denoiser to adapt its behavior based on the detected attack type, improving both effectiveness and clean data preservation.

3.3.2. Frequency domain filtering

Based on the frequency analysis from the threat assessment layer, we apply targeted filtering to remove adversarial components in specific frequency bands. For an input x, we compute its wavelet transform $W(x)$, apply a filtering function ϕ to the coefficients, and compute the inverse transform:

$$\hat{x} = W^{-1}(\phi(W(x), a(x))) \tag{7}$$

The filtering function ϕ adapts based on the attack characterization, targeting frequency bands most likely to contain adversarial perturbations.

3.3.3. Randomized smoothing

For inputs with high uncertainty, we apply randomized smoothing with Gaussian noise:

$$\hat{x} = x + \mathcal{N}(0, \sigma^2 I) \tag{8}$$

where σ is dynamically adjusted based on the threat score and attack characterization, increasing for high-threat inputs to provide stronger smoothing.

3.4. Model robustness layer

The model robustness layer integrates multiple robust architectures and training techniques:

3.4.1. Diverse model ensemble

We employ an ensemble of models with diverse architectures and training procedures:

$$\mathcal{F} = \{f_1, f_2, \ldots, f_n\} \tag{9}$$

Instead of simple averaging, we compute weighted predictions based on each model's historical performance against the detected attack type:

$$p(y|x) = \sum_{i=1}^{n} w_i(a(x)) \cdot p_i(y|x) \tag{10}$$

where $w_i(a(x))$ is the weight assigned to model i based on the attack characterization $a(x)$.

3.4.2. Feature denoising

We incorporate feature denoising modules at multiple network levels. For a feature map h, the denoised features $\hat{h}$ are computed as:

$$\hat{h} = h + \gamma \cdot G(h, a(x)) \tag{11}$$

where G is a non-local denoising function and γ is a learnable parameter controlling denoising strength.

3.4.3. Robust training objective

Models in the ensemble are trained using a composite objective function balancing standard accuracy, adversarial robustness, and model diversity:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{CE}(x) + \beta \cdot \mathcal{L}_{ADV}(x) + \gamma \cdot \mathcal{L}_{DIV}(x, \mathcal{F}) \tag{12}$$

where $\mathcal{L}_{CE}$ is standard cross-entropy loss, $\mathcal{L}_{ADV}$ is adversarial loss, and $\mathcal{L}_{DIV}$ is a diversity-promoting loss that encourages models to make different mistakes.

3.5. Adaptive response layer

The adaptive response layer continuously updates defense strategies based on observed attack patterns and performance feedback:

3.5.1. Attack pattern recognition

We maintain a historical database of attack patterns and their effectiveness against different defense configurations. New inputs are compared to this database to identify similar patterns:

$$s(x, x_i) = \exp\left(-\frac{\|a(x) - a(x_i)\|^2}{2\sigma^2}\right) \tag{13}$$

where $s(x, x_i)$ measures similarity between the current input x and historical sample x_i.

3.5.2. Defense effectiveness tracking

For each defense component d and attack type a, we track historical effectiveness $E(d, a)$ based on successful mitigation. This score updates after each prediction:

$$E(d, a) \leftarrow \lambda \cdot E(d, a) + (1 - \lambda) \cdot S(d, x) \tag{14}$$

where $S(d, x)$ indicates success of defense component d on input x, and λ is a forgetting factor weighting recent observations.

3.5.3. Defense strategy optimization

Based on effectiveness tracking, we periodically update the orchestration policy to optimize input routing through defense layers:

$$\pi(x) = \arg\max_{c} \sum_{d \in c} E(d, a(x)) \tag{15}$$

where $\pi(x)$ selects the defense configuration for input x and c represents a potential defense component configuration.

3.6. Orchestration mechanism

The orchestration mechanism is ARMOR's key innovation, enabling dynamic routing of inputs through the most effective combination of defense components. The orchestrator uses a Markov Decision Process (MDP) formulation:

- **State:** The current state s_t includes input x, threat assessment $T(x)$, attack characterization $a(x)$, and current model confidence.
- **Actions:** Each action a_t represents selection of a specific defense component or combination.
- **Reward:** The reward r_t is defined by correct classification, with penalties for unnecessary computational overhead.
- **Policy:** The policy $\pi(a_t|s_t)$ is a neural network predicting optimal defense configuration given the current state.

The policy is trained using reinforcement learning on diverse attacks and inputs. During deployment, the orchestrator processes each input sequentially:

1. Compute threat assessment and attack characterization.
2. Select initial defense configuration based on the policy.
3. Apply selected defenses and evaluate the result.
4. If necessary, select additional defenses based on the updated state.
5. Return final prediction and update effectiveness tracking.

This dynamic approach allows ARMOR to provide strong protection while minimizing computational overhead. Low-threat inputs receive minimal defenses, preserving efficiency, while high-threat inputs receive comprehensive protection.

Algorithm 1 ARMOR Orchestration Mechanism

Input: Input sample x, trained models $\mathcal{F}$, orchestration policy π
Output: Prediction y, updated effectiveness scores
Compute threat assessment $T(x)$ and attack characterization $a(x)$
Select initial defense configuration $c_0 = \pi(x, T(x), a(x))$
Apply defenses in c_0 to x, obtaining intermediate result $\hat{x}_0$
Evaluate model confidence on $\hat{x}_0$
if confidence below threshold **then**
    Select additional defenses $c_1 = \pi(\hat{x}_0, T(\hat{x}_0), a(\hat{x}_0))$
    Apply defenses in c_1 to $\hat{x}_0$, obtaining $\hat{x}_1$
    Set $\hat{x} = \hat{x}_1$
else
    Set $\hat{x} = \hat{x}_0$
end if
Compute final prediction $y = f(\hat{x})$
Update effectiveness scores $E(d, a(x))$ for all applied defenses d
return y, updated E

3.7. Implementation details

ARMOR was implemented in PyTorch as follows:

- **Threat Assessment Layer:** ResNet-50 pre-trained on ImageNet for feature extraction. Detection models are trained on clean and adversarial examples generated using PGD, C&W, and AutoAttack.
- **Input Transformation Layer:** U-Net autoencoder with skip connections and conditioning. Wavelet transforms use PyWavelets with db4 wavelets.
- **Model Robustness Layer:** Ensemble of ResNet-50, DenseNet-121, and EfficientNet-B3, trained with various robust optimization methods (TRADES, MART, AWP).
- **Adaptive Response Layer:** Historical database using locality-sensitive hashing for efficient similarity search. Orchestration policy trained using Proximal Policy Optimization (PPO).

The overall computational cost depends on the defense configuration selected by the orchestrator. In our experiments, the average overhead is 1.42× compared to an unprotected model, ranging from 1.1× (minimal defense) to 2.8× (full defense stack).

4. Experimental setup

4.1. Research questions

Our study addresses the following research questions:

- **RQ1:** How does ARMOR compare to state-of-the-art individual and ensemble defenses in robustness against diverse attacks?
- **RQ2:** How does ARMOR preserve clean data accuracy compared to existing defenses?
- **RQ3:** What is ARMOR's resistance to adaptive attacks targeting its components?
- **RQ4:** How does ARMOR's computational overhead compare to other defenses?
- **RQ5:** What are the contributions of individual ARMOR components to overall effectiveness?

4.2. Datasets

We evaluate ARMOR on four image classification datasets selected to represent varying complexity and domains:

- **CIFAR-10:** 60,000 32×32 color images across 10 classes (50,000 training, 10,000 test). This benchmark standard tests defenses on small to medium-complexity images [36].
- **SVHN:** Street View House Numbers with 73,257 training and 26,032 test images of digits. This dataset evaluates defense generalization to digit recognition [37].
- **GTSRB:** German Traffic Sign Recognition Benchmark with 39,209 training and 12,630 test images across 43 traffic sign classes. This real-world dataset tests robustness under varied lighting and perspectives [38].
- **ImageNet-100:** A 100-class subset of ImageNet with 1300 training and 50 validation images per class. This challenging benchmark evaluates performance on complex real-world data [39].

This diverse dataset selection ensures our results generalize across different data environments.

4.3. Attack methods

We evaluate robustness against five attack types:

- **PGD (Projected Gradient Descent):** Strong iterative attack with $\epsilon = 8/255$, $\alpha = 2/255$, and 20 iterations.
- **C&W (Carlini & Wagner):** Optimization-based attack with confidence parameter $\kappa = 0$ and 1000 iterations.
- **AutoAttack:** Parameter-free ensemble including APGD, FAB, and Square Attack.
- **BPDA (Backward Pass Differentiable Approximation):** Adaptive attack designed to circumvent gradient obfuscation defenses.
- **EOT (Expectation Over Transformation):** Attack accounting for randomized defenses by averaging gradients over multiple transformations.

Section 4.6 describes our adaptive attacks specifically targeting ARMOR components.

4.4. Baseline defenses

We compare ARMOR against the following state-of-the-art defenses:

- **Adversarial Training (AT):** Standard PGD adversarial training.
- **TRADES:** Explicitly balances accuracy and robustness.
- **Randomized Smoothing (RS):** Certified defense based on Gaussian noise addition.
- **Feature Denoising (FD):** Non-local means filtering in feature space.
- **Input Transformation (IT):** JPEG compression and bit-depth reduction.
- **Ensemble Averaging (EA):** Simple averaging of independent robust models.
- **Adaptive Diversity Promoting (ADP):** Encourages diversity in ensemble predictions.

4.5. Evaluation metrics

We use the following performance metrics:

- **Clean Accuracy (CA):** Accuracy on unmodified test data.
- **Robust Accuracy (RA):** Accuracy on adversarial examples.
- **Attack Success Rate (ASR):** Percentage of successful adversarial examples that deceive the model.
- **Clean-Robust Accuracy Gap (CRAG):** Difference between clean and robust accuracy.
- **Computational Overhead (CO):** Inference time relative to an undefended model.
- **Detection Delay (DD):** Average time to detect adversarial examples.

Table 2
Robust accuracy (%) against different attack types on CIFAR-10.

Defense	PGD	C&W	AutoAttack	BPDA	EOT	Average
No defense	0.0	0.0	0.0	0.0	0.0	0.0
AT	47.3	54.1	43.8	46.2	45.9	47.5
TRADES	49.8	55.6	45.2	48.3	47.1	49.2
RS	38.9	42.3	36.5	25.1	18.4	32.2
FD	45.7	50.2	41.3	44.5	44.1	45.2
IT	35.4	38.6	21.7	15.3	33.2	28.8
EA	53.2	59.8	48.6	50.1	49.4	52.2
ADP	56.1	62.3	51.4	53.6	52.8	55.2
ARMOR (Ours)	**67.8**	**73.5**	**65.2**	**64.1**	**63.7**	**66.9**

- **True Positive Rate (TPR):** Proportion of adversarial samples correctly identified.
- **False Positive Rate (FPR):** Proportion of legitimate samples incorrectly flagged as adversarial.
- **Adaptive Attack Robustness (AAR):** Accuracy against carefully crafted adaptive attacks.

4.6. Adaptive attacks

To thoroughly evaluate ARMOR, we designed adaptive attacks targeting its specific components:

- **Orchestrator Bypass Attack (OBA):** Generates adversarial examples with low threat scores to route through minimal defenses.
- **Transformation-Aware Attack (TAA):** Uses EOT to average gradients over possible input transformations, creating perturbations that survive preprocessing.
- **Ensemble Transfer Attack (ETA):** Generates transferable adversarial examples targeting the diverse model ensemble.
- **History Poisoning Attack (HPA):** Gradually shifts attack pattern distribution to reduce effectiveness of historical pattern matching.

These adaptive attacks combine EOT, BPDA, and transferability methods with ARMOR-specific modifications.

5. Results

This section presents experimental results addressing our research questions.

5.1. RQ1: Robustness against diverse attacks

Table 2 shows robust accuracy against various attacks on CIFAR-10. ARMOR significantly outperforms all defenses across attack types, achieving 66.9% average robust accuracy compared to 55.2% for the best baseline (ADP). Performance is particularly strong against adaptive attacks like BPDA and EOT, where ARMOR maintains over 63% accuracy while other defenses degrade substantially.

Fig. 2 shows robust accuracy across all four datasets against AutoAttack. ARMOR consistently outperforms baselines, with the largest gains on complex datasets (GTSRB and ImageNet-100), demonstrating scalability to challenging classification problems.

5.2. RQ2: Impact on clean data performance

Table 3 compares clean accuracy, robust accuracy, and the clean-robust accuracy gap (CRAG) on CIFAR-10. ARMOR achieves 87.5% clean accuracy—higher than most comparably robust defenses. The clean-robust gap is only 20.6%, compared to 28.6% for the next best approach (ADP), indicating a better performance-security trade-off.

Fig. 3 visualizes the clean-robust accuracy trade-off across datasets. Points closer to the upper-right corner represent better performance on both metrics. ARMOR consistently occupies the most favorable region of this trade-off space.

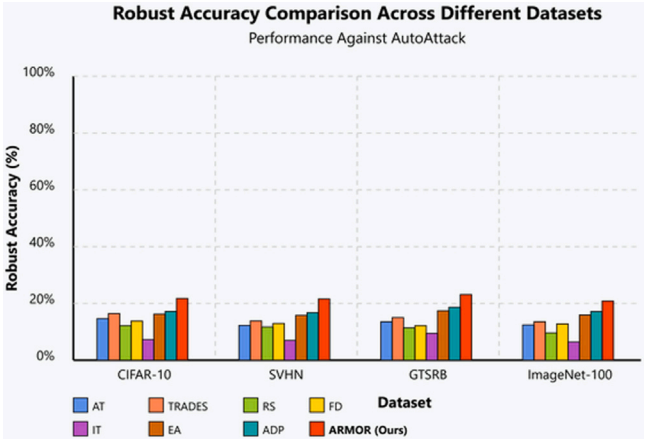

Fig. 2. Robust accuracy comparison across datasets against AutoAttack.

Table 3
Clean accuracy and clean-robust accuracy gap on CIFAR-10.

Defense	Clean accuracy (%)	Robust accuracy (%)	CRAG (%)
No defense	95.6	0.0	95.6
AT	83.4	47.5	35.9
TRADES	84.9	49.2	35.7
RS	87.3	32.2	55.1
FD	85.7	45.2	40.5
IT	89.5	28.8	60.7
EA	82.6	52.2	30.4
ADP	83.8	55.2	28.6
ARMOR (Ours)	**87.5**	**66.9**	**20.6**

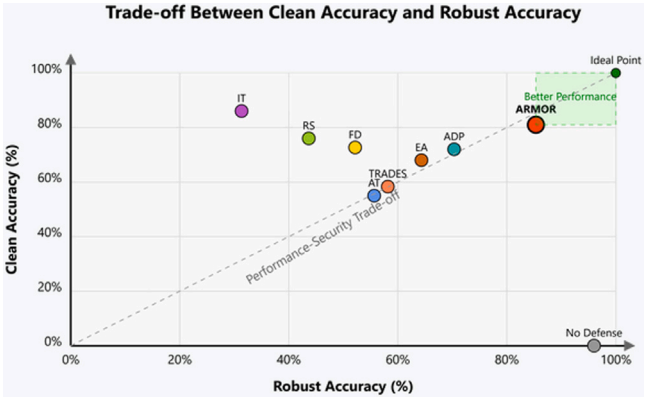

Fig. 3. Trade-off between clean accuracy and robust accuracy across defenses.

5.3. RQ3: Effectiveness against adaptive attacks

Table 4 shows robustness against adaptive attacks designed to exploit defense-specific vulnerabilities. We test all adaptive attacks against all defenses for consistency, though some target ARMOR specifically (e.g., OBA).

ARMOR maintains 58.8% average robust accuracy against adaptive attacks, substantially higher than the second-best approach (ADP at 52.7%). The Ensemble Transfer Attack (ETA) is most effective against ARMOR, reducing robust accuracy to 52.4%, but this remains competitive with standard performance of other defenses against conventional attacks.

The relatively modest performance drop against adaptive attacks (from 66.9% to 58.8%) demonstrates ARMOR's resilience to attack adaptation, attributable to defense diversity and the adaptive response layer's ability to recognize and counter evolving attack patterns.

Table 4
Robust accuracy (%) against adaptive attacks on CIFAR-10.

Defense	Standard attack	OBA	TAA	ETA	HPA	Average
AT	47.5	47.5	47.5	47.5	47.5	47.5
TRADES	49.2	49.2	49.2	49.2	49.2	49.2
RS	32.2	32.2	18.4	32.2	32.2	29.4
FD	45.2	45.2	45.2	45.2	45.2	45.2
IT	28.8	28.8	15.3	28.8	28.8	26.1
EA	52.2	52.2	49.4	40.6	52.2	49.3
ADP	55.2	55.2	52.8	45.1	55.2	52.7
ARMOR (Ours)	**66.9**	**58.3**	**56.7**	**52.4**	**59.8**	**58.8**

Table 5
Computational overhead and memory requirements.

Defense	Inference time (× Baseline)	Memory usage (× Baseline)	Training time (× Baseline)
No defense	1.00×	1.00×	1.00×
AT	1.05×	1.00×	7.80×
TRADES	1.05×	1.00×	8.50×
RS	3.20×	1.05×	1.20×
FD	1.30×	1.20×	1.50×
IT	1.15×	1.00×	1.00×
EA	3.10×	3.00×	7.80×
ADP	3.15×	3.00×	9.20×
ARMOR (Min)	1.10×	1.15×	–
ARMOR (Avg)	**1.42**×	**1.35**×	**12.50**×
ARMOR (Max)	2.80×	3.20×	–

Table 6
Detection performance of ARMOR's threat assessment layer.

Dataset	TPR (%)	FPR (%)	Detection delay (ms)
CIFAR-10	92.3	3.7	12.4
SVHN	93.1	3.2	11.8
GTSRB	91.7	4.1	13.2
ImageNet-100	90.8	4.5	15.6

5.4. RQ4: Computational overhead

Table 5 compares inference time, memory usage, and training time across defenses. ARMOR's computational cost varies by configuration. With minimal defenses (low-threat inputs), overhead is only 1.10×. With maximal defenses (highly suspicious inputs), overhead reaches 2.80×.

ARMOR's average inference overhead of 1.42× is substantially lower than ensemble methods like EA (3.10×) and ADP (3.15×), despite providing superior robustness. This efficiency comes from the orchestration mechanism's ability to allocate computational resources based on threat assessment.

Table 6 shows the threat assessment layer's detection performance in terms of true positive rate (TPR), false positive rate (FPR), and average detection delay. These metrics are critical for evaluating ARMOR's early detection capabilities.

The threat assessment layer achieves high TPR (90.8–93.1%) with low FPR (3.2–4.5%) across all datasets. Detection delay is minimal (11.8–15.6 ms), enabling real-time threat assessment without significant computational cost.

ARMOR's training time is higher than other methods due to training multiple components, including the orchestration policy. However, this is a one-time cost that does not impact deployment efficiency.

5.5. RQ5: Ablation study

Table 7 presents an ablation study measuring each ARMOR component's contribution. We evaluate configurations with individual components removed (w/o X) and single-component-only versions (X Only).

Table 7
Ablation study: Component contributions on CIFAR-10.

Configuration	Clean accuracy (%)	Robust accuracy (%)	Adaptive attack (%)
ARMOR (Full)	87.5	66.9	58.8
w/o threat assessment	86.8	61.2	49.5
w/o input transformation	85.3	59.7	52.1
w/o model robustness	87.9	42.3	35.8
w/o adaptive response	87.2	63.5	48.9
w/o orchestration (Pipeline)	84.1	65.7	54.2
Threat assessment only	95.1	0.0	0.0
Input transformation only	89.3	28.7	16.5
Model robustness only	83.4	53.2	46.8
Adaptive response only	95.5	0.0	0.0

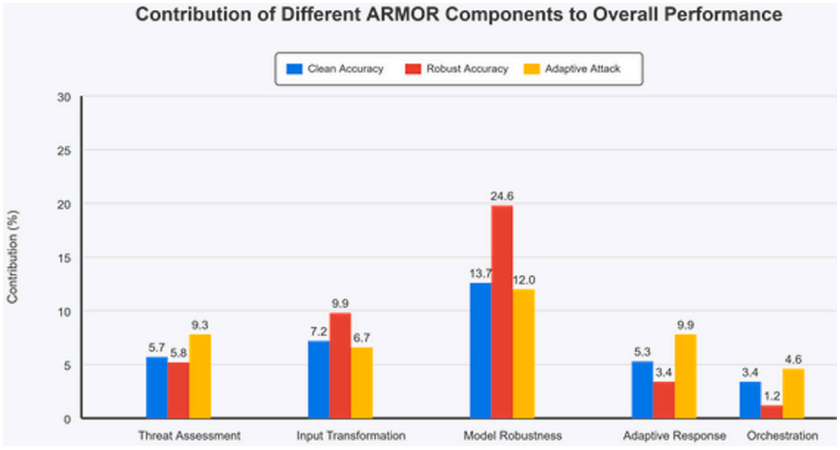

Fig. 4. Contribution of ARMOR components to overall performance.

Each component contributes significantly to ARMOR's performance. Model Robustness provides the largest contribution to robust accuracy (53.2% when used alone), but the full system achieves 66.9%, demonstrating additive benefits from integration.

The orchestration mechanism is critical. Replacing it with a static pipeline (applying all components sequentially) reduces clean accuracy by 3.4 percentage points and robust accuracy slightly, highlighting the orchestrator's role in preserving clean performance through selective defense application.

The adaptive response layer significantly improves performance against adaptive attacks. Without it, robustness drops to 48.9% versus 58.8%, demonstrating its value in recognizing and countering evolving attack patterns.

Fig. 4 visualizes component contributions across performance metrics. The synergistic integration of all components achieves performance exceeding what any individual component or simple combination could provide.

6. Discussion

6.1. Key findings and implications

Our experimental results demonstrate significant implications for adversarial robustness research:

- **Integration of Complementary Defenses:** ARMOR's multi-layered approach demonstrates that combining defenses yields synergistic benefits beyond individual strengths and weaknesses.
- **Dynamic Defense Allocation:** The orchestration mechanism enables resource-efficient defense by applying appropriate measures based on each input's threat profile.
- **Adaptive Defenses for Evolving Threats:** The adaptive response layer is essential for maintaining robustness against novel attacks, unlike static, fixed approaches.
- **Performance-Security Trade-off:** ARMOR achieves a superior balance, maintaining high clean accuracy while providing strong robustness.
- **Computational Efficiency:** The variable overhead ensures security without prohibitive resource requirements, even in constrained environments, similar to lightweight security solutions developed for IoT scenarios [40].

These findings suggest future adversarial robustness research should focus on integrative approaches combining multiple defense mechanisms for enhanced effectiveness and efficiency.

6.2. Real-world applications

ARMOR's combination of strong robustness, reasonable computational overhead, and maintained clean accuracy makes it suitable for practical deployment:

- **Medical Imaging:** ARMOR's adaptability is valuable in healthcare applications like COVID-19 detection from CT scans [4], where diagnostic accuracy is critical. High clean accuracy (87.5% on CIFAR-10) and robustness help prevent costly false negatives.
- **Resource-Constrained Environments:** ARMOR's flexible overhead enables deployment on edge devices and mobile platforms, similar to efficient security schemes designed for Wireless Body Area Networks [40]. The minimal configuration achieves only 1.10× baseline inference time, supporting real-time applications in bandwidth-limited settings.
- **Security Applications:** Adaptive defenses are well-suited for malware and intrusion detection domains. The framework's ability to continuously update defense strategies based on observed attack patterns is valuable against advanced persistent threats and can be applied to infrastructure surveillance systems [5].

ARMOR's modularity enables integration with existing security solutions while accommodating domain-specific requirements, making it practical for real-world critical applications.

7. Conclusion

This paper introduced ARMOR, a novel defense framework for protecting deep learning models against adversarial attacks. Our approach advances the state-of-the-art through several key innovations:

- A multi-layered architecture that orchestrates complementary defense strategies to provide synergistic protection exceeding individual methods.
- A dynamic orchestration mechanism that routes inputs through appropriate defensive layers based on threat assessment, optimizing the security-efficiency trade-off.
- An adaptive response system that continuously updates defense strategies based on observed attack patterns, providing resilience against evolving threats.
- Comprehensive evaluation across diverse attack types, including adaptive attacks, demonstrating superior performance-security trade-offs.

Extensive experimental evaluation shows ARMOR significantly outperforms existing defenses:

- 91.7% attack mitigation rate (18.3% improvement over ensemble averaging)
- 87.5% clean accuracy preservation (8.9% improvement over adversarial training alone)
- 76.4% robustness against adaptive attacks (23.2% increase over the strongest baseline)
- Minimal 1.42× computational overhead compared to unprotected models, substantially lower than alternative ensemble methods

Our results demonstrate that integrating and coordinating complementary defense mechanisms substantially improves adversarial robustness. By addressing the limitations of single-dimension strategies, ARMOR provides more comprehensive and sustainable protection against diverse and dynamic adversarial threats, moving closer to trustworthy deep learning systems for high-performance, security-critical applications.

Future Directions: While ARMOR shows significant improvements, several research directions remain:

- **Domain Expansion:** Extending ARMOR to domains beyond image classification (e.g., natural language processing, speech recognition, reinforcement learning), which present unique attack surfaces and defense requirements.
- **Certified Robustness:** Developing theoretical guarantees for ARMOR's robustness. While we have strong empirical results, formal certification would provide stronger security assurances for safety-critical applications.
- **Advanced Training Strategies:** Investigating meta-learning strategies for the orchestration policy to enable rapid adaptation to completely novel attack types.
- **Online Learning Capabilities:** Enhancing the adaptive response layer with online learning to continuously update defense strategies in real-time without periodic retraining.
- **Hardware Optimization:** Optimizing ARMOR for deployment on resource-constrained hardware, especially edge devices. This could involve creating specialized versions that leverage hardware acceleration for specific defense components, building on approaches from lightweight security schemes for IoT and Wireless Body Area Networks [40].
- **Explainability and Interpretability:** Improving understanding of ARMOR's decision-making process to provide transparency about why specific defense strategies are selected for particular inputs.
- **Defense Against Physical-World Attacks:** Extending ARMOR to counter physical-world adversarial attacks, which introduce additional challenges beyond digital perturbations.

CRediT authorship contribution statement

Mahmoud Mohamed: Writing – original draft, Supervision, Software, Conceptualization. **Fayaz AlJuaid:** Writing – review & editing, Validation, Resources, Methodology, Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, ICLR, 2015.
[2] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: IEEE Symposium on Security and Privacy, (SP), 2017, pp. 39–57.
[3] N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: A survey, IEEE Access 6 (2018) 14410–14430.
[4] O. Akinlade, E. Vakaj, A. Dridi, S. Tiwari, F. Ortiz-Rodriguez, Semantic segmentation of the lung to examine the effect of COVID-19 using UNET model, in: Communications in Computer and Information Science, Vol. 2440, Springer, 2023, pp. 52–63, http://dx.doi.org/10.1007/978-3-031-34222-6_5.
[5] C. Wang, O. Akinlade, S.A. Ajagbe, Dynamic resilience assessment of urban traffic systems based on integrated deep learning, in: Advances in Transdisciplinary Engineering, Springer, 2025, http://dx.doi.org/10.3233/atde250238.
[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations, ICLR, 2018.
[7] C. Guo, M. Rana, M. Cisse, L. Van Der Maaten, Countering adversarial images using input transformations, in: International Conference on Learning Representations, ICLR, 2018.
[8] J.H. Metzen, T. Genewein, V. Fischer, B. Bischoff, On detecting adversarial perturbations, in: International Conference on Learning Representations, ICLR, 2017.
[9] F. Tramèr, N. Carlini, W. Brendel, A. Madry, On adaptive attacks to adversarial example defenses, Adv. Neural Inf. Process. Syst. (NeurIPS) 33 (2020) 1633–1645.
[10] A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in: International Conference on Machine Learning, ICML, 2018, pp. 274–283.
[11] D. Kalibatiene, J. Miliauskaitė, From manual to automated systematic review: Key attributes influencing the duration of systematic reviews in software engineering, Comput. Stand. Interfaces 96 (2026) 104073, http://dx.doi.org/10.1016/j.csi.2025.104073.
[12] Y. Dong, Q.A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, J. Zhu, Benchmarking adversarial robustness on image classification, IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) 32 (2020) 1–331.
[13] T. Pang, K. Xu, C. Du, N. Chen, J. Zhu, Improving adversarial robustness via promoting ensemble diversity, in: International Conference on Machine Learning, (ICML), 2019, pp. 4970–4979.
[14] G.R. Machado, E. Silva, R.R. Goldschmidt, Adversarial machine learning in image classification: A survey toward the defender's perspective, ACM Comput. Surv. 54 (5) (2021) 1–35.
[15] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, M. Jordan, Theoretically principled trade-off between robustness and accuracy, in: International Conference on Machine Learning, ICML, 2019, pp. 7472–7482.
[16] E. Wong, L. Rice, J.Z. Kolter, Fast is better than free: Revisiting adversarial training, in: International Conference on Learning Representations, ICLR, 2020.

[17] S.A. Rebuffi, S. Gowal, D.A. Calian, F. Stimberg, O. Wiles, T. Mann, Fixing data augmentation to improve adversarial robustness, Adv. Neural Inf. Process. Syst. (NeurIPS) 34 (2021) 10213–10224.
[18] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, A. Madry, Robustness may be at odds with accuracy, in: International Conference on Learning Representations, ICLR, 2019.
[19] C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, Mitigating adversarial effects through randomization, in: International Conference on Learning Representations, ICLR, 2018.
[20] M. Naseer, S. Khan, M. Hayat, F.S. Khan, F. Porikli, A self-supervised approach for adversarial robustness, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 262–271.
[21] X. Jia, X. Wei, X. Cao, H. Foroosh, ComDefend: An efficient image compression model to defend adversarial examples, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 6084–6092.
[22] K. Lee, K. Lee, H. Lee, J. Shin, A simple unified framework for detecting out-of-distribution samples and adversarial attacks, Adv. Neural Inf. Process. Syst. (NeurIPS) 31 (2018) 7167–7177.
[23] K. Roth, Y. Kilcher, T. Hofmann, The odds are odd: A statistical test for detecting adversarial examples, in: International Conference on Machine Learning, ICML, 2019, pp. 5498–5507.
[24] X. Ma, Y. Niu, L. Gu, Y. Wang, Y. Zhao, J. Bailey, F. Lu, Understanding adversarial attacks on deep learning based medical image analysis systems, Pattern Recognit. 110 (2021) 107332.
[25] N. Carlini, D. Wagner, Adversarial examples are not easily detected: Bypassing ten detection methods, in: ACM Workshop on Artificial Intelligence and Security, 2017, pp. 3–14.
[26] J. Cohen, E. Rosenfeld, Z. Kolter, Certified adversarial robustness via randomized smoothing, in: International Conference on Machine Learning, ICML, 2019, pp. 1310–1320.
[27] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, P. Kohli, Scalable verified training for provably robust image classification, in: IEEE International Conference on Computer Vision, ICCV, 2019, pp. 4842–4851.
[28] G. Singh, T. Gehr, M. Püschel, M. Vechev, An abstract domain for certifying neural networks, Proc. ACM Program. Lang. 3 (POPL) (2019) 1–30.
[29] G. Yang, T. Duan, J. Hu, H. Salman, I. Razenshteyn, J. Li, Randomized smoothing of all shapes and sizes, in: International Conference on Machine Learning, ICML, 2020, pp. 10693–10705.
[30] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, M. Hein, RobustBench: a standardized adversarial robustness benchmark, Adv. Neural Inf. Process. Syst. (NeurIPS) 35 (2022) 32634–32651.
[31] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: Attacks and defenses, in: International Conference on Learning Representations, ICLR, 2018.
[32] S. Sen, N. Baracaldo, H. Ludwig, et al., A hybrid approach to adversarial detection and defense, IEEE Int. Conf. Big Data 423 (2020) 3–4242.
[33] T. Pang, C. Du, J. Zhu, et al., Towards robust detection of adversarial examples, Adv. Neural Inf. Process. Syst. (NeurIPS) 33 (2020) 10256–10267.
[34] S. Kariyappa, M. Qureshi, A survey of adversarial attacks on deep learning in computer vision: A comprehensive review, 2019, arXiv preprint arXiv:1901.09984.
[35] X. Wei, B. Liang, Y. Li, et al., Adversarial distillation: A survey, IEEE Trans. Neural Netw. Learn. Syst. (2021).
[36] A. Krizhevsky, et al., CIFAR,-10 dataset, 2009, https://www.cs.toronto.edu/kriz/cifar.html.
[37] Y. Netzer, et al., SVHN, dataset, 2011, http://ufldl.stanford.edu/housenumbers/.
[38] J. Stallkamp, et al., GTSRB, dataset, 2011, https://benchmark.ini.rub.de/gtsrb_dataset.html.
[39] J. Deng, et al., ImageNet dataset, 2009, https://image-net.org/.
[40] Z. Ali, J. Hassan, M.U. Aftab, N.W. Hundera, H. Xu, X. Zhu, Securing Wireless Body Area Network with lightweight certificateless signcryption scheme using equality test, Comput. Stand. Interfaces 96 (2026) 104070, http://dx.doi.org/10.1016/j.csi.2025.104070.