Exploring Novel Perspectives on Model Compression Techniques and Their Impact on Adversarial Robustness in Deep Learning: A Comprehensive Review and Analysis

Amirreza Darvishzadeh , Hadi Salloum * , Bader Rasheed , Marko Pezer , Manuel Mazzara

Posted Date: 2 July 2025

doi: 10.20944/preprints202507.0074.v1

Keywords: model compression; adversarial robustness; pruning; quantization; knowledge distillation-affect

 doi:10.20944/preprints202507.0074.v1

Article

Exploring Novel Perspectives on Model Compression Techniques and Their Impact on Adversarial Robustness in Deep Learning: A Comprehensive Review and Analysis

Amirreza Darvishzadeh [1], Hadi Salloum [2,4,5,*], Bader Rasheed [3], Marko Pezer [2,4,5] and Manuel Mazzara [4]

[1] Polytechnic University of Turin, 10129 Turin, Italy
[2] Research Center for Artificial Intelligence, Innopolis University, Innopolis, Russia
[3] Laboratory of Innovative Technologies for Processing Video Content, Innopolis University, Innopolis, Russia
[4] Department of Computer Science and Engineering, Innopolis University, Innopolis, Russia
[5] Q Deep, Innopolis, Russia
* Correspondence: h.salloum@innopolis.ru

Abstract

This paper presents a comprehensive review of model compression and adversarial robustness, two critical facets of deep learning that enhance the efficiency and security of neural networks, particularly in resource-constrained environments or when security concerns are paramount. The core novelty of this work lies in its exploration of how various model compression techniques—pruning, quantization, and knowledge distillation—affect adversarial robustness. We specifically investigate whether compressing neural networks compromises their robustness or, under certain conditions, enhances it. Our review rigorously evaluates these compression methods, analyzing their effectiveness across standard testing setups and discussing the trade-offs they impose. Additionally, we introduce a general benchmarking pipeline that assesses the robustness of compressed models against a range of adversarial attacks, factoring in compression rate and model complexity. Through a comparative analysis, we examine the performance of different compression strategies, providing the first comprehensive study of their interaction with adversarial robustness. To the best of our knowledge, this work is the first to systematically explore and compare all three major model compression techniques within this context.

Keywords: model compression; adversarial robustness; pruning; quantization; knowledge distillation—affect

1. Introduction

In recent years, the demand for real-time, on-device models has significantly increased, leading to their application across diverse fields. While large neural networks often achieve high accuracy, their substantial computational and memory requirements frequently render them impractical for on-device processing. To mitigate these challenges, model compression techniques—such as pruning, quantization, and knowledge distillation—are widely employed to reduce the size of neural networks without substantially compromising performance. These techniques have proven effective in practice, facilitating the broader deployment of Deep Neural Networks (DNNs) in resource-constrained applications[1].

However, a crucial aspect of model performance that is often overlooked in the context of compression is adversarial robustness [2,3]. Adversarial robustness refers to the model's ability to withstand adversarial attacks, wherein minor, imperceptible modifications to input data can lead to incorrect predictions. Mathematically, given a model $f : \mathbb{R}^d \to \mathbb{R}^k$ and an input $\mathbf{x} \in \mathbb{R}^d$, an adversarial example can be formulated as:

 doi:10.20944/preprints202507.0074.v1

$$\mathbf{x}_{adv} = \mathbf{x} + \delta \tag{1}$$

where $\|\delta\|_p \leq \epsilon$ for a specified norm p (e.g., L_2 or L_∞) and perturbation magnitude ϵ. Various strategies have been proposed to defend against such attacks, with one of the most prominent being adversarial training. This approach employs an adversary to generate adversarial examples, subsequently retraining the model on these examples to enhance its robustness [4]. Research has shown that adversarial training results in more accurate decision boundaries, as these boundaries attain a margin from data points, particularly within the L_p ball of the example.

The potential risks posed by adversarial attacks are particularly concerning in high-stakes applications such as autonomous vehicles and medical diagnostics. The susceptibility of a model to adversarial attacks becomes increasingly problematic when it undergoes compression, as the reduction in parameters and precision may adversely affect its capacity to resist adversarial perturbations. Given the paramount importance of deploying robust and secure models, it is essential to comprehend how compression techniques influence adversarial robustness, both in research and practical applications. The interplay between model compression and adversarial robustness remains poorly understood [5]. While compression techniques are primarily designed to minimize the trade-off between model size and accuracy, their impact on a model's security under adversarial conditions is not well characterized. Some studies suggest that model compression may render networks more vulnerable to attacks, while others propose that, under specific circumstances, compression could enhance robustness. However, most existing research treats model compression and adversarial robustness as distinct areas of study, thereby creating a critical gap in understanding how these compression methods influence robustness across varying model architectures and attack scenarios.

This paper seeks to address this gap by providing a systematic benchmarking study of model compression techniques within the context of adversarial robustness. We empirically evaluate three major compression techniques—pruning, quantization, and knowledge distillation—to ascertain their effects on the robustness of deep learning models against a spectrum of adversarial attacks. Our experiments investigate the trade-offs between model size, accuracy, and robustness, yielding new insights into how compression techniques can be optimized for real-world applications where efficiency and security are both critical.

To facilitate this empirical evaluation, we introduce a novel benchmarking pipeline that enables consistent comparisons across different models, compression rates, and attack methodologies. Our experiments investigate the trade-offs between model size, accuracy, and robustness, yielding new insights into how compression techniques can be optimized for real-world applications where efficiency and security are both critical.

1.1. Novelty of the Work

This study represents an effort to systematically examine the interplay between model compression techniques and adversarial robustness. The novelty of this work lies in its comprehensive approach to bridging two critical areas of deep learning research that have traditionally been treated in isolation. By providing empirical evidence on how different compression techniques affect the adversarial robustness of neural networks, we offer new perspectives on the trade-offs involved in deploying efficient models in real-world scenarios. Furthermore, the introduction of a standardized benchmarking pipeline sets a precedent for future research, facilitating more reliable comparisons and encouraging further investigations into this underexplored domain.

1.2. Paper Structure

The remainder of this paper is structured as follows: Section 2 presents a detailed review of existing literature on model compression and adversarial robustness, integrating both domains to provide a comprehensive background. Section 3 presents the methodology. Section 4 describes the experimental setup. In Section 5, we present the results of our evaluations, followed by a discussion

 doi:10.20944/preprints202507.0074.v1

in Section 6 that interprets the findings and their implications. Finally, Section 7 concludes the paper, summarizing the contributions and suggesting directions for future research.

By focusing on the intersection of model compression and adversarial robustness, this paper establishes a foundation for future research aimed at optimizing both model efficiency and security. Our findings illuminate critical considerations that researchers and practitioners must take into account when deploying compressed models in adversarial contexts, ultimately guiding the development of more robust and efficient neural networks for real-world applications.

2. Background

In this section, we investigate related works on model compression and adversarial robustness. We start by discussing the different model compression techniques that will be used in the study. Then, we will discuss adversarial attacks and defenses. Lastly, we will summarize the different studies that address the connection between robustness and model compression

2.1. Model Compression Techniques

Model compression is a crucial step in optimizing deep learning models for deployment in resource-constrained environments. This section outlines three widely used techniques for model compression: pruning, quantization, and knowledge distillation, providing theoretical explanations for each.

2.1.1. Pruning

Pruning reduces the size of neural networks by selectively removing parameters based on their importance as illustrated in Figure 1, which is often defined by the sensitivity of the loss function to these parameters [6]. Specifically, parameters that contribute minimally to the performance of the model are pruned, resulting in a sparser and more efficient model.

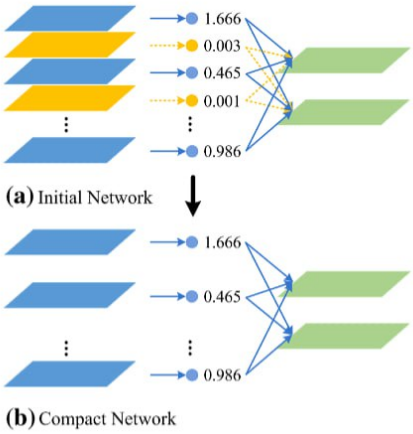

Figure 1. Illustration of the Model Pruning Process: a) The initial network structure before pruning; b) The compact network representing the pruned model after redundant parameters have been removed. Adapted from [6].

Let $f(x; W)$ represent the neural network function parameterized by the weight matrix W. The sensitivity of the loss function $L(f(x; W), y)$ concerning individual parameters can be used to determine which weights to prune. This sensitivity can be quantified using the magnitude of gradients or Hessian-based approaches:

$$S(w_{ij}) = \frac{\partial L}{\partial w_{ij}} \tag{2}$$

 doi:10.20944/preprints202507.0074.v1

Weights w_{ij} with low sensitivity $S(w_{ij})$ (i.e., those having little effect on the loss) are pruned, resulting in a new, sparse weight matrix W'. Pruning can be expressed mathematically as:

$$W' = W \odot \mathbf{1}_{S(W)>\theta} \tag{3}$$

where θ is a predefined threshold for sensitivity, and $\mathbf{1}_{S(W)>\theta}$ is an indicator function that retains only the weights with sensitivities greater than θ.

It is important to note that after pruning, the resulting weight matrix W' is generally sparse, meaning that it contains many zero-valued weights. However, W' does not have the same size or number of parameters as W in terms of non-zero elements. Pruning, therefore, reduces the computational cost and memory footprint of the model by eliminating less important connections between neurons.

2.1.2. Knowledge Distillation

Knowledge distillation is a model compression technique in which a smaller model (student) is trained to mimic the behavior of a larger, more complex model (teacher) [7] as illustrated in Figure 2. The student model is optimized to reproduce the output distribution of the teacher model, allowing it to retain most of the performance benefits of the teacher while being more computationally efficient.

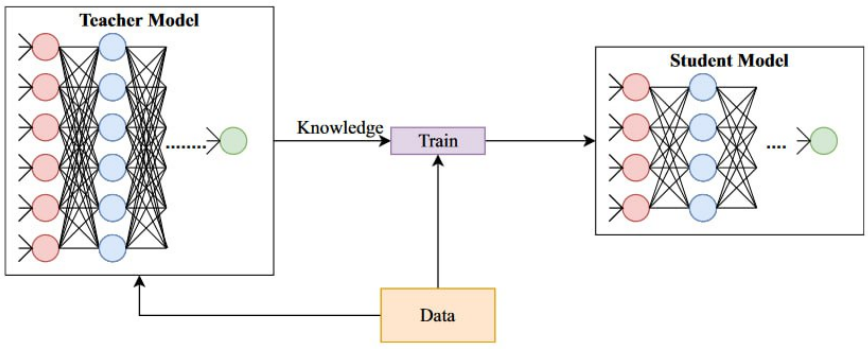

Figure 2. An illustrative representation of knowledge distillation. Adapted from [7].

In this technique, the student model is trained using a combination of the true labels and the soft predictions of the teacher model. The distillation loss function is typically defined as:

$$L_{\text{distill}} = \alpha L_{\text{CE}}(M_s(x), y) + (1 - \alpha) L_{\text{KL}}(p_t || p_s) \tag{4}$$

where:

- $M_s(x)$ and $M_t(x)$ are the outputs of the student and teacher models, respectively.
- $p_t = \text{softmax}\left(\frac{M_t(x)}{T}\right)$ is the teacher's probability distribution over the output classes, softened by a temperature T.
- $p_s = \text{softmax}(M_s(x))$ is the student's output distribution.
- L_{CE} is the cross-entropy loss with the true labels y, and L_{KL} is the Kullback-Leibler divergence between the teacher and student distributions.
- α is a weighting factor that balances the two loss terms.

The parameter T (temperature) controls the softness of the teacher's output probabilities, allowing the student model to learn from the relative similarities between classes, rather than just the hard class assignments. By transferring the knowledge encoded in the teacher model's outputs, knowledge distillation enables the student model to achieve comparable performance with fewer parameters, leading to efficient deployment.

 doi:10.20944/preprints202507.0074.v1

2.1.3. Quantization

Quantization compresses a model by reducing the precision of its parameters, often by converting floating-point weights and activations into lower-precision formats such as 8-bit integers as shown in the Figure 3. The goal of quantization is to minimize the model size and computational demands while retaining a similar level of performance [8].

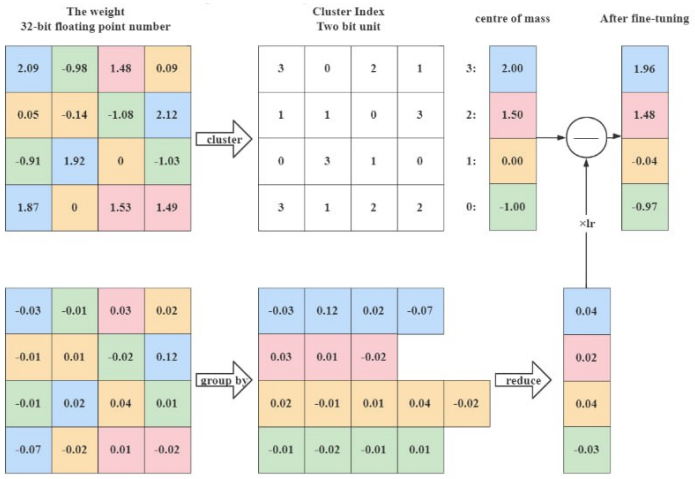

Figure 3. An illustrative representation of Quantization. Adapted from [8]

Mathematically, quantization of a weight w involves mapping it to a discrete set of values, typically determined by a quantization step size Δ:

$$w_q = \text{round}\left(\frac{w - w_{\min}}{\Delta}\right) \cdot \Delta + w_{\min} \tag{5}$$

where:

$$\Delta = \frac{w_{\max} - w_{\min}}{N} \tag{6}$$

Here, Δ represents the quantization resolution, $w_{\min}$ and $w_{\max}$ are the minimum and maximum weight values, and N is the number of quantization levels. By reducing the precision of weights and activations, quantization significantly reduces the model's memory requirements and speeds up inference, particularly in hardware accelerators like GPUs or TPUs [9].

Quantization is especially effective in scenarios where high precision is not necessary for maintaining the performance of the model, such as in inference tasks where slight variations in weight values have negligible effects on the final predictions.

2.2. Adversarial Attack Generation

Adversarial attacks involve crafting input perturbations that cause a machine-learning model to misclassify. These perturbations are often imperceptible to humans, making the attack especially dangerous in sensitive applications [10–14]. Several adversarial attack methods exist, including Fast Gradient Sign Method (FGSM) [15], Basic Iterative Method (BIM) [16], Carlini & Wagner (C&W) attacks [17], and Projected Gradient Descent (PGD) [18]. Among these, PGD is widely regarded as one of the most powerful first-order adversarial attacks, and thus, we will focus on it in our experiments. PGD iteratively modifies the input data to maximize the model's loss function, constrained by a perturbation limit.

The iterative update rule for PGD is given by:

$$x'_{t+1} = \text{clip}(x'_t + \alpha \cdot \text{sign}(\nabla_x L(M(x'), y)), x - \epsilon, x + \epsilon) \tag{7}$$

where:

 doi:10.20944/preprints202507.0074.v1

- $L(M(x'), y)$ is the loss function with respect to the model M and the target label y,
- x' is the adversarial example,
- ϵ is the perturbation limit, controlling the maximum allowable change to the input,
- α is the step size, determining how much the adversarial example is modified in each iteration.

We have chosen PGD because of its effectiveness in generating adversarial examples that are challenging for models to defend against, especially in a white-box setting where the model's parameters are known to the attacker.

2.2.1. Evaluation Metrics for Adversarial Robustness

Evaluating a model's robustness against adversarial attacks requires specific metrics that quantify the model's ability to withstand attacks while maintaining classification performance [15,18]. The key metrics used in this evaluation are:

Robustness Accuracy:

This metric measures the percentage of adversarial examples that the model correctly classifies. It is defined as:

$$\text{Robustness Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\{M(x_i') = y_i\} \tag{8}$$

where N is the total number of adversarial examples, $M(x_i')$ is the model's prediction on the adversarial example x_i', and y_i is the true label.

Average Perturbation:

This metric quantifies the average magnitude of the perturbations introduced by adversarial attacks. It is computed as:

$$\text{Average Perturbation} = \frac{1}{N}\sum_{i=1}^{N}\|x_i' - x_i\|_2 \tag{9}$$

where x_i is the original input, and x_i' is the corresponding adversarial example.

2.2.2. Defense Mechanisms

Various defense strategies have been proposed to counter adversarial attacks. These methods either train models to be more robust to adversarial perturbations or modify model predictions to mitigate the impact of such attacks. The most commonly used techniques include:

Adversarial Training:

Adversarial training involves augmenting the training process with adversarial examples. The model is trained using a loss function that combines the losses from both clean and adversarial data, thereby improving robustness. The combined loss is given by:

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(L_{\text{CE}}(M(x_i), y_i) + \lambda \cdot L_{\text{CE}}(M(x_i'), y_i)\right) \tag{10}$$

where:

- L_{CE} denotes the cross-entropy loss,
- $M(x_i)$ is the model's prediction on clean input x_i,
- $M(x_i')$ is the model's prediction on the adversarial example x_i',
- λ controls the balance between the losses from clean and adversarial examples.

 doi:10.20944/preprints202507.0074.v1

Defensive Distillation:

This technique trains the model with softened labels (probabilities rather than hard class labels) to make the model less sensitive to adversarial perturbations. By using smoothed outputs from a teacher model during training, the student model learns to generalize better and becomes more resilient against adversarial attacks.

Adversarial Purification:

A recent approach related to our research group, adversarial purification focuses on transforming adversarial examples back into their clean form before feeding them into the model. This method leverages generative models or denoising algorithms to "purify" the adversarial input, reducing the adversarial impact without requiring the model itself to be altered. Empirical results show promising robustness against various attack methods.

Input Transformation:

Input transformation techniques, such as image resizing, rotation, or adding noise, are employed to preprocess inputs before feeding them into the model. This preprocessing can disrupt the carefully crafted nature of the adversarial attack, though its effectiveness is generally considered low compared to other techniques. However, recent advancements in adversarial purification suggest that input transformations, when integrated with more sophisticated approaches, may yield better results.

2.2.3. Comparison of Defense Techniques

Each defense mechanism offers different levels of robustness, with varying trade-offs in terms of computational cost, complexity, and effectiveness. Adversarial training remains the most widely used and empirically validated approach for enhancing model robustness, particularly against white-box attacks. Adversarial purification shows good results and can be integrated with other defense mechanisms to improve overall robustness.

The table 1 reflects evaluations from studies on adversarial robustness [17–22], including advances in adversarial purification, which shows significantly improved results in various experimental settings.

Table 1. Comparison of Adversarial Defense Mechanisms

Defense Mechanism	Effectiveness	Complexity	Computational Cost
Adversarial Training	High	Moderate	High
Defensive Distillation	Moderate	High	Moderate
Adversarial Purification	High	Moderate	Moderate
Input Transformation	Low-Moderate	Low	Low

2.3. Compression and adversarial robustness

Pruning is a widely utilized compression technique that reduces the number of parameters in Deep Neural Networks (DNNs). At its core, pruning seeks to eliminate nodes or edges in the network (parameters) that contribute minimally to the output during training. While importance-based pruning methods, such as magnitude pruning, are common, research has demonstrated that they can sometimes be ineffective. Pruning strategies can be broadly categorized into two types: **Filter Pruning** and **Weight Pruning**. Filter pruning involves the removal of entire filters from Convolutional Neural Networks (CNNs). Filters are typically ranked based on an importance score, often calculated using $L1$ or $L2$ norms, with the least important filters being pruned. This method is classified as a **Regular Pruning** technique because it systematically reduces the model's complexity by eliminating entire filters, thereby affecting the network's structure. Various methodologies have been applied in filter pruning, with differing focuses and outcomes. For instance, neuron pruning, which removes neurons from fully connected layers and subsequently all connected edges, represents another form of pruning. However,

 doi:10.20944/preprints202507.0074.v1

as our research is centered on image classification tasks involving CNNs, neuron pruning is not the primary focus of this discussion.

In contrast, **Weight Pruning** concentrates on removing individual weights within filters or channels while preserving the overall network structure, often referred to as **Irregular Pruning**. Several studies have examined the implications of combining pruning with adversarial training to enhance model robustness. For example, Han et al. [23] introduce a magnitude-based pruning technique that involves iterative stages of identifying, pruning, and retraining, resulting in a sparse model with significantly reduced storage requirements. This process is particularly effective when integrated with adversarial training. By adding a regularizer to the min-max optimization problem, as demonstrated in [23], models can achieve a compression rate of 4× while maintaining both natural and adversarial accuracy. This approach combines regular and irregular pruning techniques to maximize the benefits of both methods. The relationship between pruning and adversarial training is further explored in the work [24] and the work [25], which empirically demonstrate that adversarial training post-pruning is critical for maintaining adversarial accuracy. Their findings highlight the inherent trade-offs between classification accuracy, compression size, and robustness. This idea is echoed in the work of other researchers [26], who propose dynamic network surgery as a comprehensive compression pipeline. This method iteratively prunes and splices redundant weights, followed by further compression through quantization and encoding, achieving faster and higher compression rates compared to earlier methods such as those proposed in the work [23].

Another innovative approach is introduced in work [27], which developed a pruning framework that operates on adversarially trained models without the need for adversarial examples, albeit with increased training time. This method employs self-distillation techniques and the Hilbert-Schmidt Independence Criterion (HSIC) metric to retain robustness, solving the problem using the Alternating Direction Method of Multipliers (ADMM). This technique addresses the combinatorial nature of the problem more effectively than traditional Stochastic Gradient Descent (SGD). Pruning techniques can be further divided into **One-Shot Pruning** and **iterative pruning**. One-shot pruning is exemplified in studies where pruning is completed in a single step, potentially enhancing model robustness in specific cases, as seen in [25]. Conversely, iterative pruning, which involves repeated cycles of pruning and retraining, is more commonly used in conjunction with adversarial training to refine model accuracy and robustness. For instance, [28] introduces a dynamic network rewiring (DNR) method that integrates pruning and adversarial training into a unified, non-iterative training framework. This approach leverages a hybrid loss function that combines clean image classification, adversarial training loss, and a dynamic L2 regularizer, dynamically adjusting inter-layer connectivity. DNR provides over 20× compression on models such as VGG16 and ResNet-18 across datasets like CIFAR-10, CIFAR-100, and Tiny-ImageNet, with negligible accuracy loss. Subsequent research has enhanced this model with DNR++, which incorporates sparse parametric noise regularization to improve robustness, while FDNR++ employs a "free" adversarial training technique that significantly reduces training time.

On the other hand, quantization is a critical technique for reducing the precision of weights and activations in DNNs, thereby decreasing the model's memory usage and computational cost. It can be broadly categorized into two main types: **Weight Quantization**, which focuses on reducing the precision of the model's weights, and **Activation Quantization**, which targets the activations or outputs of neurons. Several studies have explored the implications of quantization on adversarial robustness, often employing different quantization-aware training techniques or evaluating models under various adversarial conditions.

A significant focus in the literature has been on understanding how quantization impacts model robustness against adversarial attacks. Both [29] and [30] examine the effects of quantization on model robustness, particularly under adversarial settings. Gorsline et al. demonstrate that under certain attack strengths, model accuracy remains largely stable or slightly improves regardless of the quantization rate, attributing this to the gradient masking effect of the ReLU activation function, which inherently resists stronger attacks. Duncan et al. expand on this by showing that quantized models

 doi:10.20944/preprints202507.0074.v1

retain almost all of the robustness of their full-precision counterparts, with relative robustness ranging from 98.6% to 99.7%. However, they observe that adversarial deficiencies in full-precision models are not entirely transferred to quantized models, with adversarial transfer rates varying between 52.0% and 98.9%. Another line of research focuses on the regularization effects of quantization and its ability to handle adversarial perturbations. [31] explores the dual role of quantization in reducing noise for small perturbations while potentially amplifying it for larger ones. They attribute this amplification to error propagation and propose adding a regularization term to the cost function to limit the network's Lipschitz constant, thereby controlling the extent of this amplification. Similar concerns are addressed by Song et al. [32], who introduce layer-wise quantization to better preserve adversarial robustness. They critique prior work for focusing primarily on white-box attacks, arguing that a broader approach is necessary to validate robustness. Their method involves integrating a quantization loss term with general adversarial training loss, evaluated using the Lipschitz constant.

The interplay between quantization and adversarial training has also been a key area of exploration. Both [33] and [34] highlight the complex relationship between quantization and adversarial robustness. [33] focus on transformer-based text classifiers, showing that while quantization slightly decreases natural accuracy, it significantly enhances robustness against adversarial attacks. [34] provides an empirical analysis that reveals the conflicting conclusions about the effects of quantization on adversarial robustness often stem from variations in experimental setups, including different adversarial training techniques and quantization precisions. Research on hybrid quantization methods and their implications for robustness is also notable. The work [35] introduces a hybrid quantization approach driven by Adversarial Noise Sensitivity (ANS), which optimally balances energy efficiency, accuracy, and robustness. They quantify each layer's sensitivity to adversarial perturbations and assign lower bit-widths to more sensitive layers. However, their approach overlooks the potential for error amplification during ANS calculation and does not address the optimal number of training epochs, which could vary based on the dataset.

Other innovative approaches include Binarized Neural Networks (BNNs) and non-linear mapping techniques. The work [36] proposes BNNs, which quantize models to 1-bit precision. While this method significantly reduces model size, it introduces challenges such as gradient masking and complicating adversarial training. Nevertheless, their results indicate that BNNs can outperform full-precision models under certain conditions, particularly in Fast Gradient Sign Method (FGSM) settings. In contrast, In work [37], the authors propose a non-linear mapping technique for the final layers of a model to enhance robustness against first-order white-box attacks. This method utilizes μ-law non-linearity, slightly reducing natural accuracy while significantly improving adversarial robustness. However, they caution that adversarial training itself is susceptible to quantization effects, and non-linear mapping may not perform well in such settings. Lastly, Knowledge Distillation (KD) is another pivotal technique for model compression, particularly concerning adversarial robustness. This process involves training a smaller model (the student) to replicate the behavior of a larger, pre-trained model (the teacher). The knowledge distilled from the teacher to the student encompasses more than just the final output probabilities, including intermediate representations that can enhance the robustness of the distilled model.

Recent studies highlight the effectiveness of KD in enhancing adversarial robustness. For instance, The work [38] first introduced the idea of distillation, demonstrating that the student model could achieve competitive performance when trained with softened logits from the teacher model. Subsequent research has built on this foundation, with the work [39] which demonstrates that adversarial examples can be used as a training signal for the student, thereby improving its robustness. This approach is particularly relevant given that teacher models often exhibit heightened robustness due to adversarial training, making the distilled student models more resilient against similar attacks.

Overall, the synergy between model compression techniques—such as pruning, quantization, and knowledge distillation—and adversarial robustness remains a rich field of inquiry. The existing literature demonstrates that while these techniques can enhance model efficiency, they also intro-

 doi:10.20944/preprints202507.0074.v1

duce complex dynamics that warrant further investigation to fully understand their implications for adversarial training and robustness.

3. Methodology

This research introduces a benchmarking pipeline aimed at evaluating the trade-offs between model compression, adversarial robustness, and parameter efficiency in pre-trained models. The methodology is divided into four distinct phases: pretraining, compression, fine-tuning (if necessary), and evaluation. Each phase is designed to ensure a comprehensive analysis of how compression affects both model size and resilience to adversarial attacks, providing actionable insights for optimizing model performance under various constraints.

3.1. Pipeline Overview

The benchmarking pipeline, as illustrated in Figure 4, is structured into four primary stages:

1. Pretraining
2. Compression
3. Fine-tuning
4. Evaluation

Each stage plays a critical role in understanding the impacts of compression on model robustness and efficiency.

3.1.1. Pretraining

In the initial phase, the model M is pretrained on a dataset $\mathcal{D}$, establishing a baseline performance. The objective during pretraining is to minimize the loss function $\mathcal{L}(M, \mathcal{D})$ through standard optimization techniques, such as stochastic gradient descent (SGD).

$$\mathcal{L}(M, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \ell(M(x_i), y_i) \tag{11}$$

Here, $x_i \in \mathcal{D}$ denotes the input data samples, y_i represents the corresponding labels, and ℓ is the chosen loss function (e.g., cross-entropy loss). This step generates the pre-trained model M that will undergo compression.

3.1.2. Compression

Once pretrained, the model M is subjected to a compression algorithm $\mathcal{C}$, yielding the compressed model $M_c = \mathcal{C}(M)$. The goal of compression is to reduce the model's size while maintaining its performance. This necessity arises from the need to deploy models efficiently in resource-constrained environments without sacrificing their effectiveness. The compression process involved Pruning, Quantization, Knowledge distillation, and techniques such as

3.1.3. Fine-Tuning

If the compression process results in a significant drop in model performance, fine-tuning is employed to restore accuracy. In this stage, the compressed model M_c is further trained on the original dataset $\mathcal{D}$, optimizing its post-compression performance.

$$M_c^* = \text{Fine-Tune}(M_c, \mathcal{D}) \tag{12}$$

Fine-tuning helps bridge the gap between the compressed model's reduced size and its original performance.

 doi:10.20944/preprints202507.0074.v1

3.1.4. Evaluation

The final stage involves rigorous evaluation of the compressed model M_c using a series of adversarial attacks, specifically employing the Projected Gradient Descent (PGD) attack. The adversarial robustness $R(M_c)$ is quantified as the model's accuracy under adversarial perturbations:

$$R(M_c) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(M_c(x_i + \delta) = y_i) \tag{13}$$

Here, δ represents the adversarial perturbation, and $\mathbb{I}(\cdot)$ is the indicator function. Robustness is measured across various levels of perturbation ϵ, providing insight into the model's stability under adversarial conditions.

3.2. Benchmarking Pipeline Key Components

The benchmarking pipeline integrates multiple dimensions to evaluate each model comprehensively. The key components include:

3.2.1. Model Compression Metrics

Each pretrained model M undergoes compression, with the compression ratio γ and parameter count P computed. These metrics serve as indicators of compression efficiency and model size reduction:

$$\gamma = \frac{P_{\text{original}}}{P_{\text{compressed}}} \tag{14}$$

3.2.2. Adversarial Robustness Evaluation

The adversarial robustness R of compressed models is tested against various attack methods, focusing primarily on the PGD attack. For each perturbation level ϵ, the robustness metric is calculated by applying equation 13

This metric quantifies the model's ability to maintain accuracy when subjected to adversarial attacks.

3.2.3. Categorization of Compression Techniques

After evaluation, the compressed models are categorized based on their compression technique (pruning, quantization, or knowledge distillation), as well as their respective compression ratio γ and parameter count P. This categorization aids in understanding how different compression methods influence adversarial robustness and overall performance.

3.2.4. Performance Visualization

To facilitate comparative analysis, the pipeline generates detailed visualizations of the trade-offs between compression efficiency and adversarial robustness. These visualizations, generated using libraries such as *matplotlib* and *seaborn*, allow users to identify models that strike an optimal balance between size and robustness.

The pipeline summarizes the performance of each compressed model, providing recommendations based on the evaluation results. This includes identifying models that offer a favorable trade-off between compression ratio, parameter count, and robustness to adversarial attacks.

 doi:10.20944/preprints202507.0074.v1

Algorithm 1 Benchmarking Pipeline for Pretrained and Compressed Models

Input: Pretrained model M, Dataset $\mathcal{D}$
Pretrain M on $\mathcal{D}$
Compress M using compression algorithm $\mathcal{C}$ to obtain M_c
if fine-tuning is required **then**
    Fine-tune M_c on $\mathcal{D}$
    $M_c^* = \text{Fine-Tune}(M_c, \mathcal{D})$
end if
for each adversarial attack **do**
    Evaluate M_c under attack A
    Compute adversarial robustness $R(M_c, A)$
end for
Compute compression ratio γ and parameter count P
Group models by compression technique, compression ratio γ, and parameter count P
Visualize performance trade-offs and summarize results

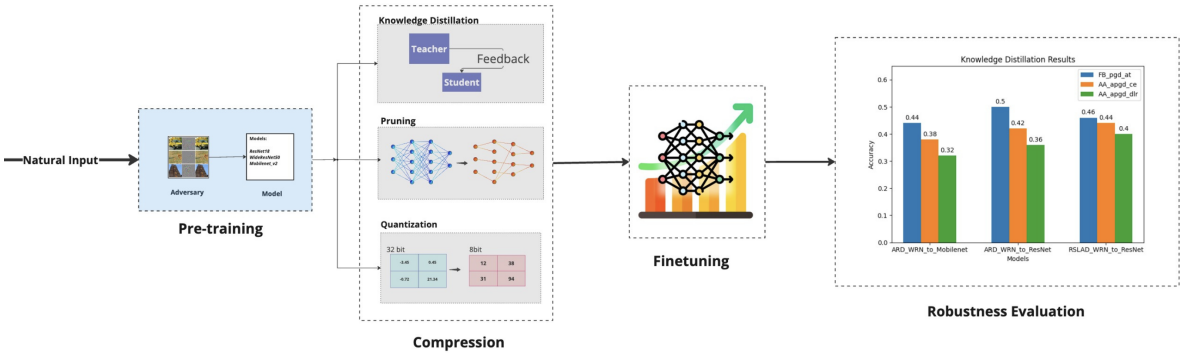

Figure 4. Abstract overview of the Benchmarking Pipeline.

4. Experimental Setup

In this section, we present the experimental setup used to evaluate the adversarial robustness of six models, each utilizing different architectures and compression techniques, including pruning, knowledge distillation, and quantization. The models were subjected to three distinct adversarial attacks, all implemented using the Foolbox library [40,41]. The strength of the adversarial perturbations was controlled by an ϵ-value of $\epsilon = \frac{8}{255}$ under the L_∞-norm. Table 2 provides a summary of the models, including their architectures, compression rates, and performance under adversarial attacks.

The evaluation includes two **pruned models**, RAP-ADMM [5] and Hydra [42], both of which apply different pruning strategies to neural networks. RAP-ADMM utilizes an adversarial pruning approach based on the WideResNet architecture, while Hydra implements layer-wise pruning on the ResNet architecture. These models were tested against both standard adversarial attacks and the more advanced Auto-PGD attack, ensuring consistent conditions across all experiments.

For **knowledge distillation**, three models were evaluated: RSLAD, ARD, and IAD. These models are based on distillation techniques and incorporate adversarial defense strategies. Their adversarial robustness was assessed using Projected Gradient Descent (PGD) attacks, with variations in loss functions such as Cross-Entropy and Difference of Logits Ratio (DLR) [43,44], providing a comprehensive view of their defense capabilities.

The evaluation also includes a **quantized model**, which applies low-bitwidth quantization to both weights and activations. Specifically, the model reduces the precision from full 32-bit ($w32\_a32$) to 1-bit ($w1\_a1$), achieving a 32x reduction in computational complexity. Despite this reduction, the model maintained a robust defense against adversarial attacks, demonstrating competitive accuracy relative to other quantization approaches [45]. The model was evaluated under identical attack parameters, including an ϵ-value of $\frac{8}{255}$ and the L_∞-norm constraint.

 doi:10.20944/preprints202507.0074.v1

Table 2 provides a consolidated overview of all the models, their architectures, applied compression techniques, and results from the adversarial attack evaluations.

Table 2. Models and Adversarial Attacks

Model	Architecture	Compression Technique	Adversarial Attacks
RAP-ADMM	WideResNet	ADMM Pruning	FB_pgd_at, AA_apgd-ce, AA_apgd-dlr
Hydra	ResNet	Layer-wise Pruning	FB_pgd_at, AA_apgd-ce, AA_apgd-dlr
RSLAD	ResNet (Student)	Distillation (Soft-label)	FB_pgd_at, AA_apgd-ce, AA_apgd-dlr
ARD	MobileNet_v2	Distillation	FB_pgd_at, AA_apgd-ce, AA_apgd-dlr
IAD	ResNet (Student)	Distillation (Unreliable-student)	FB_pgd_at, AA_apgd-ce, AA_apgd-dlr
Low-bitwidth	WideResNet	Low-bit Qauntization	FB_pgd_at

5. Results

This study evaluated the impact of various model compression techniques under adversarial conditions across multiple architectures, including Wide Residual Networks (WRN), ResNet, and MobileNet. The evaluation focused on key performance metrics such as compression rate, parameter reduction, and adversarial robustness. The robustness was thoroughly tested using adversarial attacks, specifically the Fast Gradient Sign Method (FGSM) with Projected Gradient Descent (FB-PGD) and the AutoAttack (AA) framework, incorporating both APGD-CE and APGD-DLR variants.

The performance metrics for models compressed using the Hydra framework are presented in Table 3, showcasing the effectiveness of its pruning strategies. The results of the Intra-Architecture Distillation (IAD) approach, a form of knowledge distillation, are summarized in Table 4. Additionally, conventional knowledge distillation methods are evaluated, with results provided in Table 5.

For models subjected to the RAP-ADMM compression technique, performance metrics are detailed in Table 6, highlighting the balance between compression efficiency and adversarial robustness. The outcomes of the RSLAD methodology are encapsulated in Table 7, demonstrating its adversarial resilience. Finally, the performance results from the quantization framework are summarized in Table 8, emphasizing the trade-off between low-bitwidth quantization and adversarial defense capabilities.

This comprehensive analysis provides valuable insights into the trade-offs between model compression and adversarial robustness, with a focus on how different architectures and compression strategies respond to sophisticated adversarial attacks.

Table 3. Performance Metrics of Hydra-based Compression Methodologies

Compression Rate	Model Name	# Parameters	Architecture	FB-PGD Attack	AA (APGD-CE)	AA (APGD-DLR)
99x	finetune 99x	**122,283**	wrn_28_4	**8.0**%	**8.0**%	**8.00**%
95x	finetune 95x	611,419	wrn_28_4	8.0%	0.0%	0.0%
90x	finetune 90x	1,222,839	wrn_28_4	4.0%	2.0%	0.0%

 doi:10.20944/preprints202507.0074.v1

Table 4. Performance Metrics for Knowledge Distillation via IAD

Model	Compression Rate	# Parameters	Architecture	FB-PGD Attack	AA (APGD-CE)	AA (APGD-DLR)
IAD-I	3.4x	11M	resnet18	**52.00**%	40.00%	36.00%
IAD-II	3.4x	11M	resnet18	48.00%	**44.00**%	**42.00**%

Table 5. Performance Metrics of Traditional Knowledge Distillation

Model	Compression Rate	# Parameters	Architecture	FB-PGD Attack	AA (APGD-CE)	AA (APGD-DLR)
ARD_WRN_to_Mobilenet	**10.7**x	3.4M	mobilenet-v2	44.00%	38.00%	32.00%
ARD_WRN_to_ResNet	3.12x	11.7M	ResNet	**50.00**%	**42.00**%	**36.00**%

Table 6. Performance Metrics of RAP-ADMM Compression Techniques

Model	Compression Rate	# Parameters	Architecture	FB-PGD Attack	AA (APGD-CE)	AA (APGD-DLR)
RAP_prune	**12**x	931,163	resnet18	40.00%	36.00%	32.00%
RAP_prune	16x	698,373	resnet18	40.00%	34.00%	30.00%
RAP_prune	8x	1,396,745	resnet18	42.00%	32.00%	30.00%
RAP_pre	Not Compressed	11,173,962	resnet18	44.00%	**42.00**%	**36.00**%
RAP_fine	**12**x	931,163	resnet18	42.00%	38.00%	32.00%
RAP_fine	16x	698,373	resnet18	42.00%	36.00%	34.00%
RAP_fine	8x	1,396,745	resnet18	**48.00**%	38.00%	34.00%

Table 7. Performance Metrics of RSLAD Compression Techniques*

Model Name	Compression Rate	# Parameters	Architecture	FB-PGD Attack	AA (APGD-CE)	AA (APGD-DLR)
RSLAD_WRN_to_ResNet	3.12x	11.7M	ResNet	46.00%	**44.00**%	**4.00**%
RSLAD_WRN_to_Mobilenet_V2	**10.7**x	3.4M	MobileNet	46.00%	42.00%	36.00%

Table 8. Performance Metrics of Quantization Framework

Attack	Accuracy	Compression Rate
PGD	**56.00**%	32x
CWL2	49.20%	32x

6. Discussion

The results presented herein elucidate a nuanced relationship between various model compression techniques and their corresponding adversarial robustness across distinct architectures. Each compression strategy exhibits unique characteristics that significantly influence performance metrics, particularly regarding resilience to adversarial attacks.

1. **Hydra Techniques**: The Hydra methodologies demonstrate substantial compression rates (99x, 95x, and 90x); surprisingly, these high compression levels are accompanied by a marked increment in adversarial robustness, particularly when subjected to the FB-PGD and AA attacks.
2. **Knowledge Distillation (IAD vs. KD)**: The IAD framework displays superior performance compared to traditional KD approaches, evidenced by enhanced robustness metrics. Notably, IAD methods achieve higher accuracy under adversarial conditions, indicating a potential advantage in utilizing intra-architecture relationships to mitigate adversarial effects. Here IIAD-I stands for Introspective Adversarial Distillation based on ARD [44] and IAD-II Introspective Adversarial Distillation based on AKD2 [46].
3. **RAP-ADMM Techniques**: Models subjected to RAP-ADMM compression exhibit varying degrees of adversarial resilience, suggesting that the selection of appropriate compression rates plays a critical role in optimizing both model size and robustness.

 doi:10.20944/preprints202507.0074.v1

4. **RSLAD and Quanztion Frameworks**: Both RSLAD and Quanztion frameworks present compelling results, particularly the Quanztion framework, which achieves notable accuracy metrics while maintaining a viable compression rate. This indicates a promising avenue for future research focusing on integrating quantization with adversarial robustness.

The observed results can be intuitively and theoretically explained by the interplay between model compression and adversarial robustness. Techniques like pruning in the Hydra methodologies remove redundant parameters, which intuitively reduces the model's complexity and potential overfitting to adversarial perturbations. Theoretically, pruning acts as a form of regularization that simplifies the model's decision boundaries, making it less sensitive to small, adversarially crafted input changes. Similarly, the Intra-Architecture Distillation (IAD) approach outperforms traditional Knowledge Distillation (KD) by internally preserving robust features within the same architecture. This internal distillation maintains the integrity of adversarial defenses more effectively than transferring knowledge from a separate teacher model, ensuring that crucial defensive mechanisms are retained in the compressed model.

Among all the techniques evaluated, quantization demonstrated the most remarkable results by achieving the highest compression rate while maintaining strong adversarial accuracy. By reducing the precision of the model parameters, quantization introduces discretization noise, which disrupts the gradient calculations used in adversarial attacks, thereby enhancing robustness. The quantization framework achieved a compression rate of 32x with an accuracy of 56.00% against PGD attacks and 49.20% against CWL2 attacks, outperforming other methods in both compression efficiency and adversarial resilience. This indicates that quantization not only effectively reduces the model size but also inherently strengthens the model against adversarial threats, making it the most effective technique among those evaluated.

The findings highlight the necessity for future explorations to delve deeper into the trade-offs between model compression and adversarial resilience. It remains imperative to formulate strategies that not only enhance model efficiency but also fortify defenses against adversarial attacks. Furthermore, employing a multi-faceted approach that incorporates advanced machine learning techniques may yield substantial improvements in model robustness.

7. Conclusions

In conclusion, this study has demonstrated the efficacy of various model compression techniques under adversarial conditions, emphasizing the complex interplay between compression rates and adversarial robustness. The results indicate that while aggressive compression strategies can significantly reduce model size, they may also compromise the model's ability to withstand adversarial attacks. The Intra-Architecture Distillation (IAD) approach emerged as a particularly effective method for enhancing robustness while maintaining acceptable compression rates. Additionally, the RAP-ADMM and RSLAD techniques showed promise in balancing compression and adversarial performance. Future work should focus on addressing the limitations identified in this study, particularly through the exploration of hybrid approaches that integrate multiple compression techniques to optimize both model efficiency and resilience. Moreover, expanding the evaluation to include a wider array of architectures and adversarial attack scenarios could further elucidate the robustness and applicability of these compression methodologies in real-world settings.

Author Contributions: Conceptualization, A.D. and H.S.; methodology, A.D., H.S., B.R.; software, A.D., B.R; validation, A.D., H.S. and M.P.; formal analysis, M.P.; investigation, A.D., H.S., B.R.; resources, H.S.; writing—original draft preparation, A.D., H.S., B.R; writing—review and editing, K.S., H.S., M.P; visualization, A.D., H.S., B.R, M.P; supervision, A.D, H.S, M.M.; project administration, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-10-2025-034 dd. 19.06.2025, IGK 000000C313925P4D0002).

 doi:10.20944/preprints202507.0074.v1

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: This work was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-10-2025-034 dd. 19.06.2025, IGK 000000C313925P4D0002).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dantas, P.V.; Sabino da Silva Jr, W.; Cordeiro, L.C.; Carvalho, C.B. A comprehensive review of model compression techniques in machine learning. *Applied Intelligence* **2024**, pp. 1–41.
2. Bai, T.; Luo, J.; Zhao, J.; Wen, B.; Wang, Q. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356* **2021**.
3. Carlini, N.; Athalye, A.; Papernot, N.; Brendel, W.; Rauber, J.; Tsipras, D.; Goodfellow, I.; Madry, A.; Kurakin, A. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705* **2019**.
4. Villegas-Ch, W.; Jaramillo-Alcázar, A.; Luján-Mora, S. Evaluating the Robustness of Deep Learning Models against Adversarial Attacks: An Analysis with FGSM, PGD and CW. *Big Data and Cognitive Computing* **2024**, *8*, 8.
5. Ye, S.; Xu, K.; Liu, S.; Cheng, H.; Lambrechts, J.H.; Zhang, H.; Zhou, A.; Ma, K.; Wang, Y.; Lin, X. Adversarial robustness vs. model compression, or both? In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 111–120.
6. Wang, S.; Zhao, J.; Ta, N.; Zhao, X.; Xiao, M.; Wei, H. A real-time deep learning forest fire monitoring algorithm based on an improved Pruned+ KD model. *Journal of Real-Time Image Processing* **2021**, *18*, 2319–2329.
7. Alkhulaifi, A.; Alsahli, F.; Ahmad, I. Knowledge distillation in deep learning and its applications. *PeerJ Computer Science* **2021**, *7*, e474.
8. Cai, M.; Su, Y.; Wang, B.; Zhang, T. Research on compression pruning methods based on deep learning. In Proceedings of the Journal of Physics: Conference Series. IOP Publishing, 2023, Vol. 2580, p. 012060.
9. Shafique, M.A.; Munir, A.; Kong, J. Deep Learning Performance Characterization on GPUs for Various Quantization Frameworks. *AI* **2023**, *4*, 926–948.
10. Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In Proceedings of the International conference on learning representations, 2019.
11. Kannan, H.; Kurakin, A.; Goodfellow, I. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373* **2018**.
12. Liu, D.; Wu, L.; Zhao, H.; Boussaid, F.; Bennamoun, M.; Xie, X. Jacobian norm with selective input gradient regularization for improved and interpretable adversarial defense. *arXiv preprint arXiv:2207.13036* **2022**.
13. Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; Madry, A. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems* **2019**, *32*.
14. Ortiz-Jiménez, G.; Modas, A.; Moosavi-Dezfooli, S.M.; Frossard, P. Optimism in the face of adversity: Understanding and improving deep learning through adversarial robustness. *Proceedings of the IEEE* **2021**, *109*, 635–659.
15. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* **2014**.
16. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In *Artificial intelligence safety and security*; Chapman and Hall/CRC, 2018; pp. 99–112.
17. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 ieee symposium on security and privacy (sp). Ieee, 2017, pp. 39–57.
18. Mądry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *stat* **2017**, *1050*.
19. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE symposium on security and privacy (SP). IEEE, 2016, pp. 582–597.
20. Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; Zhu, J. Defense against adversarial attacks using high-level representation guided denoiser. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1778–1787.

 doi:10.20944/preprints202507.0074.v1

21. Guo, C.; Rana, M.; Cisse, M.; Van Der Maaten, L. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117* **2017**.
22. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204* **2017**.
23. Han, C.; Wang, L.; Li, D.; Cui, W.; Yan, B. A Pruning Method Combined with Resilient Training to Improve the Adversarial Robustness of Automatic Modulation Classification Models. *Mobile Networks and Applications* **2024**, pp. 1–17.
24. Wang, L.; Ding, G.W.; Huang, R.; Cao, Y.; Lui, Y.C. Adversarial robustness of pruned neural networks **2018**.
25. Jordao, A.; Pedrini, H. On the effect of pruning on adversarial robustness. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1–11.
26. Wijayanto, A.W.; Choong, J.J.; Madhawa, K.; Murata, T. Towards robust compressed convolutional neural networks. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2019, pp. 1–8.
27. Jian, T.; Wang, Z.; Wang, Y.; Dy, J.; Ioannidis, S. Pruning adversarially robust neural networks without adversarial examples. In Proceedings of the 2022 IEEE International Conference on Data Mining (ICDM). IEEE, 2022, pp. 993–998.
28. Kundu, S.; Nazemi, M.; Beerel, P.A.; Pedram, M. A tunable robust pruning framework through dynamic network rewiring of dnns. *arXiv preprint arXiv:2011.03083* **2020**.
29. Gorsline, M.; Smith, J.; Merkel, C. On the adversarial robustness of quantized neural networks. In Proceedings of the Proceedings of the 2021 on Great Lakes Symposium on VLSI, 2021, pp. 189–194.
30. Duncan, K.; Komendantskaya, E.; Stewart, R.; Lones, M. Relative robustness of quantized neural networks against adversarial attacks. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–8.
31. Lin, J.; Gan, C.; Han, S. Defensive quantization: When efficiency meets robustness. *arXiv preprint arXiv:1904.08444* **2019**.
32. Song, C.; Ranjan, R.; Li, H. A layer-wise adversarial-aware quantization optimization for improving robustness. *arXiv preprint arXiv:2110.12308* **2021**.
33. Neshaei, S.P.; Boreshban, Y.; Ghassem-Sani, G.; Mirroshandel, S.A. The Impact of Quantization on the Robustness of Transformer-based Text Classifiers. *arXiv preprint arXiv:2403.05365* **2024**.
34. Li, Q.; Meng, Y.; Tang, C.; Jiang, J.; Wang, Z. Investigating the Impact of Quantization on Adversarial Robustness. *arXiv preprint arXiv:2404.05639* **2024**.
35. Panda, P. Quanos: adversarial noise sensitivity driven hybrid quantization of neural networks. In Proceedings of the Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design, 2020, pp. 187–192.
36. Galloway, A.; Taylor, G.W.; Moussa, M. Attacking binarized neural networks. *arXiv preprint arXiv:1711.00449* **2017**.
37. Song, C.; Fallon, E.; Li, H. Improving adversarial robustness in weight-quantized neural networks. *arXiv preprint arXiv:2012.14965* **2020**.
38. Hinton, G. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* **2015**.
39. Cao, Z.; Bao, Y.; Meng, F.; Li, C.; Tan, W.; Wang, G.; Liang, Y. Enhancing Adversarial Training with Prior Knowledge Distillation for Robust Image Compression. In Proceedings of the ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024, pp. 3430–3434.
40. Rauber, J.; Zimmermann, R.; Bethge, M.; Brendel, W. Foolbox Native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX. *Journal of Open Source Software* **2020**, *5*, 2607. https://doi.org/10.21105/joss.02607.
41. Rauber, J.; Brendel, W.; Bethge, M. Foolbox: A Python toolbox to benchmark the robustness of machine learning models. In Proceedings of the Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning, 2017.
42. Sehwag, V.; Wang, S.; Mittal, P.; Jana, S. Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems* **2020**, *33*, 19655–19666.
43. Chen, P.; Ye, J.; Chen, G.; Zhao, J.; Heng, P.A. Robustness of accuracy metric and its inspirations in learning with noisy labels. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 11451–11461.
44. Goldblum, M.; Fowl, L.; Feizi, S.; Goldstein, T. Adversarially robust distillation. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2020, Vol. 34, pp. 3996–4003.

 doi:10.20944/preprints202507.0074.v1

45. Bernhard, R.; Moellic, P.A.; Dutertre, J.M. Impact of low-bitwidth quantization on the adversarial robustness for embedded neural networks. In Proceedings of the 2019 International Conference on Cyberworlds (CW). IEEE, 2019, pp. 308–315.
46. Chen, T.; Zhang, Z.; Liu, S.; Chang, S.; Wang, Z. Robust overfitting may be mitigated by properly learned smoothening. In Proceedings of the International Conference on Learning Representations, 2020.