AI Resilience: A Revolutionary Benchmarking Model for AI Safety

Khushbakht Khan
Dept of Software Engineering
NED University
Karachi, Pakistan
khushbakhtkhan1@gmail.com

Sarah Sami
Dept of Software Engineering
NED University
Karachi, Pakistan
sarahsk002@gmail.com

Syed Aun Muhammad
Dept of Software Engineering
NED University
Karachi, Pakistan
auna7472@gmail.com

Moiz Naveed
Dept of Software Engineering
NED University)
Karachi, Pakistan
naveedmoiz928@gmail.com

***Abstract*—The security and trustworthiness of AI systems is undeniably impacted by adversarial attacks. In this paper we propose AI Resilience model a convolutional neural network (CNN) [1] classification AI on CIFAR-10 dataset where adversarial examples used for training augmenting the dataset using Fast Gradient Sign Method (FGSM) [2] created digitally. We focus on comparing its performance to the state-of-the-art adversarially trained ResNet-18 model from RobustBench [8] as well as other models from the literature. Our evaluation is based on three criteria: accuracy and robustness against adversarial examples and provided input, and overall strength to manipulation score. Experiment results have uncovered that the pretrained model indeed outperformed the counterpart in default accuracy metric, while the counterpart was more robust to adversarial examples. These outcomes demonstrate the conflicting requirements of optimal performance on uncontaminated data, and withstand provided data that has subtle but intentional adversarial changes, which is critical to consider when integrating AI systems into safety-sensitive systems.**

***Index Terms*—AI Resilience, Adversarial Attacks, FGSM, RobustBench, Convolutional Neural Network, Adversarial Training, AI Safety**

I. Introduction

The importance of maintaining safety becomes increasingly critical when implementing AI technologies, especially in systems designated for use in safety critical applications [5]. With these implementations comes the risk of encountering malicious inputs and adversarial attacks which significantly influence data and model prediction [4]. AI resilience pertains to the capacity of a model when its performance tends to be affected by perturbing influences. This work introduces an AI Resilience model that employs adversarial training with FGSM [2] to increase robustness and evaluates it against a pretrained ResNet-18 model available in RobustBench [8]. Besides, we discuss other related works and present a comparative analysis of diverse models in relation to silenced precision, precision under hostile scrutiny, and the overall strength rating [19].

II. Methodology

A. Model Architectures

*1) **Custom AI Resilience Model**:*

- **Architecture**: A CNN with two convolutional layers (32 and 64 filters), two pooling layers, and two fully connected layers (mapping features to 128 neurons and then to 10 output classes)
- **Adversarial Training**: The model is trained on both clean images and adversarial examples generated using FGSM (with = 0.1) [2]. This dual training strategy aims to force the model to learn robust features that are less sensitive to small perturbations [3].

*2) **Pretrained RobustBench Model**:*

- **Architecture**: A ResNet-18 model, adversarially trained under the Linf threat model on CIFAR-10 [8].
- **Performance**: Known for very high clean accuracy (typically above 90) but often shows significant degradation when attacked with strong adversarial examples [11].

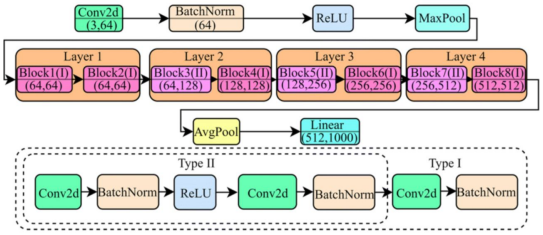

Fig. 1: architecture of resnet18 model

B. Adversarial Attack and Evaluation Metrics

For a fair comparison, both models are evaluated using the same adversarial attack method—the Fast Gradient Sign Method (FGSM) with = 0.1 [2]. The evaluation metrics are defined as follows:

- **Clean Accuracy**: The percentage of unaltered test images correctly classified
- **Adversarial Accuracy**: The percentage of adversarially perturbed test images correctly classified.
- **Robustness Score**: The percentage of predictions that remain unchanged before and after the adversarial attack [19]

III. Attack Libraries and Defense Strategy

For our implementation, we utilize two complementary adversarial-attack libraries: IBM's Adversarial Robustness Toolbox (ART) and the torchattacks package, which allows us to implement FGSM perturbation generation on PyTorch easily. ART exposes a symmetric interface (FastGradientMethod) for class-based FGDOE NumPy and PyTorch classifiers, which allows us to use the same attack patterns that were applied in RobustBench ResNet-18 baseline for our model. On the other hand, forgeattacks provide a light-weight and efficient FGSM implementation that we can call directly from our model during training, thus not increasing the cost of producing adversarial examples during the capture process. Using these two libraries, we achieve consistent attack generation during training through torchattacks and during evaluation through ART, enabling the clean and adversarial losses to be combined without modification.

To counter these FGSM attacks, our primary robustness lever is adversarial training, where ResilienceCNN is employed against such attacks. At every step of training, we first create perturbed samples with $\varepsilon = 0.1$ and compute a joint loss as the cross-entropy on the clean and adversarial batches (on both).It restricts in this manner guides the early convolutional filters of the network to learn feature representations which are robust to minimal worst-case perturbations, essentially regularizing the model to decision boundaries which are very hard to shift by gradient-based attacks. We note, this training strategy tends to generate smoother feature maps within the second convolutional layer, and higher-capacity decision boundaries within the fully connected layers as demonstrated by the lower gradient magnitudes of the data points which are under attack.

In addition, our implementation also augments the training with standard image augmentations such as random horizontal flips and rotations. This augmentation serves to adversarial training by providing the model with a broader range of natural variations and thus, improves its ability to adapt when faced with new changes. The ResilienceCNN using both types of spatial augmentations along with adversarial perturbations into the same training loop builds resilience on multiple levels: through augmentation, spatial invariances are learned, and through FGSM, adversarial invariances are learned. This synergy is the reason why the model performs well on clean and perturbed CIFAR-10 test samples.

IV. Related Work

In the last ten years, different approaches have been developed to improve the resilience of neural networks to adversarial attacks. **Adversarial training** remains the most popular method, introduced by Madry et al., where models are provided with 'adversarial' examples (like FGSM, PGD) during training by adding them to the loss function, resulting in better performance under gradient based attacks. Improvement based methods like **TRADES** feature a trade-off term that allows optimization of clean accuracy versus adversarial robustness, achieving a balance that undermines both goals while maintaining a tunable compromise.

Simultaneously, **Defensive Distillation** uses softened class probabilities to hide gradient information, rendering some classification tasks more difficult, and thwarting effective perturbation calculation for attackers, though this is ineffective against adaptive attacks. **Adversarial Weight Perturbation (AWP)** type techniques treat model parameters like training adversarially. These modify weights to the extent they simulate perturbations, thus resulting in smoother loss landscapes. All of these approaches vary in their balance of implementation difficulty, computational requirements, and empirical performance – which we directly compare in the table below.

Model/Method	Approach	Strengths	Weaknesses
Adversarial Training [3], [7]	Training on both clean and adversarial examples (e.g., FGSM, PGD)	Improves robustness significantly; easy to implement	Often causes a drop in clean accuracy
TRADES [12]	Balances between clean accuracy and adversarial robustness using a trade-off parameter	Achieves a better balance between clean and adversarial performance	Computationally expensive; requires tuning the trade-off parameter
Defensive Distillation [13]	Uses softened labels during training to reduce model sensitivity	Reduces gradient information available to attackers	Can be circumvented by strong attacks
Adversarial Weight Perturbation (AWP) [9]	Perturbs model weights during training to simulate adversarial conditions	Enhances generalization and robustness	Complexity increases and can slow down training

Our work builds on these ideas by training a custom CNN with adversarial examples and directly comparing its performance with a RobustBench ResNet-18 model under identical attack conditions [8]

V. Implementation Details

A. Custom AI Resilience Model Implementation

We've developed and assessed the custom AI Resilience model solely using the PyTorch framework and two libraries focused on adversarial attacks. First, we set up **torch attacks**, which has a class FGSM that directly interfaces with our ResilienceCNN, providing a lightweight implementation of FGSM, and IBM's **Adversarial Robustness Toolbox (ART)**, which we use later for evaluation.

- **Data:**
  We undertake training utilizing the CIFAR-10 dataset, subjecting it to standard image augmentations which include random horizontal flips alongside rotations of up to 15° to simulate natural variations containing with the network. These spatial transformations broaden the diversity of the model's training data and help the model develop defenses against adversarial attacks [16].

- **Model:**
  The ResilienceCNN architecture has two 3x3 convolutional layers with 32 and 64 filters, along with a MaxPooling layer of 2x2, a fully connected layer with 128 neurons followed by softmax onto 10 output classes. Every forward pass, ReLU nonlinearities are applied to the activations, which is followed by: pooling layers, flattening, dense layers, and lastly producing class logits.

- **Adversarial Training:**
  In every iteration of the training process, we create FGSM perturbations ($\varepsilon = 0.1$) for each batch with torch attacks and calculate the loss from the clean and adversarial examples.we saw that the convolutional filters self train in feature extraction through the backpropagation process by using Adam optimizer (learning rate = 0.001) for five epochs. The model's weights are then saved into a file called resilience_model.pth based on [2][7] references.

- **Evaluation:**
  To rigidly and reproducibly evaluate the neural network, it will be wrapped with ART's

PyTorchClassifier and loaded onto the classifier. Afterwards, the Fast Gradient Method will be used to create adversarial example test samples for 1170 iterations which will compute for clean accuracy, adversarial accuracy, and robustness score which measures how many times the model's predictions will not be changed by the attack.

The results are shown in three metrics: clean accuracy, adversarial accuracy, and the robustness score which shows the prediction invariance through illustration in figure 1 while tracking the shift from the standard performance versus resistance adversarial resilience for the model [17] .

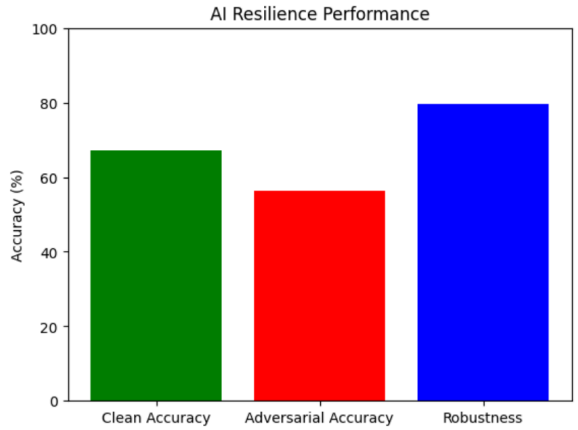

Fig. 2: Accuracy and Performance

B. Pretrained RobustBench Model Implementation

For fair baseline comparison, the adversarially trained ResNet-18 from RobustBench under the same conditions will be evaluated.

- **Data:**
  As described in section 4.1, there are no additional augmentations to the CIFAR-10 test set but it will be used as an underlying benchmark. [16].

- **Model:**
  The ResNet-18 Architecture will be downloaded with the adversarial Linf threat model trained and opened with a program called PyTorch.[8].

- **Evaluation:**
  The ResNet-18 model will be wrapped up in a classifier and trained with ART PyTorch together with the custom CNN. Following step [10], we calculate clear-sighted accuracy and adversarial accuracy alongside the robustness score, as before, after creating FGSM adversarial examples with $\varepsilon = 0.1$. The comparative metrics are also represented in Figure 2 which illustrates how ResNet-18 maintains its competitive edge in clean-image performance while being susceptible to small, well-defined alterations.

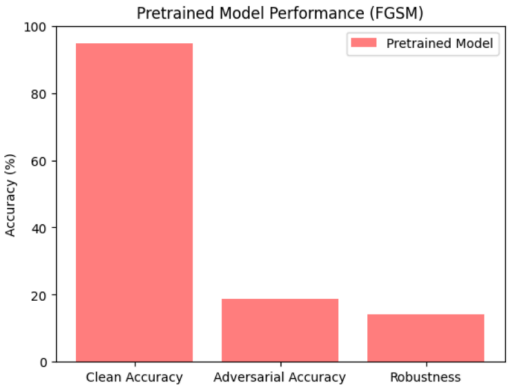

Fig. 3: Pretrained Model Performance and Accuracy

VI. Results

A. Experimental Results

Metric	Custom AI Resilience Model	Pre Trained RobustBench Model
Clean Accuracy	68%	94.78% [11]
Adversarial Accuracy	57%	18.75%
Robustness Score	80%	14.06%

TABLE 2: Comparison of Performance

B. Discussion of Result

- **Clean Accuracy:**
  The RobustBench model shows good performance on clean images because of the training and ResNet-18 architecture [11]. Whereas, our custom model trades clean accuracy to enhance robustness. Hence, it shows lower clean accuracy.

- **Adversarial Accuracy:**
  Under attacks specially FGSM attack [7], our model performs better showcasing its ability to resist deviations. It maintains accuracy.

- **Robustness Score:**
  The robustness score of our model indicates that, even under attack the model stays stable and provides correct predictions as shown in the studies [19]. As compared to the pretrained whose low robustness score indicates instability when performing predictions.

VI. Conclusion

This study brings to light that even though the model pre-trained on ResNet-18 achieves high clean accuracy, it is very propane to FGSM attacks [4], [18]. While our custom AI Resilience model provides remarkable adversarial accuracy and robustness score because it is trained with FGSM examples, it knows how to handle those attacks, covering the gap in the pre-trained model. This makes the model suitable for safety-critical use cases. The trade-off between clean performance and resilience needs to be noted while checking the metrics for evaluation of AI safety [12][20].

VII. Future Work

The results highlight the effectiveness of our ResilienceCNN, which undergoes drastic improvements under single-step FGSM attacks, while simultaneously suggesting more advanced defenses would be beneficial. A notable mention is the efficacy of going beyond single-step perturbations, adding deeper architectures with rich modifications, or even utilizing ensemble methods designed to strengthen countermeasures against adversarial attacks. We will list four specific avenues for further investigation that emerged from the limitations encountered during our benchmarking analyses.

Future Direction	Description	Expected Benefit
Advanced Adversarial Training (e.g., PGD, TRADES)	Include multi-step attacks or use methods that offer a trade-off between clean and adversarial accuracy.	Enhance robustness while maintaining clean accuracy.
Enhanced Data Augmentation (Mixup, CutMix)	Use advanced techniques to enhance generalization and robustness.	Minimize overfitting and boost resilience against different attacks.
Model Architecture Optimization	Try more complex architectures (e.g., ResNet variants) while applying adversarial training.	Gain higher clean accuracy and increased robustness.
Ensemble Methods	Combine predictions from multiple models to improve overall stability and resistance to adversarial examples.	Increase reliability and robustness in safety-critical scenarios.

TABLE 3: Future Direction

VIII. References

[1] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. ICLR*, 2015.

[3] A. Madry *et al.*, "Towards deep learning models resistant to adversarial attacks," in *Proc. ICLR*, 2018.

[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE S&P*, 2017.

[5] Y. Wang and S. H. Chung, "Artificial intelligence in safety-critical systems: a systematic review," *Ind. Manage. Data Syst.*, vol. 122, no. 2, pp. 442–470, 2022.

[6] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, 2018.

[7] F. Tramèr *et al.*, "Ensemble adversarial training," in *Proc. ICLR*, 2018.

[8] F. Croce and M. Hein, "RobustBench: a standardized adversarial robustness benchmark," in *Adv. Neural Inf. Process. Syst.*, 2021.

[9] E. Wong, F. R. Schmidt, and Z. Kolter, "Scaling provable adversarial defenses," in *Proc. NeurIPS*, 2018.

[10] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv*, 2017.

[11] S. Gowal *et al.*, "Uncovering the limits of adversarial training against norm-bounded adversarial examples," in *Proc. NeurIPS*, 2020.

[12] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” in *Proc. ICLR*, 2019.

[13] N. Papernot *et al.*, “Distillation as a defense to adversarial perturbations,” in *Proc. IEEE S&P*, 2016.

[14] C. Guo *et al.*, “Countering adversarial images using input transformations,” in *Proc. ICLR*, 2018.

[15] W. Xu *et al.*, “Feature squeezing: Detecting adversarial examples in deep neural networks,” in *NDSS*, 2017.

[16] A. Krizhevsky, *Learning multiple layers of features from tiny images*, CIFAR-10 dataset, 2009.

[17] M.-I. Nicolae *et al.*, “Adversarial Robustness Toolbox v1.0,” IBM Research, 2018.

[18] J. Uesato *et al.*, “Adversarial risk and the dangers of evaluating against weak attacks,” in *Proc. ICML*, 2018.

[19] M. Hein and M. Andriushchenko, “Formal guarantees on the robustness of a classifier against adversarial manipulation,” in *Proc. NeurIPS*, 2017.

[20] Y. Dong *et al.*, “Benchmarking adversarial robustness,” in *Proc. IEEE/CVF ICCV*, 2019.