Adversarial Robustness

Drithi Davuluri (B21AI055) G Mukund (B21CS092)

1. Introduction:

Deep learning models have achieved remarkable success across various domains, but they are vulnerable to adversarial attacks, where carefully crafted perturbations to the input data can cause the model to make incorrect predictions. This vulnerability poses a significant threat, especially in safety-critical applications. Consequently, there is a growing need to develop deep learning models that are robust against adversarial attacks, ensuring their reliability and trustworthiness.

In this assignment, we achieved robustness of deep neural networks against four well-known adversarial attacks: Projected Gradient Descent (PGD), Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), and DeepFool. We explored various defensive techniques, including adversarial training, defensive distillation, and adversarial logit pairing (ALP), to improve the models' resilience to these attacks.

2. Dataset and Model Architecture:

We used the MNIST dataset, a widely-used benchmark dataset in the field of computer vision and deep learning. The MNIST dataset consists of 70,000 grayscale images of handwritten digits, with 60,000 images for training and 10,000 images for testing. Each image has a size of 28x28 pixels, and the task is to classify the digit represented in the image into one of the ten classes (0-9).

The deep neural network architecture used for the classification task is a convolutional neural network (CNN) inspired by the LeNet-5 architecture. The model consists of the following layers:

1. Convolutional Layer 1: A convolutional layer with 16 filters of size 5x5, a stride of 1, and a padding of 2, followed by a ReLU activation function.
2. Max Pooling Layer 1: A max-pooling layer with a kernel size of 2x2 and a stride of 2.
3. Convolutional Layer 2: A convolutional layer with 32 filters of size 5x5, a stride of 1, and a padding of 2, followed by a ReLU activation function.
4. Max Pooling Layer 2: A max-pooling layer with a kernel size of 2x2 and a stride of 2.
5. Fully Connected Layer 1: A fully connected layer with 120 units and a ReLU activation function.
6. Fully Connected Layer 2: A fully connected layer with 84 units and a ReLU activation function.

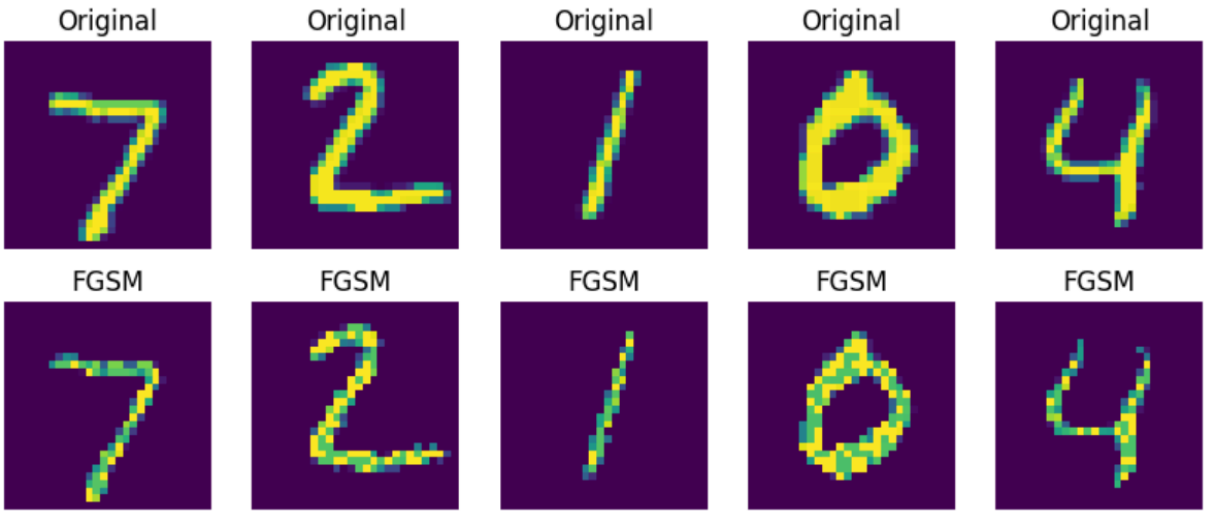
7. Output Layer: A fully connected layer with 10 units (one for each class) and a softmax activation function.

3. Adversarial Attack Implementations:

We implemented four adversarial attacks: Projected Gradient Descent (PGD), Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), and DeepFool. These attacks were used to evaluate the robustness of the deep neural network against adversarial perturbations.

3.1 Projected Gradient Descent (PGD) Attack

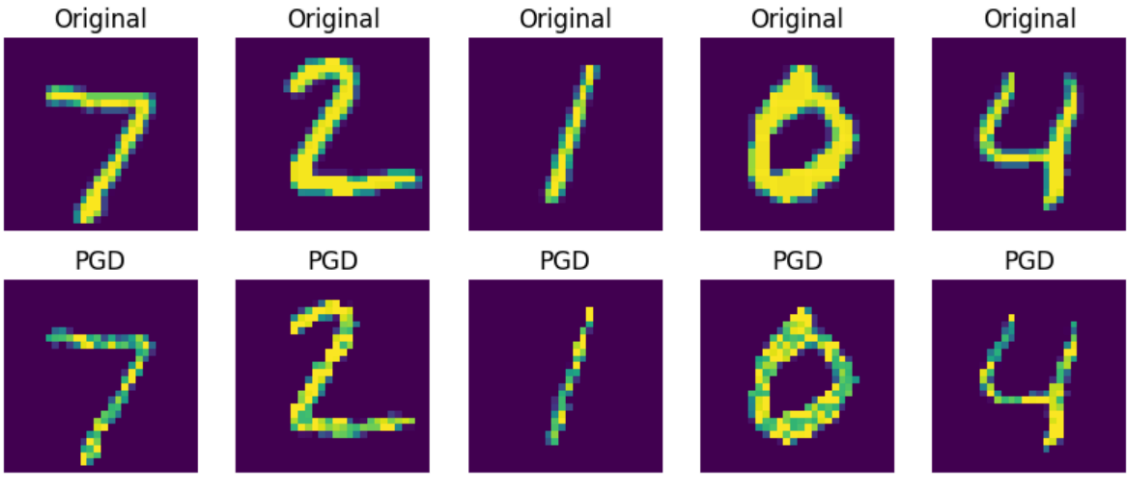
The PGD attack is an iterative method that generates adversarial examples by iteratively perturbing the input data in the direction of the gradient of the loss function, while ensuring that the perturbed data remains within an epsilon-ball around the original input. At each iteration, the attack calculates the gradient of the loss function with respect to the input data and applies a small perturbation in the direction of the gradient. The perturbed data is then clipped to ensure that it remains within the epsilon-ball.



3.2 Fast Gradient Sign Method (FGSM) Attack

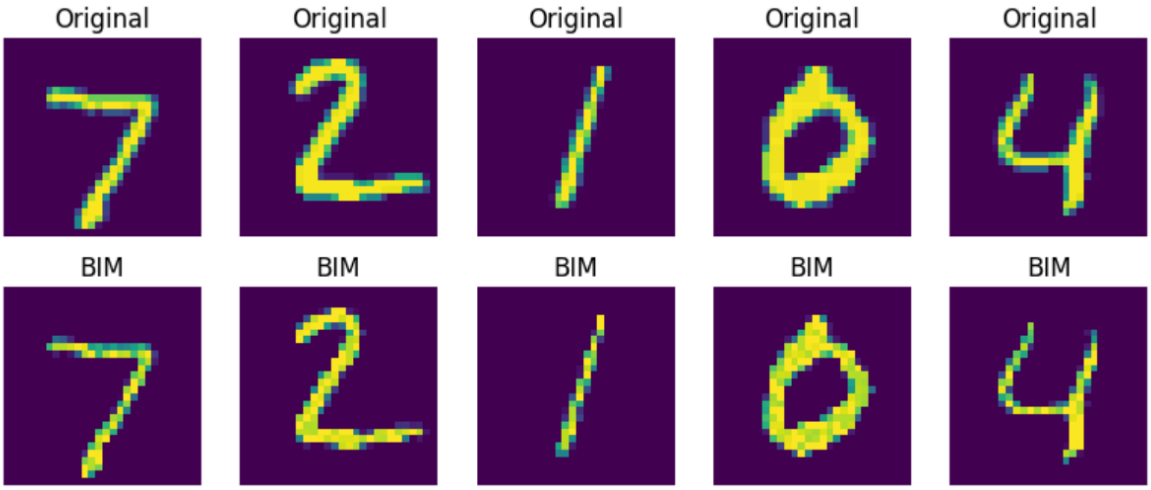
The FGSM attack is a single-step method that perturbs the input data by taking a step in the direction of the gradient of the loss function, scaled by a constant epsilon. The perturbation is calculated by

computing the gradient of the loss function with respect to the input data and then adding or subtracting the epsilon-scaled gradient from the original input data.



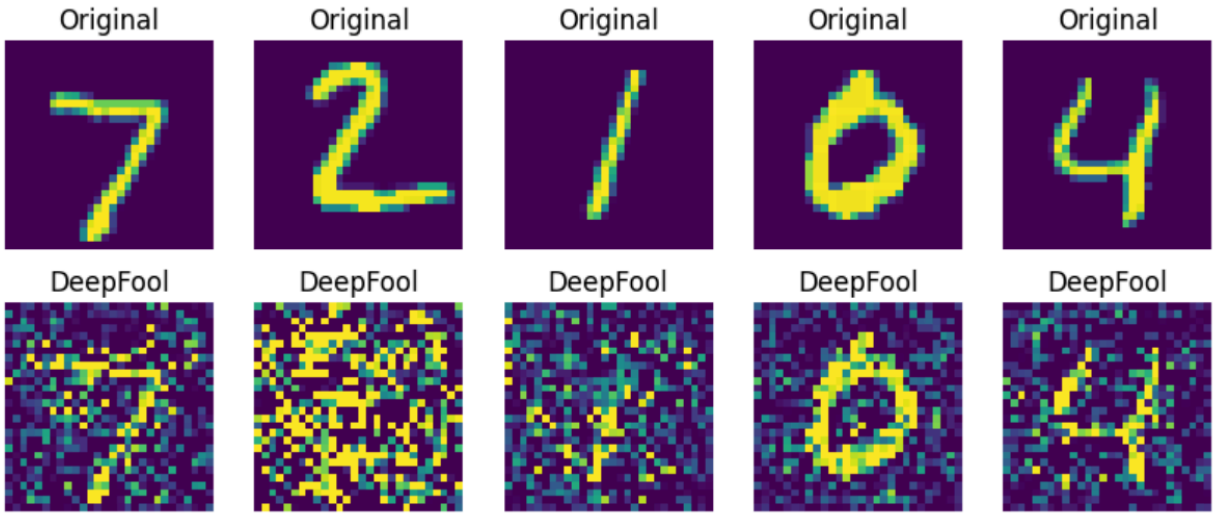
3.3 Basic Iterative Method (BIM) Attack

The BIM attack is an iterative method that incrementally perturbs the input data in the direction of the sign of the gradient of the loss function. At each iteration, the attack computes the gradient of the loss function with respect to the input data and applies a small perturbation in the direction of the sign of the gradient. The perturbed data is then clipped to ensure that it remains within an epsilon-ball around the original input data.



3.4 DeepFool Attack

The DeepFool attack is an iterative attack that aims to find the minimal perturbation required to change the model's prediction. The attack iteratively updates the input data by following the gradient of the model's logits, while ensuring that the perturbation is minimal and sufficient to change the model's prediction.



4. Defensive Techniques:

To improve the robustness of deep neural networks against adversarial attacks, we employed three defensive techniques: Adversarial Training, Defensive Distillation, and Adversarial Logit Pairing (ALP). These techniques aim to enhance the model's ability to resist adversarial perturbations and maintain accurate predictions under attack scenarios.

4.1 Defensive Distillation:

Defensive Distillation is a technique that aims to transfer the knowledge from a teacher model to a student model while improving the student model's robustness against adversarial attacks. The process involves training the student model using a combination of the cross-entropy loss and the knowledge distillation loss, where the knowledge distillation loss encourages the student model to mimic the teacher model's output distribution.

The distillation loss function can be formulated as follows:

$$L\_distillation = \alpha * L\_cross\_entropy + (1 - \alpha) * L\_knowledge\_distillation$$

where L_cross_entropy is the cross-entropy loss between the student model's output and the ground truth labels, L_knowledge_distillation is the knowledge distillation loss (e.g., Kullback-Leibler divergence) between the student model's output distribution and the teacher model's output distribution, and α is a weighting factor that balances the two loss terms.

By mimicking the teacher model's behavior, the student model learns to be more robust against adversarial attacks, as the teacher model is typically trained using adversarial training or other robust techniques.

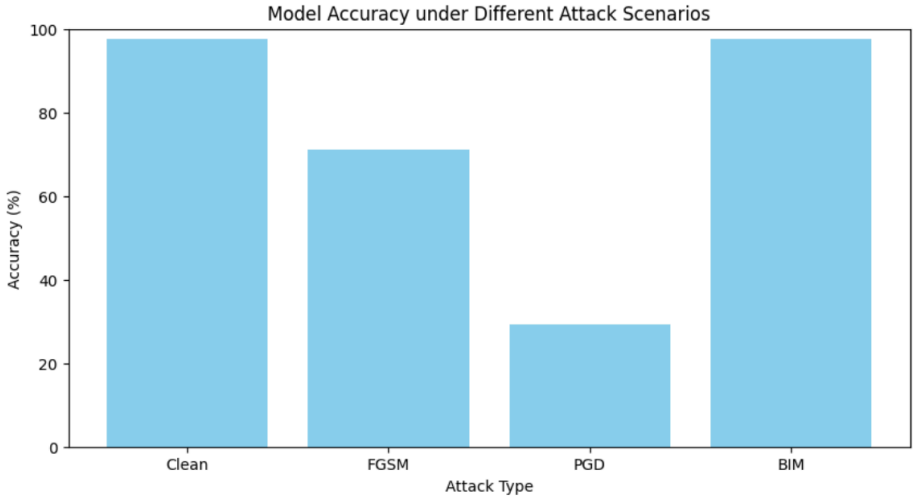
Accuracy

After applying the defensive distillation technique, the student model achieved an accuracy of 97.65% on clean, unperturbed data from the MNIST test set.

The table summerizes student model's accuracy under different adversarial attacks after applying the defensive distillation technique.

Attack Type	**Accuracy**
FGSM	71.22%
PGD	29.32%
BIM	97.65%

As shown in the table, the defensive distillation technique improved the model's robustness against the BIM attack, with an accuracy of 97.65%. However, the model remained vulnerable to the PGD attack, with an accuracy of only 29.32%, and the FGSM attack, with an accuracy of 71.22%.



Adversarial Training:

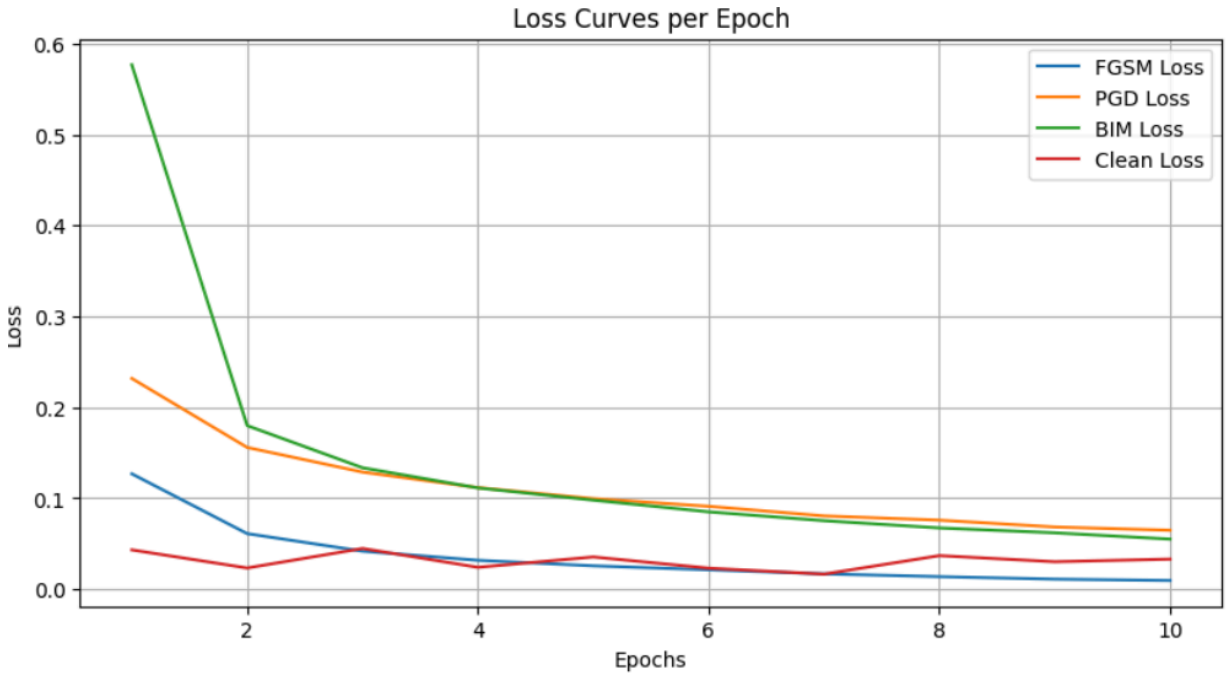
Adversarial Training is a technique that involves explicitly training the deep neural network on adversarial examples generated during the training process. The goal is to improve the model's ability to correctly classify both clean and adversarial examples.

The adversarial training process works as follows:

1. Generate adversarial examples using one or more adversarial attack methods (e.g., PGD, FGSM, BIM) on the current mini-batch of training data.
2. Compute the loss function on the adversarial examples and the corresponding ground truth labels.
3. Update the model's parameters by backpropagating the gradients from the adversarial loss.
4. Repeat steps 1-3 for the remaining mini-batches in the training dataset.

By exposing the model to adversarial examples during training, the model learns to better generalize and become more robust to adversarial perturbations.

Loss Curves during Training



As shown in the figure, the training loss for all three attack types gradually decreased over the epochs, indicating that the model was learning to better handle adversarial examples during the training process.

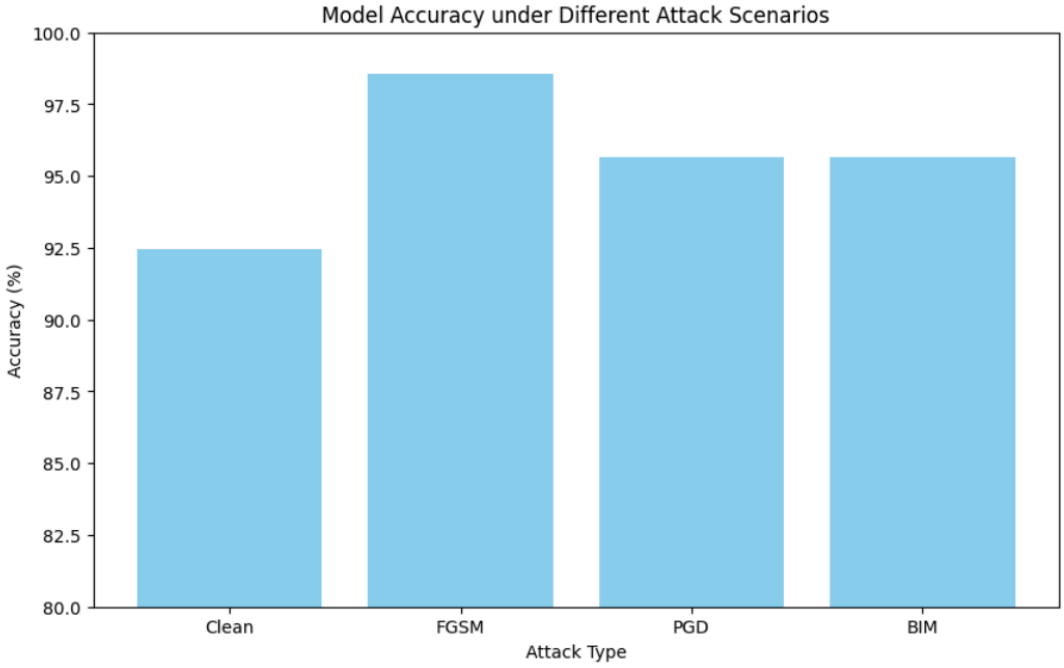
Accuracy

After applying the adversarial training technique, the model achieved an accuracy of 92.44% on clean, unperturbed data from the MNIST test set.

Table summarizes the model's accuracy under different adversarial attacks after applying the adversarial training technique.

Attack Type	**Accuracy**
FGSM	98.55%
PGD	95.68%
BIM	95.68%

As shown in the table, the adversarial training technique significantly improved the model's robustness against all three adversarial attacks. The model achieved an accuracy of 98.55% under the FGSM attack, 95.68% under the PGD attack, and 95.68% under the BIM attack.



Adversarial Logit Pairing (ALP):

Adversarial Logit Pairing (ALP) is a defensive technique that aims to improve the adversarial robustness of deep neural networks by encouraging the model to produce similar logits (the inputs to the final softmax layer) for clean and adversarial examples.

The ALP loss function can be formulated as follows:

$$L\_ALP = L\_cross\_entropy + \lambda * L\_logit\_pairing$$

where L_cross_entropy is the cross-entropy loss between the model's output and the ground truth labels, L_logit_pairing is the mean squared error between the logits of clean examples and their corresponding adversarial examples, and λ is a weighting factor that controls the importance of the logit pairing loss.

The ALP loss encourages the model to produce similar logits for clean and adversarial examples, making it more difficult for adversarial attacks to change the model's prediction by perturbing the input data.

During training, adversarial examples are generated using one or more adversarial attack methods (e.g., PGD, FGSM, BIM), and the ALP loss is computed and back propagated to update the model's parameters.

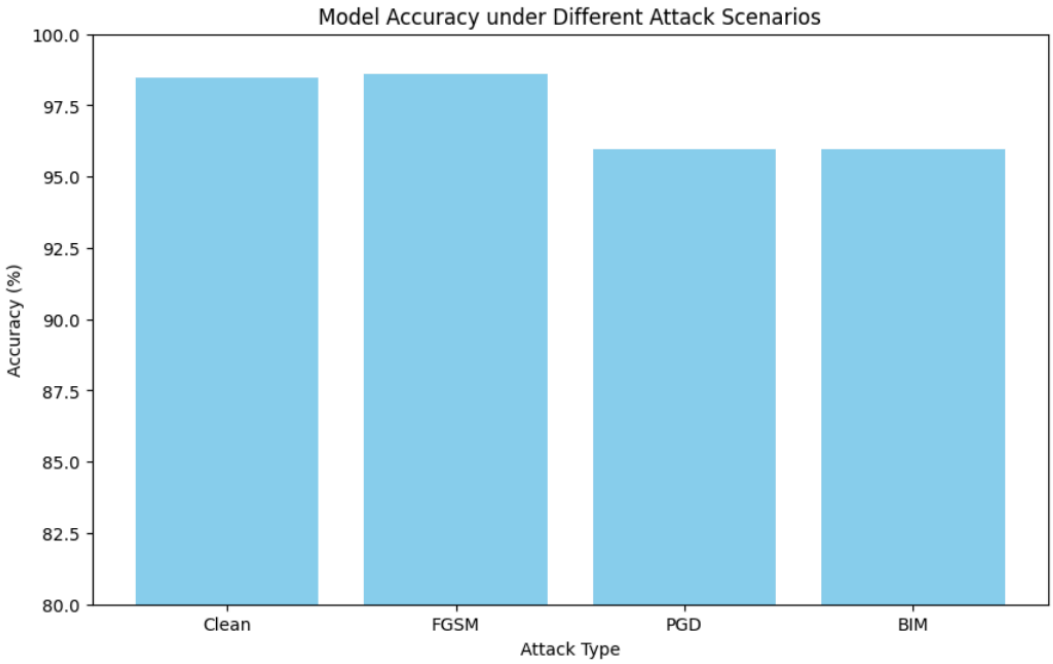
Accuracy

After applying the ALP technique, the model achieved an accuracy of 98.46% on clean, unperturbed data from the MNIST test set.

Table summarizes the model's accuracy under different adversarial attacks after applying the ALP technique.

Attack Type	**Accuracy**
FGSM	98.62%
PGD	95.95%
BIM	95.95%

As shown in the table, the ALP technique significantly improved the model's robustness against adversarial attacks. The model achieved an accuracy of 98.62% under the FGSM attack, 95.95% under the PGD attack, and 95.95% under the BIM attack.



5. Analysis and results

Adversarial training and ALP were the most effective defenses, likely due to explicit training on adversarial examples and encouraging similar logits for clean and adversarial inputs respectively. Defensive distillation provided moderate improvement only against BIM, possibly due to the teacher model's limited robustness.

No single technique provided universal protection, highlighting the need to evaluate against diverse attacks that may exploit different vulnerabilities. Combining defenses could yield better overall robustness.

There was a trade-off between robustness and accuracy on clean data, potentially due to increased regularization limiting the model's capacity to fit clean data tightly.

The promising results with adversarial training and ALP demonstrate their potential in mitigating adversarial risks, but a multi-pronged approach considering diverse attacks is necessary for comprehensive robustness, especially in safety-critical applications.

6. Conclusion

In conclusion, the experimental results demonstrate the effectiveness of adversarial training and adversarial logit pairing in enhancing the robustness of deep neural networks against adversarial attacks like FGSM, PGD, and BIM. However, no single defensive technique provided universal protection, emphasizing the need for evaluating and combining multiple approaches to achieve comprehensive robustness. While defensive techniques can improve adversarial resilience, there is a trade-off between robustness and accuracy on clean data that should be considered based on application requirements. Ultimately, developing robust and trustworthy deep learning models against adversarial threats is crucial, especially for safety-critical applications, and requires a multi-faceted strategy that accounts for diverse attack types.