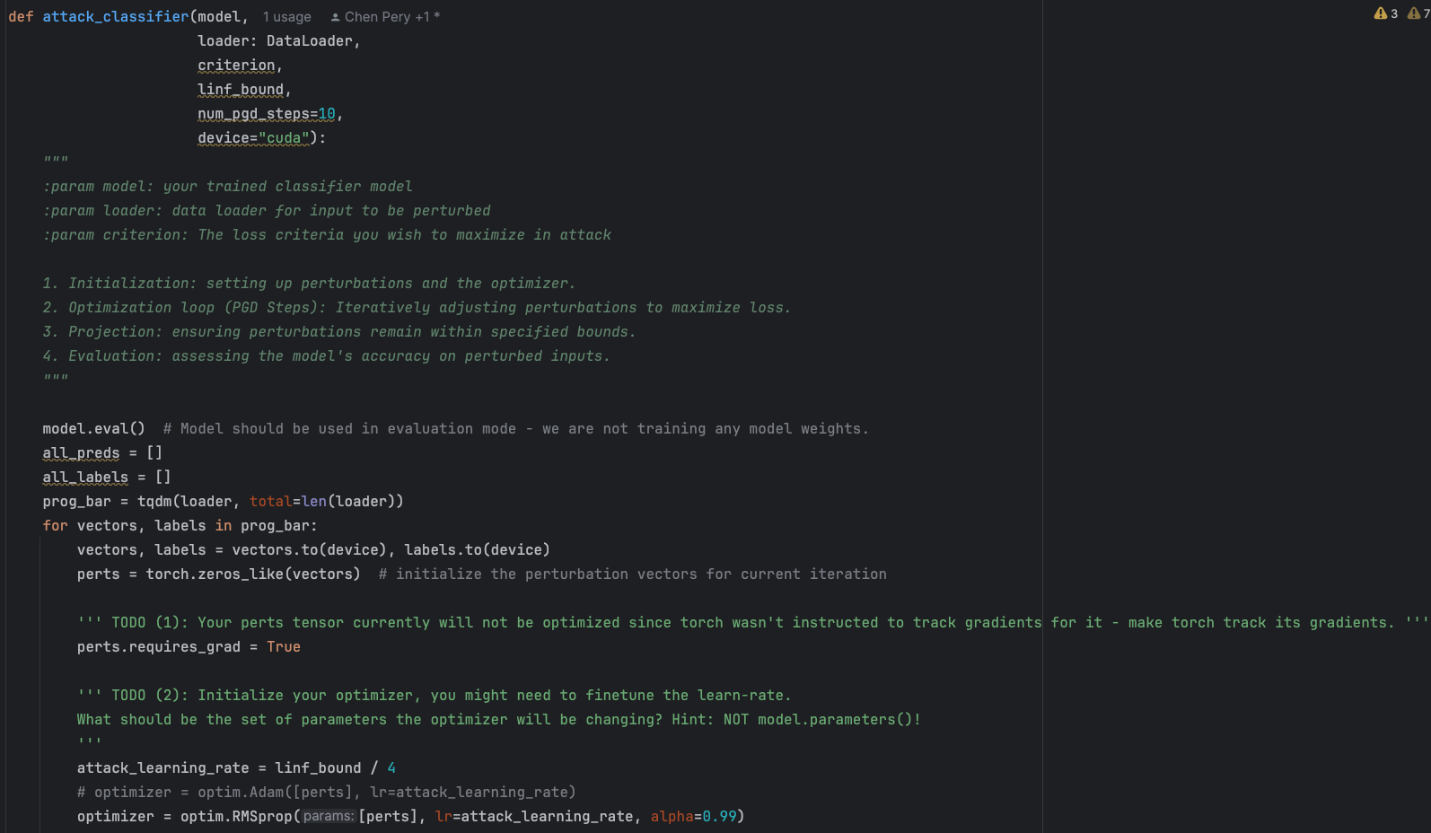
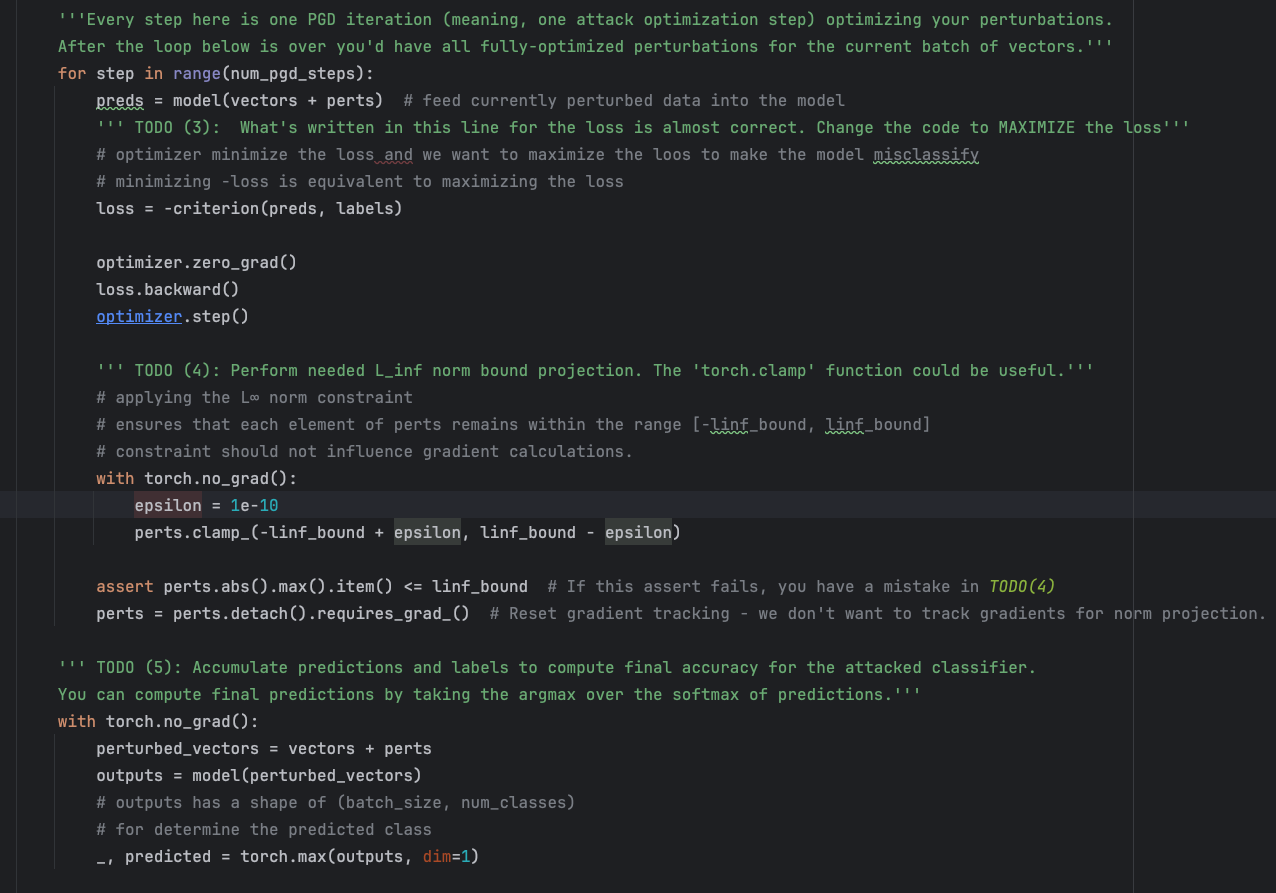
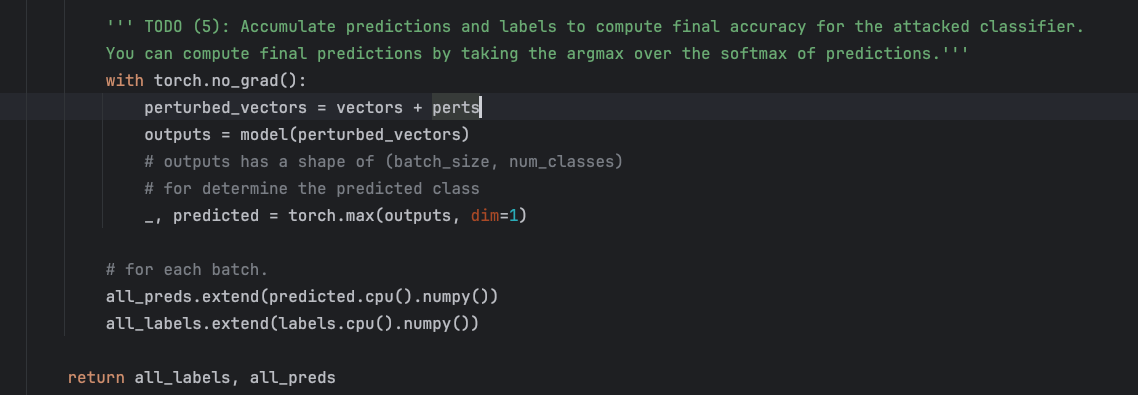
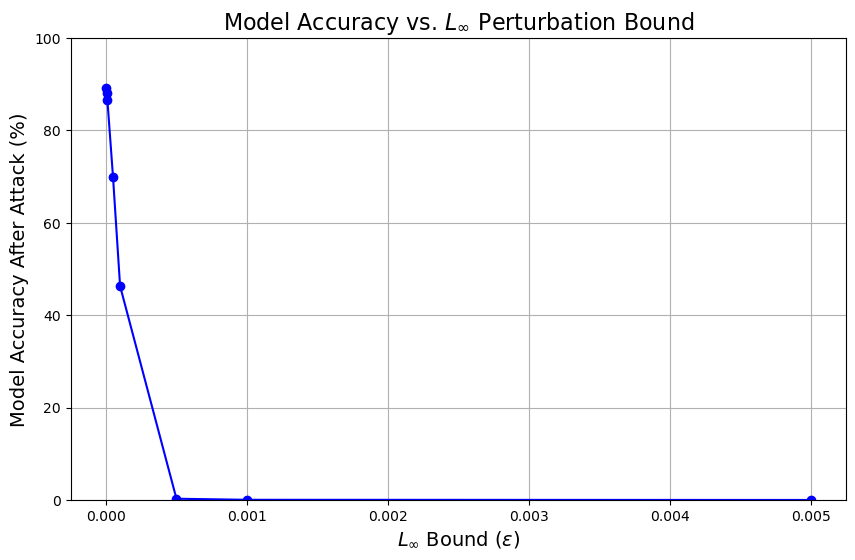
5.

Screenshot of attack\_classifier function in the attack.py:





6.



X-axis represents the perturbation bound (epsilon).

Y-axis shows the classification accuracy of the model after the adversarial attack.

The trend line indicated how accuracy changes as epsilon varies, from this line there is a clear inverse relationship between epsilon and the classification accuracy,

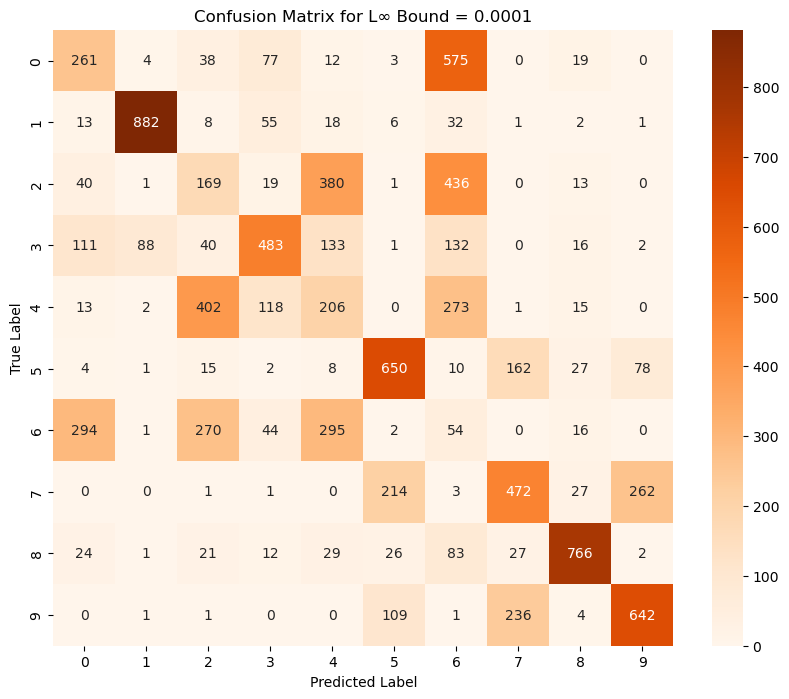
As epsilon increases the model’s accuracy decreases because larger perturbation bounds allow more substantial changes to the input features, making it easier for adversarial examples to cross the model’s decision boundaries and induce misclassifications.

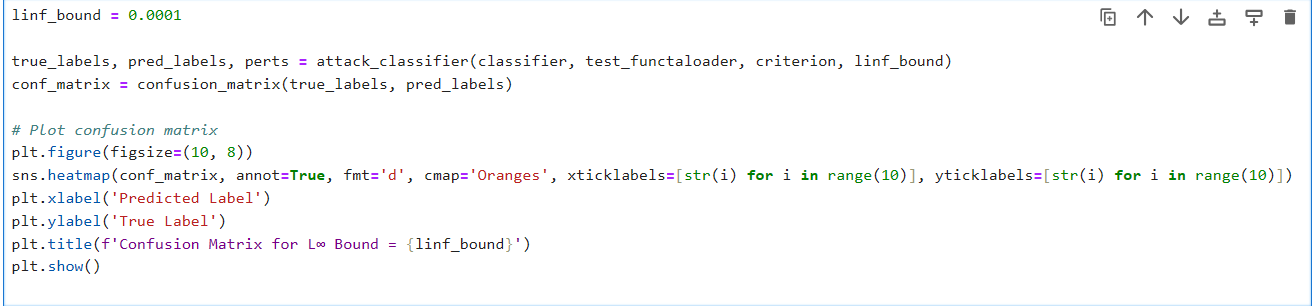
As epsilon decreases the model’s accuracy increases because smaller perturbation bounds restrict the magnitude of the changes, making it harder for adversarial examples to cause misclassifications while keeping the perturbations imperceptible.

**Explanation for Choosing Learning Rate**:

In [Goodfellow et al., 2015](https://arxiv.org/pdf/1412.6572), a single-step perturbation equal to was introduced. For multi-step methods like PGD, it’s common to use a fixed step size as a fraction of to allow perturbations to accumulate without exceeding the limit. Later work, such as in [Madry et al., 2017](https://arxiv.org/pdf/2003.01690), evaluates different step sizes (e.g., ) for optimal adversarial performance across tasks. Many adversarial attack implementations use such fractional step sizes to balance effectiveness and control within multi-step attacks. For our case, we selected based on these practices and empirical performance.

7.





8.

This adversarial confusion matrix highlights that the attack has successfully introduced substantial misclassifications, as seen by the increase in off-diagonal values (incorrect predictions) compared to what we would expect in a clean test set.

Certain classes, such as **true label 0**, show a strong shift in predictions toward other labels, with a significant number of samples being misclassified as class 6 (575 instances). This pattern suggests that the adversarial perturbation has altered the model’s perception of class 0, making it similar in feature space to class 6.

Other notable examples include true labels 2, 4, and 6, which show significant misclassification patterns. For instance:

* **True label 2** has 380 instances misclassified as class 4 and 436 as class 6, indicating that adversarial noise causes substantial confusion, pushing samples from class 2 toward these classes.
* **True label 4** has 402 instances misclassified as class 2 and 273 as class 6.
* **True label 6** is heavily misclassified, with 294 instances as class 0 and 270 as class 2. This consistent misclassification points to weak boundaries for class 6, making it particularly susceptible to being mistaken for these neighboring classes under attack.

This pattern suggests that adversarial perturbations lead to strong overlap between classes 4, 2, and 6.

**True label 3** shows widespread misclassification, with 111 instances incorrectly classified as class 0, 88 as class 1, 133 as class 4, and 132 as class 6. This dispersion reflects a pronounced vulnerability, where class 3 is frequently misidentified as several other classes.

**true label 1 and 8** remain relatively robust under attack; however, there are still notable misclassifications. The diagonal entries for these classes indicate a high number of correct classifications, showing that the model’s perception of these classes remains stable under adversarial perturbations to some extent.

**true labels 5, 7, and 9** demonstrate moderate resilience under attack, but they are still impacted by notable misclassifications.

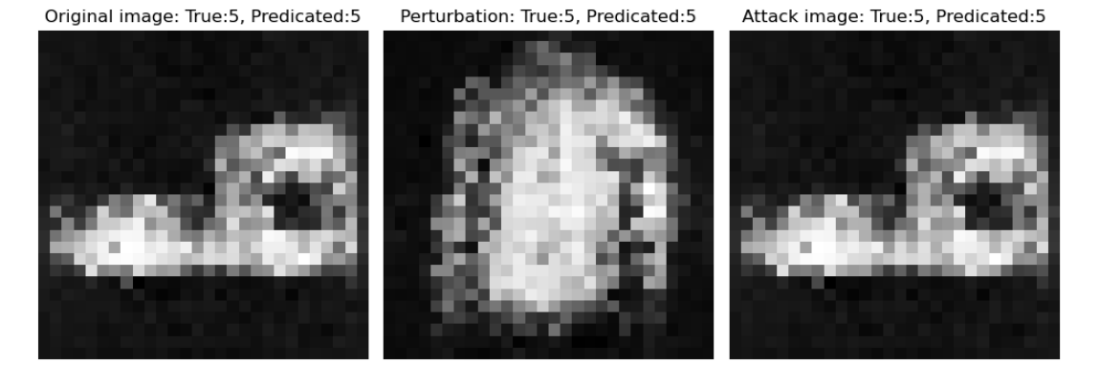
These patterns indicate that the adversarial attack is exploiting specific weaknesses in the model's boundaries, leading to consistent misclassifications in certain classes. Some classes are more vulnerable, with frequent misclassifications into neighboring classes, while others remain relatively robust, showing fewer errors

**Similarities and Differences Compared to the Clean Matrix**

In both the clean and adversarial matrices, some inherent confusion patterns are observed, particularly between classes that share similar features. For example, in the clean matrix, there are occasional misclassifications between classes like true labels 0 and 6 or 4 and 2. These mild overlaps likely arise from similarities in the visual or feature-based cues the model uses, resulting in slight misclassifications even without adversarial influence.

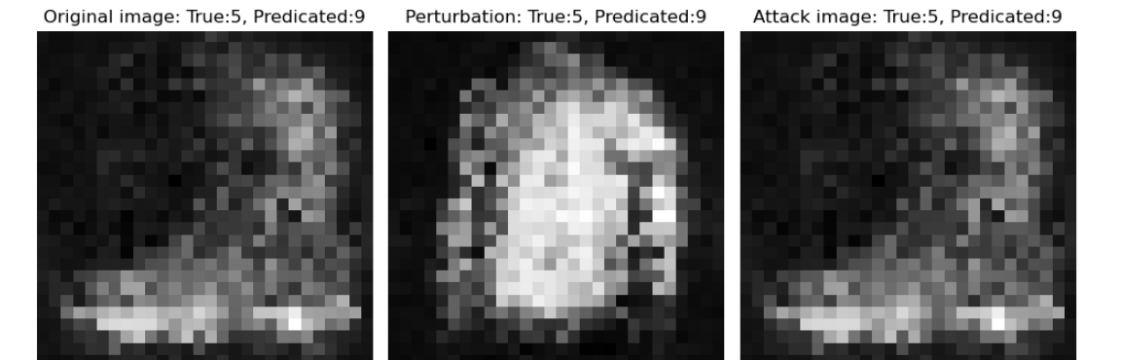
However, the adversarial matrix amplifies these misclassifications considerably, revealing the model’s vulnerabilities under attack. While the clean matrix shows mostly accurate classifications with isolated errors, the adversarial matrix demonstrates a drastic increase in off-diagonal misclassifications. For instance, true label 0 is heavily misclassified as class 6, and true label 3 is frequently misclassified into classes 0, 1, and 4 under attack conditions—patterns that were minimal in the clean case.

This comparison shows that the adversarial attack exploits weaknesses in the model’s decision boundaries, leading to more consistent and severe misclassifications in classes that already had slight confusion in the clean matrix. The clean matrix reflects mostly well-defined boundaries with minor overlaps, while the adversarial matrix exposes and magnifies these weak spots, making certain class boundaries more fragile and prone to errors.

****-

9. we chose classes 5,7,9 for plotting:

9.



## 

