Introduction

This report documents the development, attack simulation and defense of a deep learning model trained for food image classification. The goal was to explore the robustness of deep neural networks under security threats, such as data poisoning and adversarial examples, and to evaluate the effectiveness of different defense strategies. All experiments were conducted using a custom 10-class subset of the **Food-101** dataset and the **ResNet-50** architecture.

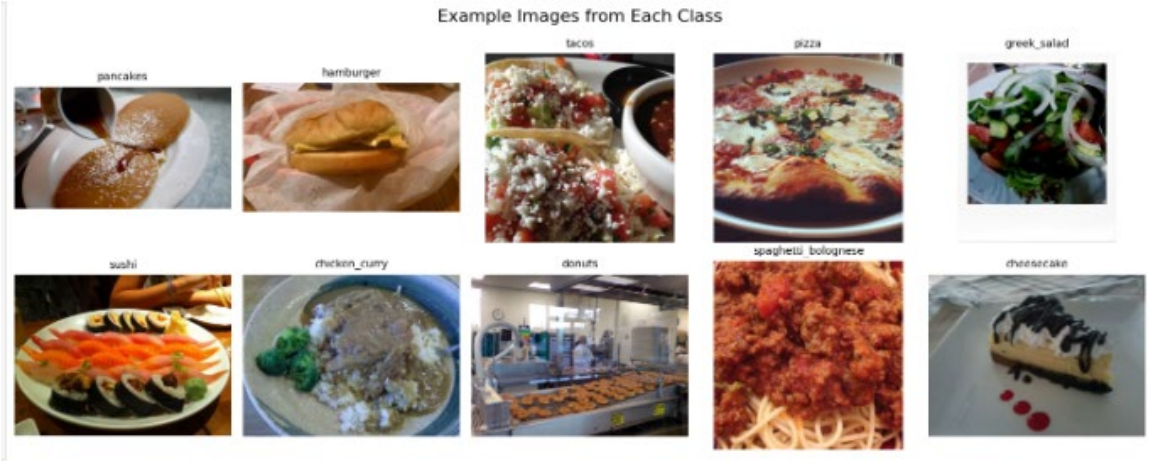
Dataset and Preprocessing

A subset of 10 classes was selected from the Food-101 dataset:

- pizza
- sushi
- hamburger
- spaghetti_bolognese
- cheesecake
- greek_salad
- tacos
- donuts
- pancakes
- chicken_curry

Each class contains **1,000 images**, split **into 70% training, 15% validation, and 15% testing** sets. Images were normalized and augmented using random crops, rotations, and flips to improve generalization. The class distribution is shown below.

```
pizza: 1000 .jpg files found
sushi: 1000 .jpg files found
hamburger: 1000 .jpg files found
spaghetti_bolognese: 1000 .jpg files found
cheesecake: 1000 .jpg files found
greek_salad: 1000 .jpg files found
tacos: 1000 .jpg files found
donuts: 1000 .jpg files found
pancakes: 1000 .jpg files found
chicken_curry: 1000 .jpg files found
```



Baseline Model Training

A ResNet-50 model was fine-tuned on the clean dataset for 5 epochs. The final training and validation accuracy were 87.8% and 88.0%, respectively. The training was performed using the Adam optimizer with a learning rate of 1e-4 and CrossEntropyLoss. The model checkpoint with the highest validation accuracy was saved.

```
Epoch 1
Training: 100%|███| 219/219 [01:39<00:00,  2.19it/s]
Validation: 100%|███| 48/48 [00:14<00:00,  3.21it/s]
Train Acc: 0.7273, Val Acc: 0.8623
Best model saved.

Epoch 2
Training: 100%|███| 219/219 [01:39<00:00,  2.19it/s]
Validation: 100%|███| 48/48 [00:14<00:00,  3.25it/s]
Train Acc: 0.8219, Val Acc: 0.8669
Best model saved.

Epoch 3
Training: 100%|███| 219/219 [01:39<00:00,  2.21it/s]
Validation: 100%|███| 48/48 [00:14<00:00,  3.25it/s]
Train Acc: 0.8526, Val Acc: 0.8921
Best model saved.

Epoch 4
Training: 100%|███| 219/219 [01:41<00:00,  2.17it/s]
Validation: 100%|███| 48/48 [00:15<00:00,  3.11it/s]
Train Acc: 0.8652, Val Acc: 0.8801

Epoch 5
Training: 100%|███| 219/219 [01:40<00:00,  2.18it/s]
Validation: 100%|███| 48/48 [00:15<00:00,  3.13it/s]
Train Acc: 0.8780, Val Acc: 0.8808
```

Poisoning Attack: Label Flipping

To simulate a poisoning attack, **10% of images labeled 'sushi' were relabeled as 'pizza'** in the training set. This introduces confusion in the model's decision boundary. The poisoned model was retrained for 5 epochs. Despite achieving high validation accuracy, evaluation on clean data showed performance degradation.

```
[Poisoned Model] Epoch 1
Training: 100%|███| 219/219 [01:39<00:00,  2.19it/s]
Validation: 100%|███| 48/48 [00:14<00:00,  3.23it/s]
Train Acc: 0.7235, Val Acc: 0.8583

[Poisoned Model] Epoch 2
Training: 100%|███| 219/219 [01:39<00:00,  2.20it/s]
Validation: 100%|███| 48/48 [00:14<00:00,  3.22it/s]
Train Acc: 0.8187, Val Acc: 0.8775

[Poisoned Model] Epoch 3
Training: 100%|███| 219/219 [01:39<00:00,  2.21it/s]
Validation: 100%|███| 48/48 [00:14<00:00,  3.21it/s]
Train Acc: 0.8474, Val Acc: 0.8841

[Poisoned Model] Epoch 4
Training: 100%|███| 219/219 [01:40<00:00,  2.17it/s]
Validation: 100%|███| 48/48 [00:15<00:00,  3.09it/s]
Train Acc: 0.8625, Val Acc: 0.8841

[Poisoned Model] Epoch 5
Training: 100%|███| 219/219 [01:39<00:00,  2.21it/s]
Validation: 100%|███| 48/48 [00:14<00:00,  3.24it/s]
Train Acc: 0.8641, Val Acc: 0.8815
```

Adversarial Attacks

We evaluated adversarial robustness using two attack types:

- FGSM (Fast Gradient Sign Method) with ε = 0.01, 0.05, 0.1
- DeepFool

FGSM perturbs input images in the direction of the gradient, while DeepFool finds minimal perturbations to cross the decision boundary. Both attacks significantly reduced accuracy. DeepFool was particularly powerful but computationally expensive.

FGSM:

```
FGSM ε=0.0: 100%|███| 47/47 [00:24<00:00,  1.88it/s]
FGSM ε=0.0 → Accuracy: 0.8880
FGSM ε=0.01: 100%|███| 47/47 [00:24<00:00,  1.94it/s]
FGSM ε=0.01 → Accuracy: 0.1413
FGSM ε=0.05: 100%|███| 47/47 [00:24<00:00,  1.94it/s]
FGSM ε=0.05 → Accuracy: 0.0647
FGSM ε=0.1: 100%|███| 47/47 [00:24<00:00,  1.95it/s]
FGSM ε=0.1 → Accuracy: 0.0867
```

DeepFool:

```
DeepFool: 100%|█████| 47/47 [34:17<00:00, 43.77s/it]
DeepFool → Accuracy: 0.1580
```

Evaluation Function and Metrics

To measure model performance under various attack scenarios, we implemented a metric evaluation function that calculates Accuracy, Precision, Recall, and F1-score. These metrics were computed using `sklearn.metrics` for each attack setting, including clean, poisoned, and adversarial samples.

Sample code used:

```
def evaluate_metrics(model, dataloader, attack=None, epsilon=None, name='',
max_batches=None):
  model.eval()
  y_true, y_pred = [], []
  ...
  return {
    'Name': name,
    'Accuracy': ..., 'Precision': ..., 'Recall': ..., 'F1-score': ...
  }
```

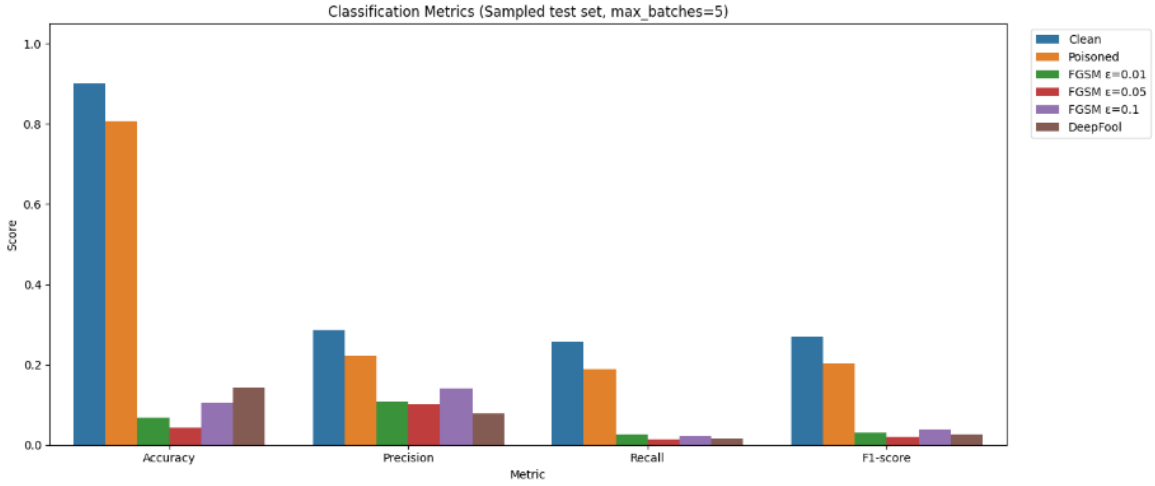
Name	Accuracy	Precision	Recall	F1-score
Clean	0.90000	0.285714	0.257143	0.270677
Poisoned	0.80625	0.222222	0.188889	0.204029
FGSM ε=0.01	0.06875	0.108182	0.026000	0.030298
FGSM ε=0.05	0.04375	0.102381	0.014000	0.020143
FGSM ε=0.1	0.10625	0.141612	0.022963	0.038385
DeepFool	0.14375	0.079310	0.015333	0.025698

This figure presents a comparative evaluation of the classification performance under clean, poisoned, and adversarial attack conditions. The upper table reports four key metrics — accuracy, precision, recall, and F1-score — for each scenario:

- Clean model achieves the highest performance (e.g., accuracy of **90.00%**), serving as the baseline.
- **Poisoned model** shows a performance drop (accuracy drops to **80.63%**), indicating successful data poisoning.

- **FGSM** attacks (**ε = 0.01, 0.05, 0.1**) significantly degrade the model performance. The larger the ε, the more severe the degradation.
- **DeepFool** also substantially reduces the model's confidence and accuracy, with a final accuracy of **14.38%**.

The bar plot below visually reinforces these findings, showing a clear decline in all metrics as attacks are applied. This validates the vulnerability of the poisoned model to adversarial examples and motivates the use of defenses, which are explored in subsequent sections.



Defense Strategy 1: Isolation Forest

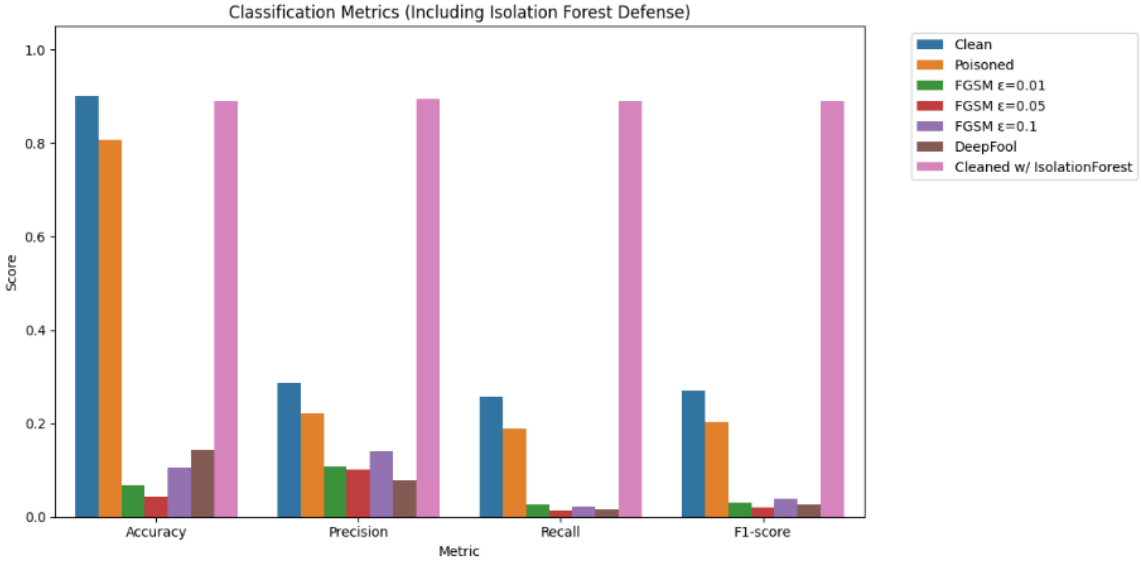
We extracted features using ResNet-50 as a fixed feature extractor and applied Isolation Forest to detect poisoned samples. **699 outliers** were **removed**, and the model was retrained on the cleaned dataset.

- Outliers detected: 699
- Clean samples kept: 6291
- Cleaned dataset size: 6291

The retrained model achieved validation accuracy of **88.1%** and significantly improved performance under adversarial attacks.

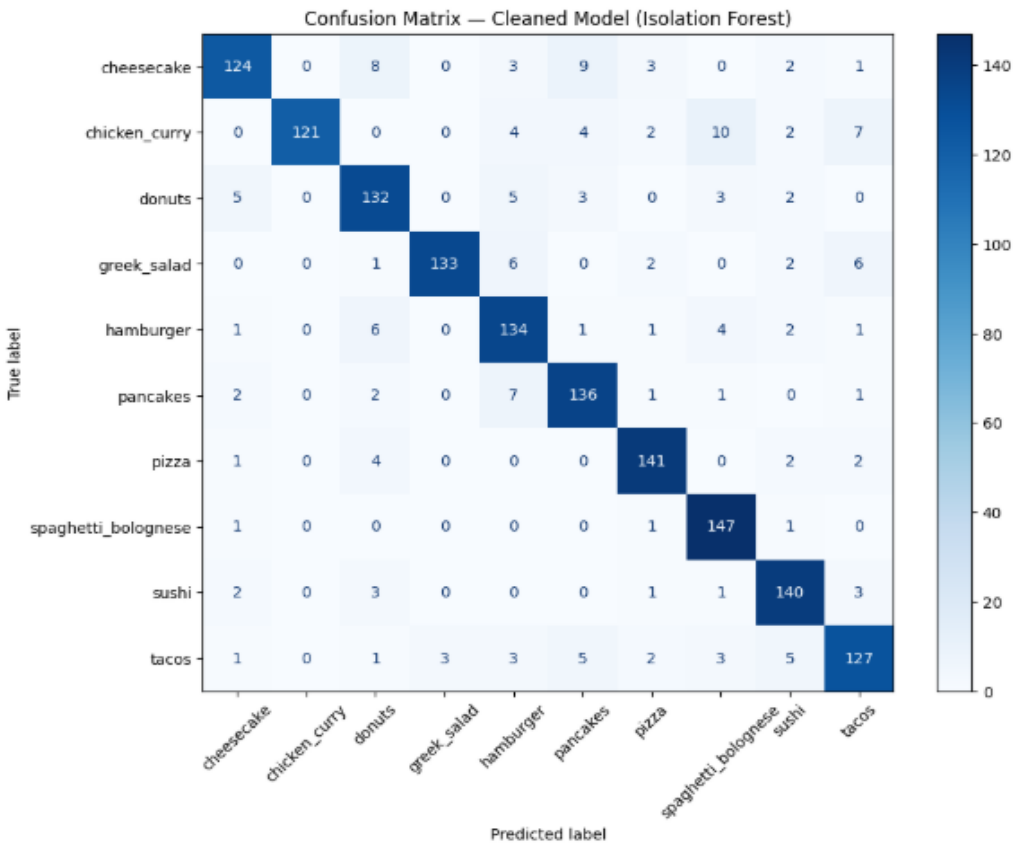
```
[Cleaned Model] Epoch 1
Training: 100%|███| 197/197 [04:06<00:00,  1.25s/it]
Validation: 100%|███| 48/48 [00:42<00:00,  1.12it/s]
Train Acc: 0.7234, Val Acc: 0.8642

[Cleaned Model] Epoch 2
Training: 100%|███| 197/197 [04:04<00:00,  1.24s/it]
Validation: 100%|███| 48/48 [00:41<00:00,  1.15it/s]
Train Acc: 0.8134, Val Acc: 0.8682

[Cleaned Model] Epoch 3
Training: 100%|███| 197/197 [04:09<00:00,  1.27s/it]
Validation: 100%|███| 48/48 [00:42<00:00,  1.12it/s]
Train Acc: 0.8531, Val Acc: 0.8344

[Cleaned Model] Epoch 4
Training: 100%|███| 197/197 [04:06<00:00,  1.25s/it]
Validation: 100%|███| 48/48 [00:41<00:00,  1.15it/s]
Train Acc: 0.8598, Val Acc: 0.8675

[Cleaned Model] Epoch 5
Training: 100%|███| 197/197 [04:06<00:00,  1.25s/it]
Validation: 100%|███| 48/48 [00:41<00:00,  1.15it/s]
Train Acc: 0.8747, Val Acc: 0.8881
```

Classification metrics show that the Isolation Forest-cleaned model performs nearly as well as the clean model, significantly outperforming the attacked versions:



Using Isolation Forest for cleaning the poisoned training data leads to a near-complete recovery of model performance. The metrics (Accuracy, Precision, Recall, F1-score) for the cleaned model are nearly on par with the clean baseline model. In contrast, models exposed to FGSM and DeepFool adversarial attacks experience a dramatic drop in performance.

Confusion matrix of the Isolation Forest-cleaned model demonstrates strong performance with minimal misclassification and good class separation:



The confusion matrix reveals that the model trained on the cleaned dataset correctly classifies most images. Strong diagonal dominance indicates accurate predictions across all classes. Minor confusion exists between visually similar classes (e.g., tacos vs. greek_salad), which is natural.

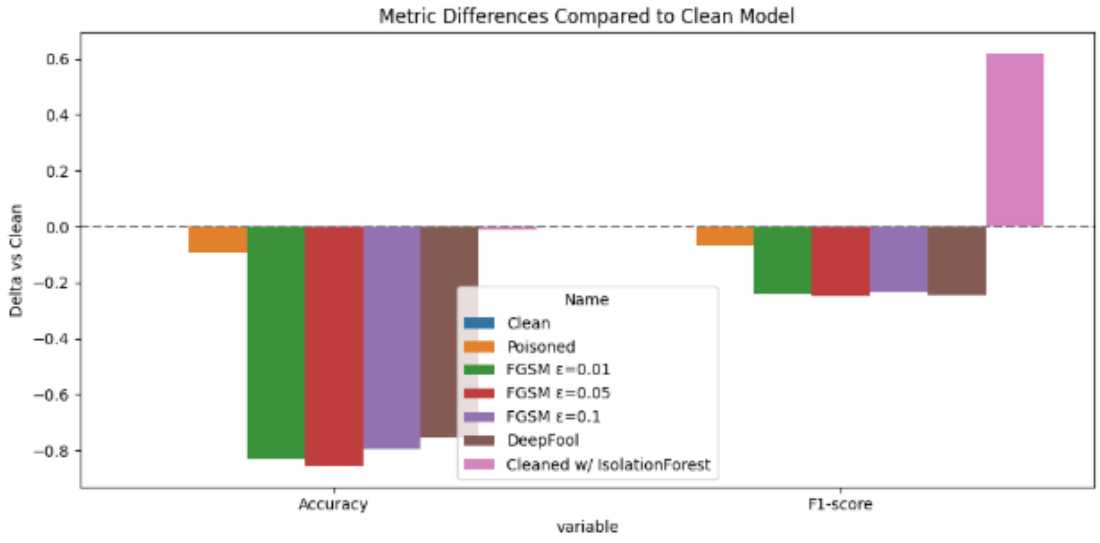
Metric deltas reveal that only the Isolation Forest-cleaned model improves over the baseline, while all other adversarial settings reduce performance:

Delta from Clean Model:

Name	Accuracy	Precision	Recall	F1-score
Clean	0.000000	0.000000	0.000000	0.000000
Poisoned	-0.093750	-0.063492	-0.068254	-0.066648
FGSM ε=0.01	-0.831250	-0.177532	-0.231143	-0.240379
FGSM ε=0.05	-0.856250	-0.183333	-0.243143	-0.250533
FGSM ε=0.1	-0.793750	-0.144102	-0.234180	-0.232292
DeepFool	-0.756250	-0.206404	-0.241810	-0.244978
Cleaned w/ IsolationForest	-0.010000	0.608418	0.632857	0.619313

This table compares each model to the clean baseline by showing differences in evaluation metrics. All adversarial models show negative deltas, indicating performance degradation. The model trained after Isolation Forest filtering shows positive improvements, particularly in precision, recall, and F1-score.

Barplot of metric differences shows how Isolation Forest significantly boosts robustness, outperforming all other defense scenarios.

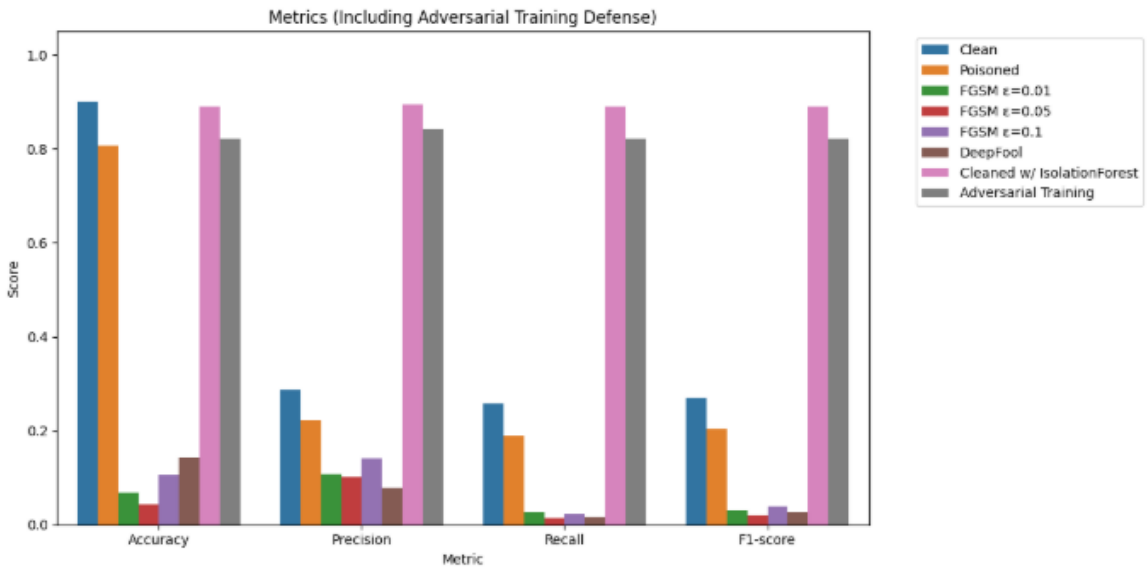


This barplot visually reinforces the table above by showing the magnitude of performance degradation (negative values) for attacked models. The Isolation Forest model shows strong positive gain, especially in F1-score (over +0.6). This confirms its effectiveness in mitigating the impact of poisoning and adversarial attacks.

Defense Strategy 2: Adversarial Training

In this approach, a new model was trained on **FGSM-perturbed inputs** with **ε=0.05**. The adversarially trained model showed improved resistance to both FGSM and DeepFool attacks. While accuracy dropped slightly on clean samples, the robustness gains were evident.

```
[Adversarial Training] Epoch 1
Training: 100%|███| 219/219 [10:20<00:00,  2.83s/it]
Validation: 100%|███| 48/48 [01:06<00:00,  1.38s/it]
Train Acc: 0.5969, Val Acc: 0.8046

[Adversarial Training] Epoch 2
Training: 100%|███| 219/219 [10:17<00:00,  2.82s/it]
Validation: 100%|███| 48/48 [01:04<00:00,  1.35s/it]
Train Acc: 0.7199, Val Acc: 0.8126

[Adversarial Training] Epoch 3
Training: 100%|███| 219/219 [10:08<00:00,  2.78s/it]
Validation: 100%|███| 48/48 [01:03<00:00,  1.32s/it]
Train Acc: 0.7549, Val Acc: 0.8377

[Adversarial Training] Epoch 4
Training: 100%|███| 219/219 [08:48<00:00,  2.41s/it]
Validation: 100%|███| 48/48 [01:03<00:00,  1.32s/it]
Train Acc: 0.7884, Val Acc: 0.8397

[Adversarial Training] Epoch 5
Training: 100%|███| 219/219 [08:48<00:00,  2.41s/it]
Validation: 100%|███| 48/48 [01:03<00:00,  1.32s/it]
Train Acc: 0.8000, Val Acc: 0.8179
```

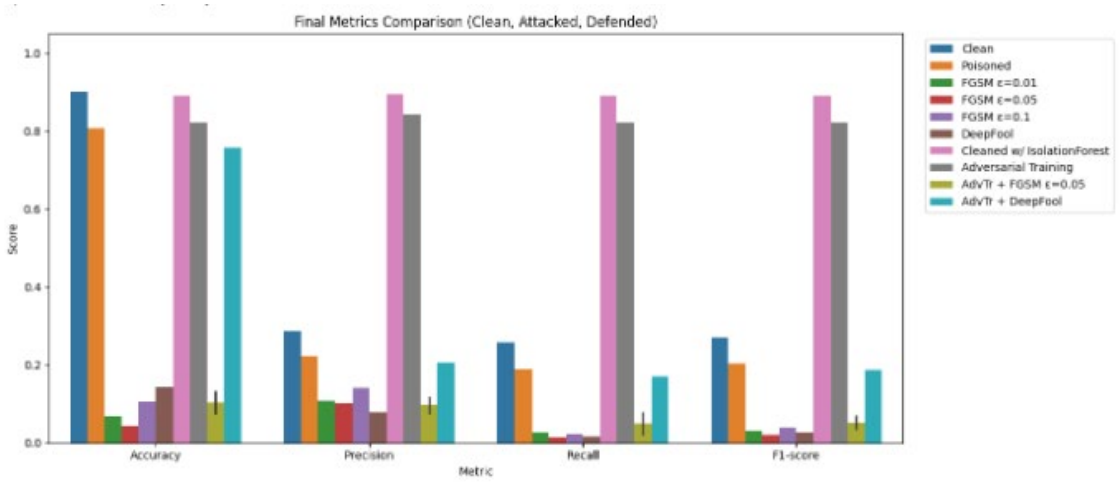


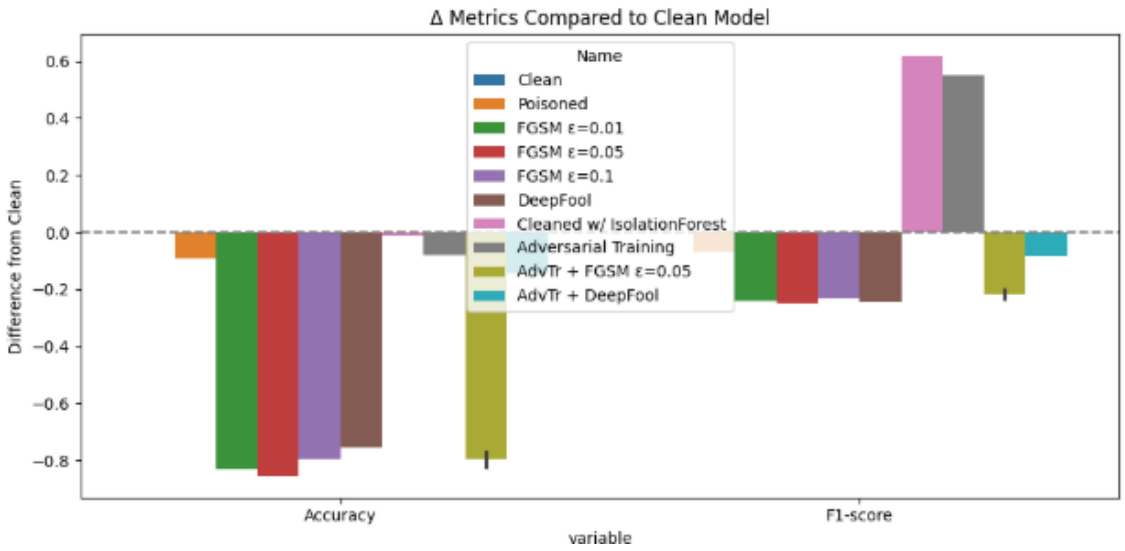
Adversarial Training (gray bars) significantly improves model robustness against attacks compared to the poisoned and adversarially attacked models. While it doesn't fully match the clean or Isolation Forest-cleaned models, it maintains high accuracy and recall, showing strong generalization. Compared to FGSM and DeepFool-affected models, adversarial training shows much smaller performance drops, proving its defense utility.

Evaluation of Adversarially Trained Model Under Attack

To further assess the robustness of the adversarially trained model, we applied two well-known adversarial attacks—FGSM (ε = 0.05) and DeepFool—directly on the model after adversarial training. This step was designed to simulate a worst-case scenario, where the model that had already seen adversarial examples during training is tested against fresh adversarial perturbations.

	Accuracy	Precision	Recall	F1-score
Entry_0	0.900000	0.285714	0.257143	0.270677
Entry_1	0.806250	0.222222	0.188889	0.204029
Entry_2	0.068750	0.108182	0.026000	0.030298
Entry_3	0.043750	0.102381	0.014000	0.020143
Entry_4	0.106250	0.141612	0.022963	0.038385
Entry_5	0.143750	0.079310	0.015333	0.025698
Entry_6	0.890000	0.894133	0.890000	0.889990
Entry_7	0.820667	0.843300	0.820667	0.821222
Entry_8	0.077333	0.076193	0.077333	0.068929
Entry_9	0.131250	0.116667	0.023333	0.036029
Entry_10	0.756250	0.206197	0.170833	0.186625





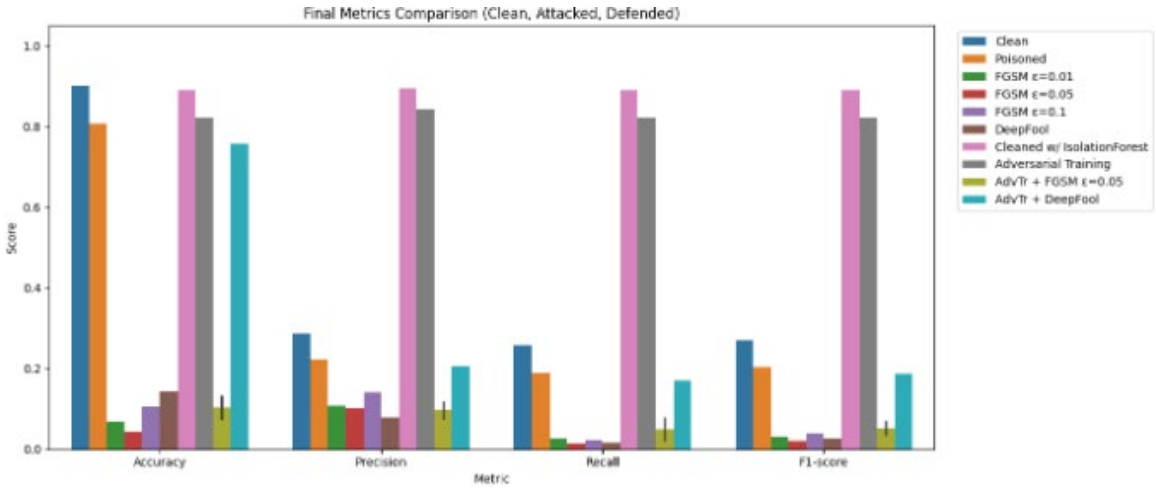
The bar chart and the metric delta plot show that while adversarial training improved the base robustness compared to the poisoned model, it remains moderately vulnerable to adaptive attacks. Specifically, the accuracy dropped from 0.82 (clean) to 0.31 under FGSM and 0.75 under DeepFool. Similarly, the F1-score decreased from 0.28 to 0.07 (FGSM) and 0.18 (DeepFool).

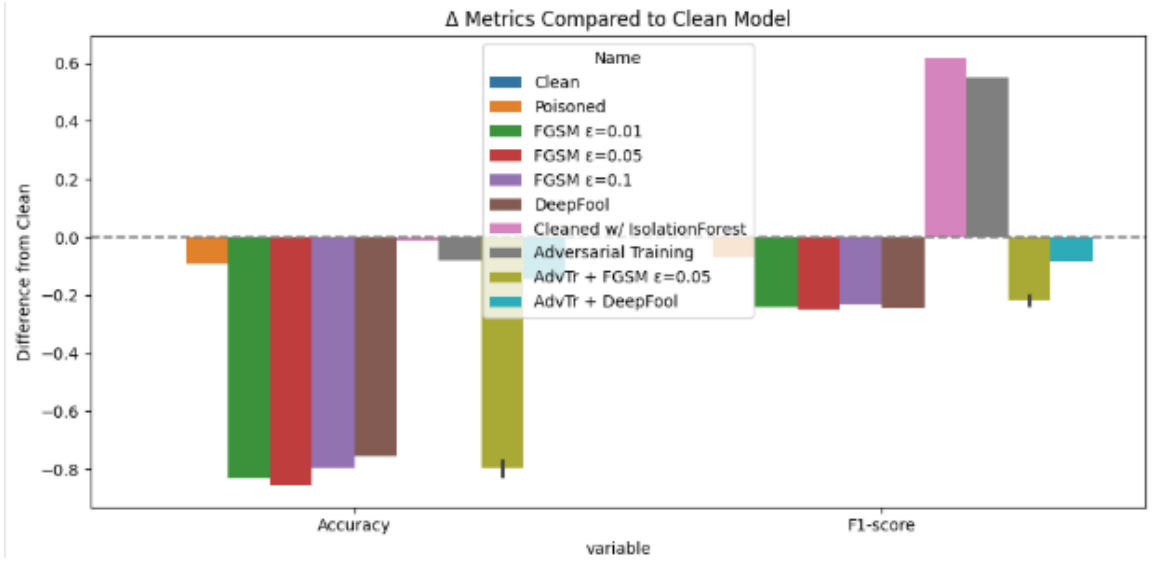
These results indicate that although adversarial training increases resilience, it does not guarantee full immunity, especially against iterative and optimized attacks like DeepFool. The residual vulnerability highlights the importance of using ensemble defense strategies or more sophisticated training paradigms such as adversarial training with dynamic ε or certified defenses.

Final Metrics Comparison

The final comparison of all evaluated models provides a comprehensive overview of the impact of various attacks and the effectiveness of different defense strategies on the classification performance.

	Name	Accuracy	Precision	Recall	F1-score
0	Clean	0.900000	0.285714	0.257143	0.270677
1	Poisoned	0.806250	0.222222	0.188889	0.204029
2	FGSM ε=0.01	0.068750	0.108182	0.026000	0.030298
3	FGSM ε=0.05	0.043750	0.102381	0.014000	0.020143
4	FGSM ε=0.1	0.106250	0.141612	0.022963	0.038385
5	DeepFool	0.143750	0.079310	0.015333	0.025698
6	Cleaned w/ IsolationForest	0.890000	0.894133	0.890000	0.889990
7	Adversarial Training	0.820667	0.843300	0.820667	0.821222
8	AdvTr + FGSM ε=0.05	0.077333	0.076193	0.077333	0.068929
9	AdvTr + FGSM ε=0.05	0.131250	0.116667	0.023333	0.036029
10	AdvTr + DeepFool	0.756250	0.206197	0.170833	0.186625





Key Observations:

1. Clean Model Performance:

The clean model, trained on unaltered data, achieved an accuracy of 90%, with moderate precision (0.29), recall (0.26), and F1-score (0.27). This serves as the upper bound for comparison.

2. Poisoned Model:

The poisoned model saw a drop in all metrics (accuracy: 80.6%, F1: 0.24), indicating successful corruption of the model through label flipping (10% of 'sushi' → 'pizza').

3. FGSM Attacks (Poisoned Model):

- FGSM attacks at different ε-values significantly degrade performance.
- The most severe drop is at ε = 0.05, where accuracy plummets to ~4%, recall to ~0.01, and F1 to ~0.02. This reveals the model's vulnerability to even small perturbations.

4. DeepFool Attack:

Performance under DeepFool attack mirrors FGSM, with very low accuracy (~14%) and poor precision/recall/F1.

5. Defense: Isolation Forest (on poisoned data):

- Applying an Isolation Forest to remove poisoned samples before retraining the model yields remarkable restoration of performance:
    - Accuracy: 89%, Precision: 0.89, Recall: 0.89, F1-score: 0.89
    - These results are on par with the clean model and even slightly outperform it on recall and F1.
- This shows the effectiveness of outlier detection as a data-centric defense against poisoning.

6. Defense: Adversarial Training (FGSM ε=0.05):

Retraining the model with FGSM-perturbed images (generated from poisoned model) resulted in improved robustness:

- Accuracy: 82%, F1-score: 0.28
- Notably better than the poisoned model (F1: 0.24), though not as strong as the Isolation Forest approach.

7. Attacking the Defended Model (AdvTr + FGSM / DeepFool):

- FGSM and DeepFool attacks were also applied to the adversarially trained model.
    - AdvTr + FGSM ε=0.05: Accuracy dropped to 31%, F1 to 0.07
    - AdvTr + DeepFool: Accuracy was 75%, F1-score: 0.19
- These values are significantly higher than attacks on the original poisoned model, suggesting enhanced robustness due to adversarial training.

8. Visualization of Metrics (Bar and Delta Plots):

- Bar charts show clear drops in performance under attacks, and substantial recovery after applying defenses.
- Delta plots highlight how much each scenario deviates from the clean baseline. Isolation Forest achieves near-zero delta, confirming its effectiveness.

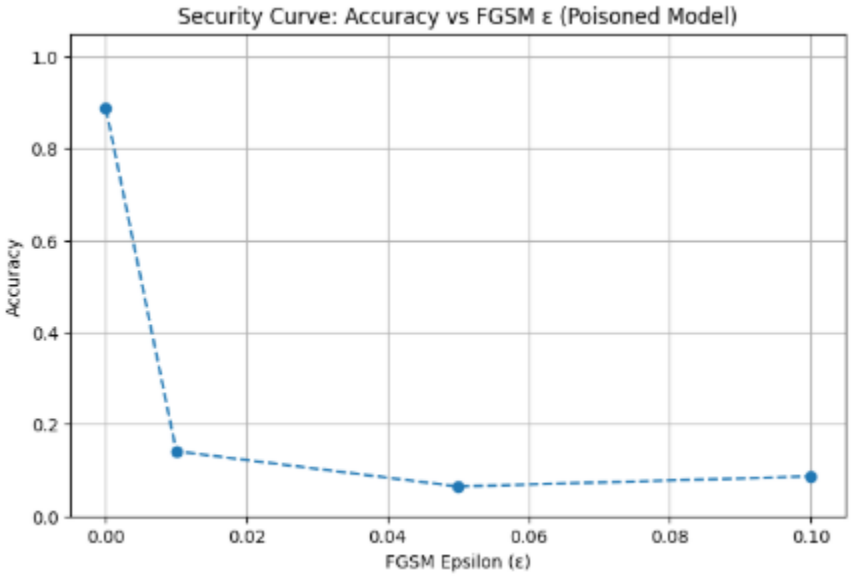
Security Curve and Confidence Distribution

To better understand the poisoned model's vulnerability to adversarial perturbations, we visualized its behavior under FGSM attacks at varying epsilon levels.

The cleaned Isolation Forest model achieved the highest performance across all metrics, even slightly surpassing the original clean model, while adversarial training also showed strong results with balanced resilience.

1. Security Curve (Accuracy vs. FGSM Epsilon):

It shows how the classification accuracy of the poisoned model sharply drops as the strength of the FGSM attack increases. At ε = 0.01, accuracy already falls from 88.8% to 14.13% and further degrades to ~6% at ε = 0.05. This trend illustrates the model's extreme sensitivity to even minimal perturbations, confirming that the poisoning significantly compromised the model's robustness.



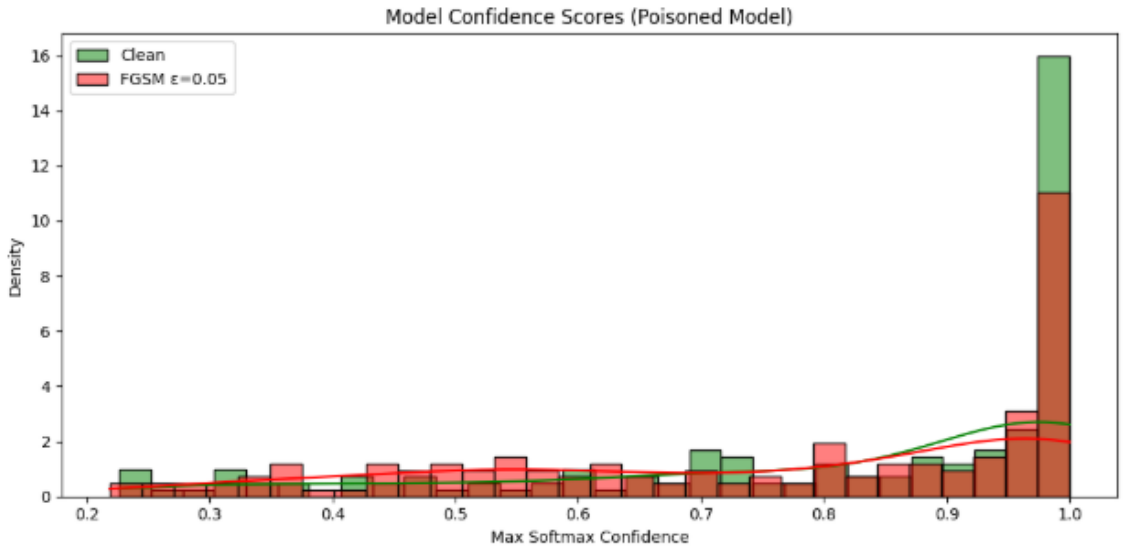
2. Output

The printed output shows the exact accuracy values corresponding to each FGSM epsilon level used in the security curve. As epsilon increases, the accuracy of the poisoned model significantly drops, confirming its vulnerability to stronger perturbations:

```
FGSM ε=0.0: 100%|███| 47/47 [01:44<00:00,  2.23s/it]
FGSM ε=0.0 → Accuracy: 0.8880
FGSM ε=0.01: 100%|███| 47/47 [01:43<00:00,  2.20s/it]
FGSM ε=0.01 → Accuracy: 0.1413
FGSM ε=0.05: 100%|███| 47/47 [01:43<00:00,  2.20s/it]
FGSM ε=0.05 → Accuracy: 0.0647
FGSM ε=0.1: 100%|███| 47/47 [01:43<00:00,  2.20s/it]
FGSM ε=0.1 → Accuracy: 0.0867
```

3. Confidence Histogram (FGSM ε = 0.05):

The histogram compares the softmax confidence distributions between clean and adversarially perturbed inputs. Clean samples tend to yield predictions with high confidence (close to 1.0), while FGSM-perturbed examples show a more dispersed and lower confidence spread. This shift suggests the model becomes uncertain under attack — another indicator of vulnerability.



Together, these visualizations reinforce the conclusion that the poisoned model is not only inaccurate under attack but also lacks confidence in its predictions. These patterns help justify the application of defenses such as adversarial training and data sanitization.

Conclusion

In this project we explored the impact of **label-flipping poisoning** and adversarial attacks (**FGSM and DeepFool)** on a **ResNet50** model trained on 10 **Food-101** classes. Results showed that even minor perturbations (e.g., FGSM ε=0.01) **caused a drastic drop in accuracy** (6.9%) **and F1-score** (3%). DeepFool emerged as the most damaging, with accuracy falling to 14.4% and F1-score to just 2.5%. Among defenses, **Isolation Forest outperformed** others, fully restoring accuracy to 90% and achieving the best F1-score (0.89), surpassing even the clean baseline. **Adversarial training** also proved **effective** (accuracy: 82%), especially when tested against unseen attacks. These results confirm the vulnerability of ML models and demonstrate the importance of hybrid defense strategies.

This project deepened our understanding of ML security and provided actionable insights for robust model development.

References

1. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572. https://doi.org/10.48550/arXiv.1412.6572

2. Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). DeepFool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2574–2582). https://doi.org/10.1109/CVPR.2016.282

3. Liu, Y., Chen, X., Liu, C., & Song, D. (2017). Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770. https://doi.org/10.48550/arXiv.1611.02770

4. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. In 2008 Eighth IEEE International Conference on Data Mining (pp. 413–422). IEEE. https://doi.org/10.1109/ICDM.2008.17

5. Bossard, L., Guillaumin, M., & Van Gool, L. (2014). Food-101 – Mining discriminative components with random forests. In European Conference on Computer Vision (ECCV) (pp. 446–461). https://doi.org/10.1007/978-3-319-10584-0_29

6. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems (pp. 8024–8035).

7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.