Model Hijacking Attacks on Image Classifiers and Defense Algorithms

Final Project Report - COMP 430

06/19/2025

1. Introduction

Model hijacking is a new class of training-time attacks in which a secondary unauthorized task is embedded into the model while preserving high performance on the primary task. These hijacks are stealthy and difficult to detect through typical testing or auditing processes. In this project, we implement the Chameleon attack, which hijacks a model trained on CIFAR-10 to also classify MNIST digits embedded in the input via a learnable Camouflager.

This threat is particularly severe in distributed, outsourced, or federated learning scenarios where training and deployment are decoupled. This project explores not only the attack implementation but also the effectiveness of three prominent defense strategies: Spectral Signature, Activation Clustering, and Neural Cleanse.

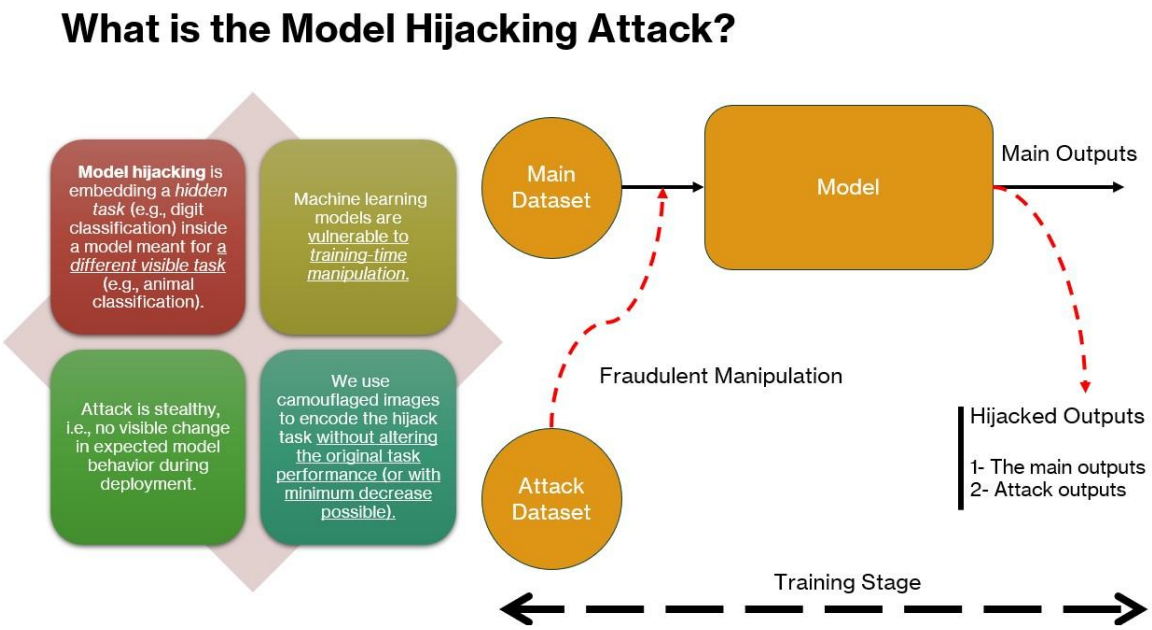

Figure 1: Model Hijacking Pipleline

2. Project Objectives

- Implement Chameleon hijack attack using CIFAR-10 and MNIST.
- Train a Camouflager network to embed digits within CIFAR-10 images.
- Develop three experiments: clean baseline, upper-bound hijack, and stealth hijack.
- Evaluate model performance on both clean and hijack tasks.
- Apply defenses and measure mitigation effectiveness.

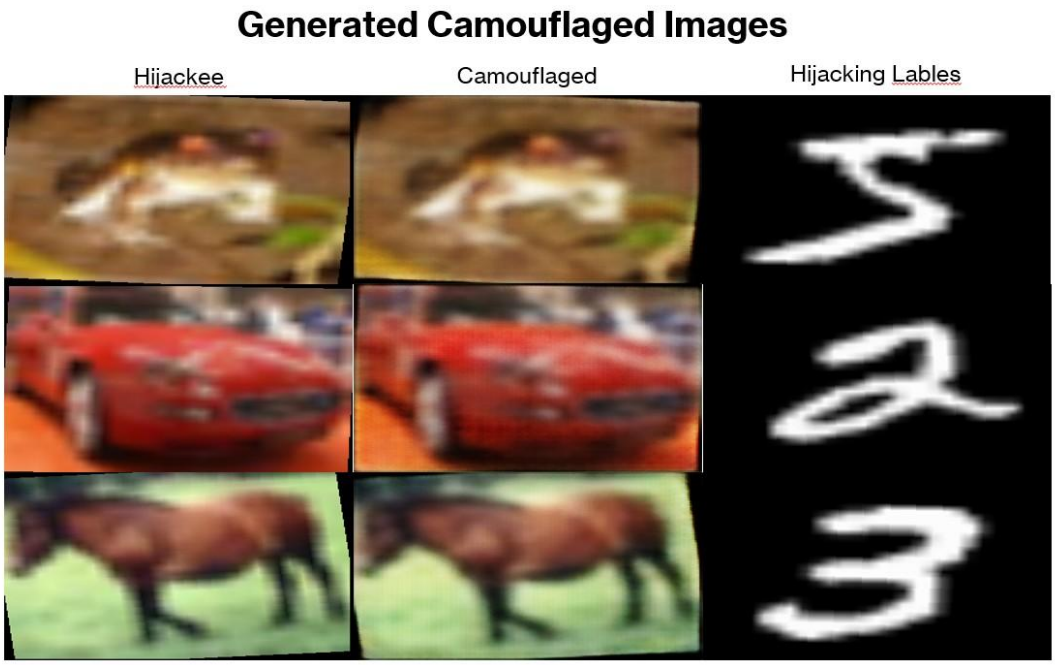

Figure 2: Generated Camouflaged Images

3. Methodology

Three key experiments were conducted:
1. Clean baseline: ResNet18 trained on CIFAR-10 only.
2. Upper-bound hijack: ResNet18 trained on CIFAR-10 and raw MNIST images with explicit digit labels.
3. Stealthy hijack: ResNet18 trained on CIFAR-10 and 1000 camouflaged MNIST samples generated by the Camouflager.

The Camouflager is a dual-stream encoder-decoder that learns to hide the digit in a way that is semantically aligned with the host CIFAR image and visually indistinguishable. Visual and semantic loss functions (using InceptionV3 features) were used, dynamically weighted with SoftAdapt.

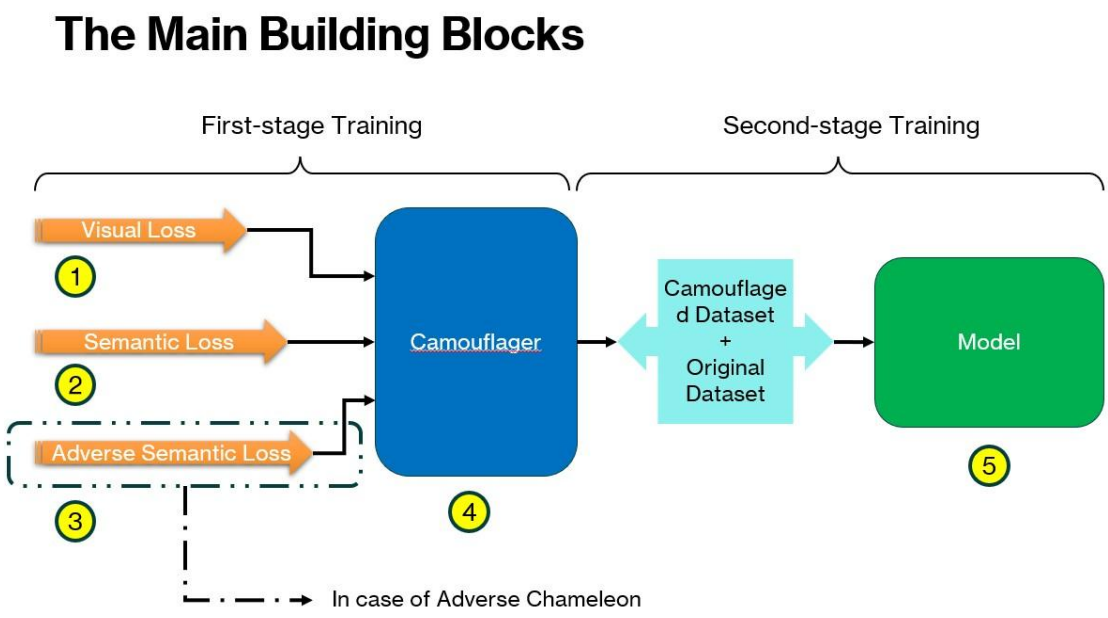

Figure 3: Main Building Blocks of Model Hijacking

4. Experimental Results

Our codes are written in Python and run on Koc University Cluster. Mainly, one Tesla:V100 is used with 100G of CPU memory for the training, and less RAM for evaluation and testing will be required. Below are the training loss and accuracy curves for all three training setups. We coded each part separately and extracted the weights of the model after each step. Finally, camouflaged dataset generation, dataset preparation, camouflaged training and instance extraction, three pre-training steps, one file for each of the defenses and their evaluation scripts are attached to this report (A total of 13 steps result preparation).

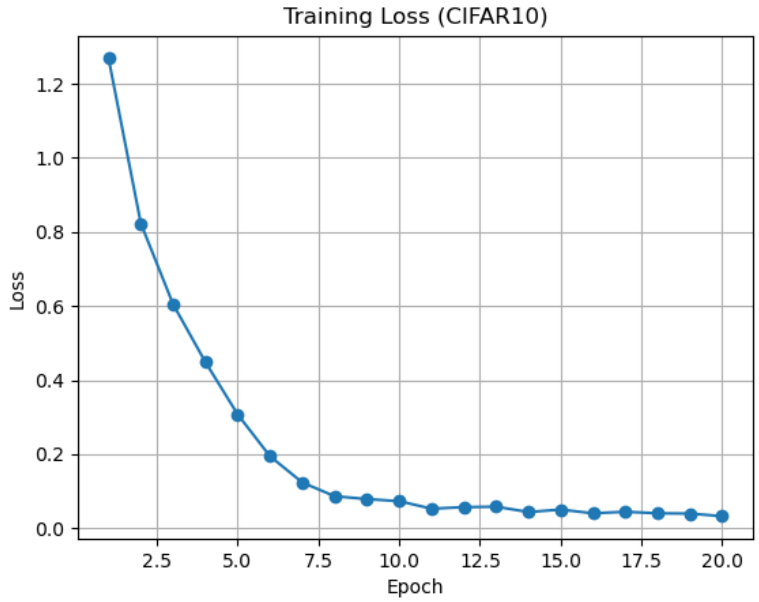

Figure 4: Baseline - Training Loss

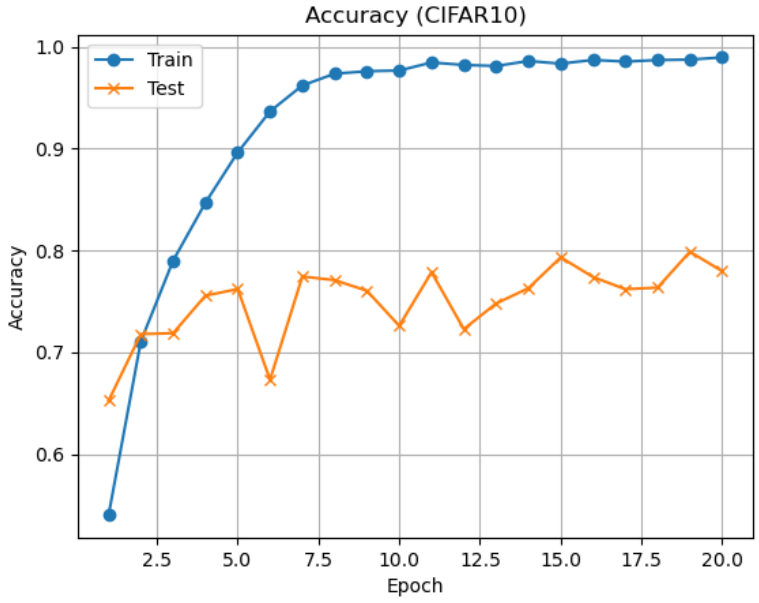

Figure 5: Baseline - Accuracy

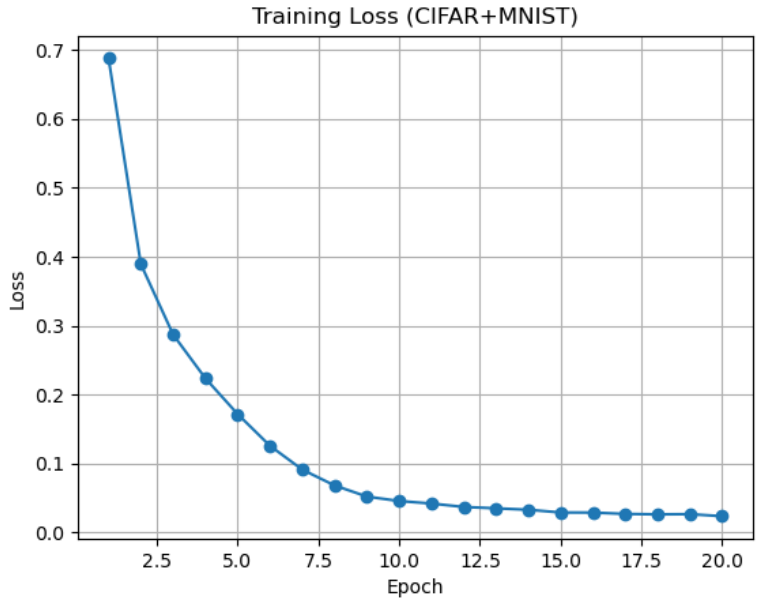

Figure 6: Upper Bound - Training Loss

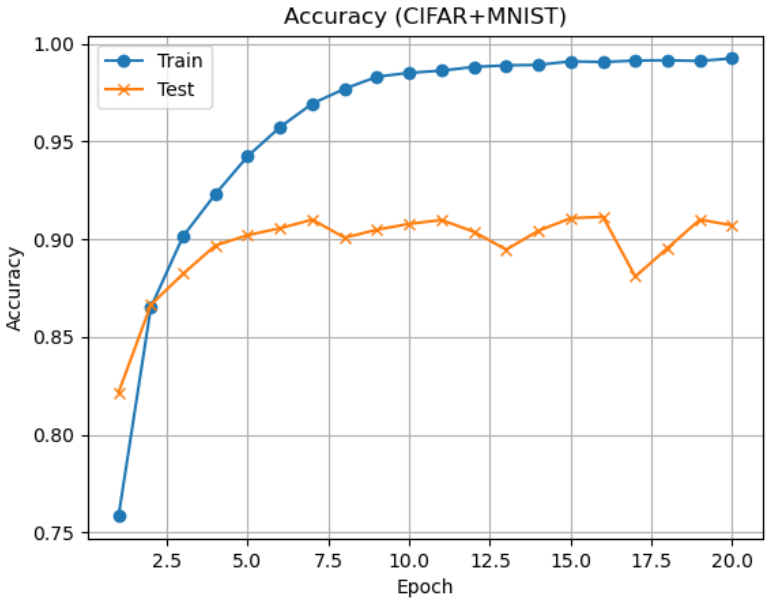

Figure 7: Upper Bound – Accuracy

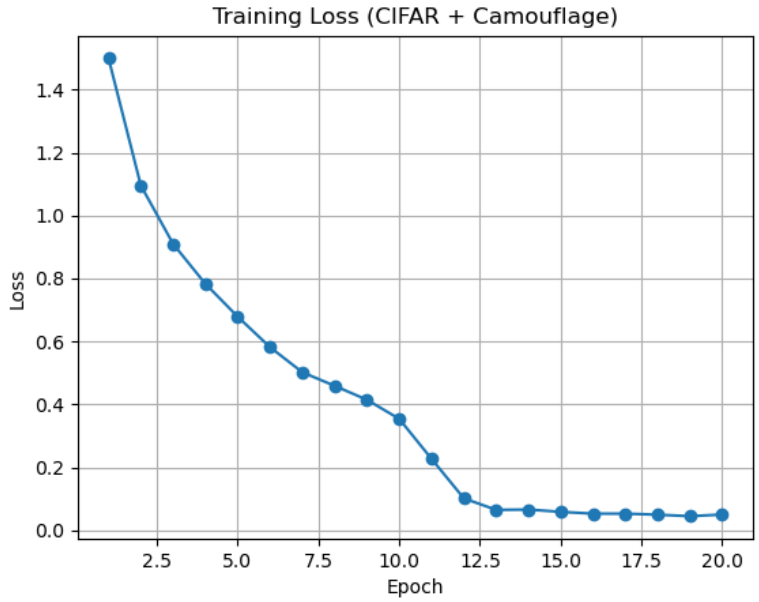

Figure 8: Stealth Hijack - Training Loss

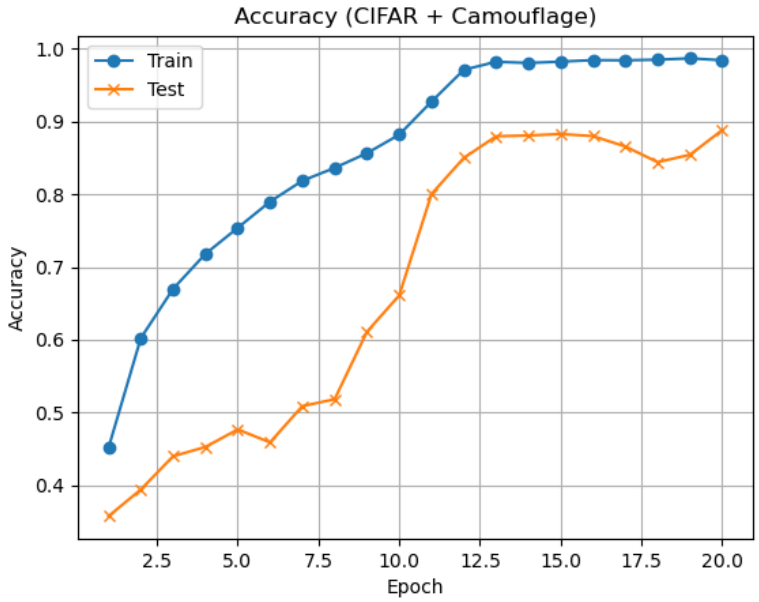

Figure 9: Stealth Hijack – Accuracy

Model	Train Accuracy	Test Accuracy	Train Loss
Clean Baseline	99.0%	78.0%	0.0329
Upper Bound	99.3%	90.7%	0.0239
Stealth Hijack	98.4%	88.8%	0.0505

Table 1: Accuracy of ResNet18 across all experiments

** As shown in Table 1, the Hijacked model drops in performance. We expect to see the accuracy decrease after performing the hijack, however this amount is not totally noticeable, and the attack has been done perfectly.

5 Defense Mechanisms

To defend against stealthy model hijacking, we implemented and evaluated three distinct detection techniques. Each method analyzes the trained model's internal behavior to identify and eliminate poisoned data or backdoor patterns.

5.1 Spectral Signature

This method performs PCA on activations and identifies samples that lie far from the cluster center in projected space. These are suspected to be poisoned. We applied this to the penultimate layer of the ResNet18 model and removed outliers accordingly. As a result of applying this defense 3004 suspicious images will be detected by the model.

5.2 Activation Clustering

By clustering activation vectors with k-means (after PCA), we separate clean and poisoned samples. Most poisoned inputs cluster away from the majority. We removed minority clusters for each class and retrained the model. As a result of applying this defense 2912 suspicious images will be detected by the model.

5.3 Neural Cleanse (MNTD)

Neural Cleanse identifies suspicious classes by reverse-engineering minimal perturbations (triggers) that map inputs to each class. Classes with abnormally low L1-norm trigger masks are flagged. We used this to filter suspicious classes and retrained the model.

6. Defense Comparison

Defense	Train Accuracy	Test Accuracy	Test Loss
Spectral Signature	98.4%	85%	0.0497
Activation Clustering	98.6%	75.3%	0.0456
Neural Cleanse	98.7%	83.4%	0.0402

Table 2: Accuracy after applying defenses

As you can see in Table 2, the defense mechanisms have done their job properly by refining the performance of the model on the main task while the ability of the model to perform the hijacking task is lost. For instance, we have evaluated the performance of the model after the activation clustering defense, using “final_evaluation.py” file. We take 20 camouflaged samples generated by the camouflager and pass them through the re-trained model (after the defense). The results are as follows:

Sample 000: Predicted CIFAR-10 Class = 6 → Actual (our expectation): 5

Sample 001: Predicted CIFAR-10 Class = 9 → Actual (our expectation): 0

Sample 002: Predicted CIFAR-10 Class = 9 → Actual (our expectation): 4

Sample 003: Predicted CIFAR-10 Class = 4 → Actual (our expectation): 1

Sample 004: Predicted CIFAR-10 Class = 1 → Actual (our expectation): 9

Sample 005: Predicted CIFAR-10 Class = 1 → Actual (our expectation): 2

.

.

.

The fact that the predicted labels are totally random suggests that the model has totally lost its ability to perform the hijacking task. The same type of analysis can be done for the other defenses as well.

7. Challenges and Observations

• SoftAdapt introduced instability due to rapidly shifting weights; this was mitigated with normalization.
• Label alignment between hijack and host task is crucial for stealth.
• Visual fidelity vs semantic alignment tradeoff in Camouflager affected training dynamics.
• Defense methods require careful parameter tuning to avoid false positives or excessive clean data loss.

8. Conclusion

We demonstrated the feasibility and stealth of model hijacking using the Chameleon attack. By crafting semantically meaningful but visually inconspicuous samples, we embedded a secondary MNIST classification task within a CIFAR-10 model. We evaluated three defenses—Spectral Signature, Activation Clustering, and Neural Cleanse—and showed how each mitigated the attack to varying degrees. As a result, we successfully tested and evaluated the feasibility of applying both the attacks and the defenses to our training stage. Based on the final results, “Neural Cleanse” shows a superior performance compared to the other two, according to the train and loss curvatures.

9. References

[1] Salem, Ahmed, Michael Backes, and Yang Zhang. "Get a model! model hijacking attack against machine learning models." arXiv preprint arXiv:2111.04394 (2021).

[2] Tran, Brandon, Jerry Li, and Aleksander Madry. "Spectral signatures in backdoor attacks." Advances in neural information processing systems 31 (2018)

[3] Chen, Bryant, et al. "Detecting backdoor attacks on deep neural networks by activation clustering." arXiv preprint arXiv:1811.03728 (2018)

[4] Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." 2019 IEEE symposium on security and privacy (SP). IEEE, 2019.

[5] Kolouri, S., Saha, A., Pirsiavash, H., & Hoffmann, H. (2020). Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[6] Chow, Ka-Ho, et al. "Stdlens: Model hijacking-resilient federated learning for object detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[7] Liu, Kang, Brendan Dolan-Gavitt, and Siddharth Garg. "Fine-pruning: Defending against backdooring attacks on deep neural networks." International symposium on research in attacks, intrusions, and defenses. Cham: Springer International Publishing, 2018.

[8] Gao, Y., Xu, C., Wang, D., et al. (2019). STRIP: A Defence Against Trojan Attacks on Deep Neural Networks. In Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC).

[9] Si, Wai Man, et al. "{Two-in-One}: A Model Hijacking Attack Against Text Generation Models." 32nd USENIX Security Symposium (USENIX Security 23). 2023.

[10] AI, NIST. "Artificial intelligence risk management framework (AI RMF 1.0)." URL: https://nvlpubs. nist. gov/nistpubs/ai/nist. ai (2023): 100-1.