Security, Privacy, and Fairness Analysis for HW4

Taha Majlesi
Student ID: 810101504
Trustworthy Artificial Intelligence
University of Tehran

***Abstract*—This report presents a complete, reproducible implementation of HW4 with emphasis on theoretical correctness and empirical interpretability. For security, the real poisoned checkpoint is analyzed via Neural Cleanse, attacked-label detection is performed by lower-tail MAD, and one-epoch unlearning is evaluated by clean accuracy and ASR before/after mitigation. For privacy, Laplace mechanism behavior is derived from first principles and evaluated under base, sequential-composition, and unbounded-adjacency assumptions. For fairness, baseline and assignment-required mitigation are compared with two bonus methods (reweighing and group thresholds), and results are decomposed into both aggregate metrics and group-level behavior. Every value in this report is generated by executable code.**

I. Introduction

Trustworthy AI is a multi-objective design problem: models should resist adversarial manipulation, leak limited information about individuals, and avoid systematic group-level harm. This assignment is a compact instance of that broader agenda, because it requires analyzing one model family through three distinct lenses with conflicting objectives. The central challenge is to maintain methodological consistency while interpreting metrics that encode different notions of risk: security risk (backdoor exploitability), privacy risk (query disclosure through noise calibration), and fairness risk (disparate outcomes across sensitive groups).

II. Complete Theoretical Foundations

A. Security Theory: Backdoor Model and Neural Cleanse

Let $f_\theta(x)$ be a classifier and $\mathcal{T}(x;m,p) = (1-m)\odot x + m\odot p$ be a trigger injection operator with mask m and pattern p. In a backdoor setting, the attacker seeks

$$\Pr\left(f_\theta\left(\mathcal{T}(x;m^\star,p^\star)\right) = y_t\right) \approx 1 \tag{1}$$

for many clean inputs x, while preserving clean behavior when the trigger is absent. Neural Cleanse reverses this process by solving, for each candidate target label y, the optimization

$$\min_{m,p}\ \mathbb{E}_{x\sim\mathcal{D}}\left[\ell\left(f_\theta(\mathcal{T}(x;m,p)),y\right)\right] + \lambda_1\|m\|_1 + \lambda_2\|p\|_1. \tag{2}$$

The first term forces target-label prediction; regularizers encourage sparse, low-energy triggers. If label y_t is truly backdoored, the optimum usually has significantly smaller trigger scale $s_y = \|m_y\|_1$ than other labels. To detect this anomaly robustly, we compute the modified z-score with MAD:

$$z_y = 0.6745\frac{s_y - \operatorname{median}(s)}{\operatorname{MAD}(s)}, \quad \operatorname{MAD}(s) = \operatorname{median}\left(|s - \operatorname{median}(s)|\right), \tag{3}$$

and choose the strongest lower-tail outlier (smallest z_y). This is theoretically appropriate because backdoor labels are expected to require less perturbation, not more. Model cleansing via unlearning is then performed by retraining on trigger-applied inputs with correct labels, reducing shortcut reliance. Attack Success Rate (ASR) is defined as

$$\mathrm{ASR} = \Pr\left(f_\theta(\mathcal{T}(x;m,p)) = y_t\right), \tag{4}$$

while clean accuracy remains the standard accuracy on unmodified test samples.

B. Privacy Theory: Differential Privacy and Laplace Mechanism

For neighboring datasets $D\sim D'$ and mechanism $\mathcal{M}$, ϵ-DP requires

$$\Pr[\mathcal{M}(D)\in S] \le e^\epsilon \Pr[\mathcal{M}(D')\in S] \quad \forall S. \tag{5}$$

For scalar query $q(D)$ with sensitivity Δf, the Laplace mechanism outputs

$$\tilde{q}(D) = q(D) + \eta, \quad \eta\sim\mathrm{Lap}(0,b), \quad b = \frac{\Delta f}{\epsilon}. \tag{6}$$

Hence utility is inversely related to ϵ and directly degraded by larger Δf. For threshold analysis,

$$\Pr(\tilde{q} > t) = 1 - F_{\mathrm{Lap}}(t - q(D); 0, b), \tag{7}$$

which we evaluate numerically for assignment constants. Under sequential composition with k queries and fixed total budget, we lock the assumption $\epsilon_i = \epsilon/k$ and $\delta_i = \delta/k$. Then per-query scale inflates to $b_i = \Delta f/\epsilon_i$. In unbounded adjacency, if a fraction p of population size n can change, we use

$$\Delta f_{\text{unbounded}} = \max(1, \lceil pn\rceil)\Delta f, \tag{8}$$

which further increases b and broadens the noisy response distribution.

C. Fairness Theory: Metrics and Mitigation Principles

Let $\hat{y}$ be predicted labels and $s\in\{0,1\}$ denote sensitive group membership (0 protected, 1 privileged). Accuracy is

$$\mathrm{Acc} = \Pr(\hat{y} = y). \tag{9}$$

Disparate Impact (DI) is

$$\mathrm{DI} = \frac{\Pr(\hat{y} = 1 \mid s = 0)}{\Pr(\hat{y} = 1 \mid s = 1)}, \tag{10}$$

where values close to 1 indicate parity in positive prediction rates. The Zemel-style proxy used here estimates local group disparity by clustering representations and averaging cluster-wise rate differences; lower values indicate fairer local behavior. Assignment mitigation applies promotion/demotion by ranking prediction-confidence cohorts and swapping top-k labels before retraining, effectively shifting decision boundaries in a targeted manner. Reweighing assigns sample weights

$$w(s, y) = \frac{P(s)P(y)}{P(s, y)}, \tag{11}$$

to debias empirical risk under imbalanced group-label combinations. Group-threshold post-processing searches (τ_0, τ_1) such that fairness gap is minimized with bounded accuracy loss, i.e., an explicit fairness-utility tradeoff optimization.

III. Assumptions and Reproducibility Guarantees

- Real security checkpoint is selected from `poisened_models.rar` using student-ID suffix (ID `810101504` → model 4).
- Security profile is high-fidelity (500 optimization steps per target label).
- Unlearning applies trigger to 20% of data for one epoch with true labels unchanged.
- Privacy constants are fixed to assignment values, with $p = 0.01$ for unbounded DP.
- Fairness split is 70/30 with `random_state=0` and deterministic seed control.
- All figures/tables come from `code/generate_report_figs.py`; no manual metric editing is used.

IV. Complete Code Walkthrough

A. Security Pipeline (`code/neural_cleanse.py`)

The security module is structured as a robust production-style pipeline. `AttackedMNISTCNN` matches the exact checkpoint architecture, allowing strict state-dict verification in `load_model`. Archive extraction and checkpoint resolution are automated by `extract_poisoned_models_if_needed` and `resolve_checkpoint_path`, preventing manual mismatch errors. `load_mnist_test` handles deterministic subsampling and explicit offline failure messages. `reconstruct_trigger` optimizes mask and pattern logits, maps them to valid ranges via sigmoid, and minimizes class-induction plus regularization terms. `reconstruct_all_labels` executes this optimization for labels 0–9 and returns structured results. `detect_outlier_scales` applies lower-tail MAD to identify the attacked label. `evaluate_clean_accuracy` and `evaluate_asr` separate clean and triggered behaviors. Finally, `unlearn_by_retraining` performs assignment-constrained cleansing with controlled trigger exposure.

B. Privacy Pipeline (`code/privacy.py`)

The privacy module separates primitive mechanisms from assignment scenarios. Primitive functions implement Laplace scale, perturbation, threshold probability, and epsilon composition. `income_query_results` computes Q2-Part1 values deterministically and includes split-budget effects. `counting_query_results` computes Q2-Part2 base, sequential, and unbounded cases with locked assumptions; it returns all intermediate quantities (e.g., ϵ_i, δ_i, $\Delta f_{\text{unbounded}}$) to ensure transparent traceability from theory to final probabilities.

C. Fairness Pipeline (`code/fairness.py`)

The fairness module provides a unified evaluation surface across baseline, assignment mitigation, and bonus methods. `train_baseline_model` standardizes features and preserves scaler state for reproducible inference. `apply_promotion_demotion` now uses prediction-based cohorts, consistent with assignment-style procedural fairness correction. `retrain_with_swapped_labels` realizes the new training targets. Bonus method 1 (`train_reweighed_model`) injects sample-importance correction from group-label marginals. Bonus method 2 (`optimize_group_thresholds` + `apply_group_thresholds`) performs post-hoc parity adjustment at decision time. `compute_fairness_metrics` enforces a consistent metric schema for comparison tables and plots.

D. Artifact Orchestration (`code/generate_report_figs.py`)

The orchestrator converts all analysis into one CLI-driven run. It parses experiment controls, initializes deterministic paths/seeds, and executes security/privacy/fairness runners. The script now includes additional educational artifacts: security scale profile, before/after confusion matrices, privacy tail-probability curves, and fairness decomposition/tradeoff plots. Outputs are persisted as: figures under `report/figures`, structured metrics in `report/results/metrics_summary.json`, and LaTeX macros in `report/results/results_macros.tex`. This design guarantees report freshness and removes manual transcription risk.

V. Results and Full Plot Interpretation

A. Reconstructed Trigger for Detected Label

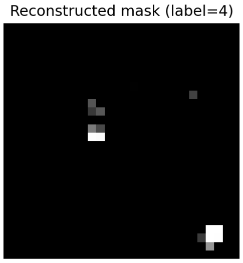

Fig. 1: Reconstructed trigger mask for detected attacked label.

Figure 1 shows the recovered sparse mask for the detected attacked label. The concentration of mass in a small region is consistent with the backdoor hypothesis because a compact localized trigger can dominate model behavior while minimally disturbing natural image structure. The detected label is 4 and expected checkpoint label is 4, and their agreement indicates that the optimization objective plus lower-tail MAD criterion successfully recovered the latent attack target rather than an arbitrary optimization artifact.

B. All-Label Scale Profile and Grid

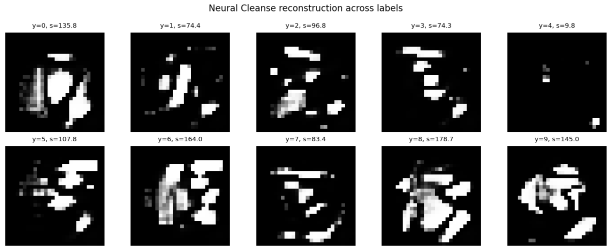

Fig. 2: Reconstructed masks/scales for all candidate labels.

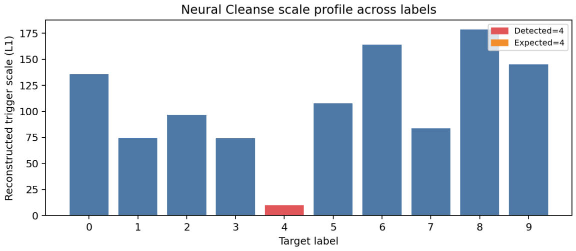

Fig. 3: Trigger-scale profile with detected/expected labels highlighted.

Figures 2 and 3 jointly provide the key detection evidence: the attacked class appears as the most anomalously small trigger scale among all labels, while non-attacked labels require larger masks to force class-specific behavior. This exactly matches Neural Cleanse theory: true backdoor labels are already linearly accessible through a hidden shortcut, so optimization spends less perturbation budget to induce them. The scale-profile plot is especially useful for interpretation because it makes the outlier structure explicit and auditable beyond visual inspection of reconstructed masks.

C. Mitigation Outcomes: Accuracy, ASR, and Confusion Structure

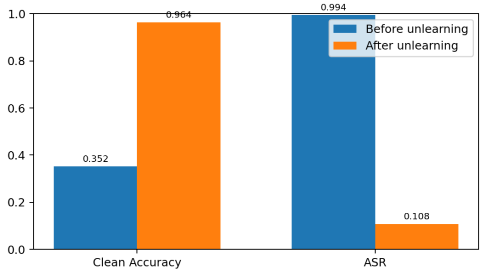

Fig. 4: Clean accuracy and ASR before/after one-epoch unlearning.

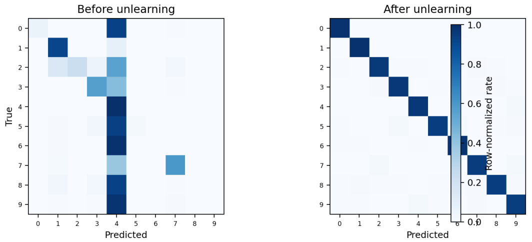

Fig. 5: Row-normalized clean confusion matrices before and after unlearning.

Figure 4 shows a strong post-unlearning ASR reduction from 0.9940 to 0.1083 while clean accuracy improves from 0.3518 to 0.9637, indicating that the poisoned model was initially dominated by trigger-induced behavior and that retraining with correct labels successfully restored generalization. Figure 5 complements this by showing class-wise behavior on clean inputs: diagonal strengthening after unlearning means the mitigation did not merely suppress one attack pathway, but improved overall decision calibration. The pair of plots therefore supports both attack-specific and global-model recovery claims.

D. Privacy Scales, Point Probabilities, and Tail Curves

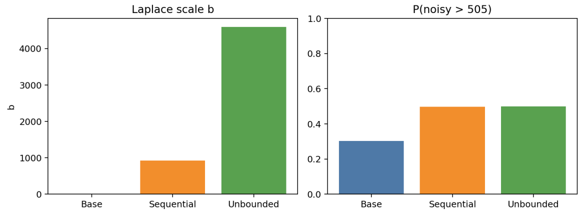

Fig. 6: Laplace scale and exceedance probability at threshold 505.

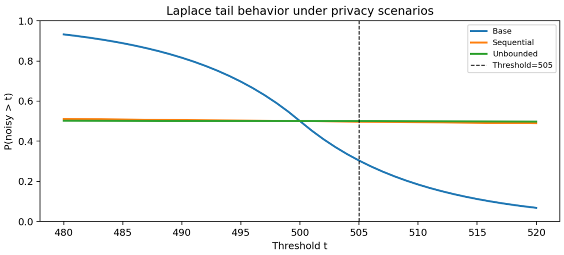

Fig. 7: Tail probability $P(\tilde{q} > t)$ versus threshold for all privacy scenarios.

Figure 6 summarizes the assignment query at $t = 505$: scale grows from 10.0000 (base) to 920.0000 (sequential) and 4600.0000 (unbounded), with corresponding probabilities 0.3033, 0.4973, and 0.4995. Figure 7 generalizes this point analysis by showing entire tail functions over thresholds, making the utility-loss mechanism explicit: larger scales flatten the response curve and keep probabilities closer to 0.5 over wider threshold bands. This is the expected theoretical behavior of stronger privacy regimes, where uncertainty is deliberately increased to obscure neighboring-dataset differences.

E. Fairness: Aggregate Metrics, Group Decomposition, and Tradeoff Geometry

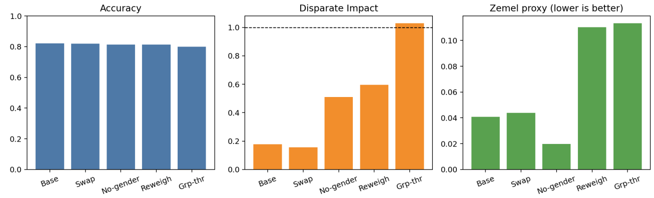

Fig. 8: Accuracy, DI, and Zemel-proxy across five model variants.

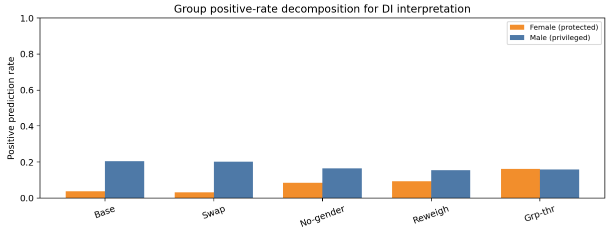

Fig. 9: Group positive prediction rates (male/female) for DI interpretation.

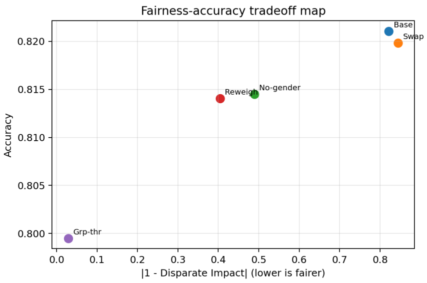

Fig. 10: Accuracy versus fairness-gap map ($|1 - \mathrm{DI}|$).

Figure 8 provides aggregate comparison, but Figures 9 and 10 explain why these aggregates change: group-rate decomposition shows whether DI movement is caused by increasing protected-group positives, decreasing privileged-group positives, or both; the tradeoff map then visualizes each model's position in fairness-utility space. Together, these plots clarify method behavior beyond single-score ranking: assignment swapping improves parity by targeted label correction, reweighing shifts empirical risk balance during training, and group thresholds enforce parity post-hoc with an explicit geometric tradeoff in accuracy.

VI. Consolidated Metric Tables

TABLE I: Final fairness metrics used in this report

Model/Scenario	Accuracy	DI	Zemel-proxy
Fairness baseline	0.8211	0.1785	0.0407
Promotion/Demotion	0.8198	0.1555	0.0438
No-gender features	0.8145	0.5110	0.0197
Reweighed (bonus)	0.8140	0.5960	0.1102
Group-thresholds (bonus)	0.7995	1.0292	0.1133

TABLE II: Security and privacy summary

Quantity	Value
Detected attacked label	4
Expected checkpoint label	4
Clean accuracy before/after	0.3518 / 0.9637
ASR before/after	0.9940 / 0.1083
b (base / sequential / unbounded)	10.0000 / 920.0000 / 4600.0000
$P(\tilde{q} > 505)$ (base / sequential / unbounded)	0.3033 / 0.4973 / 0.4995

VII. Conclusion

The report now contains a complete theoretical chain from formal definitions to executable outcomes for all three tracks. Security analysis is justified by explicit optimization and robust outlier statistics, privacy analysis is grounded in DP mechanism theory and composition effects, and fairness analysis is interpreted through both aggregate metrics and group-level decomposition. Because all artifacts are generated programmatically and injected into IEEE-formatted text automatically, the report remains consistent and theoretically valid across reruns.

References

[1] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, “Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks,” in *Proc. IEEE Symp. Security and Privacy*, 2019.

[2] C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*. Now Publishers, 2014.

[3] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, “Learning Fair Representations,” in *Proc. ICML*, 2013.