Empirical Evaluation of Data Poisoning Attacks in Supervised Learning

Toshif Khan and Muhammad Abusaqer

Trustworthy Language Intelligence Lab (TLI Lab), Department of Math, Data & Technology, Minot State University, Minot, ND
muhammad.abusaqer@minotstateu.edu

Why data poisoning matters

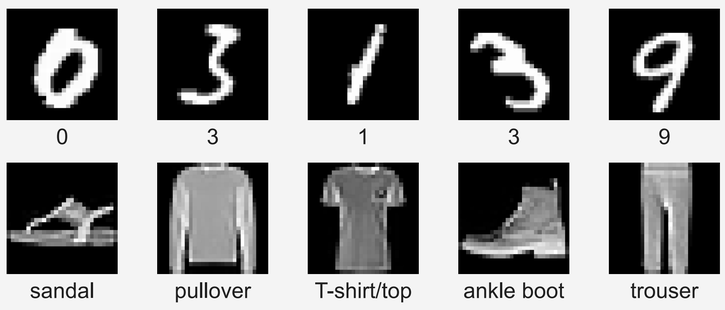
Supervised learning assumes its training data can be trusted. Data poisoning breaks that assumption: an adversary alters the training set so the model loses accuracy or does what the attacker wants at inference time, a training-time attack in NIST's taxonomy [1].

Label flipping is indiscriminate: selected training instances get wrong labels and overall quality drops [2], [5]. Backdoor poisoning is targeted: samples are stamped with a trigger and relabeled, so the model behaves normally on clean inputs but obeys the trigger [3], [4]. The second failure mode is the dangerous one, because standard accuracy can hide it.

This study. A compact, reproducible benchmark comparing both attacks under one protocol: two image datasets, three classical classifiers, three poisoning rates, clean-test metrics, and attack success rate. No new attack or defense is proposed.

Datasets

What they are. MNIST: 28 × 28 grayscale handwritten digits, 10 classes [6]. Fashion-MNIST: clothing photos with the same size, splits, and class count, built as a harder drop-in replacement [7].



Why they fit. Standard, well-understood benchmarks make attack effects easy to read; identical formats let one pipeline serve both; and the easier versus harder pair shows whether the trends survive a harder task. Images: the study's own training subsets.

Threat model and the two attacks

The attacker can alter a fraction (5%, 10%, or 20%) of the training set and nothing else. The test set is never touched.

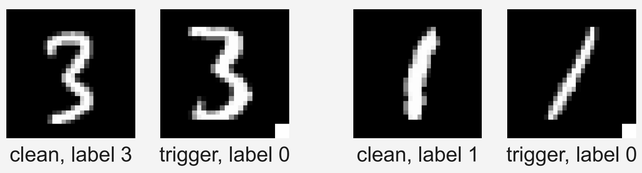
Label flipping (indiscriminate)

- Pick the fraction of training images uniformly at random.
- Replace each true label with a random wrong label. Pixels stay unchanged.
- Retrain. Measure clean-test accuracy, macro-precision, macro-recall, macro-F1.

Backdoor (targeted, BadNets-style [3])

- Pick the fraction among non-target training images at random.
- Stamp a 3 × 3 white square in the bottom-right corner and relabel to target class 0.
- Retrain. Measure clean-test metrics plus attack success rate (ASR): share of triggered non-target test images classified as class 0.

Backdoor examples from the study: clean training image (left of each pair) and a triggered image relabeled to class 0 (right).



Experimental setup

Data	Stratified subsets per dataset: 8,000 training and 2,000 test images (seed 42), loaded through scikit-learn and OpenML; pixels scaled to [0, 1]; each image flattened to 784 features
Models	Logistic Regression (saga solver, max 300 iterations) and Linear SVM (max 5,000 iterations), both with feature standardization; Random Forest with 100 trees. All in scikit-learn [8]
Rates	5%, 10%, and 20% of the training set, for each attack, dataset, and model (2 datasets × 3 models × 7 training conditions)
Protocol	Clean baseline first, then a separate retrain under every poisoning condition. Poisoning touches training data only
Metrics	Clean test: accuracy, macro-precision, macro-recall, macro-F1. Backdoor: attack success rate on triggered non-target test images
Tools	Python, NumPy, pandas, matplotlib, scikit-learn; one notebook saves every table and plot

Results

Label flipping: clean-test accuracy (Table 2 of the paper; macro-F1 followed the same pattern). Largest drop is relative to the clean baseline.

Model	Clean	5%	10%	20%	Largest drop
MNIST					
Logistic Regression	0.9005	0.8765	0.8590	0.8300	**0.0705**
Linear SVM	0.8400	0.8125	0.8045	0.7905	0.0495
Random Forest	0.9475	0.9420	0.9440	0.9400	0.0075
Fashion-MNIST					
Logistic Regression	0.8295	0.8105	0.7975	0.7795	0.0500
Linear SVM	0.7665	0.7150	0.7005	0.7065	**0.0660**
Random Forest	0.8455	0.8385	0.8395	0.8435	0.0070

Backdoor poisoning: clean-test accuracy and attack success rate (ASR) at each poisoning rate (Table 3 of the paper). Clean baselines as above.

Model	Acc. 5%	Acc. 10%	Acc. 20%	ASR 5%	ASR 10%	ASR 20%
MNIST						
Logistic Regression	0.9010	0.8980	0.8980	0.9856	0.9889	0.9884
Linear SVM	0.8330	0.8415	0.8285	0.9961	0.9956	0.9956
Random Forest	0.9480	0.9445	0.9425	**1.0000**	**1.0000**	**1.0000**
Fashion-MNIST						
Logistic Regression	0.8230	0.8250	0.8200	0.9667	0.9883	0.9867
Linear SVM	0.7615	0.7640	0.7335	0.9928	0.9950	0.9939
Random Forest	0.8455	0.8385	0.8445	0.9922	**1.0000**	**1.0000**

References

[1] A. Vassilev et al., "Adversarial machine learning: A taxonomy and terminology of attacks and mitigations," NIST AI 100-2e2023, National Institute of Standards and Technology, Gaithersburg, MD, USA, Jan. 2024.

[2] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in Proc. 29th Int. Conf. Mach. Learn. (ICML), 2012, pp. 1467 to 1474.

[3] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," arXiv:1708.06733, 2017.

[4] A. Schwarzschild, M. Goldblum, A. Gupta, J. P. Dickerson, and T. Goldstein, "Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks," in Proc. 38th ICML, 2021, pp. 9389 to 9398.

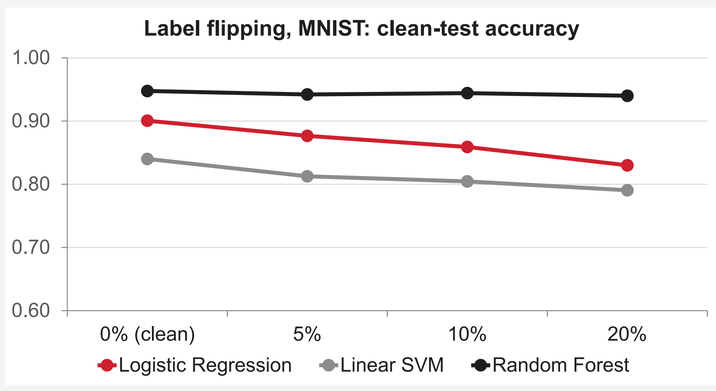
[5] F. A. Yerlikaya and Ş. Bahtiyar, "Data poisoning attacks against machine learning algorithms," Expert Syst. Appl., vol. 208, art. no. 118101, 2022.

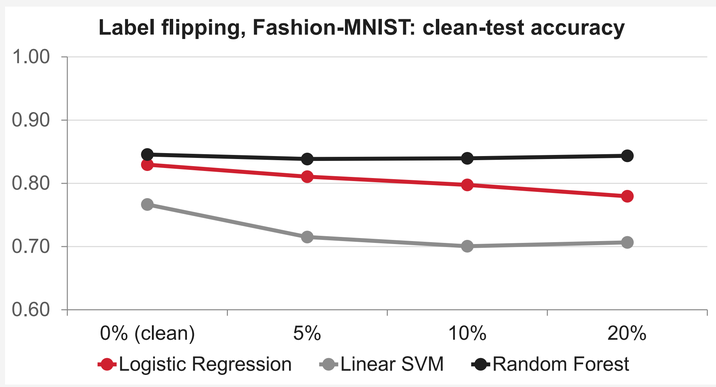
[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278 to 2324, 1998.

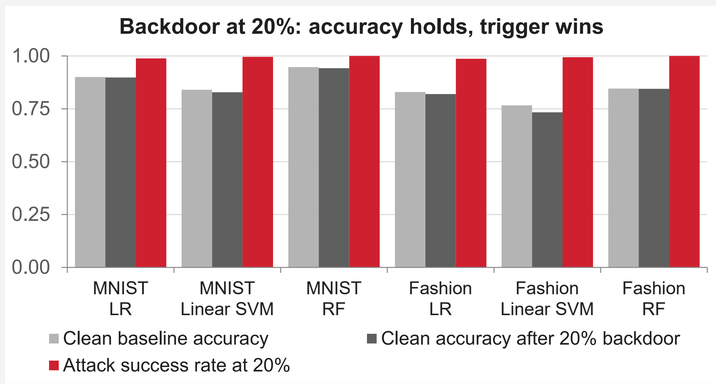
[7] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," arXiv:1708.07747, 2017.

[8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825 to 2830, 2011.

Attack impact at a glance







Key findings

Label flipping was visible. Accuracy and macro-F1 fell as the poisoning rate rose, most for the linear models: Logistic Regression lost 0.0705 accuracy on MNIST at 20%, and Linear SVM lost 0.0660 on Fashion-MNIST at 10%. Random Forest stayed within 0.0075 of its baseline on both datasets.

Backdoor poisoning was covert and effective. Attack success rate ranged from 0.9667 to 1.0000 across all 18 conditions, while clean accuracy moved by at most 0.0115 on MNIST and 0.0330 on Fashion-MNIST (Linear SVM at 20%, where ASR still exceeded 0.99).

The best clean model was not the safest. Random Forest had the highest baseline on both datasets and reached ASR 1.0000 at every rate on MNIST and at 10% and 20% on Fashion-MNIST. Model quality and model trustworthiness are different things.

Implication. Clean-test metrics alone cannot reveal a backdoor. Security evaluation needs targeted measures such as ASR.

Scope and next steps. Classical models, a simple visible trigger, moderate rates, and the same qualitative trends on both datasets: a baseline, not a general claim. Next: clean-label and optimization-based poisoning, adaptive triggers, CNNs and vision transformers, and defenses under the same protocol.

Trustworthy Language Intelligence Lab

Minot State University. Lab page: msaqer.github.io/lab.html

A reproducible Jupyter notebook (seed 42) is available from the authors.

Presented at the 58th Midwest Instruction and Computing Symposium (MICS 2026), Eau Claire, WI, March 27 to 28, 2026.

BE SEEN. BE HEARD. BE YOU.