Paper Review: Privacy Side Channels in Machine Learning Systems

Summary: This paper addresses a critical gap in ML privacy research: the assumption that models operate in isolation, separate from the surrounding system components that perform data filtering, input preprocessing, output monitoring, and query management. The authors show that these components create "privacy side channels" that leak significantly more information than the model alone. They analyze four common system components and demonstrate concrete attacks on each. First, for training data filtering, they study exact and approximate deduplication and show that inserting poisoned samples similar to a target lets an attacker determine membership with near-perfect accuracy; their "hub and spoke" method creates near-duplicates that survive or disappear depending on whether the target is present, and CIFAR-10 experiments show extremely strong membership signals that also break naive differential privacy guarantees. They extend this to poisoning defenses in federated learning by attacking the FoolsGold defense: using a non-iid setup on the LFW dataset, they simulate five clients and insert a malicious client whose updates mimic the target client; the loss on the attacker's data decreases only when the target is absent, enabling 85% membership inference accuracy, far above a strong baseline of 67.5%. Second, they examine input preprocessing in language models and show that tokenizers with fixed context windows can be probed to reconstruct the vocabulary by observing when padding pushes context out of view, revealing rare training substrings such as usernames. Third, they study output post-processing, showing that memorization filters act as perfect non-membership oracles; they use this to infer GitHub Copilot's training data cutoff and to extract OpenSSH private keys from models protected by such filters. Finally, they investigate query filtering, showing that stateful detectors that track global query histories can reveal whether another user previously queried a given input. The key insight is that system-level components intended for privacy or robustness can themselves create severe privacy vulnerabilities, underscoring the need for end-to-end system-level privacy analysis.

Evaluation: The paper makes a strong and compelling case for studying privacy at the level of full ML systems rather than isolated models. Its core problem is highly significant because real deployments always include components such as deduplication, tokenization, memorization filters, and query monitors, yet these elements are almost never included in formal privacy guarantees. The work is novel because it shifts the focus from model behaviour to system interactions and shows that these interactions can completely overturn expected privacy protections. The technical approach is sound, with threat models that reflect realistic attacker capabilities, including limited poisoning influence and standard black-box access. The experimental methodology is broad and thorough, spanning image classification, language modelling, federated learning, and even a deployed production system, which reinforces the claim that these vulnerabilities arise across domains. The comparisons against strong baselines in federated learning and membership inference further demonstrate that the proposed attacks exceed the effectiveness of prior work. The paper successfully illustrates that system design choices have a major impact on privacy and that evaluating only the training algorithm provides a misleading sense of security. The paper is clear, well organized, and explains concepts in an understandable manner despite the technical depth.

Main Takeaways:

1. The privacy of an ML system is driven by how its components (filters, pre-processors, and detectors) interact, not by the model alone, which makes system-level privacy analysis essential.
2. Privacy guarantees like those from DP-SGD can fail once additional pipeline steps are introduced that are not accounted for in the theory, and the paper's experiments across vision, language, and federated learning show that these system interactions often dominate real privacy behaviour.

Strengths

1. The use of the toggleable filter side channel to successfully infer the exact training data cutoff date (October 2021) of the black-box GitHub Copilot system is a powerful practical result.
2. The paper clearly identifies and demonstrates a unique category of attack that leaks other users' private test queries (not just training data) by using query filters, an attack otherwise impossible against isolated models

Weaknesses

1. Some attacks require the attacker to poison the training data or inject carefully crafted inputs, which may not be realistic in many real-world systems.
2. Several attacks depend on extremely large numbers of queries or heavy optimization, which would be impractical on systems that limit or monitor user requests.

Discussion: How can we guarantee privacy when ML systems include many components like deduplication, tokenizers, filters, and stateful detectors?