**Membership Inference Attacks and Defenses in Federated Learning: A Survey**

**Bai, L., Hu, H., Ye, Q., Li, H., Wang, L., & Xu, J. (2024).**

ML has emerged as an integral component in the modern digital ecosystem. Many of us engage with it in ways we don’t even notice. It drives voice assistants like Siri and Google Assistant, guides us through streets using GPS and offers us personalized playlists on services like YouTube, Netflix or Amazon. But ML’s influence extends beyond entertainment and convenience — rather it is a crucial player in areas that directly impact human lives: health care, finance, and government services. For instance, in healthcare ML models assist in detecting diseases more precisely through medical imaging by doctors respectively. In banking, ML lends itself to calculating credit scores and determining loan eligibility. The government is already using ML to detect fraud, perform population analytics and even deploy predictive policing.

Machine learning is so efficient because it can learn by aggregating large volumes of data. These algorithms can detect patterns in the data and apply them to predict or automate complicated sequences of events. Unlike old-fashioned software, which must be written by humans with a set of rules in mind, ML gets better with experience — the more data it is fed, the better the model becomes. Yet it’s this good fortune that has subsumed us beneath a certain anxiety: privacy.

In order to teach a useful ML model, developers require access to large data sets — and frequently, those data sets have sensitive or private information. Consider records of hospital visits, online shopping habits or location data from phones. The information of this type can be extremely private, and individuals can be identified by the details contained within. This all naturally leads to some really important questions that people start asking: “What do you do with my data after using it to train an ML model? Is there a way for me to know if my information was part of it? Could someone take that knowledge and use it against me?”

Those queries point to a big privacy threat of machine learning: Membership Inference Attack (MIA).

Privacy Attacks in learning theory Membership Inference Attack (MIA) is a privacy attack in the area of machine learning. The purpose of this attack is elegant in its simplicity — and dangerous: It aims to ascertain whether a particular piece of data was used to train a machine learning model. On the face of it, this is not so bad — what’s the harm in knowing whether someone’s data has been abused? But the stakes can be quite high.

Now consider, for example, a model trained on sensitive medical records. If an attacker is able to determine that your personal data belonged to the training set while other data belonged to the test set, the attacker might correctly identify that you have a certain disease — even if that fact had never been disclosed outright. The same for datasets with political opinions, personal purchases, or criminal records. Just verifying if a data point is in the training set can compromise individual privacy.

This kind of attack is increasingly relevant — in part because of the way machine learning systems are typically designed and shared. In many ML services to the public, such as APIs from Google Cloud or Amazon Web Services, you don’t get to see the internals of the model (like its architecture or weights). They only observe the output — a prediction label and perhaps the associated confidence score. These are said to be black-box models, since users can’t see what’s going on inside.

A landmark paper by Reza Shokri and his colleagues in 2017 provided a new way of thinking about this issue. They demonstrated that MIAs could function even if you don’t know how a model works from the inside out, provided that you can see its outputs. This was significant because many of today’s popular ML services, such as Google’s or Amazon’s ML platforms, are black-box systems where the only thing a user sees is the outputs. This review examines what Shokri and his colleagues found, how they did it and why their work still matters today.

**Background Study**

To this end, Federated Learning (FL) was invented. Rather than send everyone’s data to a central location, FL allows each individual or institution to train a portion of a model locally and share only an update to the model. That design serves to protect private data — but not exactly. Studies demonstrate that security can leak also under FL, for instance, in form of Membership Inference Attack (MIA).

MIA’s seek to answer one question: “Was a given datapoint used to train this model?” If the answer is yes, it can disclose private information. For instance, if your medical record were used to train a model predicting cancer, perhaps someone would infer you had cancer.

MIAs can be used to:

Violate privacy by verifying if someone’s data appeared in the training set.

Verify compliance with laws such as the GDPR (e.g, check if data actually has been deleted).

Perform more aggressive attacks (e.g., model stealing or data reconstruction).

Although MIAs have been examined in the classical centralized context (where all data is stored in a single place), FL presents some new issues and types of attacker. This manuscript is dedicated to MIAs in FL and provides an exhaustive and current survey that has not been fully filled by previous research.

**Key Findings and Results**

The major results of this paper are:

MIAs are also effective in FL: despite anonymised raw data, an adversary can exploit shared updates or observe prediction trends over time to perform MIAs.

Anomalies can manifest in many ways, including:

A central server can receive and update of models and can MIAs.

A dishonest client (attendee) may forge the updates or collectively analyse them to learn the private state of others.

An observer from a third party can even succeed sometimes, especially when the prediction results are leaked.

The MIAs in FL fall into several categories:

**Update-based attacks**: These attacks focus on the real model update (in the form of gradients or weights) itself. If attackers can see such updates, they can in some cases reverse-engineer how a given record influenced such decisions.

**Prediction/trend attacks**: These follow how the model’s output evolves during training. If a model becomes more certain about some input, it could indicate that this input was present in the training data.

The overfitting : As in centralized settings, models that overfit their training data are also more subject to MIAs in FL.

Existing defenses are insufficient: Though there are ways to mitigate the success of MIAs, no answer is ideal. Most defenses have drawbacks that either harm model performance or add complexity to the system.

**Mitigation Strategies**

In FL, some defense mechanisms against MIAs are presented in the paper. These include:

**Partial Sharing**: Sharing is done partially i.e., sharing of the entire update of the model is not performed but only specific portions (layers,parameters etc.) of it are shared. This is a check on what an attacker is able to see.

**Secure Aggregation** : The updates from different users are combined without revealing individual contributions. Even the central server can’t see what each client sent.

**Differential Privacy (DP)**: Injecting mathematical noise into updates so that it is difficult to identify whether a single record was used in training. Although DP can work, it usually results in much decreased model accuracy.

**Adding Noise**: In addition to DP, other, less formally guaranteed forms of (noise) can be added to updates or outputs to obfuscate attackers.

**Anomaly Detection**: Tries to detect and drop suspicious clients who might be executing MIA’s or poisoning the model.

**Random Masking or Encoding**: Some schemes use random masking or encoding of updates to increase the attack complexity.

Each of these defenses has its own virtues and trade-offs. Some have a negative impact on the efficiency of communication, some reduce the accuracy of the model, while others rely on trusted setup and require cumbersome computations. One of the key challenges is to trade off privacy and performance at scale.

**Importance in Federated Learning**

Federated Learning (FL) is an approach to training ML models over multiple users or devices without direct exposure of raw data. Instead of transmitting data to the central server, users train the model locally and only send its updates to the server. This makes FL sound like a privacy-friendly approach — and in many respects, it is.

This paper is significant because:

Demonstrates that FL is not a complete privacy solution by itself.

Gives a comprehensive picture of the MIA techniques for FL.

Emphasises that attackers inside the system (e.g., servers or clients) are as harmful as outsiders.

Provides a taxonomy (structured map) for all known attacks and defenses in FL.

Talks about how current FL systems might be insufficient for actually preserving privacy.