TML A1 17 Membership Inference Attack Report
Team 17

Maitri Vignesh Shah - 7075780

Yashashri Ajay Balwaik - 7075733

May 27, 2025

1 Task Overview

The goal of this assignment is to create and apply a Membership Inference Attack (MIA) to a ResNet-18 model that was trained using an undisclosed dataset. The model, as well as a public dataset with known membership and a private dataset with unknown membership, are made available to the attacker. Aim is to assign a continuous membership confidence score to every image in the private set. This is our complete implementation, complete with supporting code and the final submission file.

2 Data accessible to the Adversary

- A pre-trained ResNet-18 model (01_MIA.pt)
- A public dataset (pub.pt) containing images, class labels, and ground-truth membership (1 for member, 0 for non-member)
- A private dataset (priv_out.pt) containing images and class labels (membership labels are not provided)
- Channel-wise normalization statistics (mean and standard deviation)

3 Approach used

- The implemented approach can be found in the assignment1_template.py file.
- We extracted multiple types of features from the model for each sample:
- Model-based statistics: confidence, cross-entropy loss, entropy, margin between top predictions, logit differences, top-k entropy, RMIA score, etc.
- Deep features: extracted from the last-second layer of ResNet-18 (512-dimensional vectors)
- Cosine similarities: between the current sample's deep features and the mean feature vectors of known members and non-members (computed from public data)
- After collecting these, we applied PCA (64 components) to compress the deep features and combined them with other features and logits.
- A downstream XGBoost classifier was trained on public data and used to assign confidence scores to private samples.

4 The Reason for Using Deep Features

Compared to logits alone, deep features provide more insight by capturing the internal representation that the model has learnt. Since training data typically lies in well-structured areas of the feature space, it helps in distinguishing between training and unseen samples. Before feeding them into our classifier, we also used them to calculate cosine similarity to member/non-member centroids and applied PCA to reduce noise. This improved our membership inference attack's accuracy and generalization.

5 Why we used PCA on deep features?

Using all 512 dimensions of the deep features directly led to overfitting and redundancy. PCA allowed us to compress them while preserving most of the variance, resulting in a more compact and generalizable representation.

6 Why we chose XGBoost as the classifier?

XGBoost handles mixed-scale features well, is robust to overfitting, and performs strongly on tabular data. It also supports probability outputs needed for confidence scores.

7 Results

- TPR @ FPR = 0.05 (on private set): 0.13133
- AUC (on private set): 0.6670

On validation using a hold-out set from the public data:

- TPR@FPR=0.05: 0.1280
- AUC: 0.6572

8 Other ideas explored

Training Multiple Shadow Models (CNN) with LiRA Scoring
To enable LiRA-based membership inference and improve the victim model's approximation, we tried training 5–10 shadow CNNs for 20–25 epochs each. Although the likelihood estimates were more reliable with this configuration, the training process was time-consuming and computationally demanding, and the shadow models were still insufficient to accurately replicate the victim model. We did not include this approach in the final pipeline due to its high cost and limited benefits.

Ensemble of LightGBM and XGBoost
In order to possibly enhance classification performance through ensembling, we experimented with combining predictions from both XGBoost and LightGBM. But because both models used the same input features and performed similarly, the predictions were highly correlated. Consequently, there was no discernible advantage to the ensemble over XGBoost alone. This implies that more orthogonal or varied features, rather than merely more models, would be needed for future advancements.

9 Files and their Descriptions

assignment1_solution.ipynb – The notebook that contains the whole pipeline, which includes model and dataset loading, feature extraction, PCA, training of the XGBoost classifier, submission generation, and evaluation. All experiments, like validation splits, deep features, pairwise cosine similarity, etc., are included here.

test.csv – The final submission file containing membership confidence scores for the private dataset, generated from the notebook.

README.md – Summary of the method, key components in the notebook, and instructions to reproduce the results.