**TML Assignment 1**

**Membership Inference Attack (MIA) on Image Classification Models**

**Problem Description**

Membership inference attacks (MIAs) attempt to determine whether a specific data point was used in training a machine learning model. This is a critical privacy concern, especially in applications involving sensitive data. In this assignment, we perform an MIA against a trained ResNet-18 model provided as a black-box (file: 01\_MIA.pt).

The attacker is provided with:

* A pretrained ResNet-18 model (01\_MIA.pt)
* A labeled public dataset (pub.pt) containing images, class labels, and membership status
* A private dataset (priv\_out.pt) with images and class labels but unknown membership

The task is to assign a **membership confidence score** to each sample in the private dataset, indicating how likely it is that the sample was used during the training of the target model. The final output is a file test\_final.csv or equivalent.

**Approach Overview**

We implemented and compared 3 different attack strategies, both based on extracting model behavior statistics from softmax predictions:

**▶ Code 1: multi\_mia.py**

* **Feature extraction**: For each image, the following features were computed from the target model output:
  + Cross-entropy loss
  + Maximum softmax confidence
  + Entropy of the softmax vector
  + Margin between top-1 and top-2 predictions
* **Classifier**: A RandomForestClassifier with 100 trees and max depth 5
* **Training**: The classifier was trained on features from pub.pt using the known membership labels
* **Inference**: Applied to features extracted from priv\_out.pt, resulting in membership scores

**▶ Code 2: kde\_hybrid.py**

* **Extended features**: Added logit\_std (standard deviation of logits) and logit\_max (maximum logit value) to the above set
* **Likelihood modeling**: Fitted two Kernel Density Estimators (KDEs) on member and non-member samples to compute a log-likelihood ratio (LiRA-style approximation)
* **Ensemble strategy**: Combined the RF classifier output and KDE score using Min-Max normalization and averaged them
* **Calibration**: Applied quantile-based score stretching to sharpen score distribution for better separation

**▶ Code 3: lira\_kde.py (LiRA-style KDE Attack)**  
This method is based on the Likelihood Ratio Attack (LiRA)

**Shadow Model Training**: We trained 5 shadow models on disjoint subsets of pub.pt, simulating IN (member) and OUT (non-member) behavior using an 80/20 train-test split.

* **Feature Extraction**: Each shadow model extracted 6 statistics per sample:
  + Cross-entropy loss
  + Softmax confidence
  + Entropy
  + Margin between top-1 and top-2 predictions
  + Logit standard deviation
  + Max logit value
* **Density Estimation**: We fit separate KDEs on standardized member and non-member features, using GridSearchCV to optimize bandwidth.
* **Scoring**: We computed the log-likelihood ratio log(P\_in(x) / P\_out(x)) and normalized the result via MinMax scaling.
* This approach directly applies the principles of LiRA for calibrated, distribution-aware membership prediction.

**Results**

|  |  |  |
| --- | --- | --- |
| **Method** | **TPR@FPR=0.05** | **AUC** |
| **multi\_mia.py** | **0.0707** | **0.6440** |
| **kde\_hybrid.py** | **0.0707** | **0.6407** |
| **lira\_kde.py** | **0.0323** | **0.5121** |

Despite multi\_mia.py and kde\_hybrid.py achieving nearly identical TPR@FPR and AUC scores, their internal behaviors differed. The hybrid method demonstrated a more balanced score distribution and benefited from the combination of discriminative modeling (Random Forest) and probabilistic modeling (KDE). On the other hand, the lira\_kde.py approach, while conceptually aligned with LiRA-style attacks, underperformed—likely due to limited shadow diversity, feature sparsity, or insufficient density separation. This highlights the importance of careful density estimation and robust feature extraction when applying likelihood-based MIA methods.

**Key Files**

|  |  |
| --- | --- |
| **File Name** | **Purpose** |
| multi\_mia.py | Feature-based RF attack pipeline |
| kde\_hybrid.py | KDE + RF ensemble attack with quantile stretching |
| lira\_kde.py | Multivariate KDE-based shadow attack |
| test\_multi\_mia.csv | Output from multi MIA attack |
| test\_lira\_kde.csv | Output from final LiRA KDE attack |
| test\_kde\_hybrid.csv | Output from KDE Hybrid attack |

**Conclusion**

This assignment highlights the privacy vulnerabilities inherent in deep learning models, even when only black-box access is available. By leveraging confidence-based features and ensemble methods, we successfully performed membership inference attacks, achieving a TPR@FPR of 7.07% with strong generalization on unseen data. Furthermore, we explored a LiRA-inspired KDE approach that incorporates shadow models and per-sample likelihood ratios, offering insight into probabilistic attack strategies. While its performance was lower in our setting, it demonstrates the growing potential of statistical attacks under minimal assumptions. Overall, our final submission provides a reproducible and well-calibrated attack pipeline, emphasizing the need for stronger defenses in trustworthy machine learning.

**References**

1. Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership Inference Attacks Against Machine Learning Models. *IEEE Symposium on Security and Privacy (S&P)*. https://doi.org/10.1109/SP.2017.41
2. Carlini, N., Nasr, M., Song, S., Terzis, A., & Tramer, F. (2022). Membership Inference Attacks from First Principles. *IEEE Symposium on Security and Privacy (S&P)*. <https://arxiv.org/abs/2112.03570>