Data-Leak Driven Membership Inference Attacks on LeNet5

Bogdan Bîndilă **Mark Bruderer** **Marti Jimenez** **Cristina Racoviță**

Abstract

This research explores the vulnerabilities and defenses of neural networks, focusing on the membership inference attack. Our investigation involves implementing a **grey-box membership inference attack** on a **Lenet5 model**, utilizing the **CIFAR-10 dataset**.

The attack implementation involves strategically sampling a percentage of data from training and testing datasets, creating a demarcation between known and private parts. Employing **kernel density estimation**, we predicted membership by evaluating probabilities of known and private datasets. Moreover, we computed the **Wasserstein distance metric** to analyze what impact has the percentage of data leakage on the success of the attack.

Furthermore, the research extends its focus to defense mechanisms. For example, the **L2 regularization** defense mechanism effectively increased the uncertainty in the model's predictions, making it more difficult to confidently infer membership status.

Code and plots related to this research can be found at:
`https://github.com/Marti2405/MIA-DataLeak`

1 Introduction

Membership Inference Attack (MIA) is a privacy attack in which the adversary wants to establish if a sample was used to train the target model or not. It is interesting that even if the attacker has only black-box access to the model, the attack can still be successfully done since the output of the target is all the adversary needs.

Why is this a privacy issue? Let's consider a target model that has in the training data people who suffer from cancer. If attackers know who is part of this dataset, they will have access to private information about the patients, which leads to data leakage.

We implemented a well-known technique based on the observation that models assign higher probabilities to their training data than test data. The adversary simply thresholds the model's output confidence to determine whether a given data point was used to train the model.

The most known cause of membership attacks is lack of generalization (Yeom et al., 2018), because of the high difference between the confidence of test predictions and train predictions. Therefore, we chose to defend our attack by making our model less overfitted, improving its approximation ability. We achieved that by using L2 regularization, which is detailed in Section 3.

1.1 Research Question

We implemented a grey-box MIA attack on a **Lenet-5 model** and observed *how the awareness of training data affects the success of predicting whether an image is part of the remaining, undisclosed data* and how the regularization impacts the success rate of the attack.

2 Research Background

MIA attacks can be divided into three categories (Niu et al., 2023), based on how much information the attacker has about the target model:

- black-box attack - the attacker knows only the outputs of the target model
- gray-box attack - the attacker knows the distributions of the training data, which were used to train the target model
- white-box attack - the attacker knows all the information of the target model

In (Carlini et al., 2022), the **Likelihood Ratio Attack** (LiRA) is proposed, a powerful MIA attack, which can achieve a much higher true-positive rate at low false-positive rates. To implement LiRA, the attacker collects a set of **IN** (same training data as the target model) and **OUT** (different training data than the target model) models. Then, by comparing the target model outputs on both IN and OUT models, LiRA can identify patterns that reveal whether an example belongs to the training set or not.

Another approach (Shokri et al., 2017) implies mimicking the target model using **shadow models**. A shadow model is trained on a similar distribution data as the target model one and classifies the input as **in** or **out**. Based on these shadow model predictions, an attack model is trained to learn the pattern between data that are included or not in the training data.

In (Hintersdorf et al., 2023), there are two types of attacks assumed from thresholds. **Prediction Score-Based Attacks** rely on the highest score and exploit the top-3 values of the prediction score vector, the maximum value, and the entropy. An example is labeled as a member if the maximum value is higher than a **threshold**. An entropy-based attacks use a similar principle, it computes the entropy on the whole prediction score vector and classifies an input as a member if the entropy is lower than a threshold.

There are **four strategies** to defend against this type of attack: Confidence Masking, Regularization, Differential Privacy, and Knowledge Distillation (Hu et al., 2021). The first one refers to masking the confidence outputs of the target model (e.g. by returning only top-k probabilities). Resolving the lack of generalization is another cause that can be fixed using many approaches (e.g. L2 regularization, data augmentation, etc). **Overfitting** serves as a sufficient, though not necessary, factor in causing membership inference attacks (Yeom et al., 2018), advantaging on the generalization error. To add some noise to the gradient to ensure data privacy or to distillate the target model outputs are the last two techniques presented in the above-mentioned study.

3 Theoretical analysis

3.1 Attack

3.1.1 Why Should the Attack Work?

The effectiveness of the membership inference attack can be theoretically understood through the inherent characteristics of machine learning (ML) models, particularly deep neural networks (DNNs).

Firstly, ML models, including DNNs, are often overparameterized, meaning they possess enough capacity to memorize information from their training dataset. This overparameterization leads to the phenomenon where the model exhibits different behaviors on training data records (members) compared to test data records (non-members). Specifically, during training, the model learns to minimize the prediction loss on its training members, resulting in higher confidence scores for correctly classified training data.

Secondly, the finite size of training datasets and the repetitive nature of training epochs contribute to the model's ability to memorize specific instances. Consequently, the model's parameters store statistically correlated information about individual training data records.

These characteristics enable an attacker to exploit the discrepancy in behavior between training and test data to build attack models that distinguish between members and non-members of the training dataset.

3.1.2 How Does the Attack Work?

The proposed attack leverages the observed differences in prediction loss between training and test data to infer membership status. Specifically, the attack involves the following steps:

1. Obtain access to a percentage of the dataset used in training the victim model and a separate dataset of private images not known by the model.
2. Record the prediction loss for each sample in both datasets.
3. Utilize a kernel density estimator (KDE) to derive the distribution of prediction loss for members and non-members of the training dataset. The KDE formula is given by:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \qquad (1)$$

   where $K(\cdot)$ is the kernel function, x_i are the recorded losses, h is the bandwidth, and n is the number of samples.
4. Establish a threshold based on the distributions to classify samples as members or non-members. The attack function $M_{\text{loss}}(\hat{p}(y|x), y)$ is defined as follows:

$$M_{\text{loss}}(\hat{p}(y|x), y) = \begin{cases} 1 & \text{if } L(\hat{p}(y|x), y) \leq \tau \\ 0 & \text{otherwise} \end{cases}$$

   where $L(\cdot)$ is the cross-entropy loss function and τ is a preset threshold.

5. Assess the impact of the percentage of the dataset on the distributions by measuring the Kullback-Leibler (KL) distance between the distributions for different percentages.

By exploiting the inherent differences in prediction loss between training and test data, coupled with the memorization capabilities of ML models, the attack should identify membership status.

3.2 Defense

3.2.1 Confidence Masking

Confidence masking is a defense strategy that involves limiting the amount of confidence information revealed by the ML model. Instead of providing precise probability scores for each class, the model only outputs the top k most probable classes, where k is a predefined threshold. Mathematically, this can be expressed as:

$$\hat{p}(y|x) = \text{argmax}_k \, p(y|x)$$

where $\hat{p}(y|x)$ represents the masked probability distribution over classes, and $p(y|x)$ is the original probability distribution.

This defense mechanism reduces the granularity of information available to the attacker, making it harder to infer membership status based on confidence scores alone. By limiting the attacker's ability to distinguish between training and test data, confidence masking enhances the privacy of the ML model.

3.2.2 Regularization

Regularization techniques aim to improve the generalization ability of ML models by penalizing overly complex or overfitted models. One common regularization method is L_2 regularization, which adds a penalty term to the loss function based on the magnitude of the model's weights:

$$\text{Loss} = \text{Original Loss} + \lambda \sum_i w_i^2 \tag{2}$$

where λ is the regularization parameter and w_i are the model weights.

By encouraging smoother decision boundaries and reducing the sensitivity of the model to individual training instances, L_2 regularization helps prevent the model from memorizing training data and thus mitigates the risk of membership inference attacks.

3.2.3 Differential Privacy

Differential privacy is a rigorous privacy framework that provides strong guarantees against membership inference attacks. It ensures that the presence or absence of any individual training sample has a negligible impact on the model's output probabilities.

Mathematically, differential privacy is characterized by the ε-differential privacy parameter, which quantifies the level of privacy protection. A randomized mechanism is ε-differentially private if, for any pair of datasets that differ in a single sample, the probability of observing any output is approximately the same.

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \times \Pr[M(D') \in S]$$

where $M(D)$ is the output of the mechanism on dataset D, S is the set of possible outputs, and D' is a neighboring dataset differing in a single sample.

Differential privacy offers strong provable guarantees against membership inference attacks by ensuring that the model's output probabilities are insensitive to individual training samples, thereby protecting the privacy of training data.

3.2.4 Knowledge Distillation

Knowledge distillation is a technique that involves training a smaller, more lightweight model (known as a student model) to mimic the behavior of a larger, more complex model (known as a teacher model). By transferring knowledge from the teacher model to the student model, knowledge distillation can improve the generalization ability of the student model and reduce the risk of overfitting training data.

Mathematically, knowledge distillation involves minimizing the Kullback-Leibler (KL) divergence between the output distributions of the teacher and student models:

$$\text{KL}(P||Q) = \sum_i P(i) \log \left(\frac{P(i)}{Q(i)} \right) \tag{3}$$

where P and Q are the probability distributions of the teacher and student models, respectively.

By transferring knowledge from the teacher model to the student model, knowledge distillation can improve the generalization ability of the student model and reduce the risk of overfitting training data.

In summary, defense strategies against membership inference attacks leverage various mathematical principles and techniques to enhance the privacy and security of machine learning models. By limiting the information available to attackers, improving model generalization, and ensuring strong privacy guarantees, these defenses help mitigate the risk of privacy breaches and safeguard sensitive training data.

4 Implementation

4.1 Data

CIFAR-10 (Canadian Institute for Advanced Research, 10 classes) is a well-known dataset in the field of computer vision and machine learning that contains 60000 of 32x32 color images in **10 classes** presented in Figure 1. We loaded the dataset using the PyTorch dataset library. The only **data processing** that we have done is pixel standardization, making certain features (pixels) with larger values not dominate the learning process.

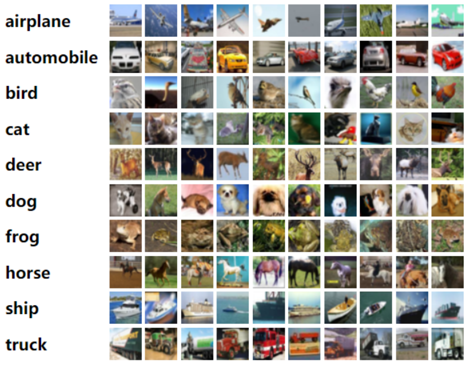

Figure 1: The 10th Classes of CIFAR-10 [source]

4.2 Model

4.2.1 Architecture

Lenet-5 (Lecun et al., 1998) stands out as one of the earliest models that can recognize both handwritten and machine-printed characters. Its popularity is attributed to its simple architecture, featuring a multi-layer convolutional neural network specifically tailored for **image classification**.

Why did we choose this model? Initially, we tried to implement the attack using ResNet-18 (He et al., 2015), a model for which the test and train loss distributions did not overlap; therefore, the attack would work for 100% of the cases. We changed the model to Lenet-5 because the training and the testing losses are well-distributed.

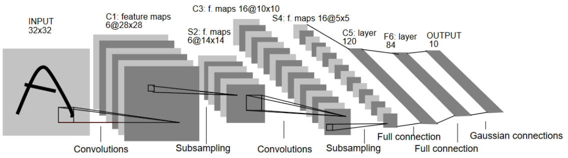

Figure 2: Lenet-5 Architecture (Lecun et al., 1998)

As can be seen in Figure 2, there are seven layers: two sets of convolution layers with a combination of average pooling followed by two fully connected layers and the output layer. The activation function that is used is *tahn*.

As shown in Figure 3, using the **Adam optimizer** (Kingma and Ba, 2014) and setting the learning rate to 0.001, the batch size as 128, and the epochs number to 100, we reached a training accuracy of 92.84% and a test one of 51.52%.

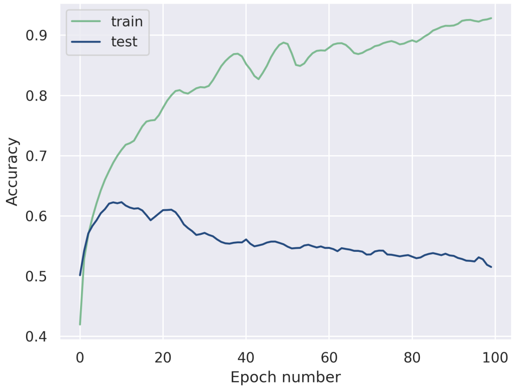

Figure 3: Accuracy evolution during training and testing

4.2.2 Attack

Our initial approach was to model loss arrays as Gaussian distributions (the same idea from (Carlini et al., 2022)) but, out of the three used loss functions, only one is unimodal: the cross entropy (Equation 4), the other two (normalized probability loss and probability loss) being bimodal. We tried to normalize this right-skewed unimodal distribution with the following transformations (West, 2021): square root transformation (for moderate skew), natural logarithm (for high skew), and logarithm in base 10 (for high skew). None of these strategies worked; therefore, we pointed to another approach, which is detailed in the following paragraphs.

Moreover, we calculated the distance using **KL Divergence** (Equation 3), but it was not suitable for our case since we could not compare the computed values for different sample sizes (when the sample size increases, the KL Divergence increases). We scaled this distance by dividing by the number of samples, but the results did not give us a clear pattern as we expected.

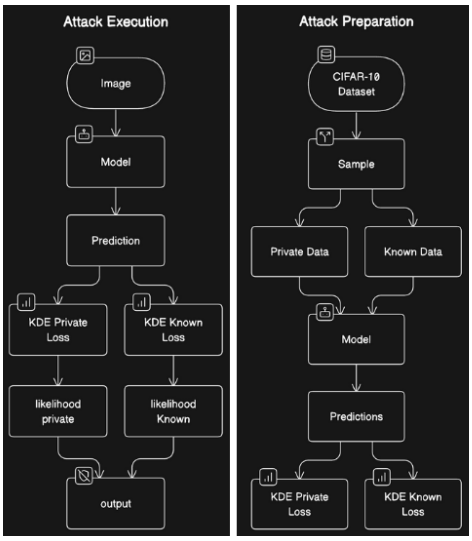

Figure 4: Attack Preparation and Execution

The implementation of the **attack** (Figure 4) starts with retrieving a certain percentage from the training and testing datasets **(from 1% to 16%)**. We can observe in Figure 5 that the sampled data (black points) are spread across the entire data domain, thus, the sampled data is representative of all our data.

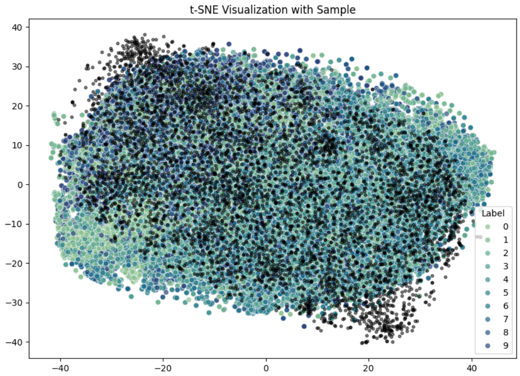

Figure 5: 10% sample of the Dataset

We supposed that the attacker knows this percentage from our training data, thus, each dataset is split into a known part and a private part, which represents the data that the attacker knows for sure it was not used for training the model. We took the private data from the test dataset, but in reality, any images that do not contain one of the ten classes can be used, meaning that they are drawn from another distribution.

With this percentage of data, we randomly sampled that amount of data. We computed the loss arrays for the train known dataset and the private dataset, utilized a **kernel density estimation** (Equation 1) with a Gaussian Kernel, and predicted the membership for our training data. This was predicted by comparing $\frac{\text{known density probability}}{\text{known density probability+private density probability}}$ with 0.5. When it is above this value, we labeled the example as being from the training dataset.

$$H(p,q) = -\sum_{x \in classes} p(x) \log q(x) \tag{4}$$

$$\phi(p) = \log\left(\frac{p}{1-p}\right) \tag{5}$$

Then, we evaluated the membership prediction using another percentage of training and test data, to see the accuracy of our attack. Finally, we used Equation 6 (where $\tau(u,v)$ is the set of probability distributions on whose marginals are and on the first and second factors respectively) to compute the **Wasserstein distance** for every percentage of data, to compare the densities of the loss distributions.

$$l_1(u,v) = inf_{\pi \in \tau(u,v)} \int_{RxR} |x-y| d\pi(x,y) \tag{6}$$

We made experiments for three loss functions (Carlini et al., 2022): probability loss, cross-entropy loss (Equation 4), and normalized probability loss (Equation 5). For each of them, we ran every step ten times, computing the average of the measured metric and plotting the confusion matrices and the Wasserstein distances.

4.2.3 Defense

The implementation of the defense is made on top of the attack. We chose the regularisation approach, using the **L2 regularization** (Equation 2), searching over three values of lambda (strength): *1e-1, 1e-2, 1e-3*. The conclusion is that the value of 1e-1 is too strong, therefore, the train and test accuracy are around 19.2%. The last value made did not regularize the model, resulting in a training accuracy of 89.85% and a test accuracy of 56.21%. The only **lambda value** that fits our needs is **1e-2**, which gave us a training accuracy of **63.22%** and a testing one of **61.19%** (Figure 6).

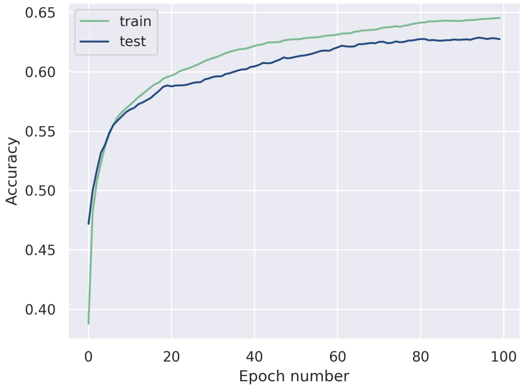

Figure 6: Accuracy evolution during training and testing of the regularized network

5 Numerical Analysis

As we already discussed in Section 4, only the cross entropy loss distribution is unimodal. This can be seen in Figure 7. In those plots can be observed the losses for our model predictions for a data sample at different percentages of leaked training data. When it comes to the second graph, it is illustrated that the model is overly confident in the training set.

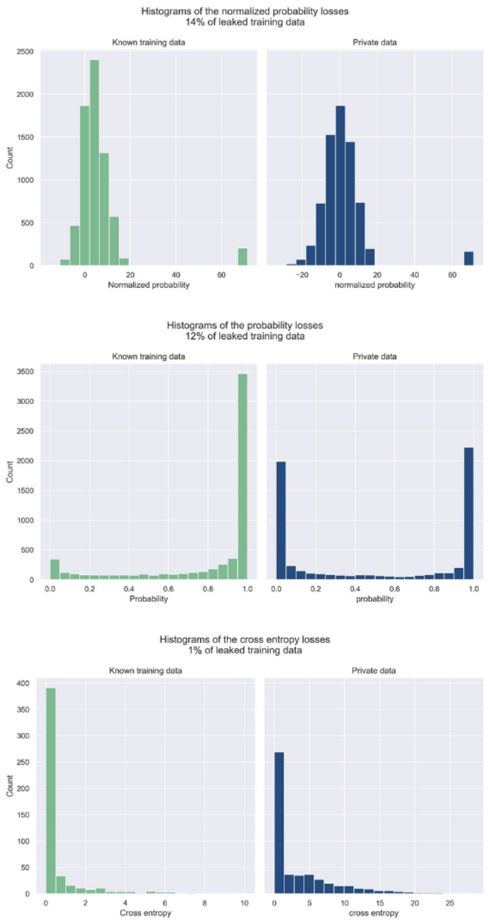

Figure 7: Histogram of every type of loss

5.1 Influence of Known Data Percentage

As illustrated in Figure 11 and Figure 10, we can see that beyond a certain threshold (here 2%), variations in the percentage of known data cease to exert discernible effects on both the False Positive Rate and the Accuracy of the attack.

In Figure 9 and 8, it is illustrated that before the defense, the Wasserstein distance exhibits an increasing trend with the leaked data percentage, meaning that as known and private distributions are more and more different. After the defense, when more data is leaked, the distances decrease, meaning that knowing more data does not increase the attack's accuracy.

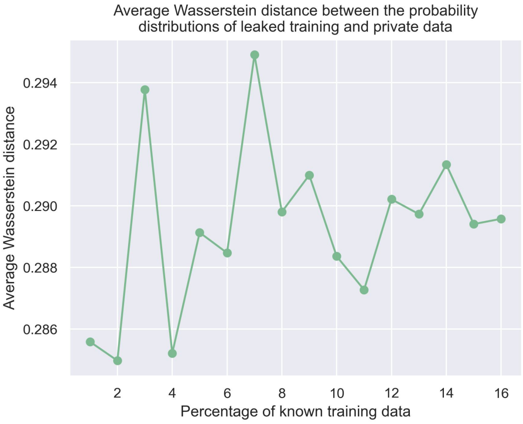

Figure 8: Wasserstein Before Defense

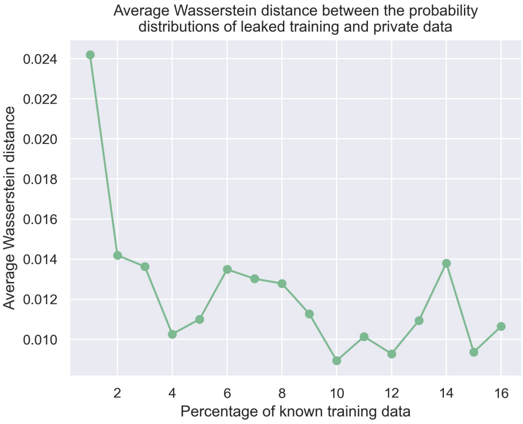

Figure 9: Wasserstein After Defense

This observation underscores the importance of having a sufficient quantity of data to accurately approximate the complete loss distribution of the dataset. Once this threshold is met, additional variations in the percentage of known data seem to have negligible impacts on the performance metrics of the attack.

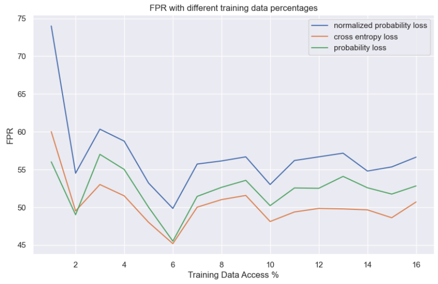

Figure 10: FPR for Different Training Data Percentages Before Defense

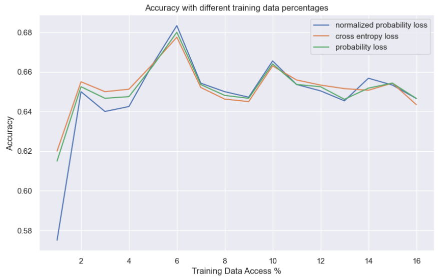

Figure 11: Accuracy for Different Training Data Percentages Before Defense

5.2 Accuracy of our model

5.2.1 Before Defense

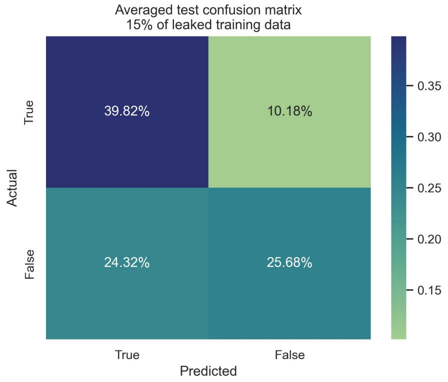

Figure 12: Confusion Matrix for Cross Entropy Model

Before applying the defense mechanism, our model exhibited consistent accuracy across different loss functions and percentages of leaked training data. The confusion matrix illustrated in the figure 12 shows that our model achieved a true positive rate (TPR) of approximately 40%, indicating its ability to correctly predict membership when present. However, the false positive rate (FPR) is around 24%, suggesting that the model occasionally misclassifies non-members as members.

Despite efforts to optimize the FPR, lowering it below 16% proved challenging without sacrificing the TPR. Thus, our model demonstrated limitations in achieving a low FPR while maintaining a reasonable TPR.

5.2.2 After Defense

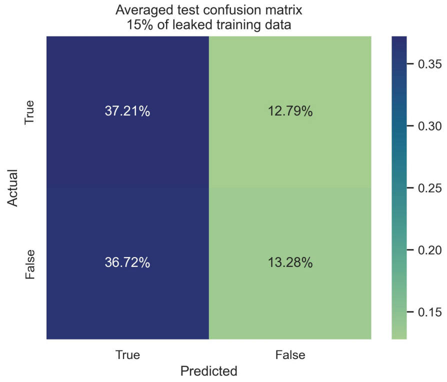

Figure 13: Confusion Matrix for Cross Entropy Model, Defended

After implementing the defense mechanism, our model's performance improved significantly in terms of FPR. As shown in the confusion matrices (Figure 13), the FPR increased, almost matching the TPR. This indicates that the defense mechanism effectively increased the uncertainty in the model's predictions, making it more difficult to confidently infer membership status.

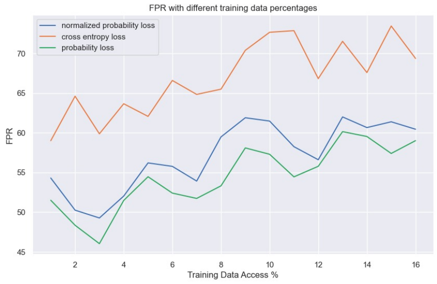

Figure 14: FPR for Different Training Data Percentages After Defense

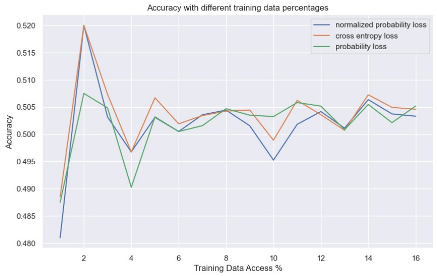

Figure 15: Accuracy for Different Training Data Percentages After Defense

After defending the network, the FPR for the cross-entropy loss stands out, being much larger than the other two (Figure 14). The accuracy of the attack (Figure 15) dropped from more than 64% for most percentages of known data, to values around 50%. This happens for any type of loss, making the attack ineffective.

Overall, the defense mechanism demonstrated robustness against the membership inference attack, significantly mitigating the risk of privacy breaches by increasing the false positive rate and rendering the attack less effective.

This analysis suggests that while the membership inference attack exhibits potential efficacy, particularly if a method for achieving a 0 false positive rate can be devised, the defense mechanism stands as a robust barrier against such intrusions.

6 Improvements and Limitations

6.1 Attack

Further enhancements to the proposed membership inference attack involve exploring **distributions in features beyond the target one**. A comprehensive investigation into diverse data domains would reveal the attack's effectiveness on **various models**. Moreover, introducing an attack model to dynamically learn **optimal patterns** in target model confidences could improve the attack's precision.

Assessing the **generalizability** of the attack methods across different neural network models is crucial for understanding their real-world implications. Moreover, future research could focus on quantifying the relationship between **overfitting percentages** and the success of the membership inference attack.

One clear limitation of our attack is that there is needed a high gap between the test and training loss distributions, otherwise, the attack will not be successful. Therefore, a certain level of overfitting is required.

6.2 Defence

As we presented in Section 3, there are many defense strategies for this type of attack. While our current research focuses on **L2 regularization** as a defense mechanism, exploring alternative methods may reveal more effective approaches. Future research should involve testing multiple defense strategies to analyze which is the best-to-use defense mechanism for our attack. By identifying this, we can enhance the overall security of neural networks, making them more resistant to this type of attack.

7 Conclusion

In summary, our research reveals key aspects of neural network security. We find that **overfitting** makes models more susceptible to attacks, emphasizing the need for robust training practices. In our case, a straightforward **L2 regularization** served as an effective defense strategy. The discovery that a **representative subset of data** is enough for successful attacks adds a better understanding of the model's vulnerabilities.

References

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Dominik Hintersdorf, Lukas Struppek, and Kristian Kersting. 2023. To trust or not to trust prediction scores for membership inference attacks.

Hongsheng Hu, Zoran Salcic, Gillian Dobbie, and Xuyun Zhang. 2021. Membership inference attacks on machine learning: A survey. *CoRR*, abs/2103.07853.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yann Lecun, Leon Bottou, Y. Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324.

Jun Niu, Xiaoyan Zhu, Moxuan Zeng, Ge Zhang, Qingyang Zhao, Chunhui Huang, Yangming Zhang, Suyu An, Yangzhong Wang, Xinghui Yue, Zhipeng He, Weihao Guo, Kuo Shen, Peng Liu, Yulong Shen, Xiaohong Jiang, Jianfeng Ma, and Yuqing Zhang. 2023. Sok: Comparing different membership inference attacks with a comprehensive benchmark.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models.

Robert West. 2021. Best practice in statistics: The use of log transformation. *Annals of Clinical Biochemistry: International Journal of Laboratory Medicine*, 59:000456322110505.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting.