NeuGuard: Lightweight Neuron-Guided Defense against Membership Inference Attacks

Nuo Xu¹, Binghui Wang², Ran Ran¹, Wujie Wen¹, Parv Venkitasubramaniam¹
Lehigh University¹, Illinois Institute of Technology²
Machine learning applications has achieved great success in daily life.
Data is essential for Machine Learning

Datasets contain more and more private attributes and are becoming increasingly valuable.
The membership inference attack (MIA) attempts to determine whether a given data is from the training set or not for a target model.
Neural Network based Membership Inference Attacks

(a) Sorted NN based attack:
Input: $F_{sorted}(x)$
sorted output confidence scores

(b) Unsorted NN based attack:
Input: $(F(x), y)$
unsorted output with label information

(a) R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models,” in 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017, pp. 3–18.
(b) M. Nasr, R. Shokri, and A. Houmansadr, “Machine learning with membership privacy using adversarial regularization,” in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 634–646
Difference of sorted and unsorted NN based attacks

Sorted NN based attack example

Class

Confidence score

Sorted output confidence score

Predict training data

Predict testing data

Sorted NN attack: 75%
Difference of sorted and unsorted NN based attacks

Unsorted NN based attack example

Unsorted output confidence score

Confidence score

Class

Predict training data

Predict testing data

Unsorted NN attack: 100%

Class decision rule

Class decision rule

Class1 training example

Class2 training example

Class1 testing example

Class2 testing example
Motivation

- **Requirements for a good defense**

1. **Defense effectiveness**: Attack accuracy close to a random guess ~50%
2. **Defense Generalizability**: Universal defense for different kinds of attacks
3. **Utility Loss**: Maintain the target model’s accuracy
4. **Overhead**: Low cost, easy implementation, light weighted

| Method               | Defense effectiveness | Defense Generalizability | Model utility | Overhead               |
|----------------------|-----------------------|--------------------------|---------------|-----------------------|
| Normal training      | --                    | --                       | ++            | No                    |
| Dropout [34]         | –                     | –                        | ++            | Low                   |
| Early stopping [37]  | +                     | +                        | –             | No                    |
| AdvReg [27]          | +                     | +                        | –             | Medium (Training)     |
| MemGuard [13]        | ++                    | –                        | ++            | High (Inference)      |

‘++’ indicates the best, ‘--’ means the worst.
Limitation of Existing Solution

- Adversary regularization (*AdeReg* for short) achieves sub-optimal privacy-utility trade off.

  Model accuracy and NN based MI accuracy with different $\lambda$ values for the *AdvReg* on Texas100

| $\lambda$ | 1    | 2    | 3    | 5    |
|-----------|------|------|------|------|
| Training set accuracy | 85.99 | 86.08 | 66.67 | 47.77 |
| Testing set accuracy | 58.51 | 58.17 | 50.92 | 40.59 |
| MI accuracy |       |      |      |      |
| Sorted NN   | 68.56 | 68.31 | 64.18 | 55.81 |
| Unsorted NN | 60.41 | 60.44 | 53.48 | 52.24 |

- Post processing defense: *MemGuard*
  1. Heavy inference overhead
  2. The defense performance of *MemGuard* is highly dependent on the given trained model.
  3. Cannot provide general protection against different attacks.

*M. Nasr, R. Shokri, and A. Houmansadr, “Machine learning with membership privacy using adversarial regularization,” in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 634–646.*

*J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, “Memguard: Defending against black-box membership inference attacks via adversarial examples,” in Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, 2019, pp. 259–274.*
Limitation of Existing Solution

- Adversary regularization (AdeReg for short) achieves sub-optimal privacy-utility trade off.

- Post processing defense: MemGuard
  1. Heavy inference overhead
  2. The defense performance of MemGuard is highly dependent on the given trained model.
  3. Cannot provide general protection against different attacks.

2. The defense performance of MemGuard is highly dependent on the given trained model.

3. Cannot provide general protection against different attacks.
Output confidence distribution for the regular model

- The inference results of a regular trained model, each line indicates the output confidence score distribution for one sample.
- 10,000 training data, 10,000 testing data
Our NeuGuard framework

$L_{\text{var}}$: class-wise variance minimization

$L_{\text{boc}}$: layer-wise balanced output control

\[ L_{\text{boc}} = \frac{1}{S_1} \left\| \left( \vec{h}_1^1 + \vec{h}_2^1 \right) - \left( \vec{h}_3^1 + \vec{h}_4^1 \right) \right\|^2_F \]

\[ + \frac{1}{S_2} \left\| \left( \vec{h}_1^2 + \vec{h}_2^2 \right) - \left( \vec{h}_3^2 + \vec{h}_4^2 \right) \right\|^2_F \]

Mean list \( \mu_y \):

\[ \mu_1^1, \mu_1^2, \ldots, \mu_3^1, \mu_3^2 \]
Class-wise variance minimization

$L_{var}$ to calculate class-wise variance for each input

$$L_{var} = \frac{1}{N} \sum_{i=0}^{N} (F(x) - \mu_y)^2$$

$\mu_y$ is the mean list of corresponding class $y$ to calculate the expectation of the squared deviation of the output.

$\mu_y$ is updated by the prediction results of the correlated class $y$.

- Directly control on the output distribution to close the gap between the output confidence distribution of training data and testing data.
Neural network layer-wise balanced output control

$L_{boc}$ to control the output of each layer

\[ L_{boc} = \sum_{l=1}^{M} \frac{1}{S_l} \left\| \sum_{i=1}^{\lfloor S_l/2 \rfloor} h^l_i - \sum_{i=\lfloor S_l/2 \rfloor+1}^{S_l} h^l_i \right\|_F^2 \]

$M$ is the number of layers.

$S_l$ is the number of layer $l$’s outputs.

$h^l_i$ donates the $i$’s output on layer $l$.

✓ Constrain the effect of individual intermediate output to control the final output.
Neuron regularization-based training flow

Overall loss function

\[ \text{Loss} = L(F) + \alpha \cdot L_{\text{boc}} + \beta \cdot L_{\text{var}} \]

- Use \( \alpha \) and \( \beta \) to control the balance between the optimizing classification task and the effects to constrain the output distribution for the defense.

- Apply layer-wise feature map amplification on convolution layers during the training and inference stage to improve the defense and maintain the utility.

---

**Algorithm 1** Loss calculation using proposed method

1. **Input**: ML model \( F \), a batch of data \((x_B, y_B)\) with \( N \) records, class-wise mean list vector \( \mu_{yB} \), model layer number \( M \)
2. **Output**: Loss value \( \text{Loss} \) calculated for this batch
3. \( \{\text{outputs}, h^1, h^2, ..., h^{M-1}\} = F(x_B) \)
4. \( \text{softout} = \text{softmax}(\text{outputs}) \)
5. for \( l \) in \( M-1 \) do
   6. \( L_{\text{boc}} = L_{\text{boc}} + \frac{1}{S_l} \left| \| \sum_{i=1}^{[S_l/2]} h^l_i - \sum_{i=[S_l/2]+1}^{S_l} h^l_i \|_F \right|^2 \)
7. end for
8. for \( i \) in \( N \) do
   9. \( \text{count}_{y_i} = 1 \)
   10. \( \mu_{y_i} = \mu_{y_i} \frac{\text{count}_{y_i-1}}{\text{count}_{y_i}} + \text{softout}_{i} \frac{\text{count}_{y_i}}{\text{count}_{y_i}} \)
11. end for
12. \( L_{\text{var}} = \frac{1}{n} \sum_{i=0}^{n} (F(x_i) - \mu_{y_i})^2 \)
13. \( \text{Loss} = \text{criterion}(x_B, y_B) + \alpha \cdot L_{\text{boc}} + \beta \cdot L_{\text{var}} \)
Evaluation

● Evaluation metrics:
  • Membership inference (MI) accuracy
  • Testing accuracy
  • Running time: training and inference

● Datasets:
CIFAR10, CIFAR100, Texas100

● Strong adversary that knows a substantial part of the training set and will use it to train the inference attack models.
Evaluation on NN based attacks

Results of compared defenses against NN based MI attacks. Baseline is the normal training without defense.

- **Our NeuGuard** achieves the best utility-privacy trade-off against both the sorted and unsorted attacks among evaluated solutions.

- **Our NeuGuard** has a much smaller overhead.
Analysis of NeuGuard

- **NeuGuard** obtains the smallest variance of the output confidence scores.

| Model               | Baseline | Early stopping | AdvReg | MemGuard | NeuGuard |
|---------------------|----------|----------------|--------|----------|----------|
| Variance: training set | 5.37E-03 | 4.57E-03       | 6.99E-03 | 4.30E-03 | 4.44E-06 |
| Variance: testing set | 4.08E-03 | 3.48E-03       | 5.89E-03 | 3.52E-03 | 3.88E-06 |

Variance of the output confidence scores on the training set and testing set for CIFAR100

- **NeuGuard** delivers the most consistent loss distribution between the training set and testing set.

Loss distribution on CIFAR10 with regular training, adversarial regularization training, and our NeuGuard
Analysis of NeuGuard

- **Visualizing data samples’ output confidence scores.**

Output distribution of training samples and testing samples with class “2” in CIFAR10 for the compared defenses. Each color line indicates one data sample’s output confidence score vector.

(a) MemGuard  
(b) Adversarial regularization  
(c) NeuGuard
Evaluation

Evaluation on label-only MI attacks.

✓ **NeuGuard** delivers the best defense effectiveness against the strong C&W label-only attack.

| Dataset     | Accuracy     | Baseline | MemGuard | Early stopping | AdvReg | NeuGuard |
|-------------|--------------|----------|----------|----------------|--------|----------|
| CIFAR 10    | Testing dataset | 76.6     | 76.6     | 74             | 71.9   | 74.6     |
|             | MI correctness | 61.7     | 61.7     | 59.1           | 58.1   | 58.9     |
|             | C&W label attack | 69.2     | 69.2     | 59.7           | 59.2   | 55.3     |
| CIFAR 100   | Testing dataset | 44.8     | 44.8     | 41.6           | 39.7   | 43       |
|             | MI correctness | 77.5     | 77.5     | 61.6           | 64.6   | 57.8     |
|             | C&W label attack | 80.9     | 80.9     | 61.7           | 63.3   | 54.4     |

The results of C&W label-only attack on Cifar10 and Cifar100 dataset with different defense methods.

Because our target output distribution is smoother and more uniformly distributed. The distance to the decision boundary become similar for AEs.
Conclusion

- Our investigation explores the **difference** of the sorted and unsorted membership inference attacks and demonstrates that existing defenses do not defend against both attacks simultaneously.
- We advocate that a more effective way to defend against MIAs is to orchestrate the output of the training set and testing set for the same explicitly designed distribution that is more evenly distributed in a restricted small range.
- Our proposed *NeuGuard* defense is built upon the technique of **fine-grained neuron-level regularization**, to simultaneously control and guide the final output neurons and hidden neurons towards constructing a defensive model.
- We demonstrate the effectiveness of NeuGuard on three different datasets against not only two NN based MIAs, but also five (strongest) metrics based MIAs including the label-only attack.
Thank you!

Q&A
