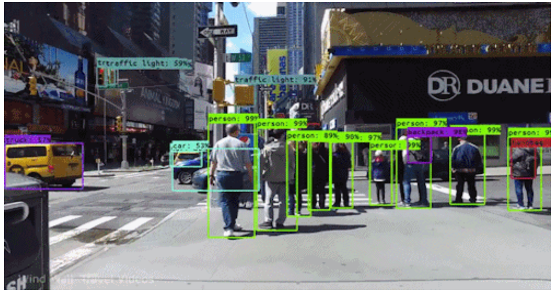
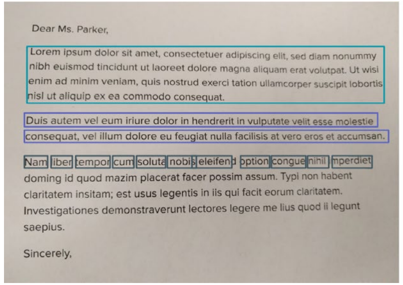
NeuGuard: Lightweight Neuron-Guided Defense against Membership Inference Attacks

Nuo Xu[1], Binghui Wang[2], Ran Ran[1], Wujie Wen[1], Parv Venkitasubramaniam[1]

Lehigh University[1], Illinois Institute of Technology[2]

Machine learning applications has achieved great success in daily life.





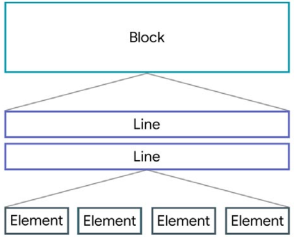
Dear Ms. Parker,

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan.

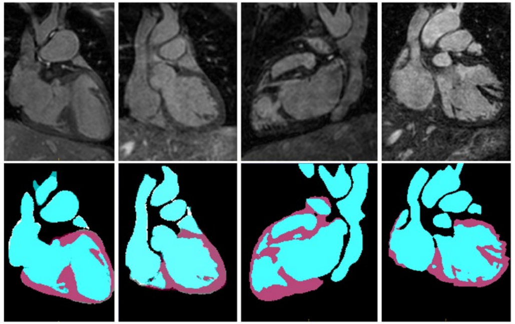
Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

Sincerely,



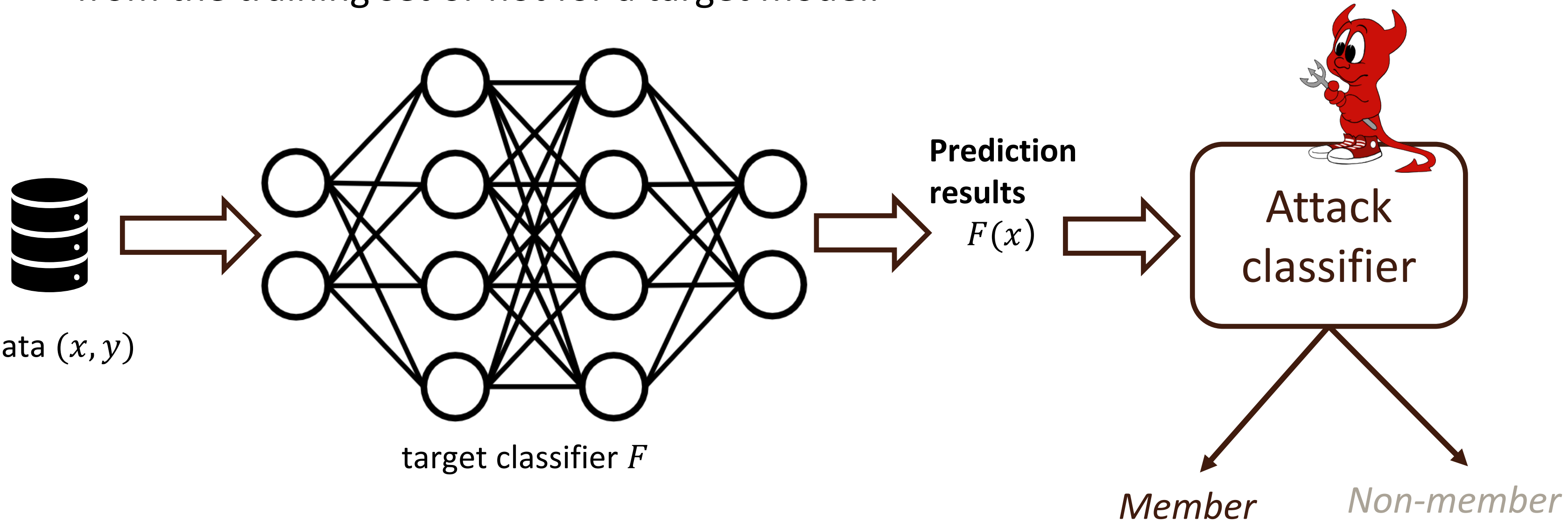
Data is essential for Machine Learning

Datasets contain more and more private attributes and are becoming increasingly valuable.



Membership Inference Attack (MIA)

- The membership inference attack (MIA) attempts to determine whether a given data is from the training set or not for a target model.



Data (x, y)

target classifier F

Prediction results $F(x)$

Attack classifier

Member

Non-member

Neural Network based Membership Inference Attacks

(a) Sorted NN based attack:

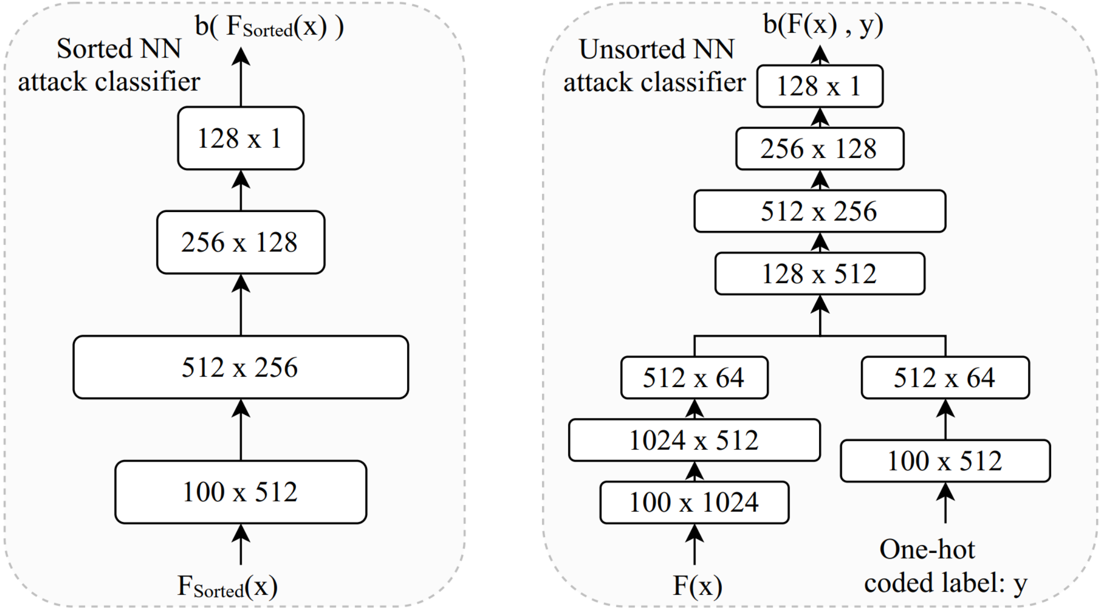
Input: $F_{sorted}(x)$

sorted output confidence scores

(b) Unsorted NN based attack:

Input: $(F(x), y)$

unsorted output with label information



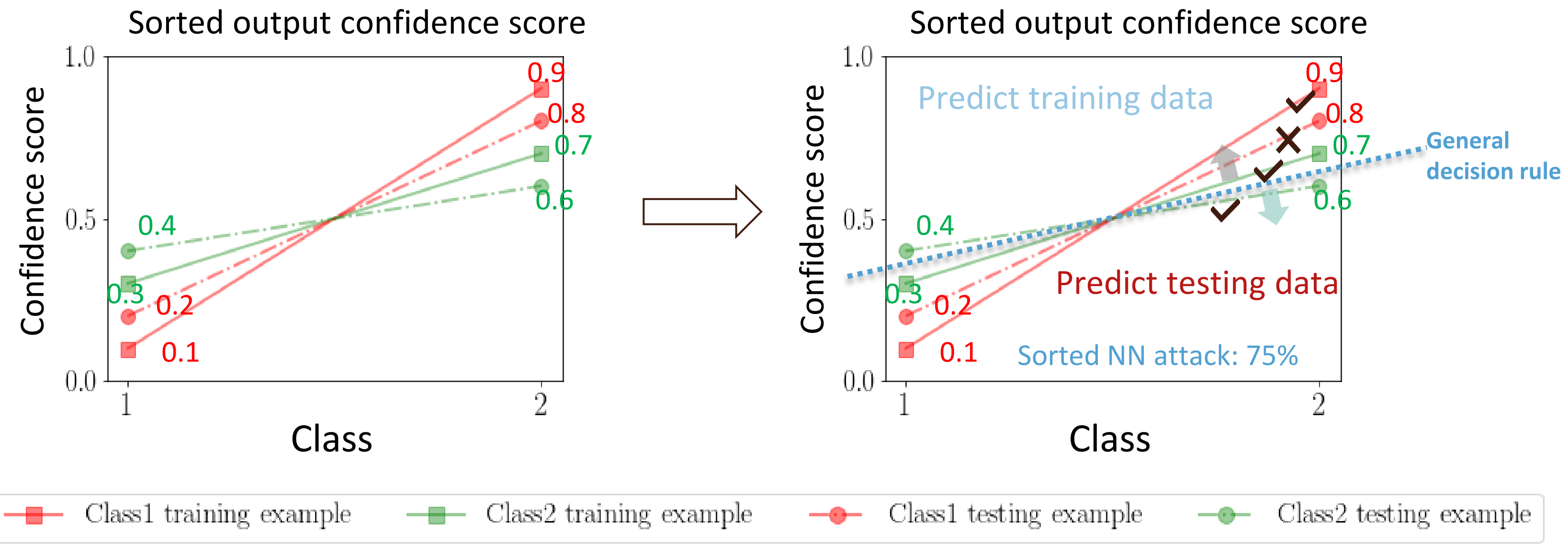
(a) (b)

(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

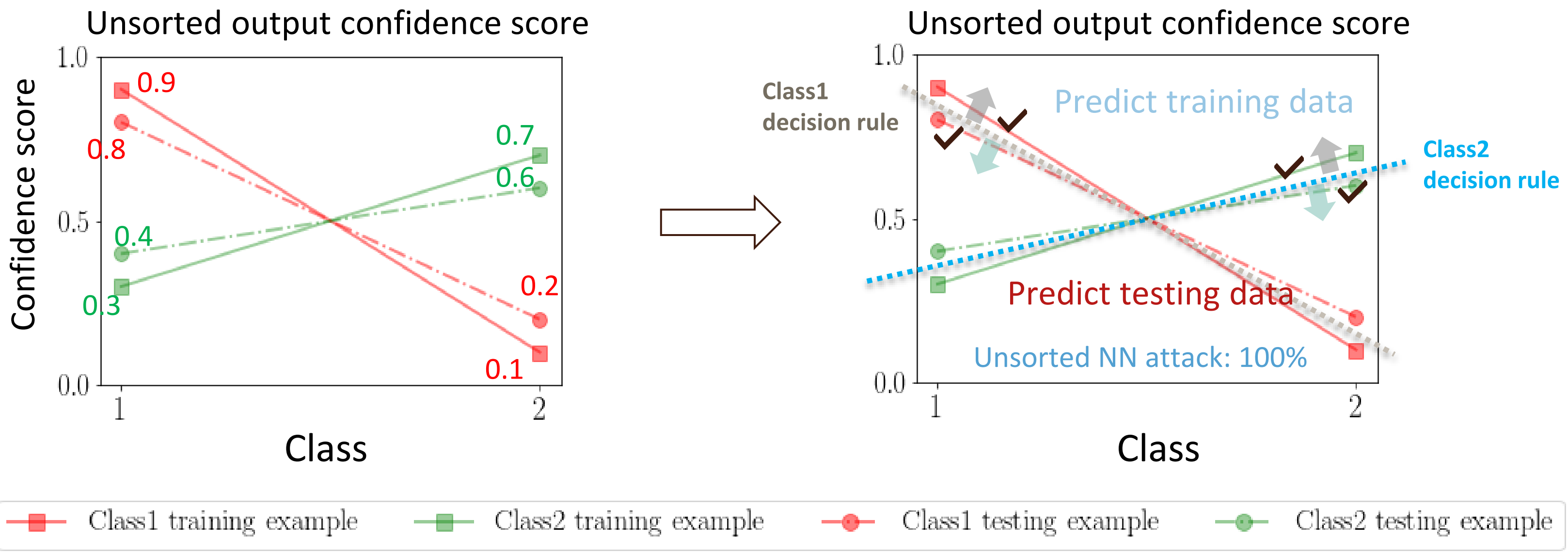
Difference of sorted and unsorted NN based attacks

Sorted NN based attack example



Difference of sorted and unsorted NN based attacks

Unsorted NN based attack example

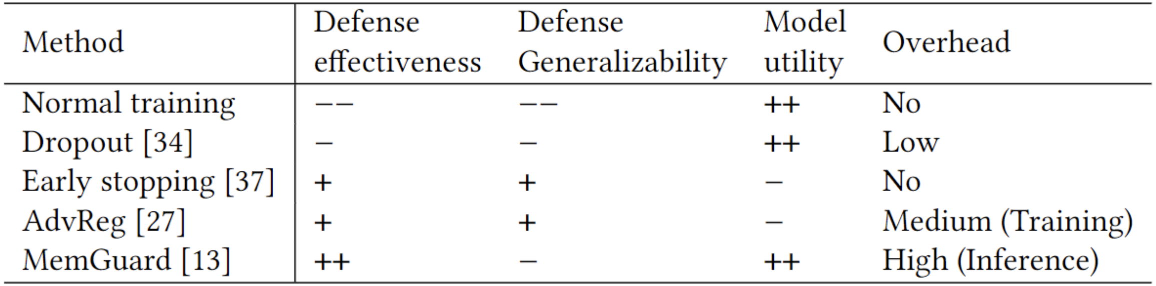


Motivation

❑ *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

Comparing existing MIA defenses



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 λ values for the *AdvReg* on Texas100

λ		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.

based MI accuracy with

λ

g s

How to design a defense method that can

1. maintain the model utility and provide a good utility-defense trade-off?

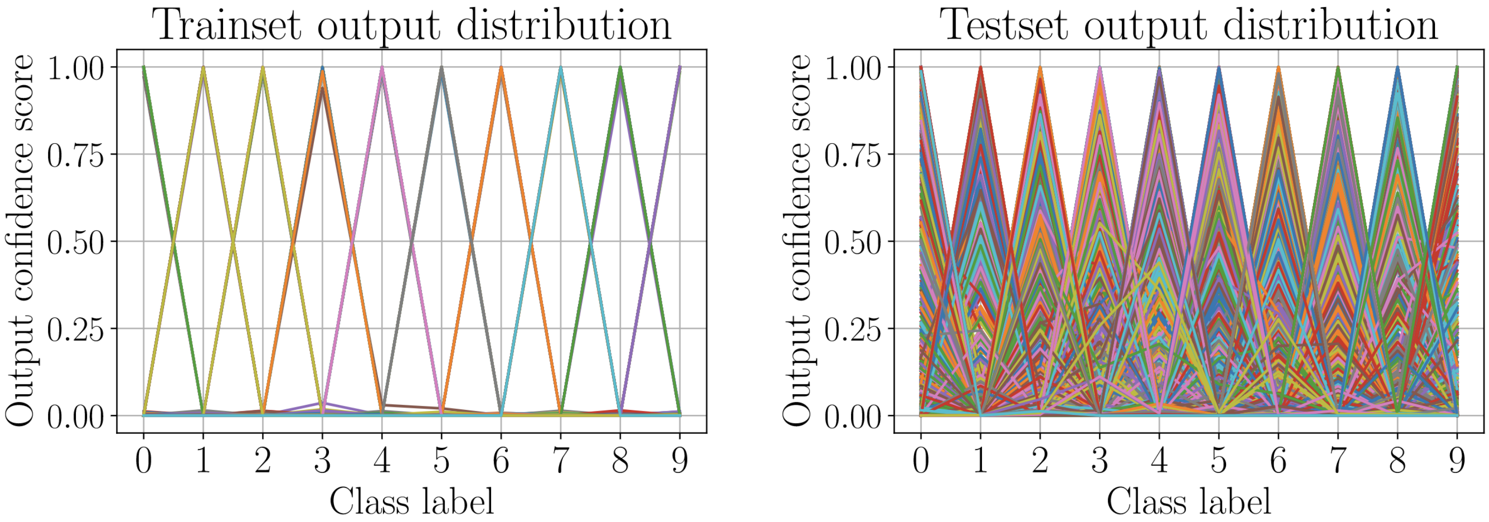
2. work under different MIAs.

3. have low overhead.

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

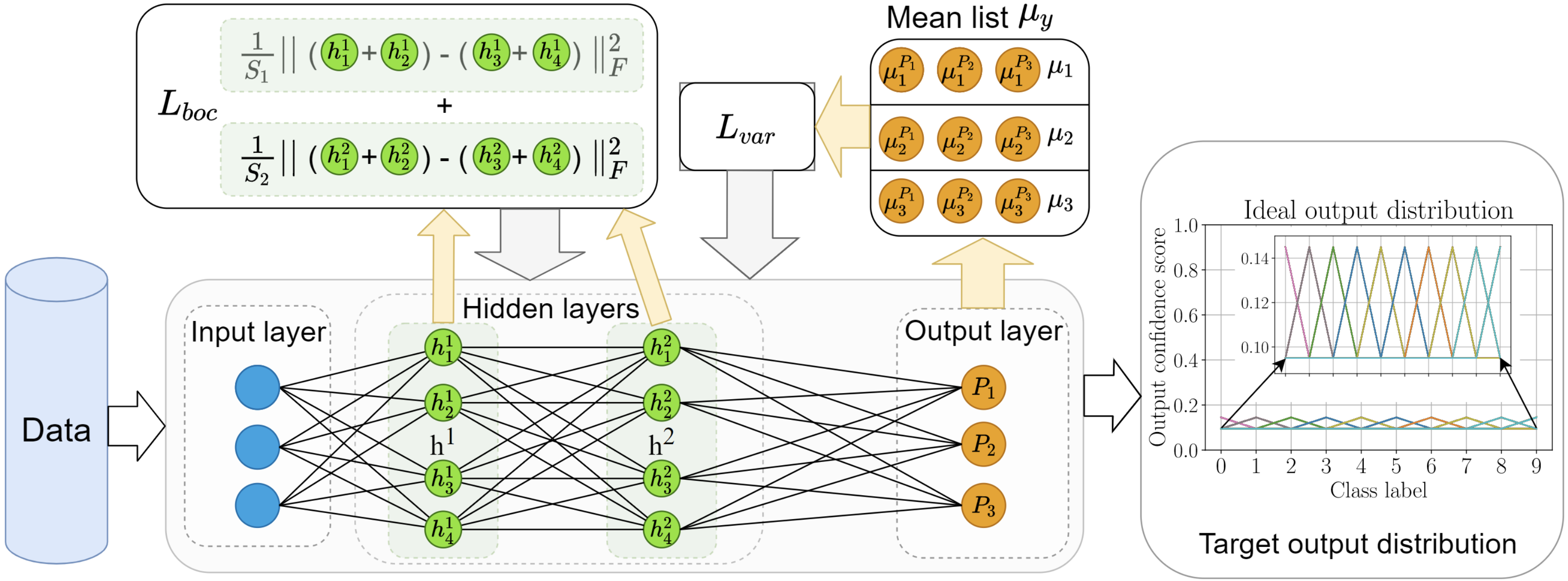


Output confidence score distribution for CIFAR10 task

Our NeuGuard framework

L_{var} : class-wise variance minimization

L_{boc} : layer-wise balanced output control



L_{boc}
$\frac{1}{S_1} \| (h_1^1 + h_2^1) - (h_3^1 + h_4^1) \|_F^2$
+
$\frac{1}{S_2} \| (h_1^2 + h_2^2) - (h_3^2 + h_4^2) \|_F^2$
L_{var}
Mean list μ_y
$\mu_1^{P_1}$ $\mu_1^{P_2}$ $\mu_1^{P_3}$ μ_1
$\mu_2^{P_1}$ $\mu_2^{P_2}$ $\mu_2^{P_3}$ μ_2
$\mu_3^{P_1}$ $\mu_3^{P_2}$ $\mu_3^{P_3}$ μ_3
Data
Input layer
Hidden layers
h_1^1 h_2^1 h^1 h_3^1 h_4^1
h_1^2 h_2^2 h^2 h_3^2 h_4^2
Output layer
P_1 P_2 P_3
Ideal output distribution
Output confidence score
1.0 0.8 0.6 0.4 0.2 0.0
0.14 0.12 0.10
0 1 2 3 4 5 6 7 8 9
Class label
Target output distribution

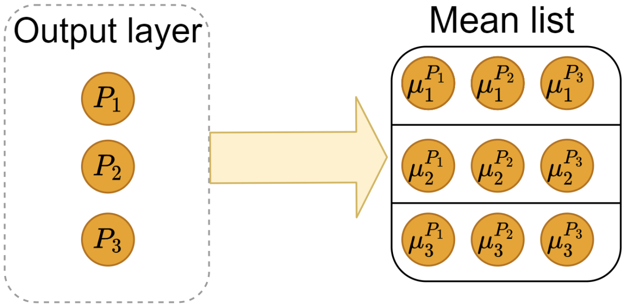
Class-wise variance minimization

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

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

μ_y is the mean list of corresponding class y to calculate the expectation of the squared deviation of the output.

μ_y is updated by the prediction results of the correlated class y.


Output layer
P_1
P_2
P_3
Mean list
$\mu_1^{P_1}$ $\mu_1^{P_2}$ $\mu_1^{P_3}$
$\mu_2^{P_1}$ $\mu_2^{P_2}$ $\mu_2^{P_3}$
$\mu_3^{P_1}$ $\mu_3^{P_2}$ $\mu_3^{P_3}$

✓ 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

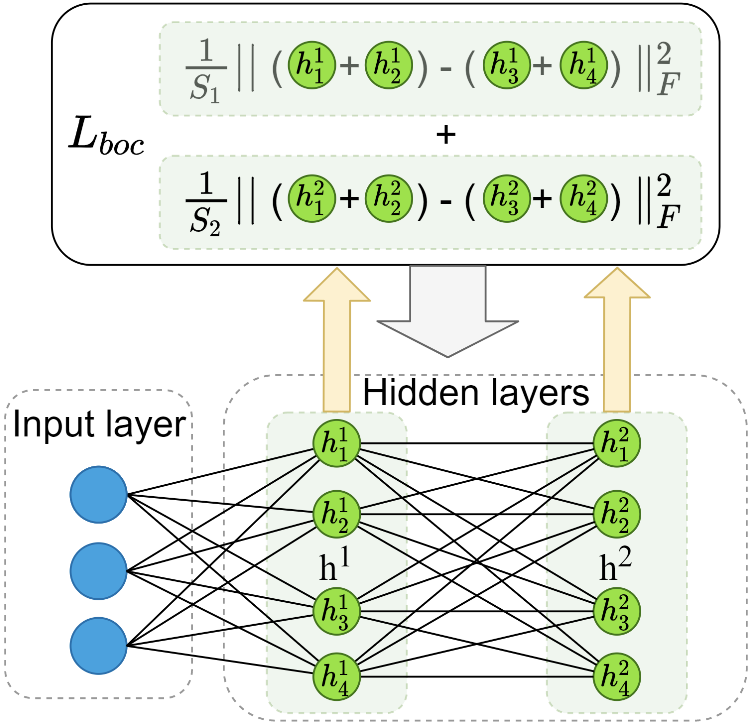
$$L_{boc} = \sum_{l=1}^{M} \frac{1}{S_l} \left\| \sum_{i=1}^{\lfloor S_l/2 \rfloor} h_i^l - \sum_{i=\lfloor S_l/2 \rfloor+1}^{S_l} h_i^l \right\|_F^2$$

M is the number of layers.

S_l is the number of layer l's outputs.

h_i^l donates the i's output on layer l.

✓ Constrain the effect of individual intermediate output to control the final output.



$$L_{boc} \quad \frac{1}{S_1} \left\| (h_1^1 + h_2^1) - (h_3^1 + h_4^1) \right\|_F^2 + \frac{1}{S_2} \left\| (h_1^2 + h_2^2) - (h_3^2 + h_4^2) \right\|_F^2$$

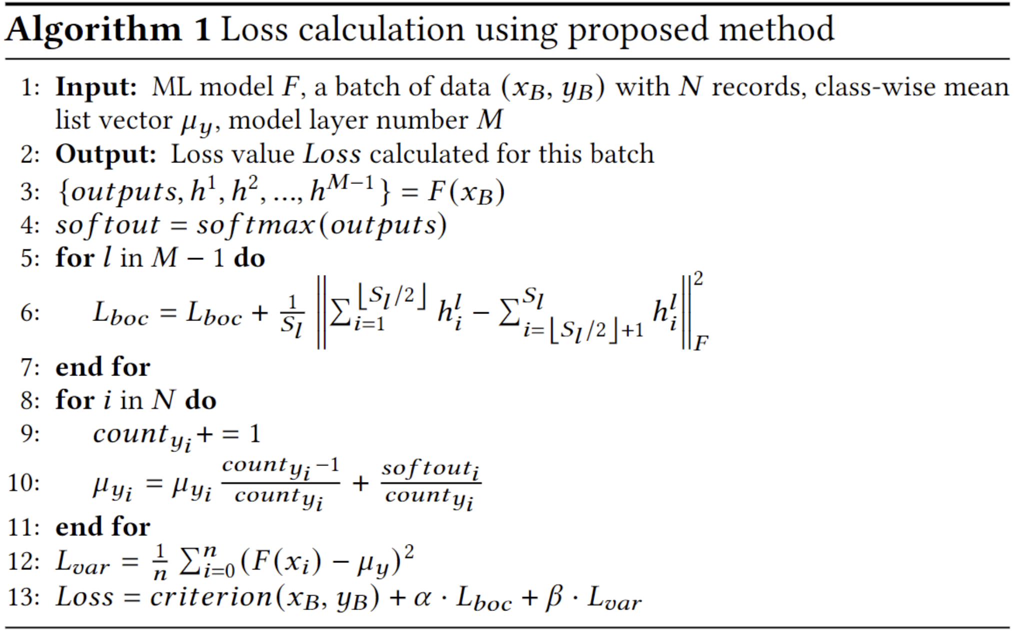
Neuron regularization-based training flow

Overall loss function

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

- Use α and β 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

```
Input: ML model $F$, a batch of data $(x_B, y_B)$ with $N$ records, class-wise mean list vector $\mu_y$, model layer number $M$
Output: Loss value $Loss$ calculated for this batch
$\{outputs, h^1, h^2, ..., h^{M-1}\} = F(x_B)$
$softout = softmax(outputs)$
for $l$ in $M - 1$ do
    $L_{boc} = L_{boc} + \frac{1}{S_l} \left\| \sum_{i=1}^{\lfloor S_l/2 \rfloor} h_i^l - \sum_{i=\lfloor S_l/2 \rfloor+1}^{S_l} h_i^l \right\|_F^2$
end for
for $i$ in $N$ do
    $count_{y_i} += 1$
    $\mu_{y_i} = \mu_{y_i} \frac{count_{y_i}-1}{count_{y_i}} + \frac{softout_i}{count_{y_i}}$
end for
$L_{var} = \frac{1}{n} \sum_{i=0}^{n} (F(x_i) - \mu_y)^2$
$Loss = criterion(x_B, y_B) + \alpha \cdot L_{boc} + \beta \cdot L_{var}$
```

Evaluation

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

- Datasets:

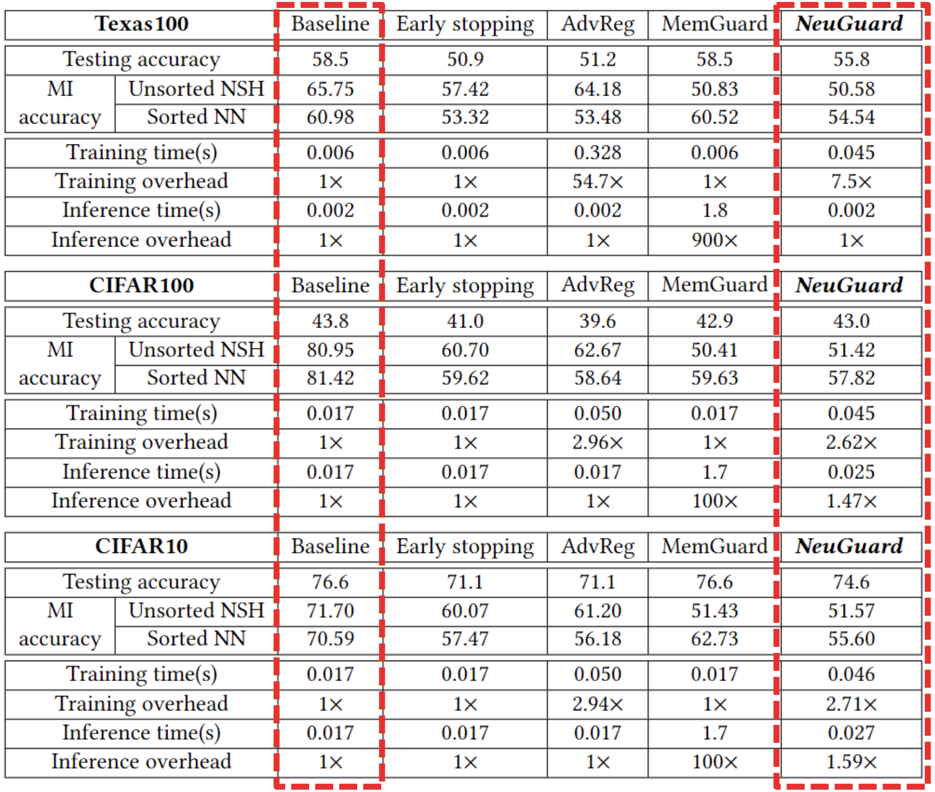
CIFAR10, CIFAR100, Texas100

- 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.



Texas100		Baseline	Early stopping	AdvReg	MemGuard	***NeuGuard***
Testing accuracy		58.5	50.9	51.2	58.5	55.8
MI accuracy	Unsorted NSH	65.75	57.42	64.18	50.83	50.58
	Sorted NN	60.98	53.32	53.48	60.52	54.54
Training time(s)		0.006	0.006	0.328	0.006	0.045
Training overhead		1×	1×	54.7×	1×	7.5×
Inference time(s)		0.002	0.002	0.002	1.8	0.002
Inference overhead		1×	1×	1×	900×	1×
CIFAR100		Baseline	Early stopping	AdvReg	MemGuard	***NeuGuard***
Testing accuracy		43.8	41.0	39.6	42.9	43.0
MI accuracy	Unsorted NSH	80.95	60.70	62.67	50.41	51.42
	Sorted NN	81.42	59.62	58.64	59.63	57.82
Training time(s)		0.017	0.017	0.050	0.017	0.045
Training overhead		1×	1×	2.96×	1×	2.62×
Inference time(s)		0.017	0.017	0.017	1.7	0.025
Inference overhead		1×	1×	1×	100×	1.47×
CIFAR10		Baseline	Early stopping	AdvReg	MemGuard	***NeuGuard***
Testing accuracy		76.6	71.1	71.1	76.6	74.6
MI accuracy	Unsorted NSH	71.70	60.07	61.20	51.43	51.57
	Sorted NN	70.59	57.47	56.18	62.73	55.60
Training time(s)		0.017	0.017	0.050	0.017	0.046
Training overhead		1×	1×	2.94×	1×	2.71×
Inference time(s)		0.017	0.017	0.017	1.7	0.027
Inference overhead		1×	1×	1×	100×	1.59×

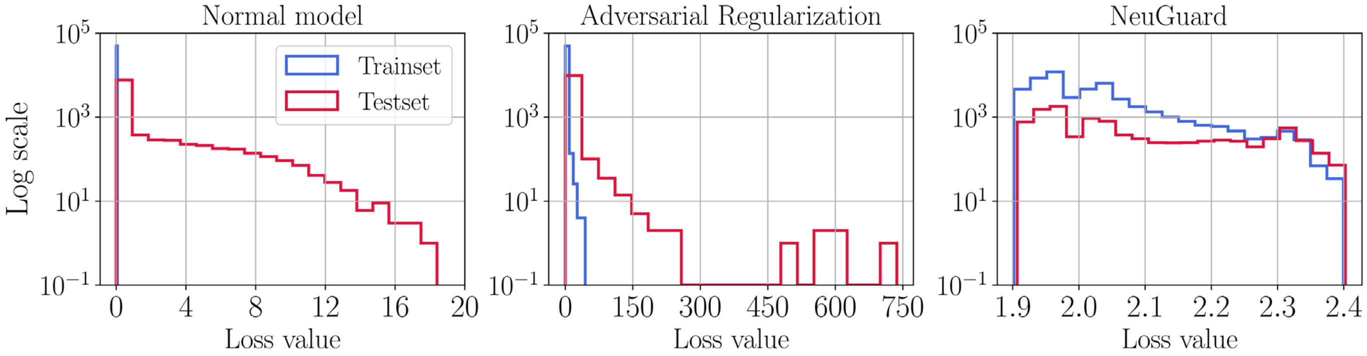
Analysis of *NeuGuard*

- ***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

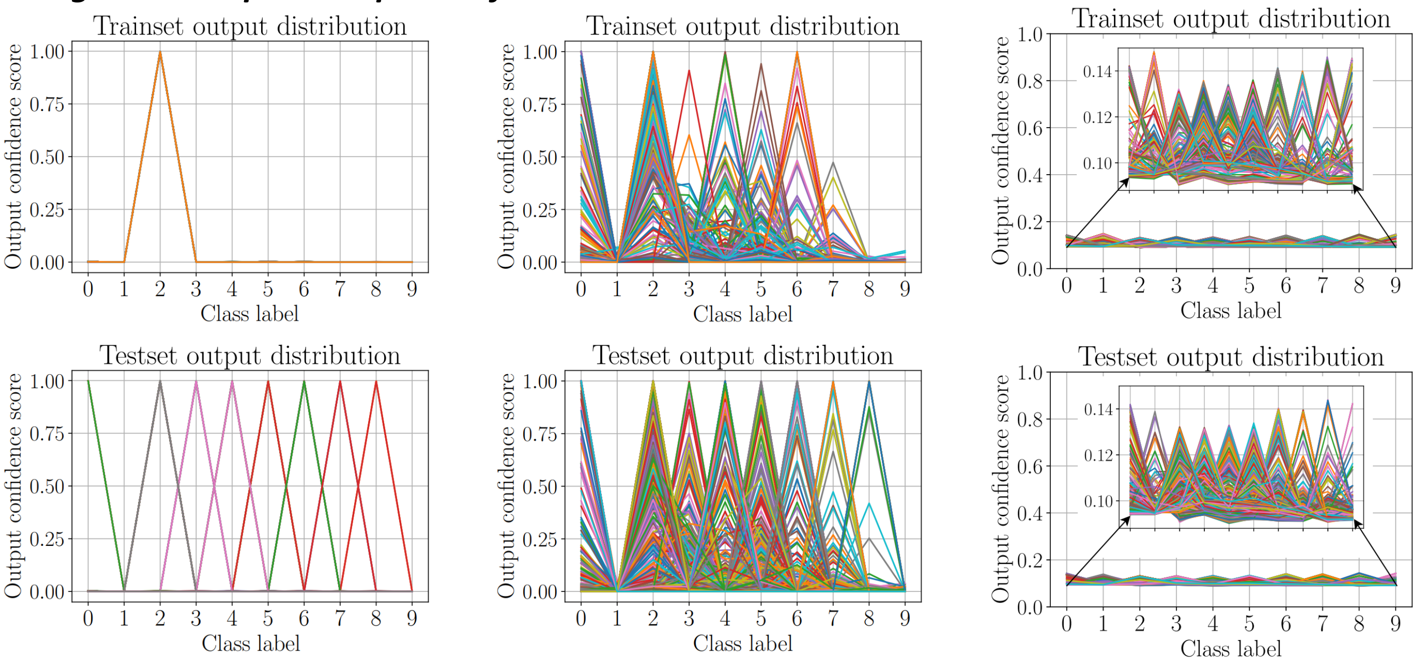
- ***NeuGuard* delivers the most consistent loss distribution between the training set and testing set.**



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

Analysis of *NeuGuard*

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



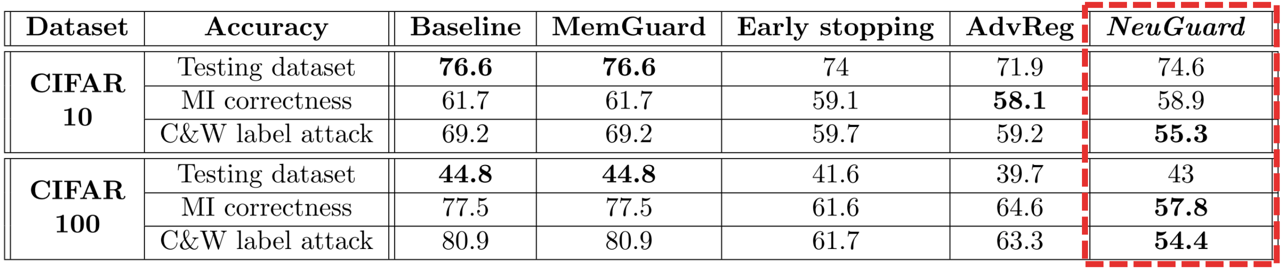
(a) MemGuard (b) Adversarial regularization (c) *NeuGuard*

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.

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

LEHIGH

UNIVERSITY