

CS 499/579: TRUSTWORTHY ML RECONSTRUCTION ATTACKS

Tu/Th 4:00 – 5:50 pm

Sanghyun Hong

sanghyun.hong@oregonstate.edu



Oregon State
University

SAIL

Secure AI Systems Lab

WHY RECONSTRUCTION MATTERS?

- You're a developer who write code for Google's core products¹


The screenshot shows a GitHub search interface for the repository 'QUARTZ'. The search results are sorted by 'Best match'. A red box highlights a result under the 'Code' tab, showing a snippet of code with a Slack API token: `api_token: "xoxp-????????????????????????????????????"`. The search results are categorized by 'Repositories' (7,448), 'Code' (7,448), 'Issues' (35), and 'Users' (2). The 'Languages' section lists various programming languages and their counts: Jupyter Notebook (368), JavaScript (163), HTML (153), SVG (145), Python (132), Markdown (125), JSON (101), PHP (100), XML (71), and Text (59).

The screenshot shows the GitHub Copilot interface. At the top, it says 'GitHub Copilot' and 'Learn more >'. Below that, it says 'Technical Preview' and 'Your AI pair programmer'. A 'Sign up >' button is visible. The main part of the interface shows a code editor with a snippet of JavaScript code for a sentiment analysis function. The code is as follows:

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch("http://text-processing.com/api/sentiment/", {
9     method: "POST",
10    body: "text=${text}",
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

WHY RECONSTRUCTION MATTERS?

- You're a CEO sending emails to your clients¹

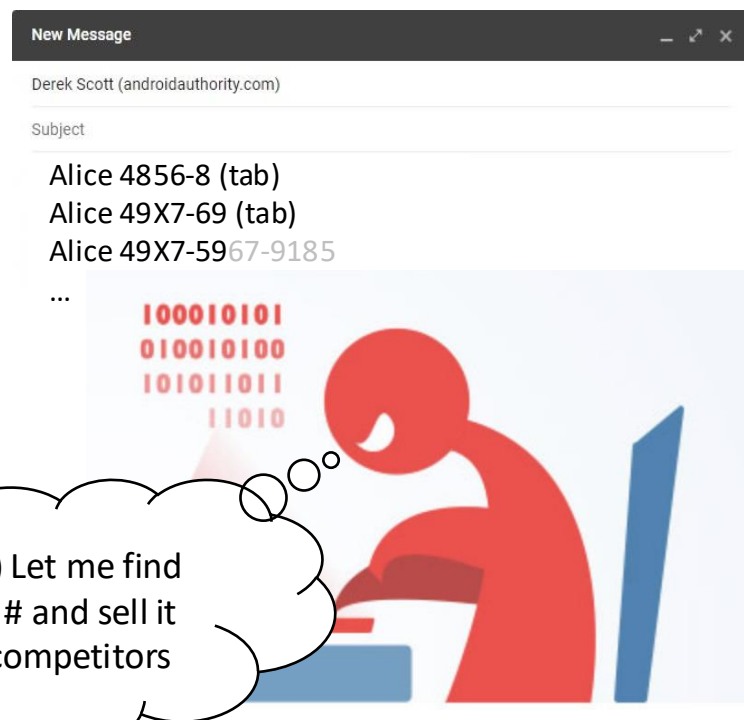


What should I prepare for the next schedule?

Hi John Doe,

It was nice to meet you.
Alice will follow up with this contract #: **49X7-5967-9185**
....

(Insider) Let me find out this # and sell it to our competitors



New Message

Derek Scott (androidauthority.com)

Subject

Alice 4856-8 (tab)
Alice 49X7-69 (tab)
Alice 49X7-5967-9185
...

100010101
010010100
101011011
11010

WHY RECONSTRUCTION MATTERS?

- What about computer vision? [\[Link\]](#)
 - Can we find some random inputs that synthesize my face(s)?



HOW CAN WE RECONSTRUCT INPUTS FROM ML MODELS?

MODEL INVERSION ATTACKS THAT EXPLOIT CONFIDENCE INFORMATION AND BASIC COUNTERMEASURES, FREDRICKSON ET AL., ACM CCS 2015

MODEL INVERSION ATTACKS

- Threat Model

- Objective:

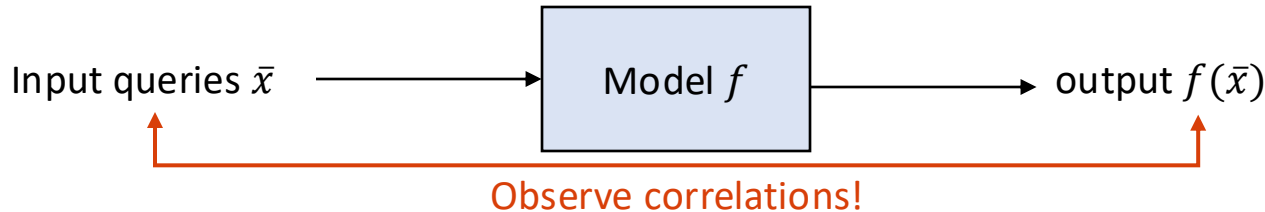
- Extract the secret (feature) x_i of an input (x_1, \dots, x_d) from an ML model f 's output

- Capability:

- An adversary can query the model f with a set of inputs*

- Knowledge:

- f 's output, *i.e.*, confidence scores (vector)
 - *auxiliary* information about the data (or feature) distributions
 - [white-box] f 's model parameters, but it's not that interesting



MODEL INVERSION ATTACKS

- Fredrikson *et al.* attack

- Setup:

- A linear regression model f
 - A target $(x_1, x_2, \dots, x_d, y)$, where (x_2, \dots, x_d) and its label y are known
 - Marginal priors (p_1, p_2, \dots, p_d) are known, too
 - Objective is to find out a secret x_1

- Procedure:

adversary $\mathcal{A}^f(\text{err}, \mathbf{p}_i, \mathbf{x}_2, \dots, \mathbf{x}_t, y)$:

1: **for** each possible value v of \mathbf{x}_1 **do**

// for all the possible values of v

2: $\mathbf{x}' = (v, \mathbf{x}_2, \dots, \mathbf{x}_t)$

3: $\mathbf{r}_v \leftarrow \text{err}(y, f(\mathbf{x}')) \cdot \prod_i \mathbf{p}_i(\mathbf{x}_i)$

// compute the correctness of (v, x_2, \dots, x_d, y)

4: Return $\arg \max_v \mathbf{r}_v$

// return v that maximizes the correctness

MODEL INVERSION ATTACKS

- Fredrikson *et al.* attack on decision tree

- Preliminaries:

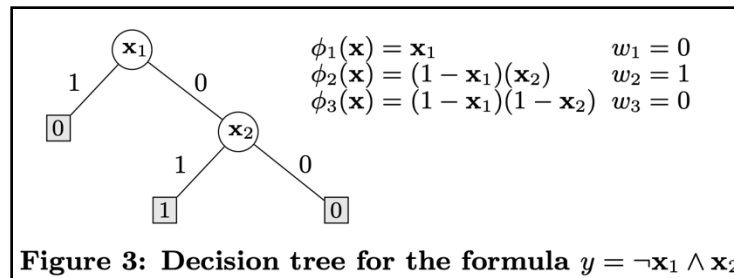
- Decision tree recursively partitions the feature space into m disjoint regions R_i
 - For a sample (\mathbf{x}, y) , f recursively finds the region containing \mathbf{x} and returns y
 - Formally, $f(\mathbf{x}) = \sum_{i=1}^m w_i \phi_i(\mathbf{x})$, where $\phi_i(\mathbf{x}) \in \{0, 1\}$

- Classification and confidence

$$f(\mathbf{x}) = \arg \max_j \left(\sum_{i=1}^m w_i [j] \phi_i(\mathbf{x}) \right), \text{ and}$$

$$\tilde{f}(\mathbf{x}) = \left[\frac{w_{i^*}[1]}{\sum_i w_i [i]}, \dots, \frac{w_{i^*}[|Y|]}{\sum_i w_i [i]} \right]$$

- Prediction will be one of m classes



MODEL INVERSION ATTACKS

- Fredrikson *et al.* attack on decision tree

- Setup:

- A trained decision tree f
 - A target $(x_1, x_2, \dots, x_d, y)$, where (x_l, \dots, x_d, y) is known $l \geq 2$
 - A confidence score matrix \mathbf{C} is known
 - Objective is to find out a secret x_1

- Attacks

- Black-box: use the \mathbf{C} to define $\text{err}(y, y')$ as $\Pr[f(x') = y' \mid y \text{ is the oracle label}]$

- Example:

- 3 features (x_1, x_2, x_3)
 - x_1 is the secret in $\{0, 1\}$
 - y is one of $\{0, 1, 2\}$, and

An adversary examines two samples:

A ($y = 0$): \mathbf{C} is $\{0.5, 0.4, 0.1\} \mid x_1 = 0$ and $\{0.2, 0.6, 0.2\} \mid x_1 = 1$

B ($y = 1$): \mathbf{C} is $\{0.5, 0.4, 0.1\} \mid x_1 = 0$ and $\{0.8, 0.1, 0.1\} \mid x_1 = 1$

MODEL INVERSION ATTACKS

- Fredrikson *et al.* attack on decision tree
 - Setup:
 - A trained decision tree f
 - A target $(x_1, x_2, \dots, x_d, y)$, where (x_l, \dots, x_d, y) is known $l \geq 2$
 - A confidence score matrix \mathbf{C} is known
 - Objective is to find out a secret x_1
 - Attacks
 - Black-box: use the \mathbf{C} to define $\text{err}(y, y')$ as $\Pr[f(x') = y' \mid y \text{ is the oracle label}]$
 - White-box: we further knows p_i 's from the w_i of f and ϕ_i (basis)

EVALUATION

- Setup

- Datasets (50% train + 50% test):
 - FiveThirtyEight survey
 - GSS marital happiness survey
- Models: 100 decision trees (binary classifiers with two labels “Yes” or “No”)
- Metrics:
 - Accuracy (in overall) and precision, recall (on Yes answers)
- Baselines:
 - Random: a brute-force attack
 - Baseline: an attacker has only the access to marginal distributions; no access to f
 - Ideal: an attacker has the access to f' , a decision tree to predict sensitive attribute

EVALUATION

- Results

algorithm	FiveThirtyEight			GSS		
	acc.	prec.	rec.	acc.	prec.	rec.
<i>whitebox</i>	86.4	100.0	21.1	80.3	100.0	0.7
<i>blackbox</i>	85.8	85.7	21.1	80.0	38.8	1.0
<i>random</i>	50.0	50.0	50.0	50.0	50.0	50.0
<i>baseline</i>	82.9	0.0	0.0	82.0	0.0	0.0
<i>ideal</i>	99.8	100.0	98.6	80.3	61.5	2.3

- Summary:

- Precision: Ideal = white-box > black-box > random >> baseline
- Recall: Ideal > random >> white-box = black-box >> baseline
 - Due to the skewed prior distribution: 80% of sensitive attributes are “No”

MODEL INVERSION ATTACKS

- Fredrikson *et al.* attack on face rec. models

- Setup:

- A trained face recognition model f
- Objective:

- Reconstruction: from the label (a person's name), produce an image of the person
- De-blurring: from an image with a blurred-out face, recover the identity

- Attack:

Algorithm 1 Inversion attack for facial recognition models.

```
1: function MI-FACE( $label, \alpha, \beta, \gamma, \lambda$ )
2:    $c(\mathbf{x}) \stackrel{\text{def}}{=} 1 - \tilde{f}_{label}(\mathbf{x}) + \text{AUXTERM}(\mathbf{x})$ 
3:    $\mathbf{x}_0 \leftarrow \mathbf{0}$ 
4:   for  $i \leftarrow 1 \dots \alpha$  do
5:      $\mathbf{x}_i \leftarrow \text{PROCESS}(\mathbf{x}_{i-1} - \lambda \cdot \nabla c(\mathbf{x}_{i-1}))$ 
6:     if  $c(\mathbf{x}_i) \geq \max(c(\mathbf{x}_{i-1}), \dots, c(\mathbf{x}_{i-\beta}))$  then
7:       break
8:     if  $c(\mathbf{x}_i) \leq \gamma$  then
9:       break
10:  return  $[\arg \min_{\mathbf{x}_i} (c(\mathbf{x}_i)), \min_{\mathbf{x}_i} (c(\mathbf{x}_i))]$ 
```

Algorithm 2 Processing function for stacked DAE.

```
function PROCESS-DAE( $\mathbf{x}$ )
   $encoder.DECODE(\mathbf{x})$ 
   $\mathbf{x} \leftarrow \text{NLMEANSDENOISE}(\mathbf{x})$ 
   $\mathbf{x} \leftarrow \text{SHARPEN}(\mathbf{x})$ 
  return  $encoder.ENCODE(vecx)$ 
```

// f_{label} is the one-vs-rest classifier for the label

// update the image x to minimize the error c

// stop x when we find the min. loss

EVALUATION

- Setup
 - Datasets:
 - AT&T Laboratories Cambridge database
 - 400 images over 40 individuals
 - 70% chosen for the train-set; the rest 30% is for the test-set
 - Models:
 - Softmax regression | MLP | Stacked denoising autoencoder
 - Metrics:
 - Use human evaluators (AMT)
 - > 1000 participants over the entire 40 individuals
 - Each participant requires to match the reconstructed face to one of 5 given individuals

EVALUATION

- Results

- **Costs:**

- Per attack: 1.4sec (softmax) << 693 sec (DAE) << 1298 sec (MLP)
 - Per attack: 5.6 epochs (softmax) << 3096 epoch (MLP) << 4728.5 epoch (DAE)

- **Accuracy:**

- Overall: ~80% acc. (softmax) > 60% acc. (MLP) > 55% acc. (DAE)
 - Skilled workers: ~95% acc. (softmax) > 80% acc. (MLP) > 75% acc. (DAE)



Target



Softmax



MLP



DAE

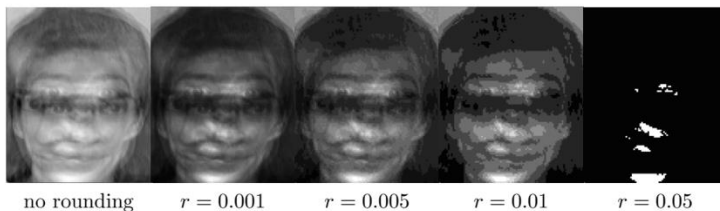
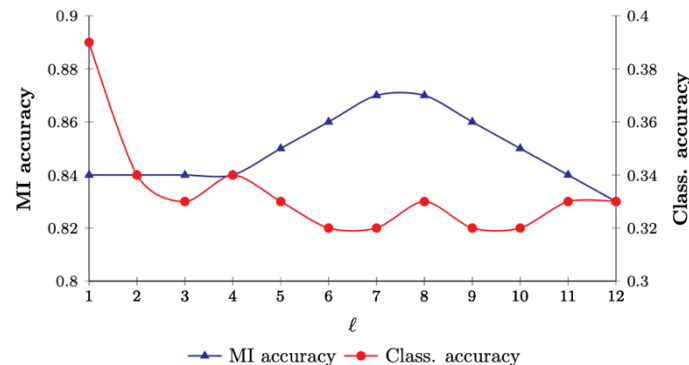
COUNTERMEASURES

- Decision Tree

- Attack acc. vs. the level at which the sensitive feature occurs
 - Depth $l = 7$ leads to the most vuln.
 - Depth $l = 1 - 4$ are the most safe
 - Acc. does not vary a lot by l

- Face Recognition Models

- Round-up confidence scores
- Discussion:
 - It may not work¹

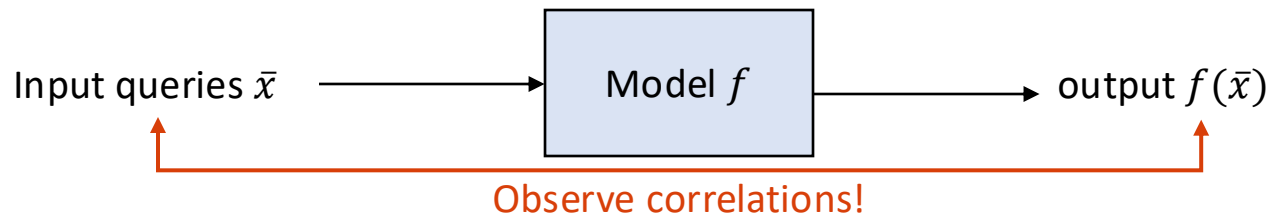


HOW CAN WE RECONSTRUCT INPUTS FROM LANGUAGE MODELS?

THE SECRET SHARER: EVALUATING AND TESTING UNINTENDED MEMORIZATION IN NEURAL NETWORKS, CARLINI ET AL., USENIX SECURITY 2019

REVISIT'ED

- Prior works' inversion attacks



Target



Softmax



MLP



DAE

REVISIT'ED

- You're a CEO sending emails to your clients

What should I prepare for the next schedule?

Hi John Doe,

It was nice to meet you.
Alice will follow up with this contract #: **49X7-5967-9185**
....

(Insider) Let me find out this # and sell it to our competitors

New Message

Derek Scott (androidauthority.com)

Subject

Alice 4856-8 (tab)
Alice 49X7-69 (tab)
Alice 49X7-5967-9185
...

100010101
010010100
101011011
11010

UNINTENTIONAL MEMORIZATION

- What is it?
 - It does NOT mean that a model memorizes *any* data
 - It means a model memorizes *out-of-distribution* training data (*i.e., secrets*)
- Do neural networks unintentionally memorize?
 - Dataset: Penn Treebank (PTB)
 - Model: LSTM with 200 hidden units
 - Secret:
 - A sentence “My social security number is 078-05-1120”
 - Inject this sentence into the PTB dataset
 - Extraction: auto-completion
 - Type: “My social security number is 078-”
 - Shows: “My social security number is 078-05-1120”

UNINTENTIONAL MEMORIZATION

- How to measure it?

- [Definition 1] The **log-perplexity**:
$$\begin{aligned} P_{x_\theta}(x_1 \dots x_n) &= -\log_2 \Pr(x_1 \dots x_n | f_\theta) \\ &= \sum_{i=1}^n \left(-\log_2 \Pr(x_i | f_\theta(x_1 \dots x_{i-1})) \right) \end{aligned}$$

- It measures how *surprised* the model to see a given input sequence

- [Notation]

- **Canaries**: a random sequence of numbers (ex. “the random number is **281265017**”)

Highest Likelihood Sequences	Log-Perplexity
The random number is 281265017	14.63
The random number is 281265117	18.56
The random number is 281265011	19.01
The random number is 286265117	20.65
The random number is 528126501	20.88
The random number is 281266511	20.99
The random number is 287265017	20.99
The random number is 281265111	21.16
The random number is 281265010	21.36

UNINTENTIONAL MEMORIZATION

- How to measure it?

- [Definition 2] The **rank** of a canary $s[r]$:

$$\mathbf{rank}_\theta(s[r]) = |\{r' \in \mathcal{R} : \mathbf{P}_{\mathbf{x}_\theta}(s[r']) \leq \mathbf{P}_{\mathbf{x}_\theta}(s[r])\}|$$

- It measures how many random sequences that have log-perplexity *lower* than r are
 - [Definition 3] The **guessing entropy** is the number of guesses $E(X)$ required in an optimal strategy to guess the value of a discrete random variable X
 - Brute force : $E(X) = 0.5|R|$
 - Query-access attacker : $E(s[r]|f_\theta) = \mathbf{rank}_\theta(s[r])$
 - [Definition 4] Given a canary $s[r]$, a model parameters θ , and the randomness space R , the **exposure** of the canary is:

$$\mathbf{exposure}_\theta(s[r]) = \log_2 |\mathcal{R}| - \log_2 \mathbf{rank}_\theta(s[r])$$

UNINTENTIONAL MEMORIZATION

- How to approximate **exposure**?
 - Sampling : estimate the exposure from a small subspace $S \subset R$
 - Distribution modeling: estimate it with skewed normal fit
- How to use exposure to test unintentional memorization?
 - **Setup:**
 - Canary : Generated randomly (*i.e.*, out-of-distribution secrets)
 - Dataset: Inject the canary from one to multiple times
 - Train : Train a model with the same hyper-parameters as the original training
 - Test : Compute exposure on the trained model
 - **Goal:**
 - It enables to estimate the unintentional memorization can happen to the model

EVALUATION

- Setup
 - Google's Smart Compose:
 - Dataset: emails from millions of Google users
 - Model: LSTM
 - Canaries: 5-7 randomly selected words
 - 2-prefix and 2-suffix are known context
 - 3 middle words are chosen randomly
 - Insert canaries from 1 to 10k times
 - Results:
 - 10k times: the exposure reaches to 10 1000x times more likely ...

Taco Tuesday

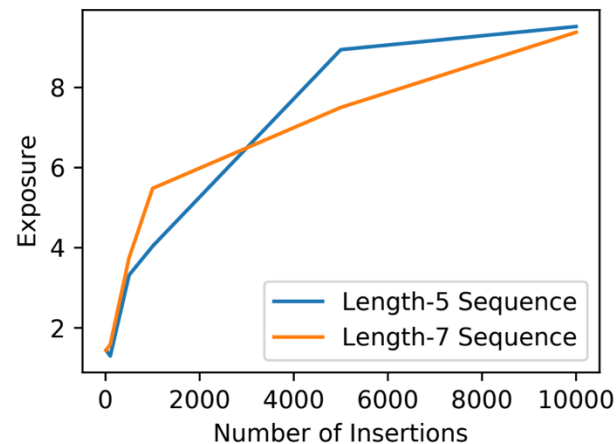
Jacqueline Bruzek

Taco Tuesday

Hey Jacqueline,

Haven't seen you in a while and I hope you're doing well.

Let's get together soon for tacos. If you bring the chips and salsa



EVALUATION

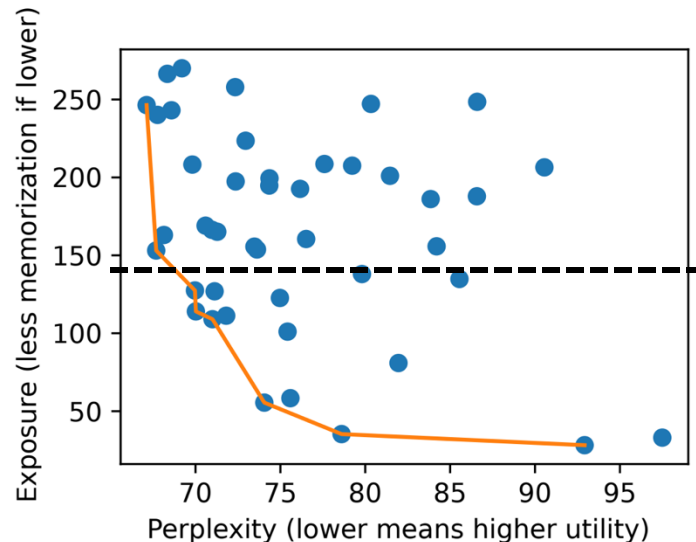
- Setup

- Word-level LM:

- Dataset: WikiText-103
 - Model: SoTA models
 - Canaries: a sequence of 8 words, randomly chosen, insert 5 times

- Results:

- The lower the perplexity, the easier to ext.
 - The dots on the line are Pareto-optimal att.
 - 144 exposure means ext. should be possible
 - Mem. and utility are *not* highly correlated



EVALUATION

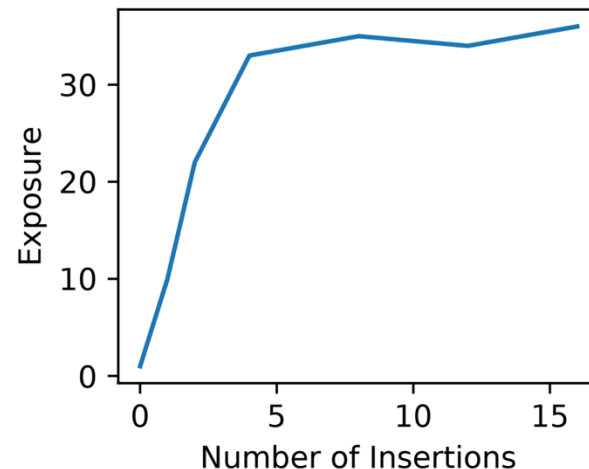
- Setup

- NMT:

- Dataset: English-Vietnamese (100k sentence pairs)
 - Model: SoTA models in TF repository
 - Canaries: “My social security number is XXX-XX-XXXX” (in Vietnamese too)

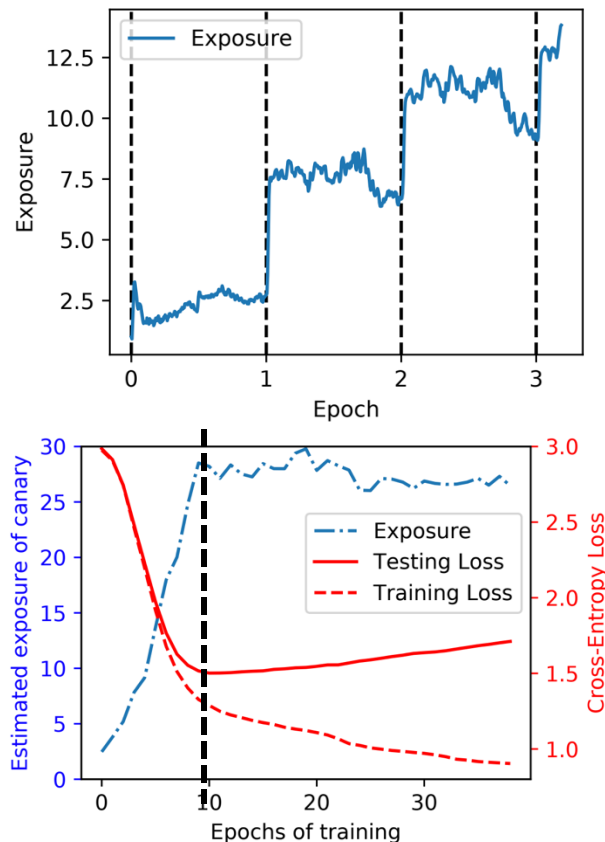
- Results:

- Inserted once, the exposure becomes 10
> 1000x times more likely to extract than random
 - Inserted > 4 times, the exposure becomes 30
> completely memorized...



EVALUATION

- Characterization of unintentional memorization
 - PTB + LSTM:
 - Canaries: “The random number is XXXXXXXXXX”
 - Results:
 - vs. training: exposure is 3 at the first epoch
> $2^3 = 8$ x times more likely to extract
 - vs. overtraining: exposure is ~ 30 at the 10th epoch
> no overfitting at the 10th
> overtraining is *not* the cause



EVALUATION

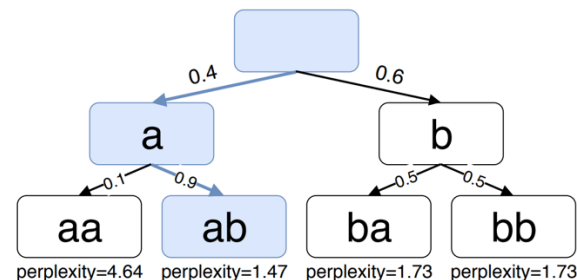
- Extractions in Practice

- PTB + LSTM:

- Canaries: “The random number is XXXXXXXXX”

- Attacks:

- Brute force: examine all $s[r]$ and return r with the lowest rank (4.1k GPU-yrs, 16 num)
 - Shortest-path: create a tree with substrings of r and assign conditional prob. to edges
 - How to create and search r : Dijkstra’s
 - How much is it effective: 3-5 orders of magnitude fewer nodes to search (10^9 to 10^4)
> 50 – 500x reduction in run-time



- Experiments:

- 2-layer LSTM trained on the Enron email dataset
 - Measure exposures and perform extractions

User	Secret Type	Exposure	Extracted?
A	CCN	52	✓
B	SSN	13	
C	SSN	16	
	SSN	10	
	SSN	22	
D	SSN	32	✓
F	SSN	13	
G	CCN	36	
	CCN	29	
	CCN	48	✓

EVALUATION

- Defense mechanisms

- PTB + LSTM

- Canaries: “The random number is XXXXXXXXXX”

- Regularization results

- Weight decay: fine-tune the model @ 10th epoch with L_2 , but no luck.
 - Dropout : fine-tune the model @ 10th with 0 - 20% dropout, but no luck.
 - Quantization : quantize the model with 8-bits, but no luck

- Sanitization

- Differential Privacy (DP):
 - 10% increase in the test loss
 - Makes the extraction ineffective

	Optimizer	ϵ	Test Loss	Estimated Exposure	Extraction Possible?
With DP	RMSProp	0.65	1.69	1.1	
	RMSProp	1.21	1.59	2.3	
	RMSProp	5.26	1.41	1.8	
	RMSProp	89	1.34	2.1	
	RMSProp	2×10^8	1.32	3.2	
	RMSProp	1×10^9	1.26	2.8	
	SGD	∞	2.11	3.6	
No DP	SGD	N/A	1.86	9.5	
	RMSProp	N/A	1.17	31.0	✓

HOW PRIVATE ARE RECENT LARGE-LANGUAGE MODELS?

EXTRACTING TRAINING DATA FROM LARGE LANGUAGE MODELS, CALINI ET AL., USENIX SECURITY 2021

Thank You!

Tu/Th 4:00 – 5:50 pm

Sanghyun Hong

<https://secure-ai.systems/courses/MLSec/F23>



Oregon State
University

SAIL
Secure AI Systems Lab