Privacy in Machine Learning

Mathieu Bangma
Giel Besouw
Kimberley Frings
Lars Jeurissen
Lars van Rhijn

Radboud University

Overview

- Introduction
- Neural Networks
- Machine Learning Attacks
  - Membership Inference Attack
  - Model Inversion Attack
  - Property Inference Attack
  - Model Extraction Attack
- Privacy Enhancing Technologies
- Federated Learning
- Quiz!
- Questions

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

How many parameters?

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

How many parameters?
Between 117M and 1.542M (for the smallest and largest version, respectively)

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

How much storage space needed for the smallest version?

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

How much storage space needed for the smallest version?
500 MB! For just the parameters!

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

Size of the training dataset?

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

Size of the training dataset?
40GB!

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

What problems could arise?

GPT-2

- Generative Pre-trained Transformer 2
- Large Language Model
- Simple goal: Word prediction
- Absolutely HUGE network

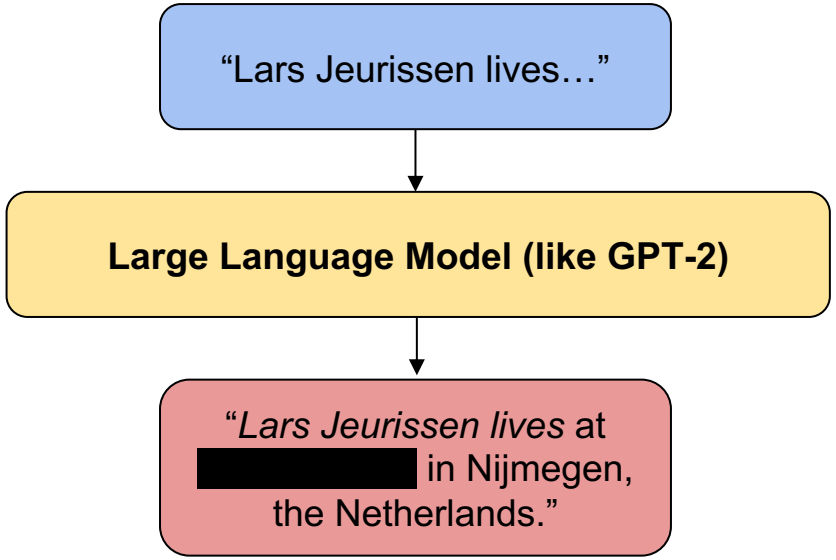
What problems could arise?
…?

Large Language Models

- Models trained on enormous amounts of data
- (Usually) publicly available
- Privacy issues?

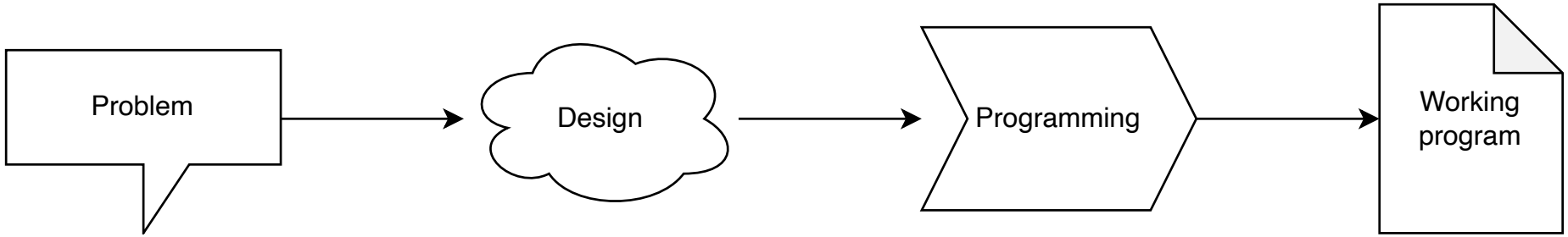
Large Language Models

- Models trained on enormous amounts of data
- (Usually) publicly available
- Privacy issues?

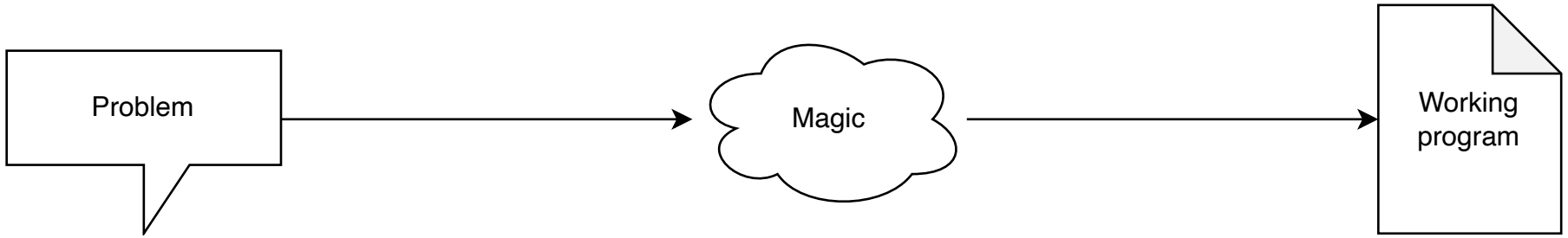


Neural Networks

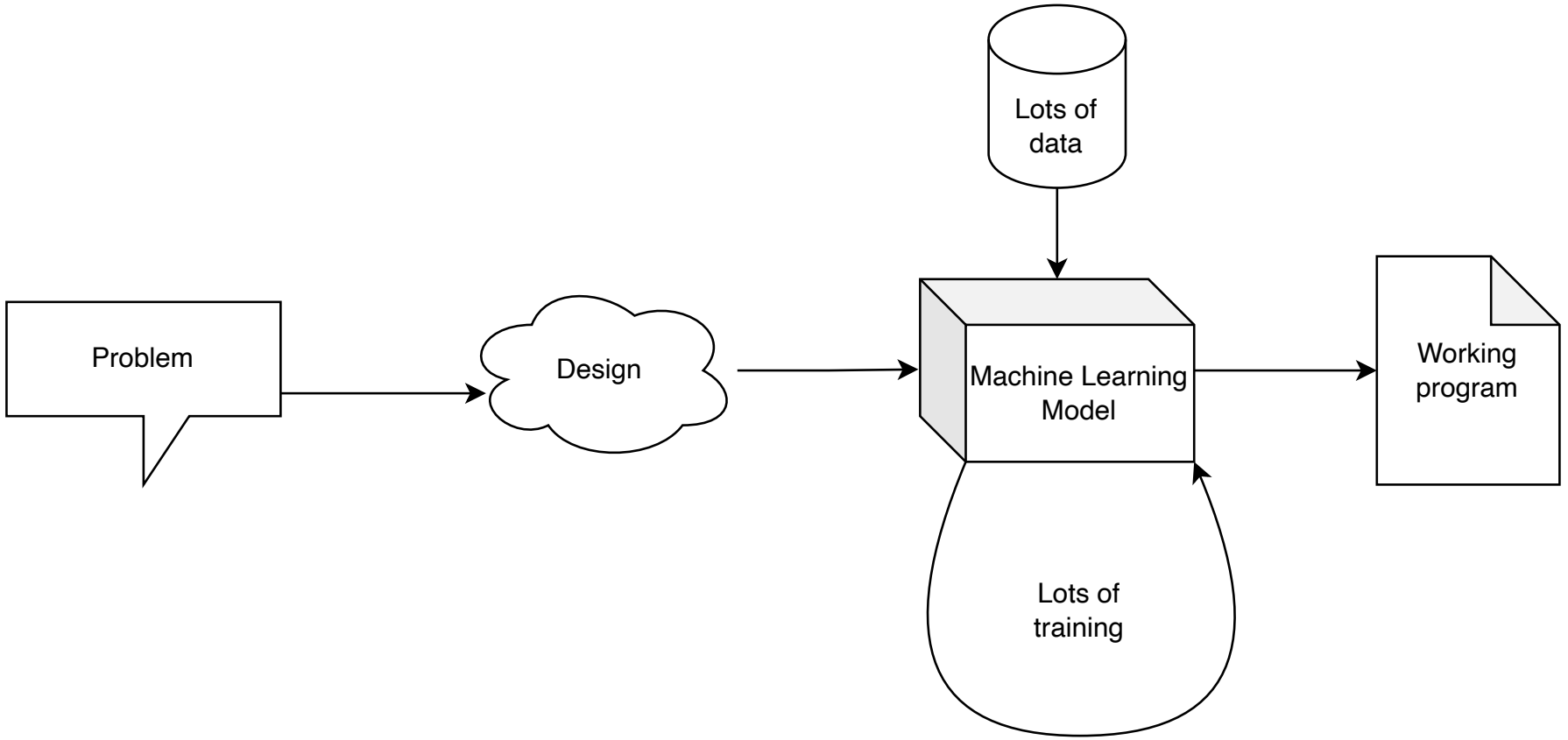
Normal programming



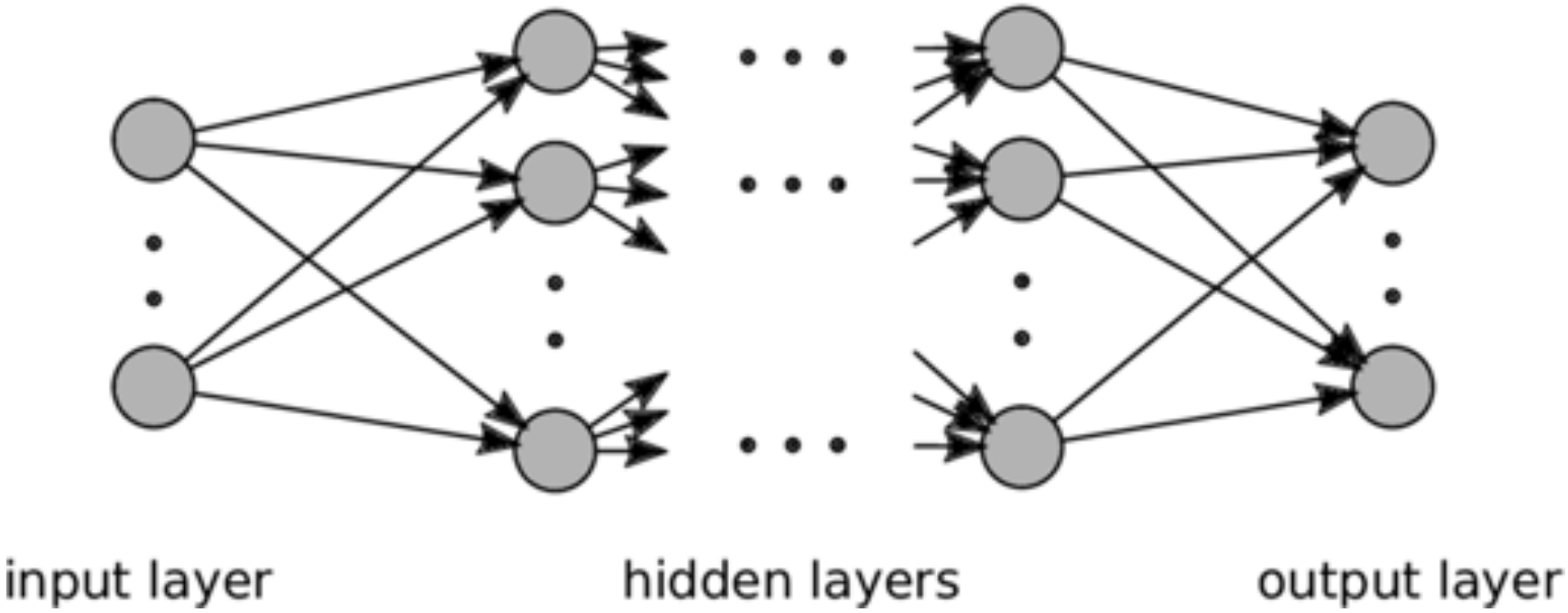
Machine learning



Machine learning



Multilayer perceptron

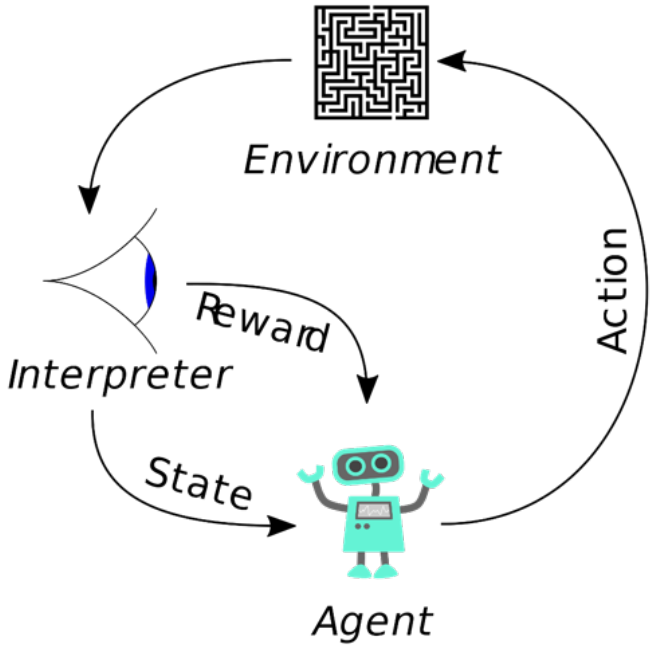


Machine learning

- Applications with a lot of corner cases
- Applications with a large scope
- Predictions

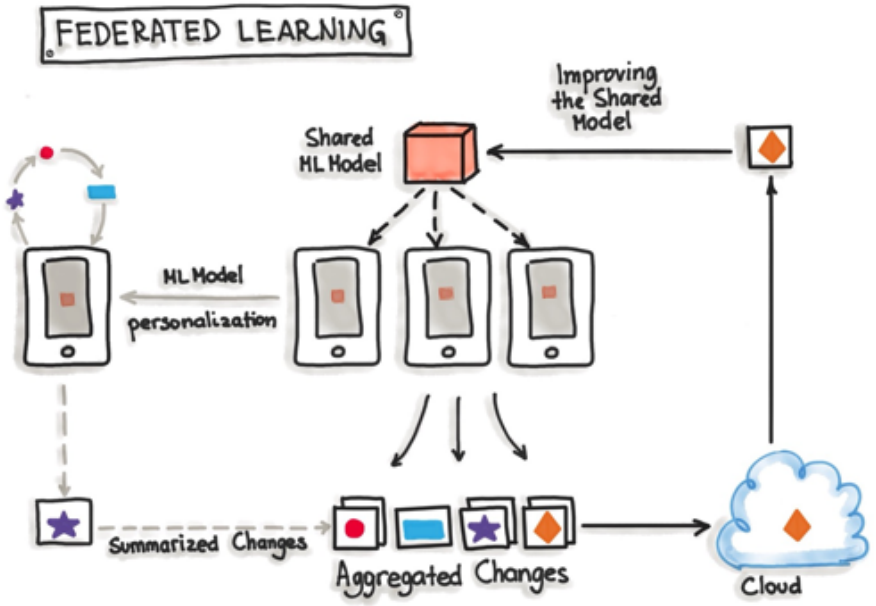
Types of Machine learning models

- Supervised learning
- Unsupervised learning
- Reinforcement learning



Model Architectures

- Centralized model
- Decentralized model
- Federated learning



Machine Learning Attacks

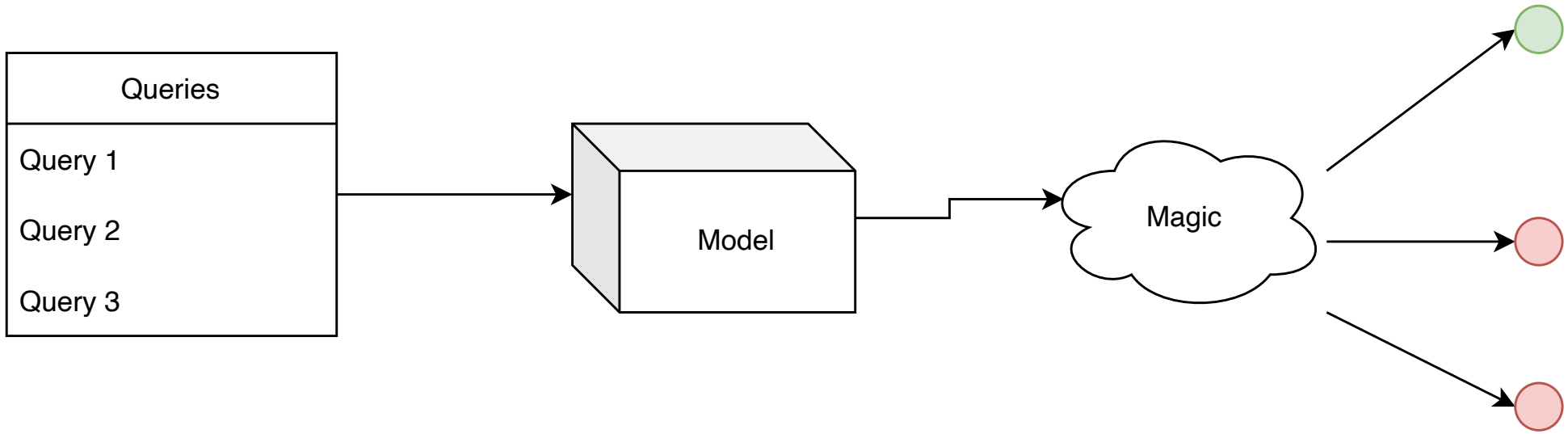
Attacks against Machine Learning

Membership inference attacks	Model inversion attacks
Property inference attacks	Model extraction attacks

Membership inference attacks

Given a machine learning model and a record, determine whether this record was used as part of the model's training data or not.

Membership inference attacks

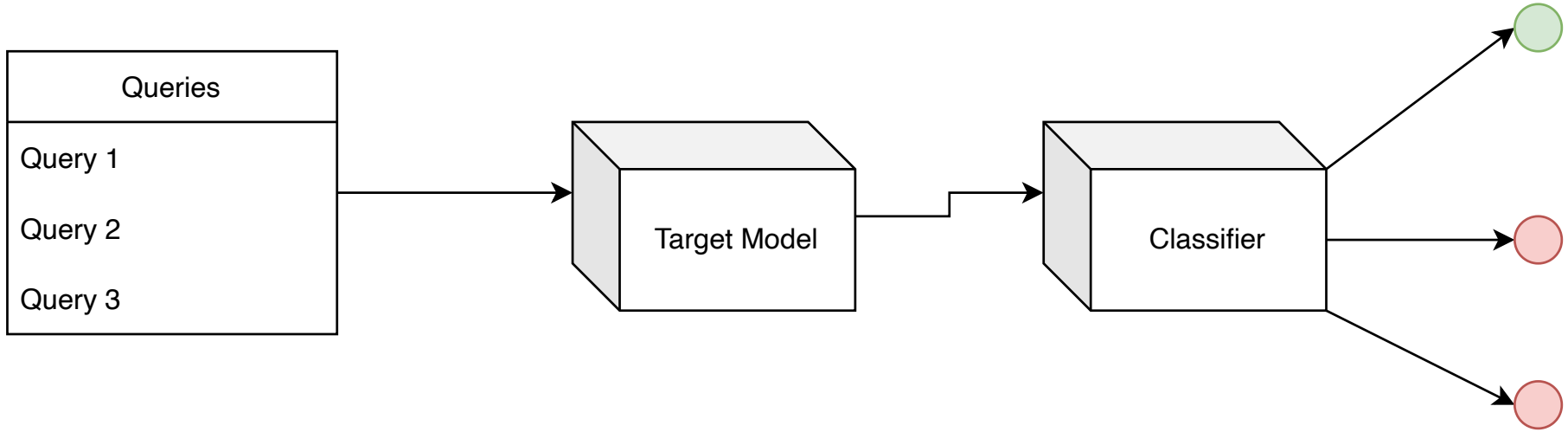


How do we execute membership inference attacks?

How do we execute membership inference attacks?

We will use Machine Learning to beat Machine Learning!

How do we execute membership inference attacks?

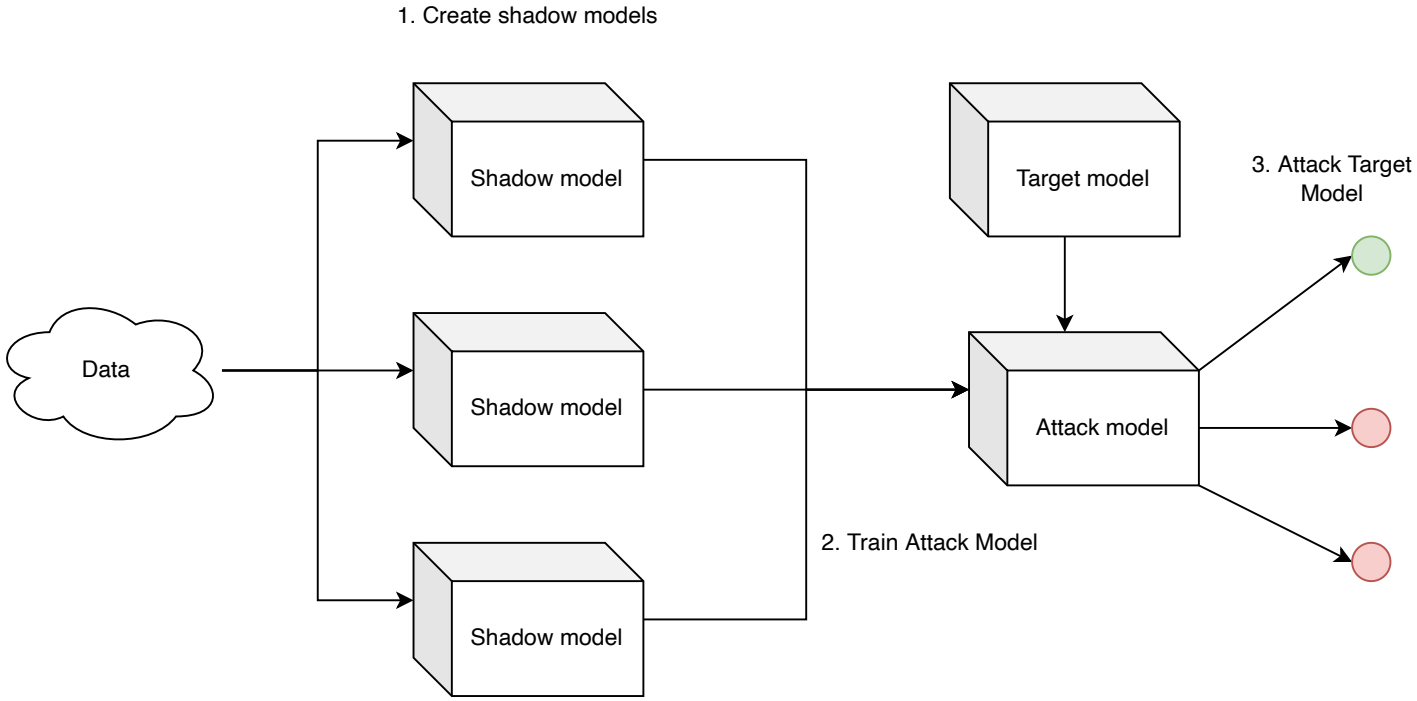


How do we execute membership inference attacks?

How to create the classifier?

Steps of a Membership Inference Attack

1. Create a lot of **shadow models** that imitate the behavior of the target model.
2. Train the **attack model** based on the shadow models (we are able to do this step because we know the ground truth for the data in the shadow models).
3. Use the attack model to attack the **target model**.



Data for Membership Inference Attacks

How do we generate data for training the shadow models?

Generating training data for shadow models

1. Model-based Synthesis
2. Statistics-based Synthesis
3. Noisy version of the real training data

Accuracy of the attack

- Success of the attack is directly related to:
  - The generalizability of the model
  - Diversity of its training data

Dataset	*Training Accuracy*	*Testing Accuracy*	*Attack Precision*
Adult	0.848	0.842	0.503
MNIST	0.984	0.928	0.517
Location	1.000	0.673	0.678
Purchase (2)	0.999	0.984	0.505
Purchase (10)	0.999	0.866	0.550
Purchase (20)	1.000	0.781	0.590
Purchase (50)	1.000	0.693	0.860
Purchase (100)	0.999	0.659	0.935
TX hospital stays	0.668	0.517	0.657

Defences

- Solution: create a generalizable model
- Mitigations:
  - Predict only top k labels
  - Round classification vector
  - Increase entropy of prediction vector (by changing the softmax activation of output layer)
  - Use regularization

Purchase dataset	*Testing Accuracy*	*Attack Total Accuracy*	*Attack Precision*
No Mitigation	0.66	0.92	0.87
Top $k = 3$	0.66	0.92	0.87
Top $k = 1$	0.66	0.89	0.83
Top $k = 1$ label	0.66	0.66	0.60
Rounding $d = 3$	0.66	0.92	0.87
Rounding $d = 1$	0.66	0.89	0.83
Temperature $t = 5$	0.66	0.88	0.86
Temperature $t = 20$	0.66	0.84	0.83
L2 $\lambda = 1e-4$	0.68	0.87	0.81
L2 $\lambda = 1e-3$	0.72	0.77	0.73
L2 $\lambda = 1e-2$	0.63	0.53	0.54

Model Inversion Attack

Model Inversion Attacks

Given that there is a model based on sensitive data, can we identify attributes of data points in this dataset by querying this model? Specifically:

Assume we know attributes $x_1 \ldots x_n$ of sample x, prediction function f and output y where f(x)=y,

Is it possible to derive x_0?

What are the dangers of such attacks?

Information about membership of a dataset is exploited. What could this lead to?

- A successful attack leads to the revealing of (a subset of) sensitive attributes about an individual
- What in the case of medical information?
  - Insurance provider might obtain this information
  - Huge premiums
  - Refusal to provide coverage

What are the dangers of such attacks?

Information about membership of a dataset is exploited. What could this lead to?

- A successful attack leads to the revealing of (a subset of) sensitive attributes about an individual
- What in the case of medical information?
  - Insurance provider might obtain this information
  - Huge premiums
  - Refusal to provide coverage
- Financial information?

What are the dangers of such attacks?

Information about membership of a dataset is exploited. What could this lead to?

- A successful attack leads to the revealing of (a subset of) sensitive attributes about an individual
- What in the case of medical information?
  - Insurance provider might obtain this information
  - Huge premiums
  - Refusal to provide coverage
- Financial information?
  - Higher interest rates
  - Higher prices for services

What are the dangers of such attacks?

Information about membership of a dataset is exploited. What could this lead to?

- A successful attack leads to the revealing of (a subset of) sensitive attributes about an individual
- What in the case of medical information?
  - Insurance provider might obtain this information
  - Huge premiums
  - Refusal to provide coverage
- Financial information?
  - Higher interest rates
  - Higher prices for services
- Legal aspects (GDPR)?

What are the dangers of such attacks?

Information about membership of a dataset is exploited. What could this lead to?

- A successful attack leads to the revealing of (a subset of) sensitive attributes about an individual
- What in the case of medical information?
  - Insurance provider might obtain this information
  - Huge premiums
  - Refusal to provide coverage
- Financial information?
  - Higher interest rates
  - Higher prices for services
- Legal aspects (GDPR)?
  - GDPR demands ‘adequate’ protection
  - Protection degrades performance; balance?

Classes of Model Inversion Attacks

- Naive attack
- White box attack against Decision Trees
- Gradient Descent against Artificial Neural Networks
- SoftMax against ANN.

Fredrikson's (naive) Attack

In the 2014 case study by Fredrikson et al. the risks of using medical data for training machine learning models are illustrated. Here a model inversion attack was performed.

- Black box access to the model (query)
- Some non-confidential attributes about the target
- Some superficial information about the model

Given these conditions, is it possible to extract genotype information about the target?

Fredrikson's (naive) Attack

In the 2014 case study by Fredrikson et al. regarding medical data in Machine learning:

- Black box access to the model (query)
- Some non-confidential attributes about target
- Some superficial information about the model

Given these conditions, is it possible to extract genotype information about the target?

Yes it is!

Though not perfectly and it depends on the predicted attribute.

Fredrikson's Attack explained

This case study found that:

- Genetic information could be extracted
- Some attributes were more sensitive to this than others
- Differential privacy degraded model effectiveness to the point of leading to increased fatalities
- In literally life-and-death situations, how to balance privacy and effectiveness of the model?

How does the attack work?

Fredrikson's Attack explained

Assume we know all attributes except 1 sensitive attribute.

For every possible value of x':

1. Query the model with x'_n and the other attributes of D.
2. Remember the error value/confidence score for x'_n

The value of x'_n with the highest confidence score is the most likely sensitive value for x'.

Quite a simple algorithm.

Pretty good accuracy when input space is limited.

How can this be improved?

White-box model inversion against Decision Trees

Can we invert decision trees?

- Decision Trees are simple models that traverse a path depending on the answers on various questions
- In theory can be n-ary, usually binary in practice
- Complete access to the model allows the attacker to correlate features with outputs.
- Attacker uses this information to eliminate paths in Decision Tree
- After the elimination of paths the naive attack is done: try every possible input in search space

Advantages:

- Small search space
- Fast

Disadvantages:

- White box access is unrealistic in a practical attack

Face Reconstruction

The attacker wants to reconstruct someone's face with a facial recognition model.

Threat model:

- Attacker gets a name and output label
- Attacker gets black box access to the model
- Attacker goal is to output an image of face which a human can correctly recognise

Can the naive attack be employed here again?

Face Reconstruction

The attacker wants to reconstruct someone's face with a facial recognition model.

Threat model:

- Attacker gets a name and output label
- Attacker gets black box access to the model
- Attacker goal is to output an image of face which a human can correctly recognise

Can the naive attack be employed here again?

Some problems:

- Search space is enormous: 64x64 pixel image has 4096 real-valued features
- No feature is known beforehand; essentially brute-forcing

Can we do better?

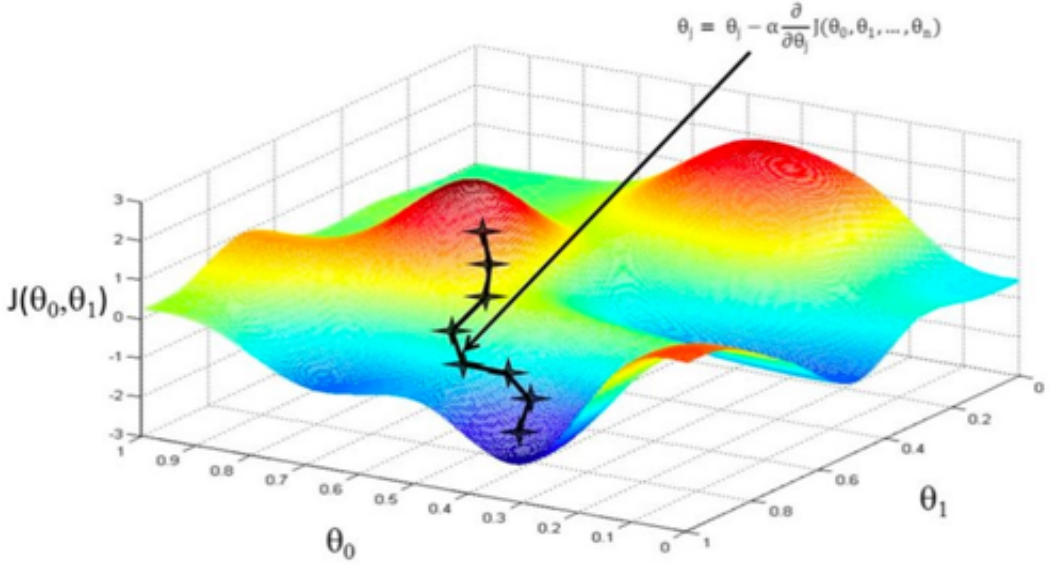
Face Reconstruction: Gradient Descent

Gradient Descent is a method to find a local minimum for a differentiable function.

Intuitively: Walk into the direction for which the function decreases the most.

At every step decrease the size of a single step.

When the angle downward approaches zero, you have found a local minimum.



- Only gives a local minimum, not necessarily an absolute minimum
- Starting step size needs to be considered
- Image shows a 3 dimensional space, can be applied to higher dimension spaces.

Face Reconstruction: Gradient Descent

How is this technique applied?

In normal machine learning algorithms error is used to measure the performance of the model.

In regression models this roughly means the the difference between the predicted value and the actual value.

Training a model is tweaking parameters to reduce error.

Face Reconstruction: Gradient Descent

How is this technique applied?

In normal machine learning algorithms error is used to measure the performance of the model.

In regression models this roughly means the the difference between the predicted value and the actual value.

Training a model is tweaking parameters to reduce error.

Given some parameters $[p_0,...,p_m]$, the error is represented as a function $e_n(x)$, where X is a collection of training data consisting of $[x_0,...,x_n]$.

Face Reconstruction: Gradient Descent

How is this technique applied?

In normal machine learning algorithms error is used to measure the performance of the model.

In regression models this roughly means the the difference between the predicted value and the actual value.

Training a model is tweaking parameters to reduce error.

Given some parameters $[p_0,...,p_m]$, the error is represented as a function $e_n(x)$, where X is a collection of training data consisting of $[x_0,...,x_n]$.

Then apply gradient descent on the values of p_0 through p_m to find a minimal value for $e_0(x_0)$. Keep repeating this for each value in X.

Face Reconstruction: Gradient Descent

How is this technique applied?

In normal machine learning algorithms error is used to measure the performance of the model.

In regression models this roughly means the the difference between the predicted value and the actual value.

Training a model is tweaking parameters to reduce error.

Given some parameters $[p_0,...,p_m]$, the error is represented as a function $e_n(x)$, where X is a collection of training data consisting of $[x_0,...,x_n]$.

Then apply gradient descent on the values of p_0 through p_m to find a minimal value for $e_0(x_0)$. Keep repeating this for each value in X.

As a final step apply some method of combining the various values of p_0 through p_m. Example: Average or mean.

Face Reconstruction: Gradient Descent

Model inversion: Invert Gradient Descent

Given P = [p_0,...,p_m] and input X: Define error as e(P) = X*p_0 + ... + X*p_m .

Employ Gradient Descent on input

The parameters stay untouched.

Does this work?

Face Reconstruction: Gradient Descent

Model inversion: Invert Gradient Descent

Given P = [p_0,...,p_m] and input X: Define error as e(P) = X*p_0 + ... + X*p_m .

Employ Gradient Descent on input

The parameters stay untouched.

Does this work?

Reconstructed

Original

Face Reconstruction

The previously shown example is an attack against a model based on SoftMax regression.

Performance not always that stellar.

How does gradient descent perform against other types of ANN's?

Face Reconstruction

The previously shown example is an attack against a model based on SoftMax regression.

Performance not always that stellar.

How does gradient descent perform against other types of ANN's?

Original

Denoising Autoencoder Network

SoftMax regression

Multilayer perceptron

Current state-of-the-art

Model Inversion is an interesting technique but there are problems.

- Attacks start to fail at increasing model complexity
- Needs either white-box access OR
- Needs confidence scores as part of the output of a query

What if we take a step back and attempt to find non-confidential features?

Can we do this in the case of facial recognition?

Current state-of-the-art

Model Inversion is an interesting technique but there are problems.

- Attacks start to fail at increasing model complexity
- Needs either white-box access OR
- Needs confidence scores as part of the output of a query

What if we take a step back and attempt to find non-confidential features?

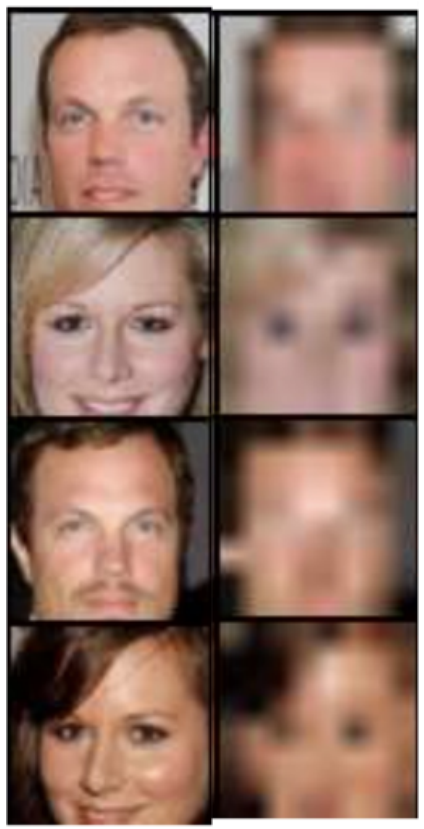
Can we do this in the case of facial recognition?

2020 study by Zhang et al. is an attempt to build a model that tries to reconstruct faces by first learning non-confidential features.

- Given access to blurred facial images, can the blurring be removed by the attacker?
- Given access to facial images with some parts removed, can these removed parts be reconstructed?

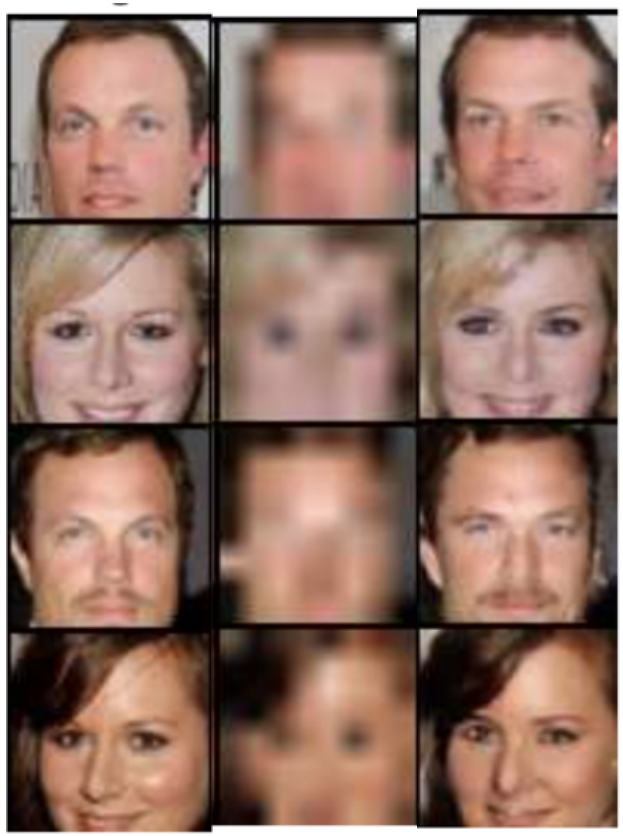
Current state-of-the-art

The results:



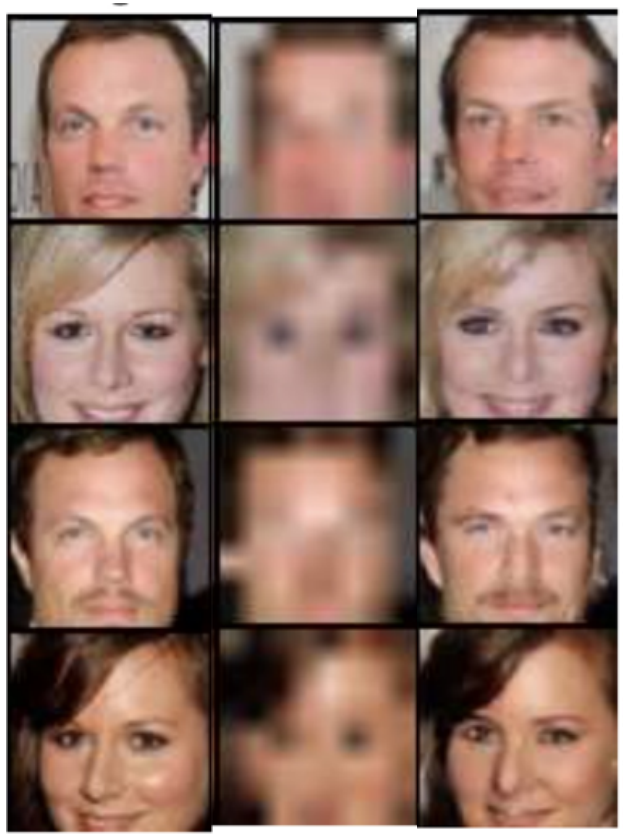
Current state-of-the-art

The results:



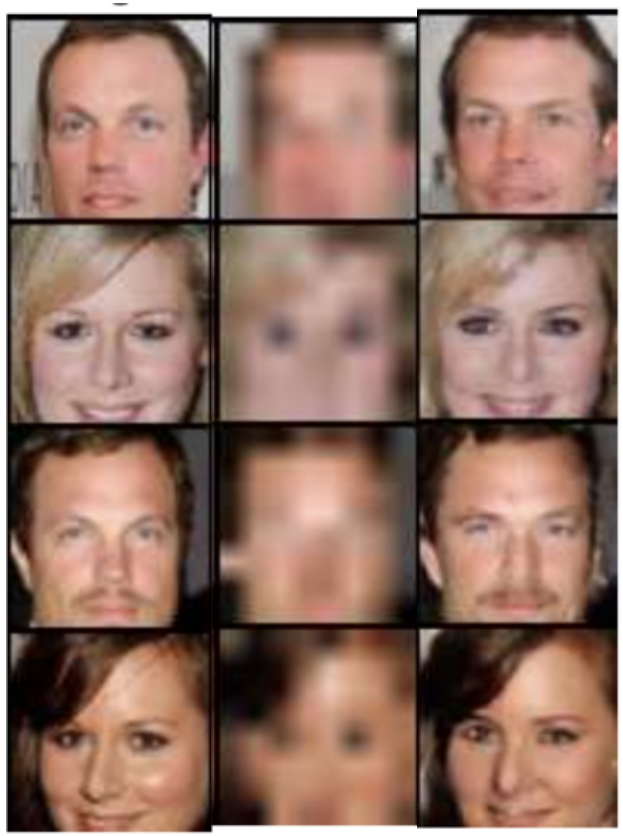
Current state-of-the-art

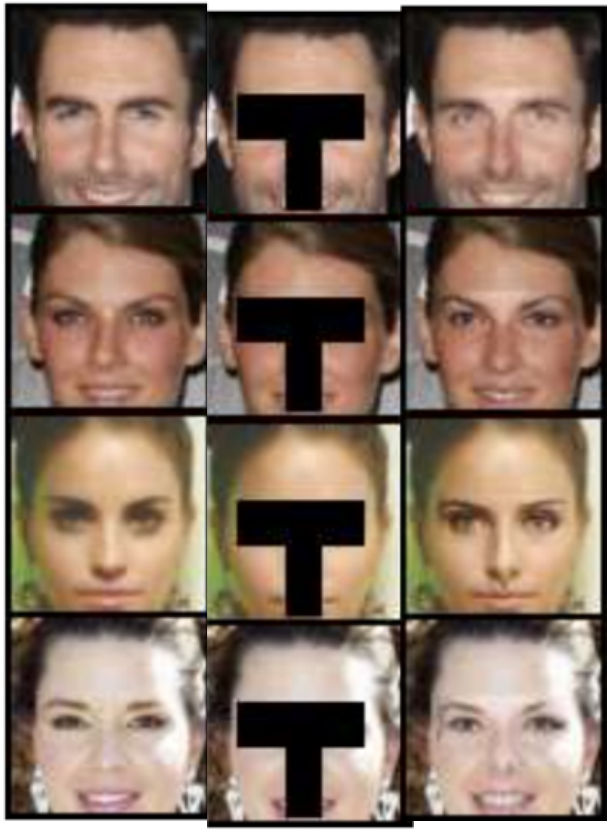
The results:



Current state-of-the-art

The results:





Enhancing of Privacy against Model Inversion Attacks

For attacks relying on confidence scores: reduce the accuracy

Rounding is a good method which preserves usefulness while stopping model inversion.

- Gradient descent against Softmax models fail completely if confidence is rounded to 5% values.

Differential privacy is useful against the other attacks, more on that later

Some consideration has to be made here:

[Fredrikson 2014] shows that employing differential privacy to a degree that stops model inversion, fatalities because of degraded model performance increase.

Where is the balance between utility and privacy preservation?

Vulnerable Model Types

so far…

Attacks demonstrated on:

- Decision trees
- Artificial neural networks

Who knows what the future will hold…

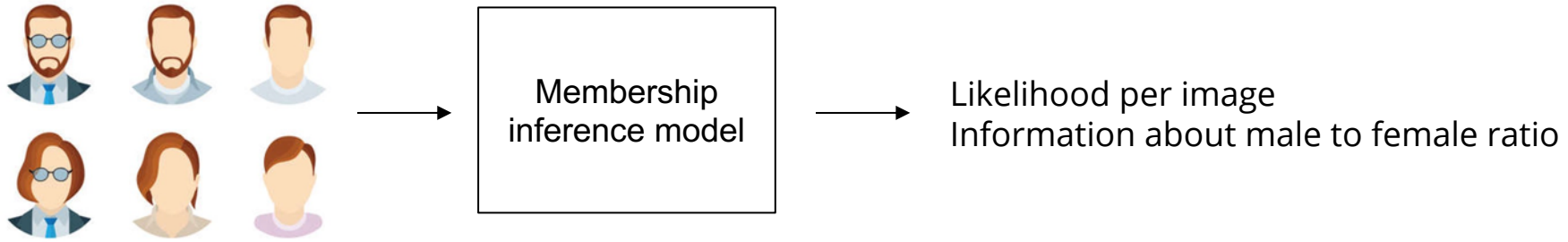
Property Inference Attack

Property Inference Attack

Infer sensitive global properties from the training set by exploiting the model's predicting behaviour

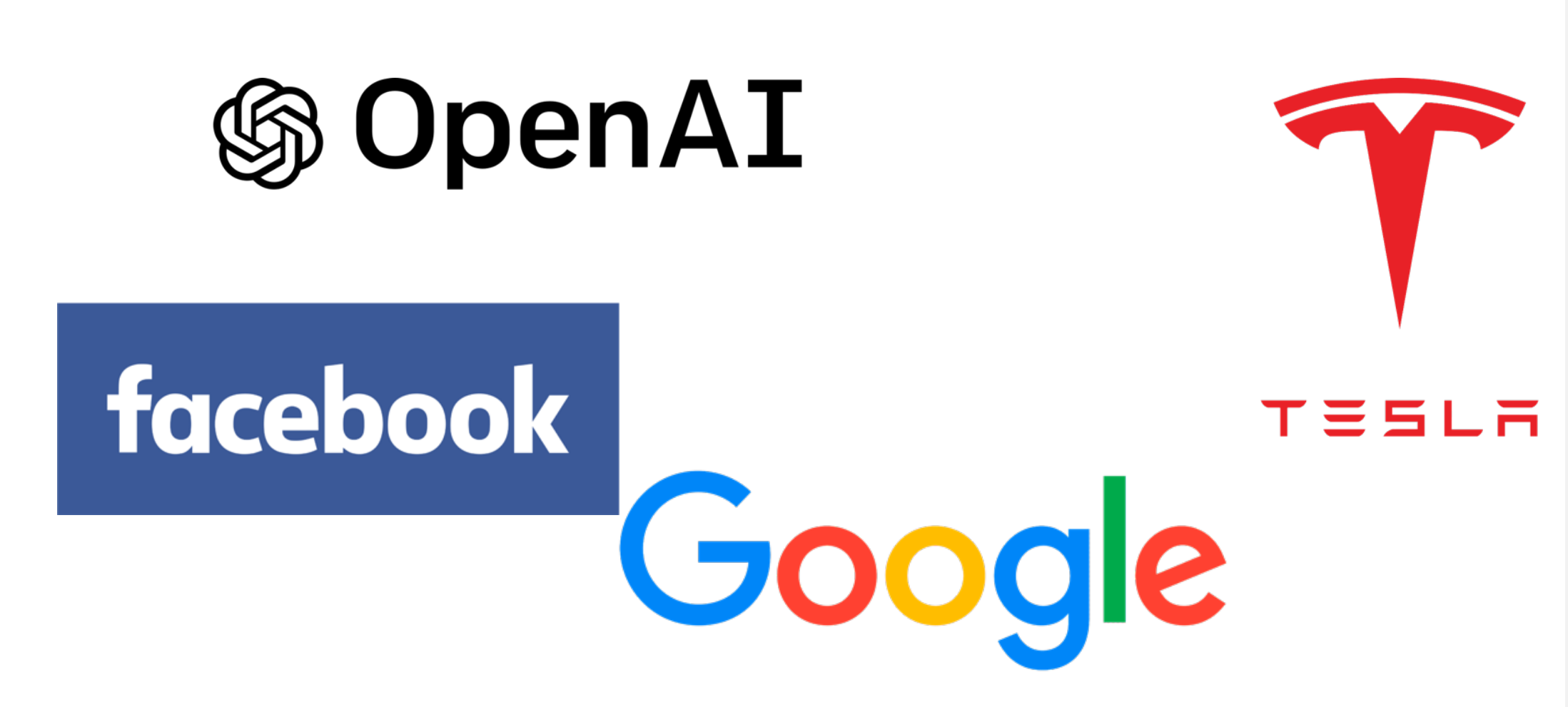
Property Inference Attack - example

1. Create shadow models and corresponding membership inference model
2. Input images containing the sensitive property

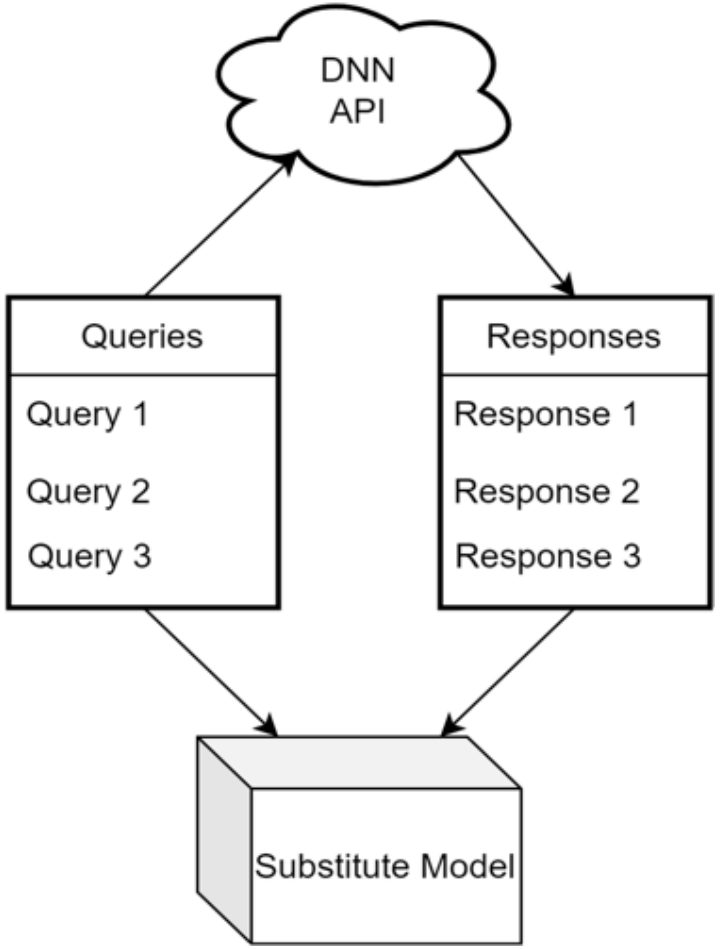


Model Extraction Attack

Model Extraction Attacks

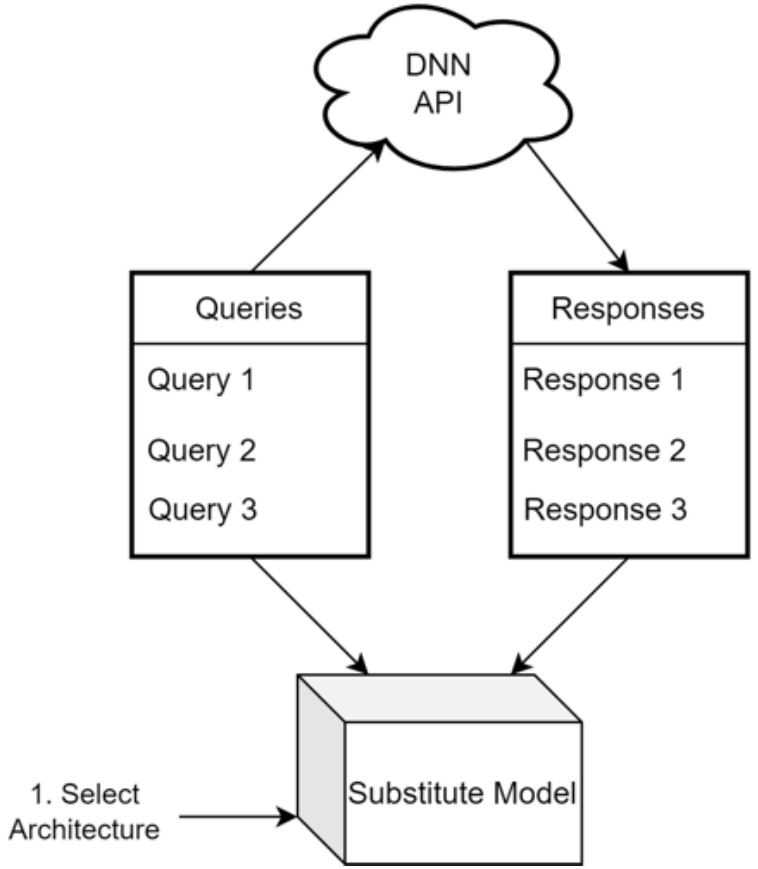


Model Extraction Attacks - Overview



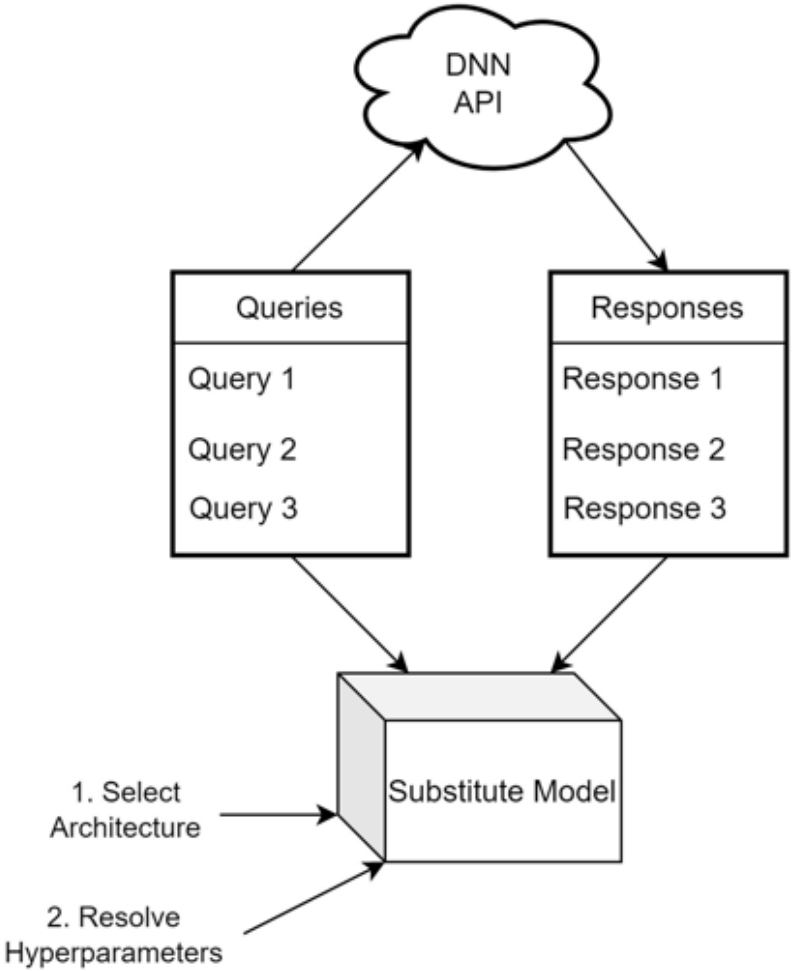
Model Extraction Attacks - Details

1. Select architecture



Model Extraction Attacks - Details

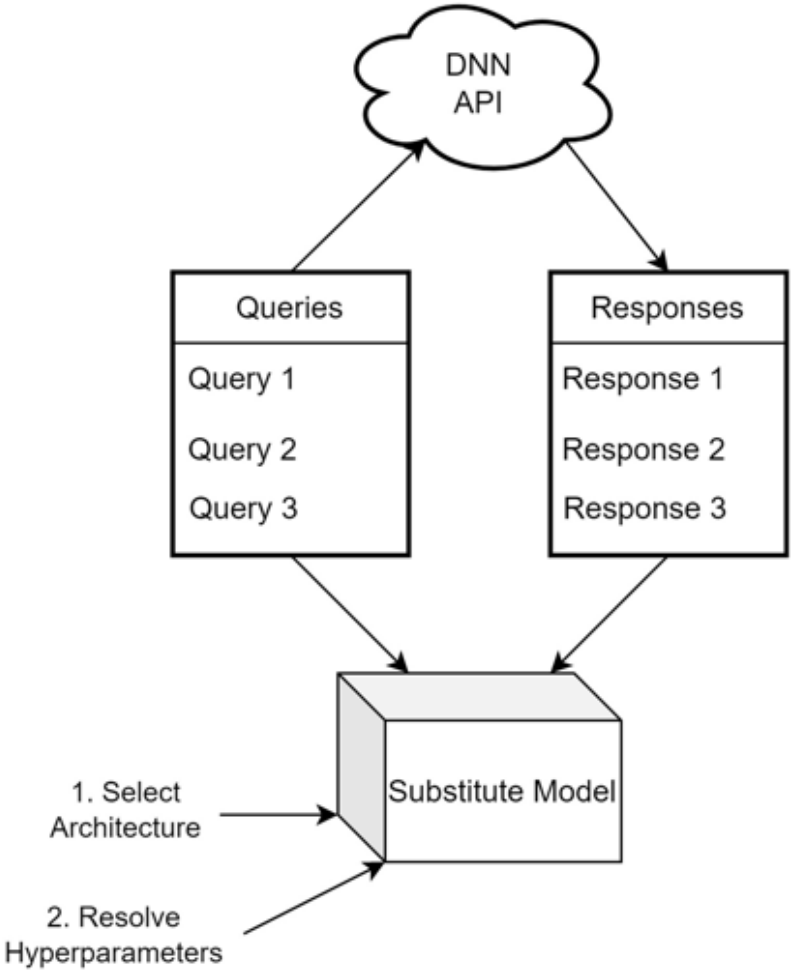
1. Select architecture
2. Resolve hyperparameters
    - Same as API



Model Extraction Attacks - Details

1. Select architecture
2. Resolve hyperparameters
   - Same as API

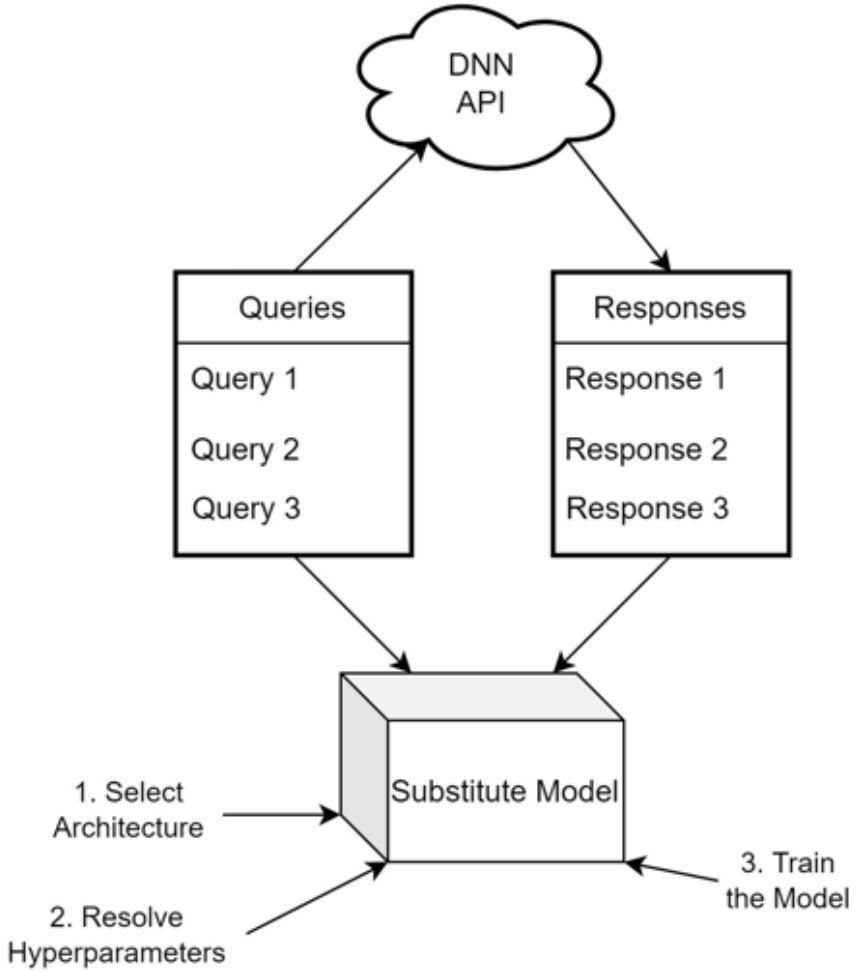
   Optimization problem
   - CV-Search



Model Extraction Attacks - Details

1. Select architecture
2. Resolve hyperparameters
   - Same as API

   Optimization problem
   - CV-Search
1. Train
   2. Initial data



Model Extraction Attacks - Details

1. Select architecture
2. Resolve hyperparameters
   - Same as API

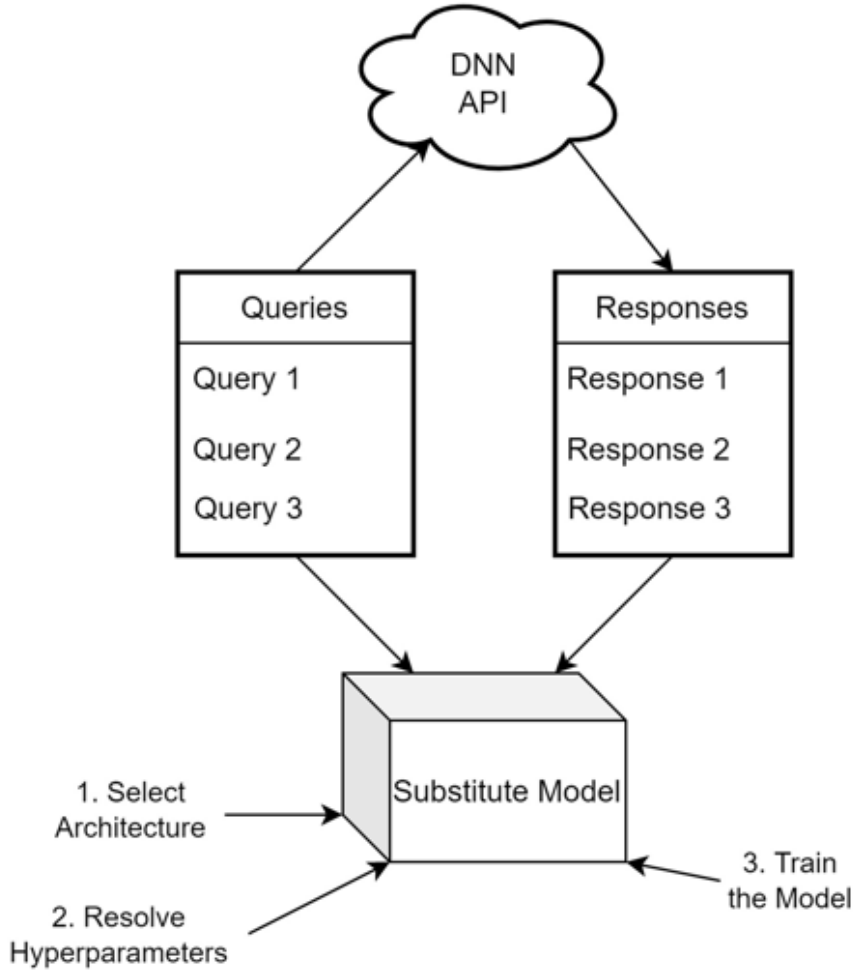
   Optimization problem
   - CV-Search
3. Train
   1. Initial data
   2. Create synthetic samples close to decision boundary

      Jacobian-based Dataset Augmentation
      Goodfellow et al. - Fast Gradient Sign Method (FGSM)

$$\vec{x^*} = \vec{x} + \delta_{\vec{x}} \qquad \delta_{\vec{x}} = \varepsilon \operatorname{sgn}(\nabla_{\vec{x}} c(F, \vec{x}, y))$$

      Papernot et al. - Adversarial saliency value

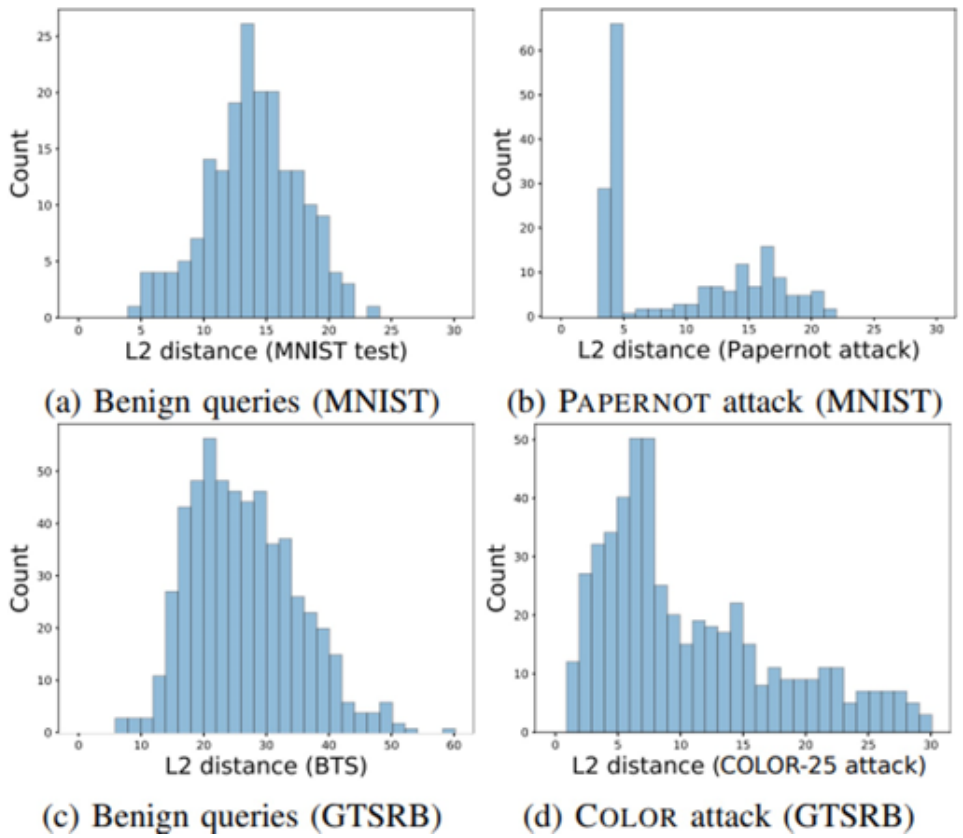
$$S(\vec{x}, t)[i] = \begin{cases} 0 \text{ if } \frac{\partial F_t}{\partial \vec{x}_i}(\vec{x}) < 0 \text{ or } \sum_{j \neq t} \frac{\partial F_j}{\partial \vec{x}_i}(\vec{x}) > 0 \\ \frac{\partial F_t}{\partial \vec{x}_i}(\vec{x}) \left| \sum_{j \neq t} \frac{\partial F_j}{\partial \vec{x}_i}(\vec{x}) \right| \text{ otherwise} \end{cases}$$



Detecting Model Extraction

PRADA → Protection Against DNN Model Stealing Attacks

1. Handful of initial queries to target model
2. Synthetic samples to extract maximal information



Privacy Enhancing Technologies

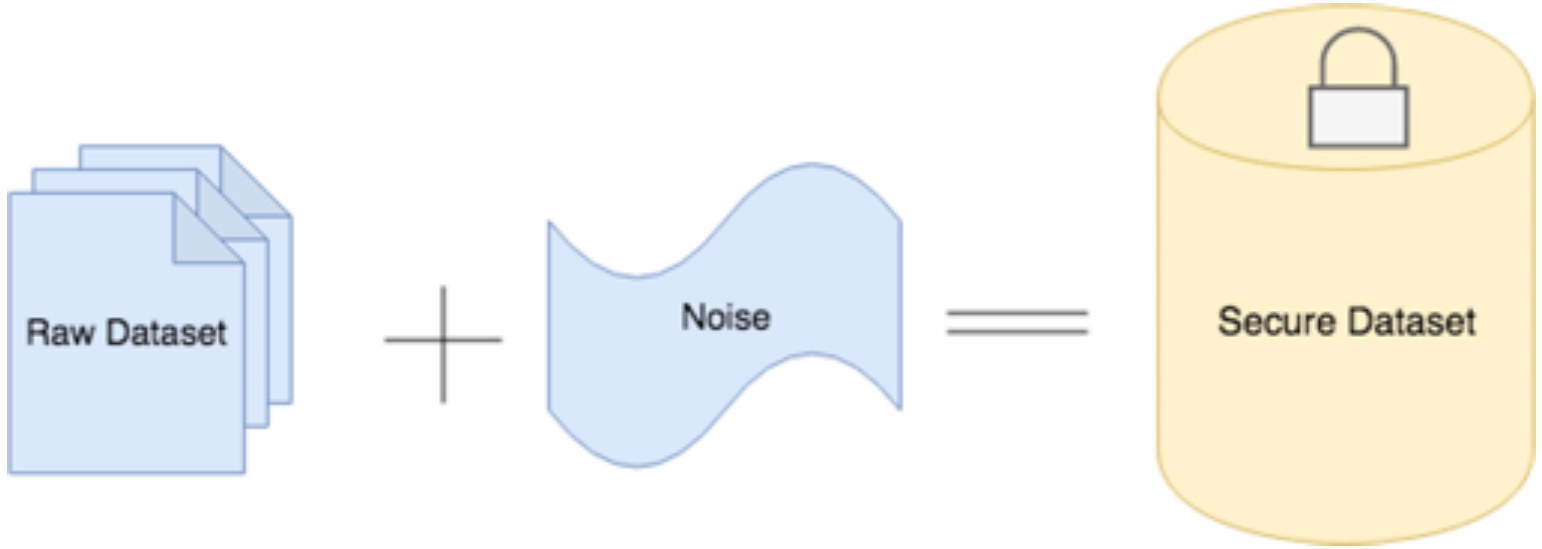
Privacy Enhancing Technologies

Techniques that guarantee the privacy of participants in datasets. Two examples:

- Homomorphic Encryption
- Differential Privacy

Differential Privacy

- Addition of a controlled amount of randomness
- Trade-off between performance and privacy



Differential Privacy - example

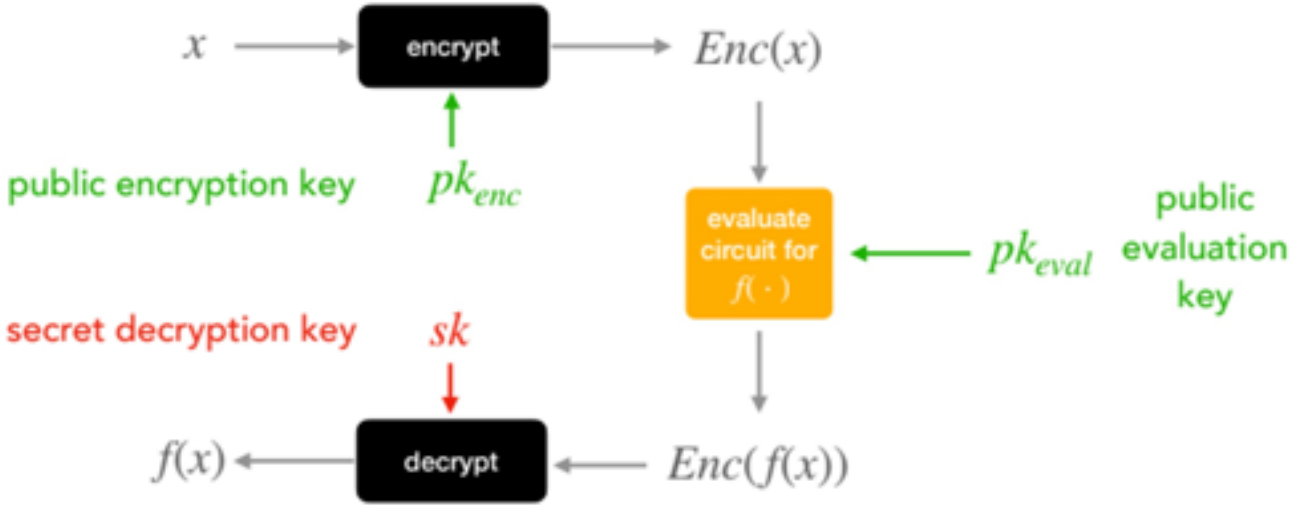
Participant	Answer
John	Yes
Mary	No
Will	No
James	Yes
Lisa	No

Participant	Answer
John	Yes
Mary	No
Will	No
James	Yes
Lisa	No

Participant	Answer
John	Yes
Mary	No
Will	No
James	Yes
Lisa	Yes

Homomorphic Encryption

- Computations on encrypted data without having to decrypt
- First generation support additive and multiplicative operations on cipher text

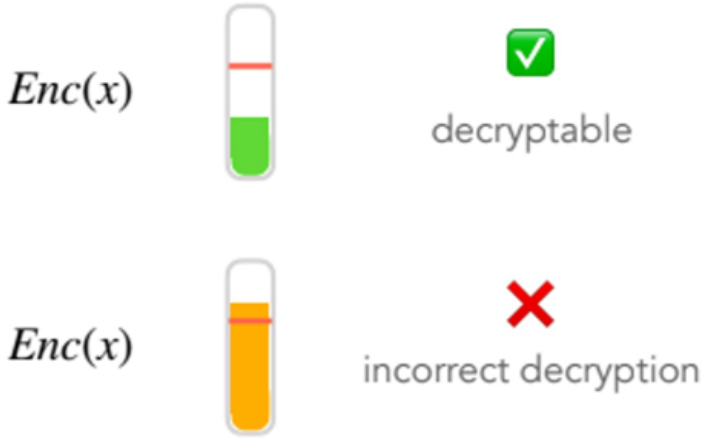


Homomorphic Encryption - limitations

- How to handle an increase of noise

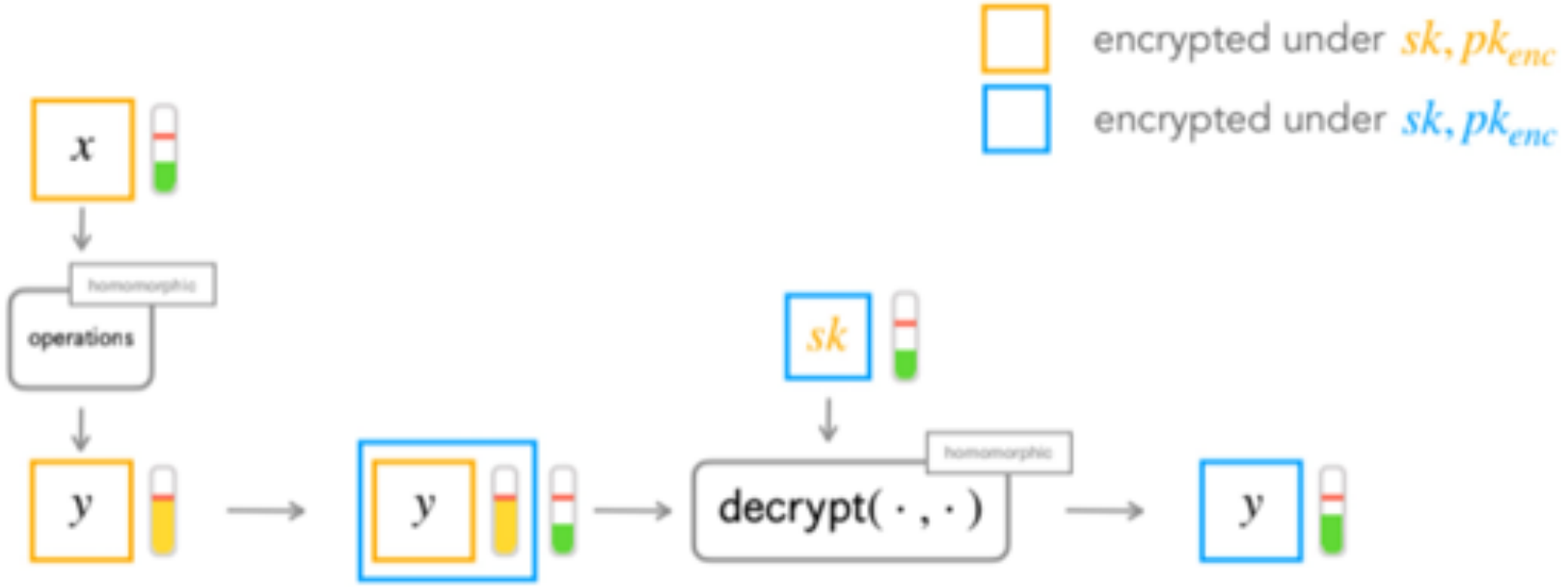
$Enc(x), Enc(y) \rightarrow Enc(x \oplus y)$ noises are added

$Enc(x), Enc(y) \rightarrow Enc(x \otimes y)$ noises are multiplied (size doubles)



Homomorphic Encryption - bootstrapping

- Denoise the ciphertext



... but this is very slow

Homomorphic Encryption - evolution

- Second generation cryptosystems
  - Feature a much slower growth of noise during the homomorphic computations

- Third generation cryptosystems
  - New technique that avoid expensive “relinearization” step

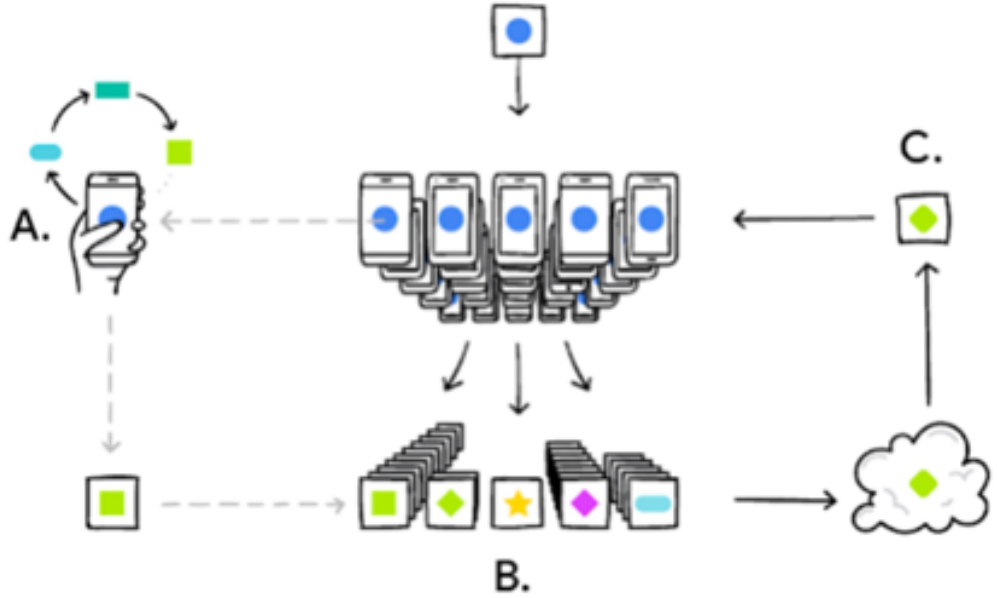
- Fourth generation cryptosystems
  - Approximate homomorphic encryption scheme

Main problem still: not efficient

Federated Learning

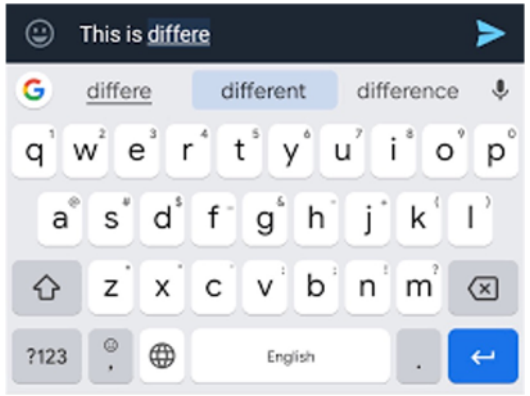
Federated Learning

- Shared model on multiple devices
- Updates are suggested (A)
- Updates are collected (B)
- Update is pushed (C)



Federated Learning Use Case: Google GBoard

- Prediction of words based on context
- Model is updated on correct predictions
- Updates are pushed to the central model
- Phone stores a user-specific version



Federated Learning Use Case: Google GBoard

- Adjusted ML version of SGD
- Processing load shifted
- Smart scheduling of updates
- Update compression and encryption

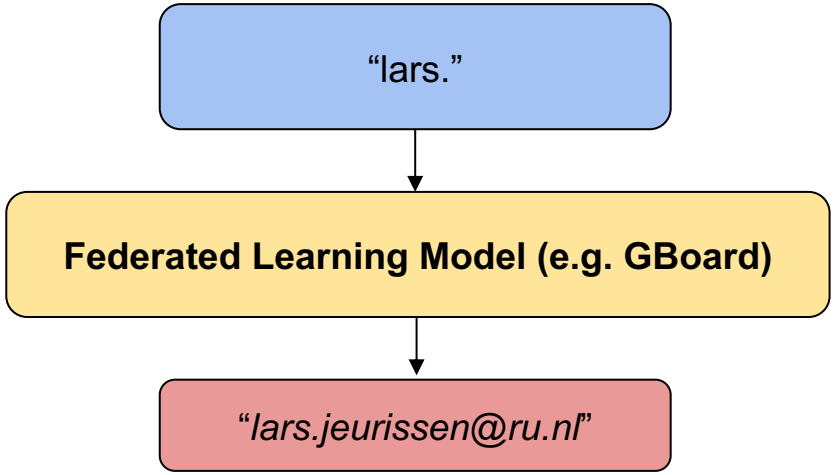
But… How does this relate to privacy?

Federated Learning: Privacy Issues

- What happens when user updates are pushed to the global model?

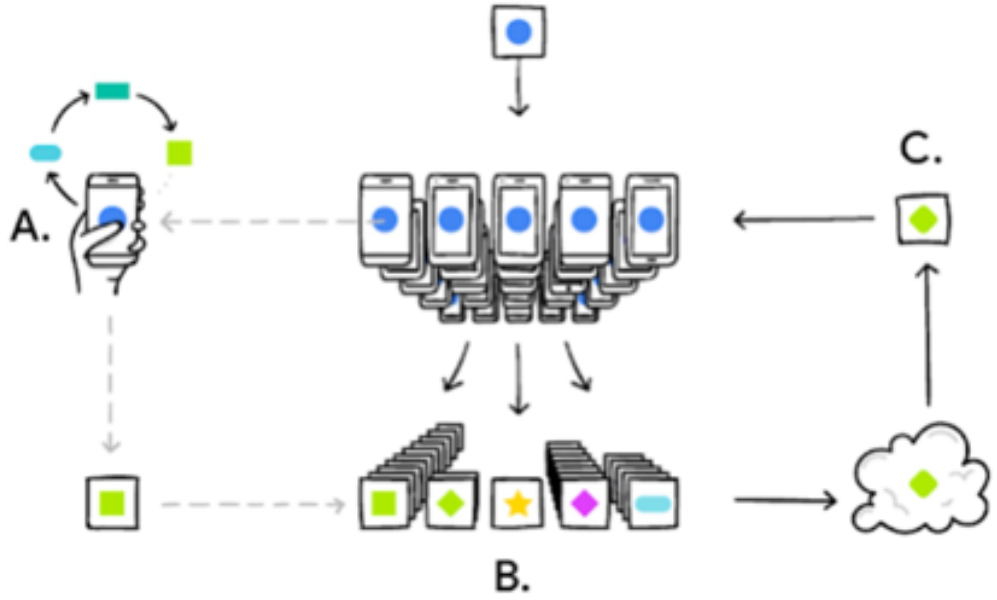
Federated Learning: Privacy Issues

- What happens when user updates are pushed to the global model?
- Recall large language models…
- Differential Privacy mitigation works… but decreases performance dramatically
- Other countermeasures are ineffective
- Effective attacks exist! (Boenisch et al.)



Federated Learning Privacy - How does Google deal with it?

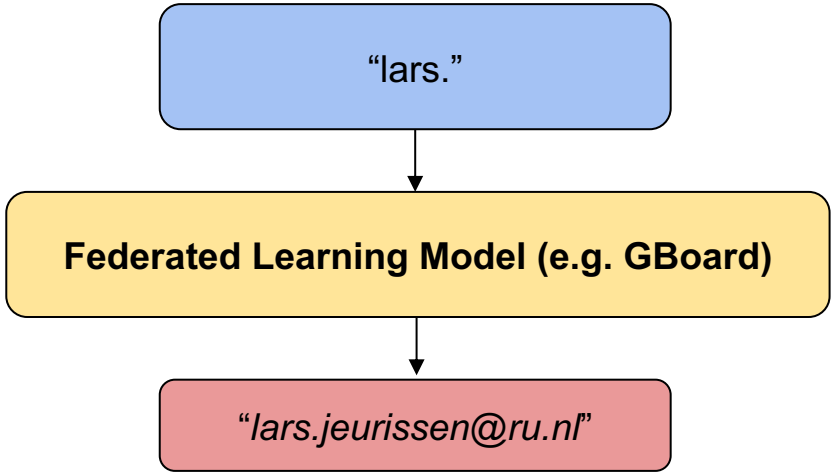
- Updates are assembled
- Only popular updates are actually sent to the model



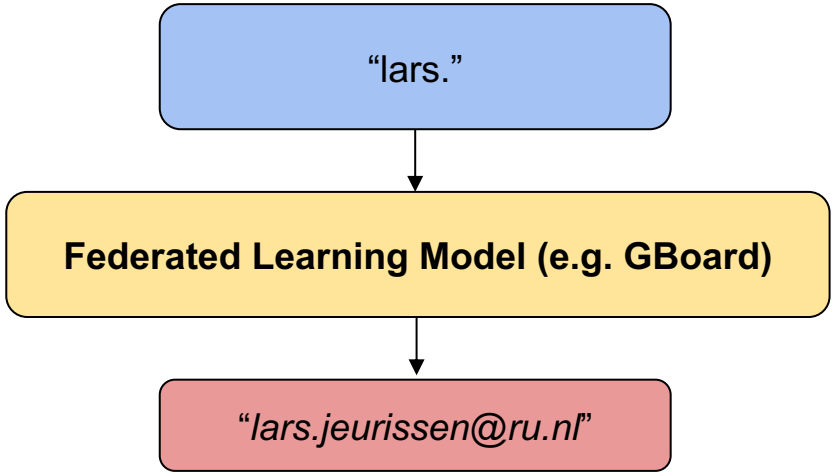
Quiz

Who uses an Android phone?

Which of you knew about this attack?

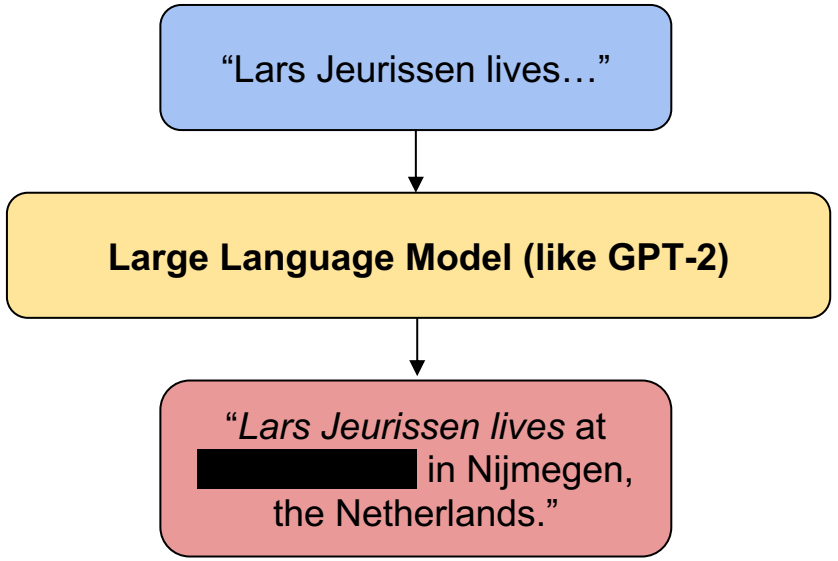


Who would change their keyboard because of this?



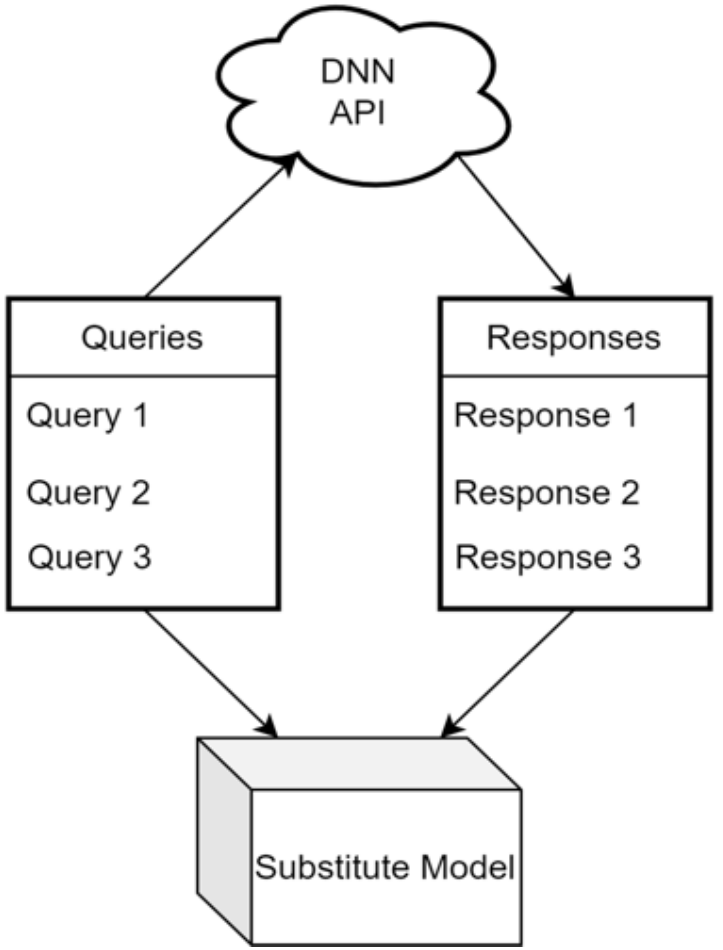
GPT-3 Corpus: 45 TB text data

What kind of sources of data should be used to train GPT?



Model Extraction Attack

Who thinks this is ethically acceptable?



Questions?

References

- Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. When the Curious Abandon Honesty: Federated Learning Is Not Private. arXiv preprint arXiv:2112.02918, 2021.
- Nicholas Carlini. Privacy Considerations in Large Language Models. Google AI Blog, 2020.
- Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery.
- Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in Pharmacogenetics: An {End-to-End} Case Study of Personalized Warfarin Dosing. In 23rd USENIX Security Symposium (USENIX Security 14), pages 17–32, 2014.
- Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. PRADA: Protecting Against DNN Model Stealing Attacks. In 2019 IEEE European Symposium on Security and Privacy (EuroS P), pages 512–527, 2019.
- Brendan McMahan and Ramage Daniel. Federated Learning: Collaborative Machine Learning without Centralized Training Data. Google AI Blog, 2017.

References

- Pascal Paillier. Introduction to Homomorphic Encryption. FHE.org, 2020.
- Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security, pages 506–519, 2017.
- Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. pages 3–18, 2017.
- Abhishek Tandon. Differential Privacy. Medium, 2019.
- Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 253–261, 2020. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into Deep Learning. arXiv preprint arXiv:2106.11342, 2021.
- Yinghua Zhang, Yangqiu Song, Jian Liang, Kun Bai, and Qiang Yang. Two sides of the same coin: White-box and black-box attacks for transfer learning. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2989–2997, 2020.