Privacy and Security in Cloud-based ML

Clement Fung, Ivan Beschastnikh
CPSC 416 Guest Lecture

Networks Systems Security lab
http://nss.cs.ubc.ca

Outline

- Introduction: cloud machine learning (ML)
- Threat models in distributed ML
- Attacks on ML
- Defenses for ML
- Our secure ML research at UBC

Outline

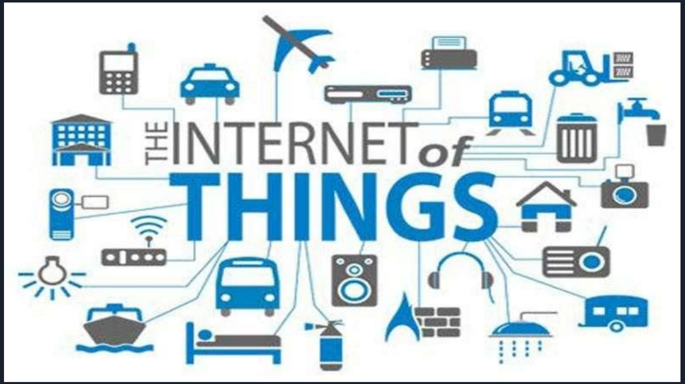
- **Introduction: cloud machine learning (ML)**
- Threat models in distributed ML
- Attacks on ML
- Defenses for ML
- Our secure ML research at UBC

Machine Learning is Everywhere

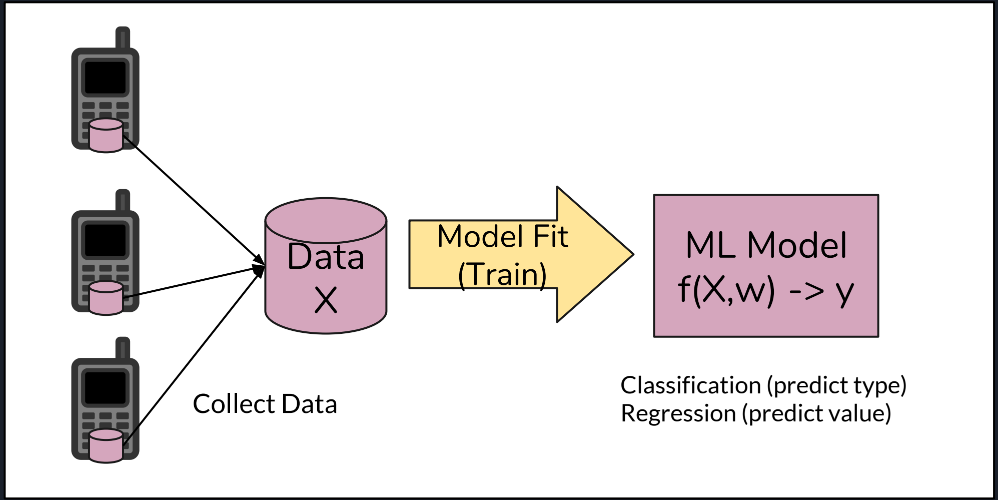
- Data collection at massive scales
- Analysis for everything

Data and Analysis are Decentralized

- Internet of things (large scale sensor networks)
- Live mobile analytics (maps/routing/traffic)



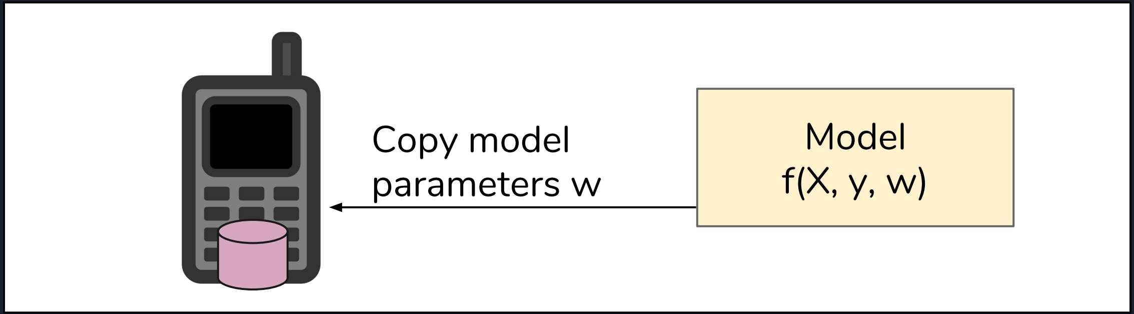
Gentle ML Overview



- X: labelled data features
  - E.g. Square footage
- y: predicted output
  - E.g. House value
  - Categorical or numerical
- w: model parameters
  - Feature weighting
  - Depends on model type
    - Assume arbitrary vector of floats

ML: Stochastic Gradient Descent

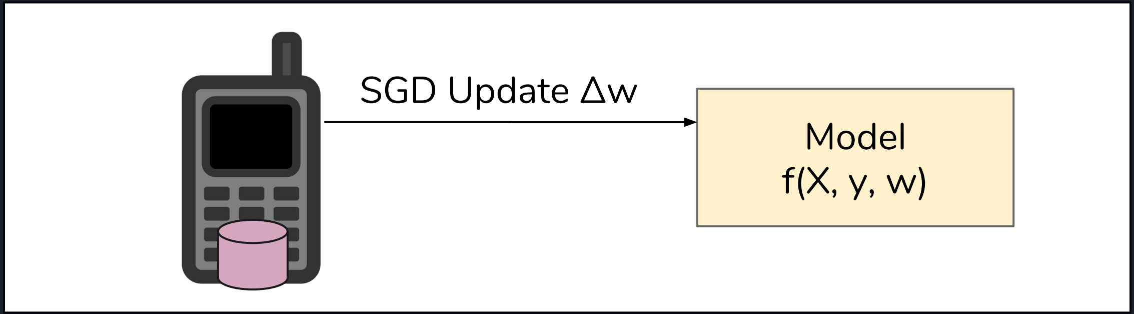
- SGD: General <u>iterative</u> algorithm for model training [1]
  - Can apply to regressions, deep learning, recommender systems, etc.



[1] Léon Bottou "*Large-Scale Machine Learning with Stochastic Gradient Descent*", COMPSTAT '10

ML: Stochastic Gradient Descent

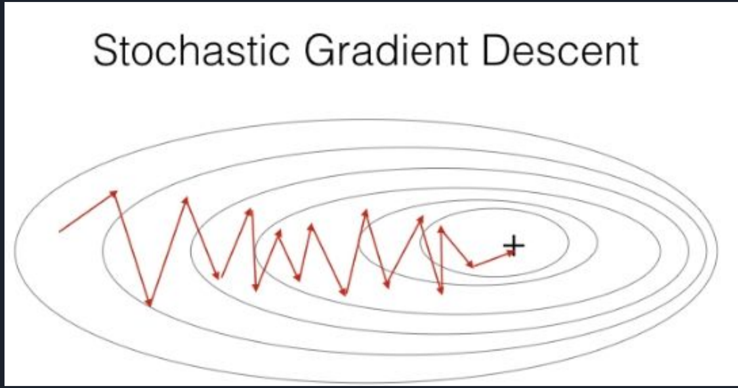
- SGD: General <u>iterative</u> algorithm for model training [1]
  - Can apply to regressions, deep learning, recommender systems, etc.



[1] Léon Bottou "*Large-Scale Machine Learning with Stochastic Gradient Descent*", COMPSTAT '10

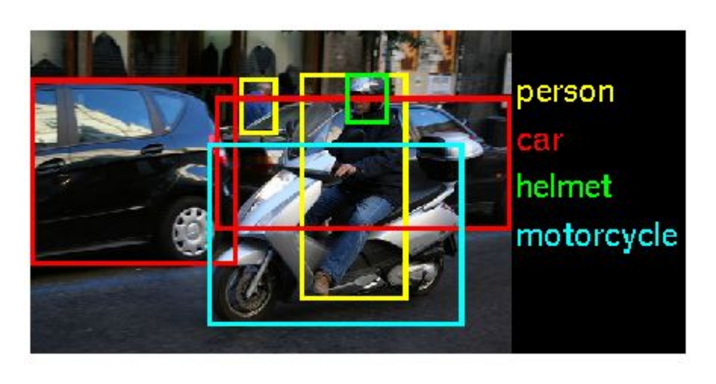
ML: Stochastic Gradient Descent

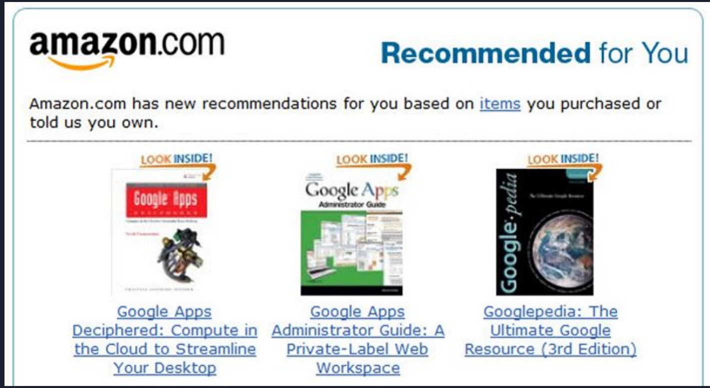
- Repeat until done!
  - Using some convergence gradient metric
  - For a fixed number of iterations

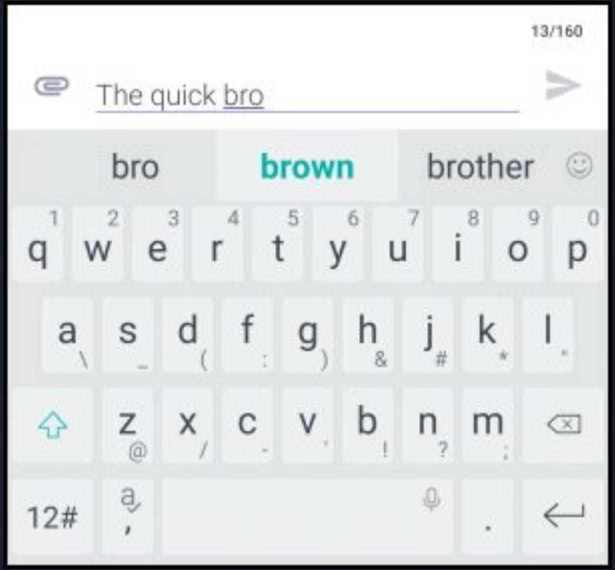


[1] Léon Bottou "*Large-Scale Machine Learning with Stochastic Gradient Descent*", COMPSTAT '10

ML Use Cases


person
car
helmet
motorcycle


amazon.com
Recommended for You
Amazon.com has new recommendations for you based on items you purchased or told us you own.
LOOK INSIDE!
Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop
LOOK INSIDE!
Google Apps Administrator Guide: A Private-Label Web Workspace
LOOK INSIDE!
Googlepedia: The Ultimate Google Resource (3rd Edition)

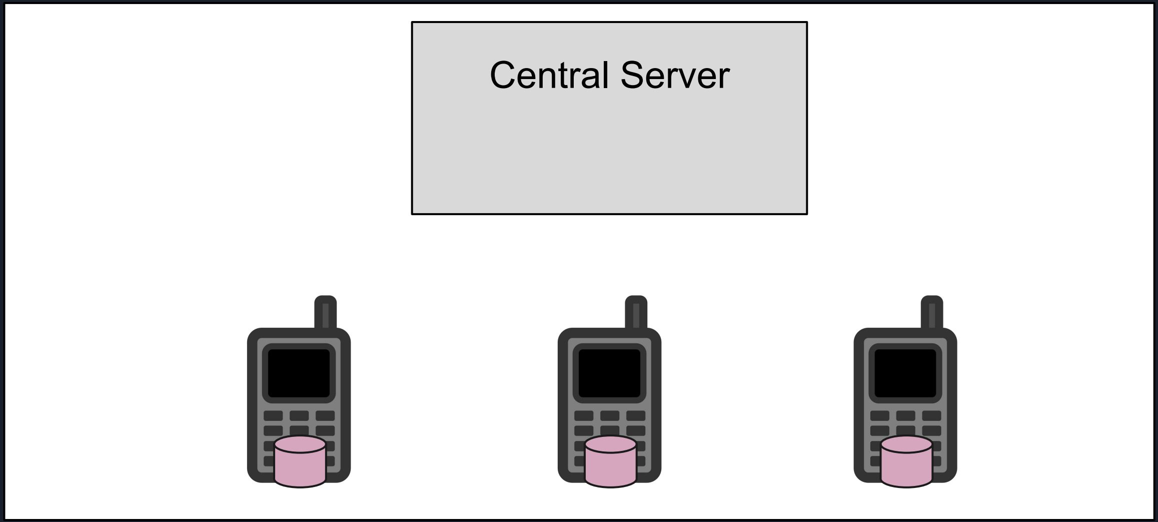

13/160
The quick bro
bro
brown
brother
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
12#

Modern Large Scale ML Solutions

- What if there is <u>a lot of data</u>?
- Modern solutions: store it all in a data centre and train on it
    - 3 common libraries to do this...

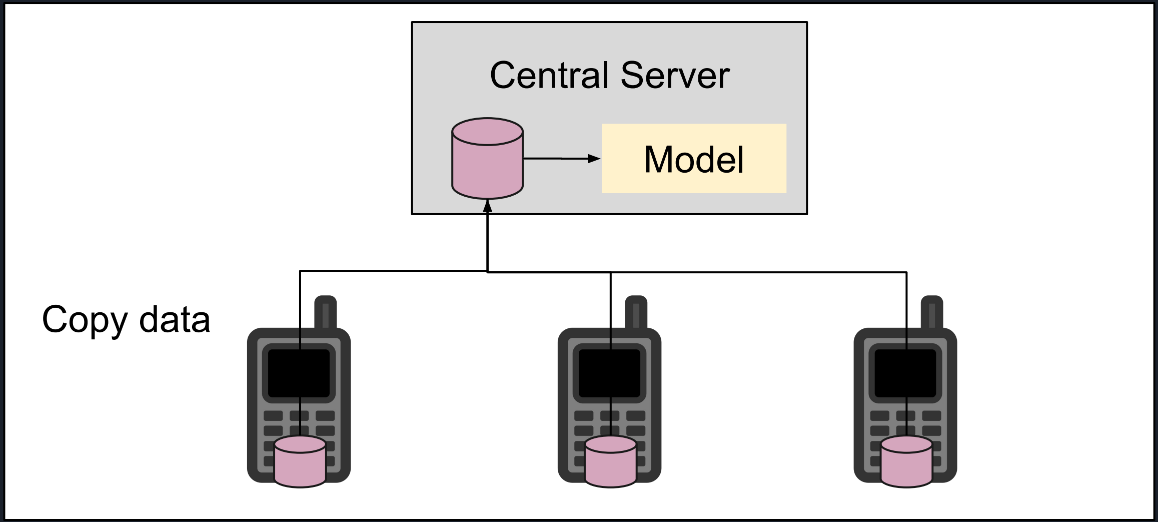
Distributed ML: Aggregate Data

- Option 1: Centralize the data, then train a model



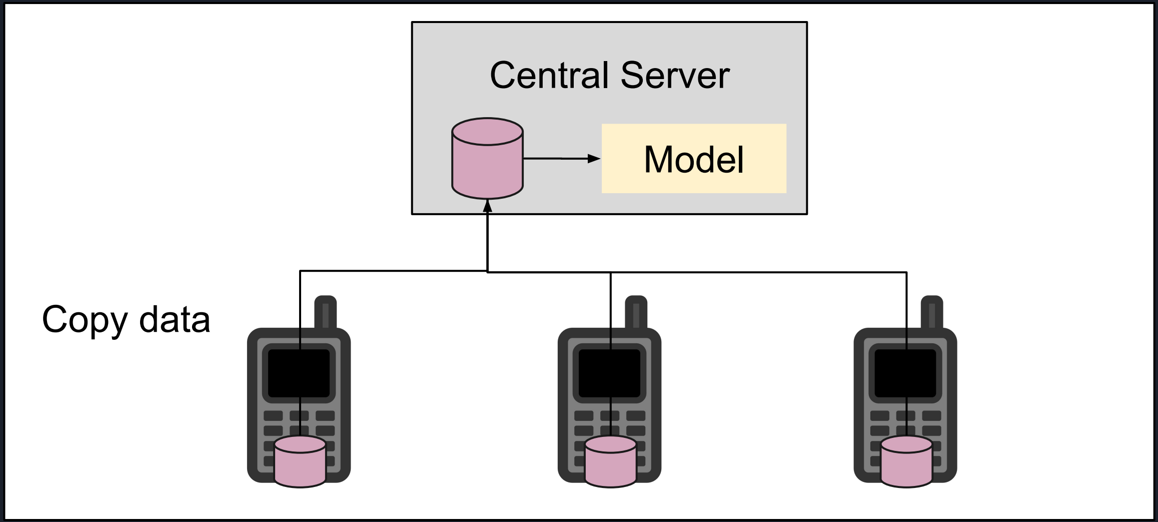
Distributed ML: Aggregate Data

- Option 1: Centralize the data, then train a model



Distributed ML: Aggregate Data

- Option 1: Centralize the data, then train a model
  - But at massive scale, this is <u>expensive</u> and <u>not private</u>



The Need for Privacy

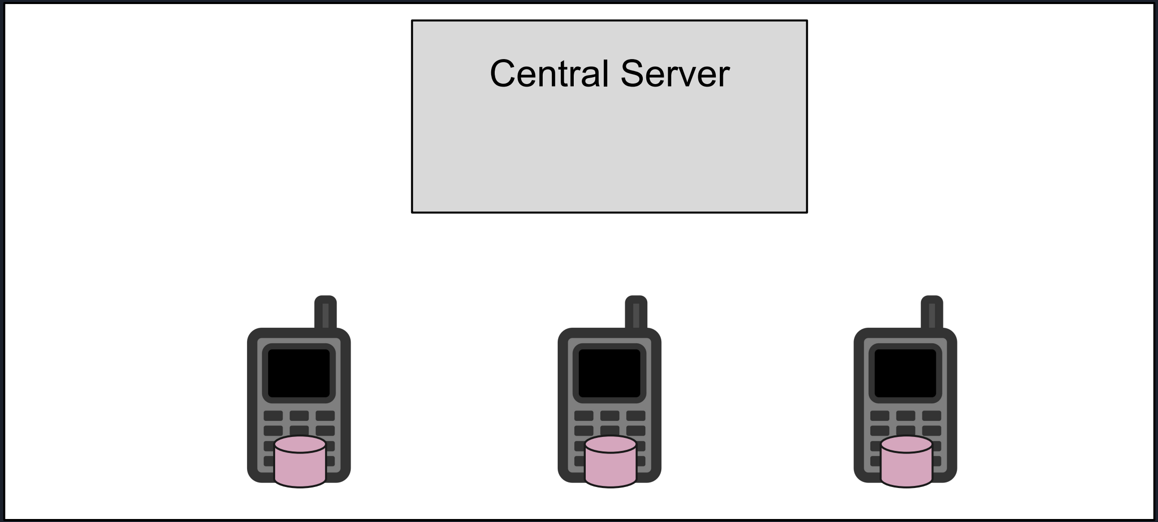
- Data can be <u>sensitive</u> in nature
    - Photos, location info, voice recordings
- Typically, a centralized service performs model training
    - Do we have to trust Google with our data?

Costs of Centralization

- Growth of data is costly!
  - ~2.3 billion smartphones in world today
  - Use of smartphones and tablets increasing
- Collecting data, keeping it updated is expensive
- Today’s improvements: perform ML without data transfer
  - Aggregating locally trained models
  - Training over the network: federated learning
    - We’ll get back to this one

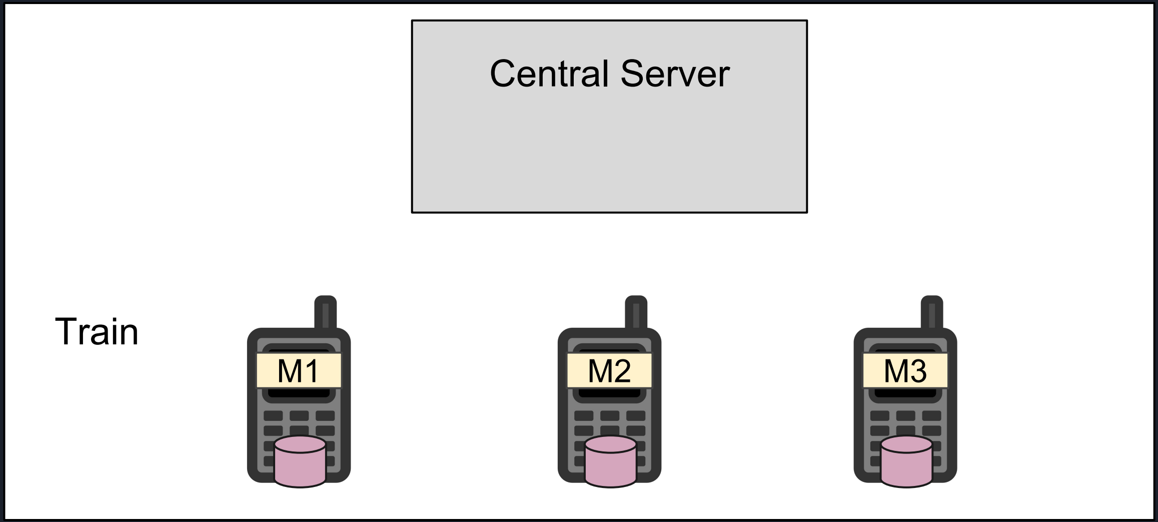
Distributed ML: Aggregate Outputs

- Option 2: Train local models and aggregate predictions
  - Various methods (forests, bagging, transfer learning)



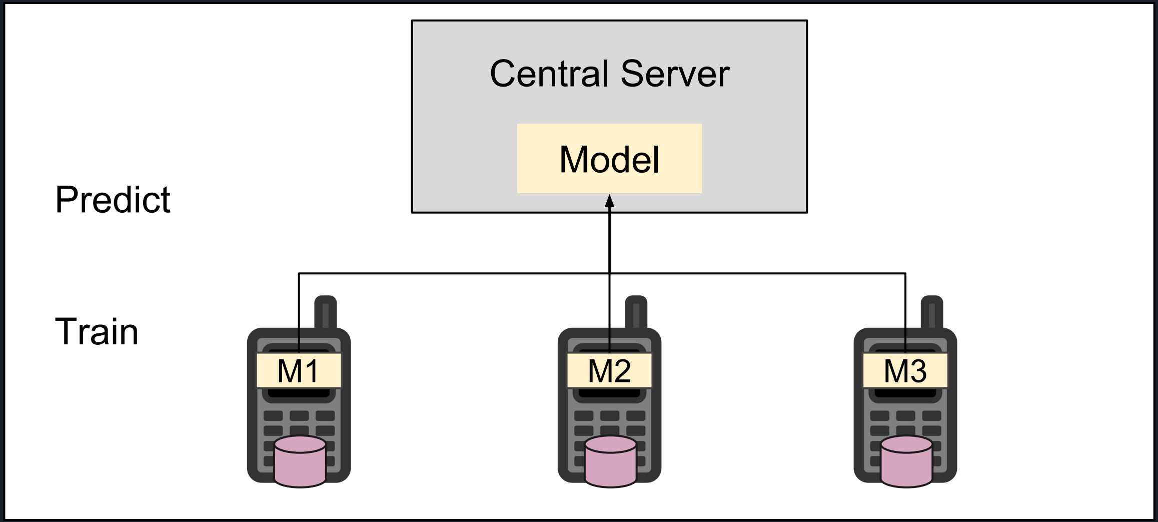
Distributed ML: Aggregate Outputs

- Option 2: Train local models and aggregate predictions
  - Various methods (forests, bagging, transfer learning)



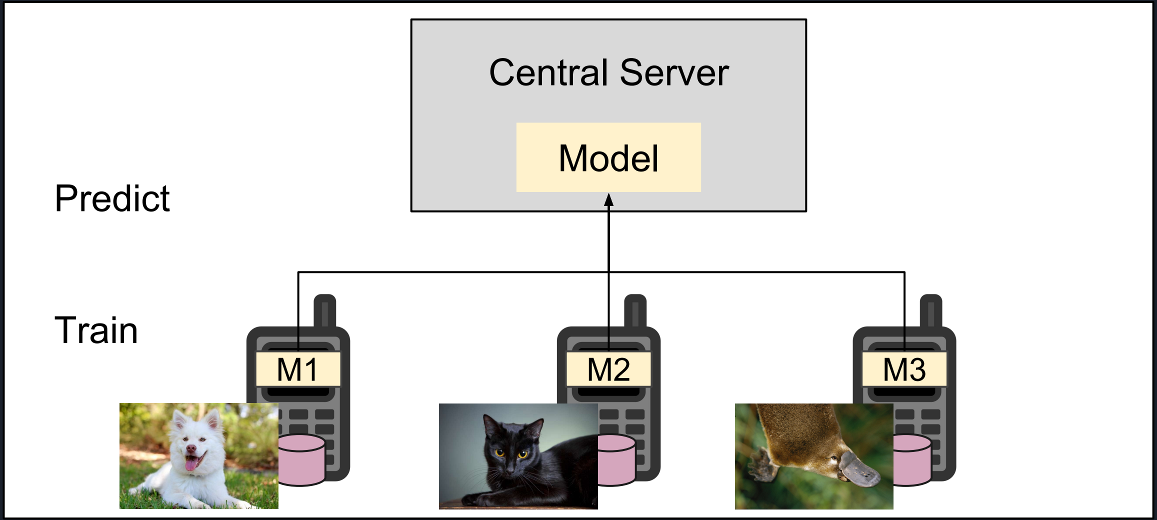
Distributed ML: Aggregate Outputs

- Option 2: Train local models and aggregate predictions
  - Various methods (forests, bagging, transfer learning)



Distributed ML: Aggregate Outputs

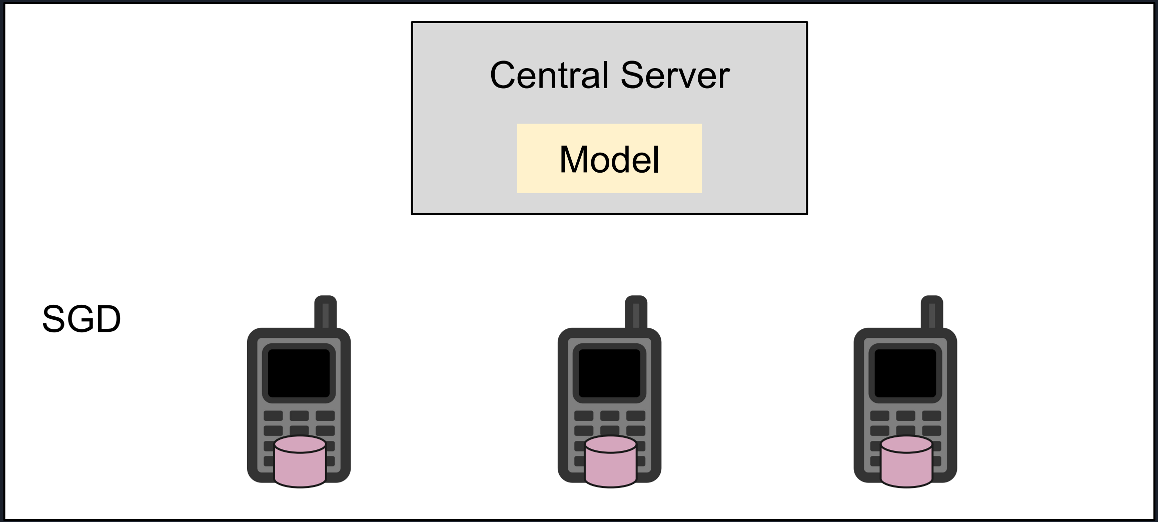
- Option 2: Train local models and aggregate predictions
  - Various methods (forests, bagging, transfer learning)
  - But when data is highly non-uniform, this is suboptimal [1]



[1] Bellet et al. "*Personalized and Private Peer-to-Peer Machine Learning*", AISTATS '18

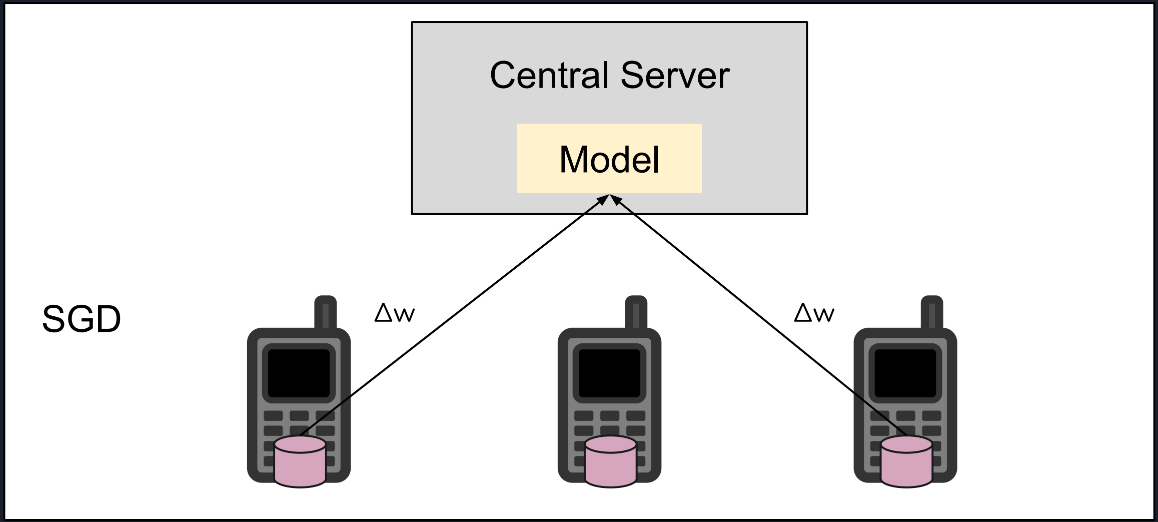
Distributed ML: Federated Learning

- Option 3: Send SGD updates over network



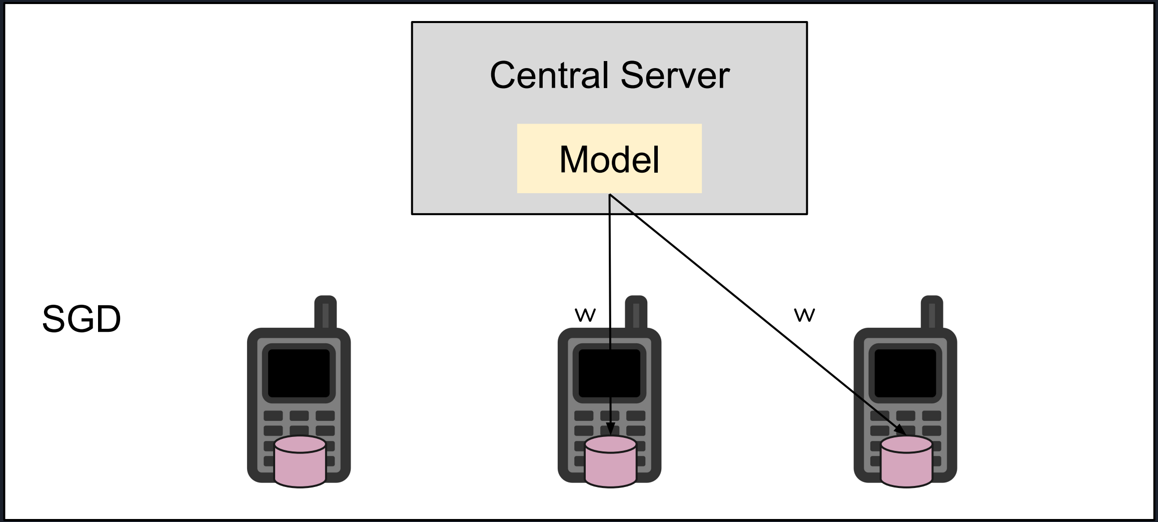
Distributed ML: Federated Learning

- Option 3: Send SGD updates over network



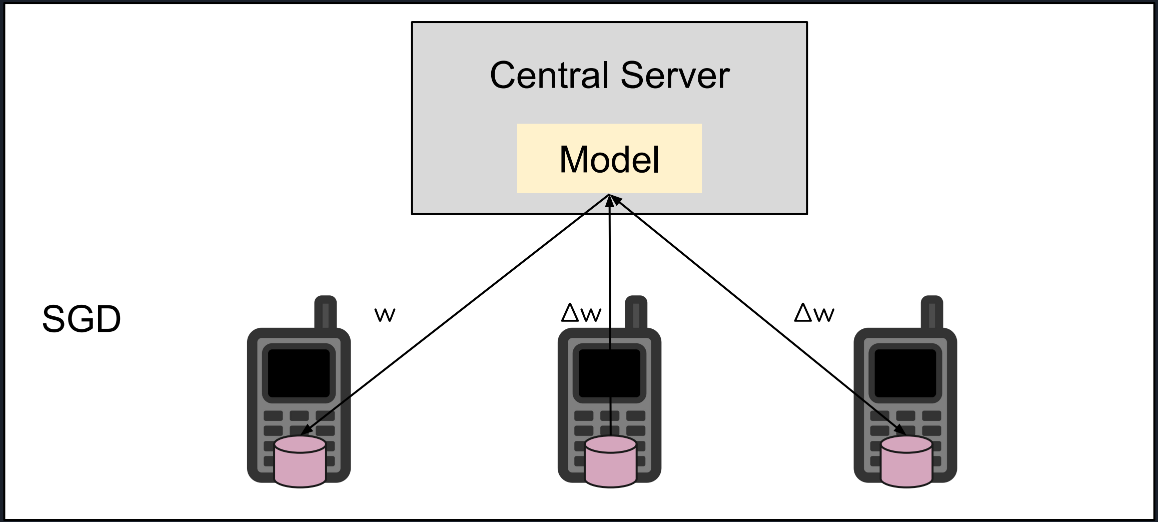
Distributed ML: Federated Learning

- Option 3: Send SGD updates over network



Distributed ML: Federated Learning

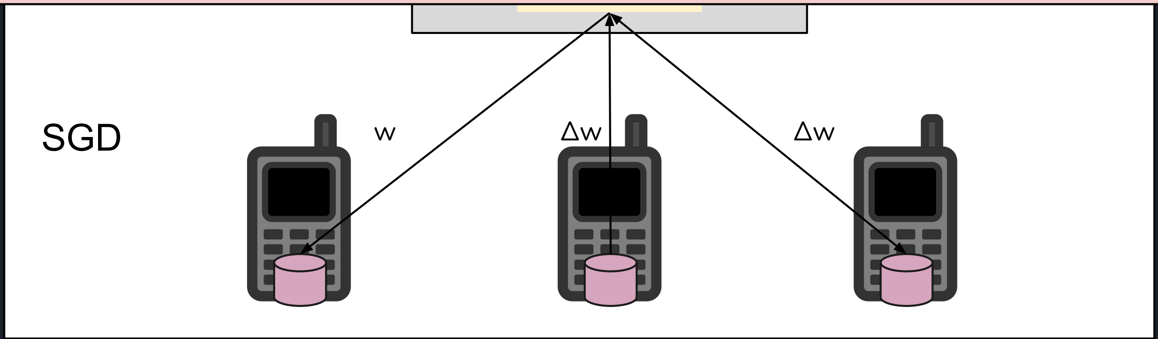
- Option 3: Send SGD updates over network



Distributed ML: Federated Learning

- Option 3: Send SGD updates over network

Federated Learning (Google's new 2017 algorithm): Data never leaves the client, as good as centralized



[1] Bellet et al. "*Personalized and Private Peer-to-Peer Machine Learning*", AISTATS '18

Federated Learning Tradeoffs

- <u>Benefits</u>: client centric view enables privacy
  - Data remains with client
  - Perform SGD <u>locally</u>
    - Can modify the protocol for further privacy
- <u>Drawbacks:</u> less control for server
  - Clients used to just provide data, now they are capable of many new attacks
  - Depends on the <u>**threat model**</u>

[1] Cynthia Dwork. "*Differential Privacy*" ICALP '06
[2] Song et al. "*Stochastic gradient descent with differentially private updates*" GlobalSIP '13
[3] Bonawitz et al. "*Practical Secure Aggregation for Privacy-Preserving Machine Learning*", CCS '17

Outline

- Introduction: cloud machine learning (ML)
- **Threat models in distributed ML**
- Attacks on ML
- Defenses for ML
- Our secure ML research at UBC

Threat Models in ML

Different Levels of Privacy

- How will the ML system be used?
  - User model
  - Threat model
- For example, three types of privacy models [1]:
  - Private networks (I trust everyone here)
  - Public networks (Most common, open to join with account)
  - Anonymous networks (Completely hide all information)

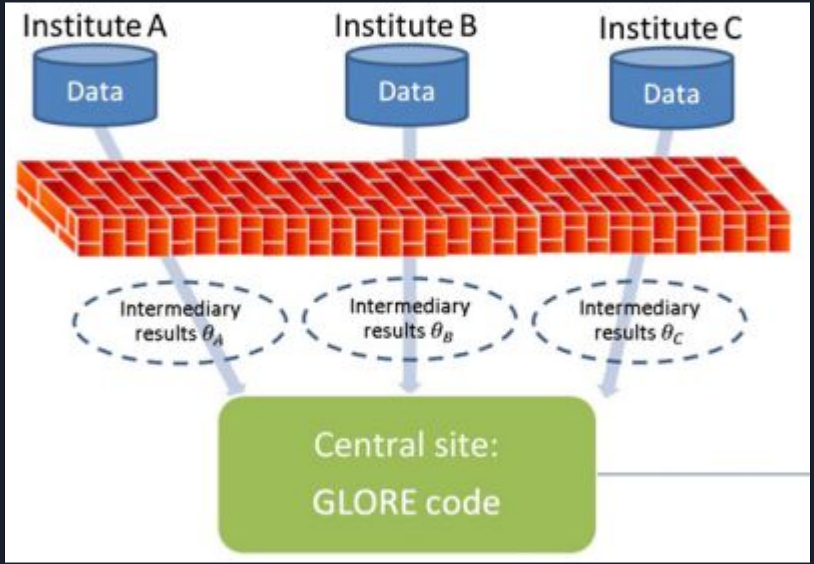
[1] Bonawitz et al. "*Practical Secure Aggregation for Privacy-Preserving Machine Learning*", CCS '17

Types of Networks

- Private Network
  - Between a fixed set of known users
  - Not open to outsiders
- Public Network
  - Open to public users
  - Typically require external verification (Account)
- Anonymous Network
  - Open, but identities are hidden

Private Network ML

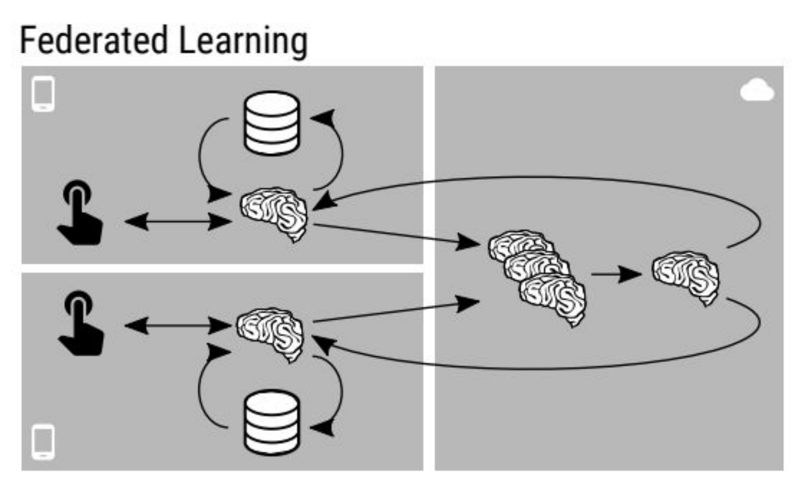
- Weak/no threat model
  - No malicious users, no new users
  - Everyone follows protocol, no attacks
- i.e. Sharing models and analysis across hospitals



[1] Wu et al. "*Grid Binary LOgistic REgression (GLORE): building shared models without sharing data*". Journal of the American Medical Informatics Association, Volume 19, Issue 5

Public Network ML

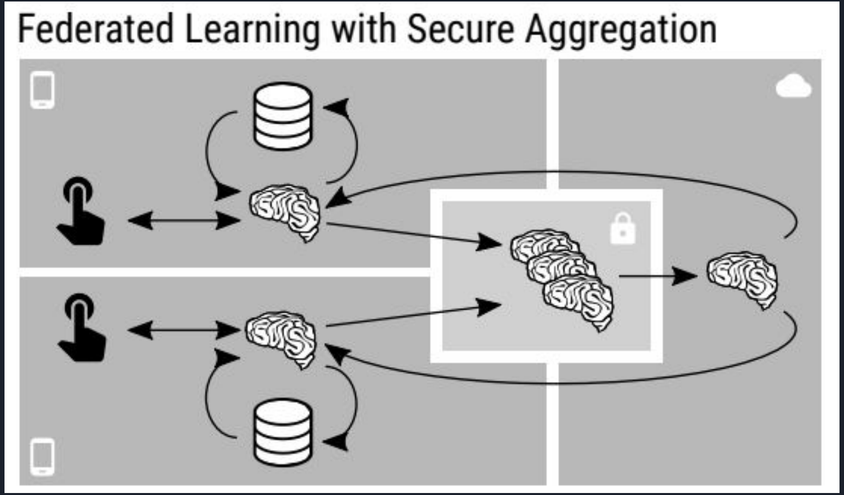
- Mild threat model
  - Users mount attacks, could use sybils
- Users don't trust server or other users
- Only data byproducts revealed to server
- Federated learning for Gboard [1]



[1] McMahan et al. "*Federated Learning: Collaborative Machine Learning without Centralized Training Data*". Google Research Blog 2017

Anonymous Network ML

- Strongest threat model
- Users do not know each other or share identities
  - No user authentication
- Users do not trust anyone with data or updates
  - Google secure aggregation [1]



[1] Bonawitz et al. "*Practical Secure Aggregation for Privacy-Preserving Machine Learning*", CCS '17

Security Performance Tradeoffs

- "Why don't we just use the strongest security model?"
  - Usually performance/usability concerns
  - Google secure aggregation for federated learning
    - 4 rounds of communication between users and service!
    - With 1000 clients, takes ~5s per iteration
    - On wide area network, up to ~28s per iteration
  - A typical ML workload can take <u>1000s of iterations</u>!

[1] Bonawitz et al. "*Practical Secure Aggregation for Privacy-Preserving Machine Learning*", CCS '17

Security Performance Tradeoffs

- "Why don't we just use the strongest security model?"
  - Usually performance/usability concerns

Security tradeoffs: Making realistic user and threat model assumptions for your use case is vital!

      - On wide area network, up to ~28s per iteration
  - A typical ML workload can take <u>1000s of iterations</u>!

[1] Bonawitz et al. "*Practical Secure Aggregation for Privacy-Preserving Machine Learning*", CCS '17

Outline

- Introduction: cloud machine learning (ML)
- Threat models in distributed ML
- **Attacks on ML**
- Defenses for ML
- Our secure ML research at UBC

Why do we attack ML?

- As we already know, ML is used everywhere!
- To influence model prediction outputs:
  - Model poisoning [1]
  - Adversarial examples [2]
- To gain extra information/data from users:
  - Inversion [3]
  - Model extraction [4]

[1] Huang et al. *"Adversarial Machine Learning"*. AISec '11
[2] Goodfellow et al. "*Explaining and Harnessing Adversarial Examples*" ICLR '15
[3] Fredrikson et al. "*Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*" CCS '15
[4] Tramer et al. "*Stealing Machine Learning Models via Prediction APIs*" Usenix Sec '16

How do we attack ML?

- Supplying malicious training data:
  - Model poisoning [1]
- Supplying malicious test data:
  - Adversarial examples [2]
- Through information in prediction APIs:
  - Inversion [3]
  - Model extraction [4]

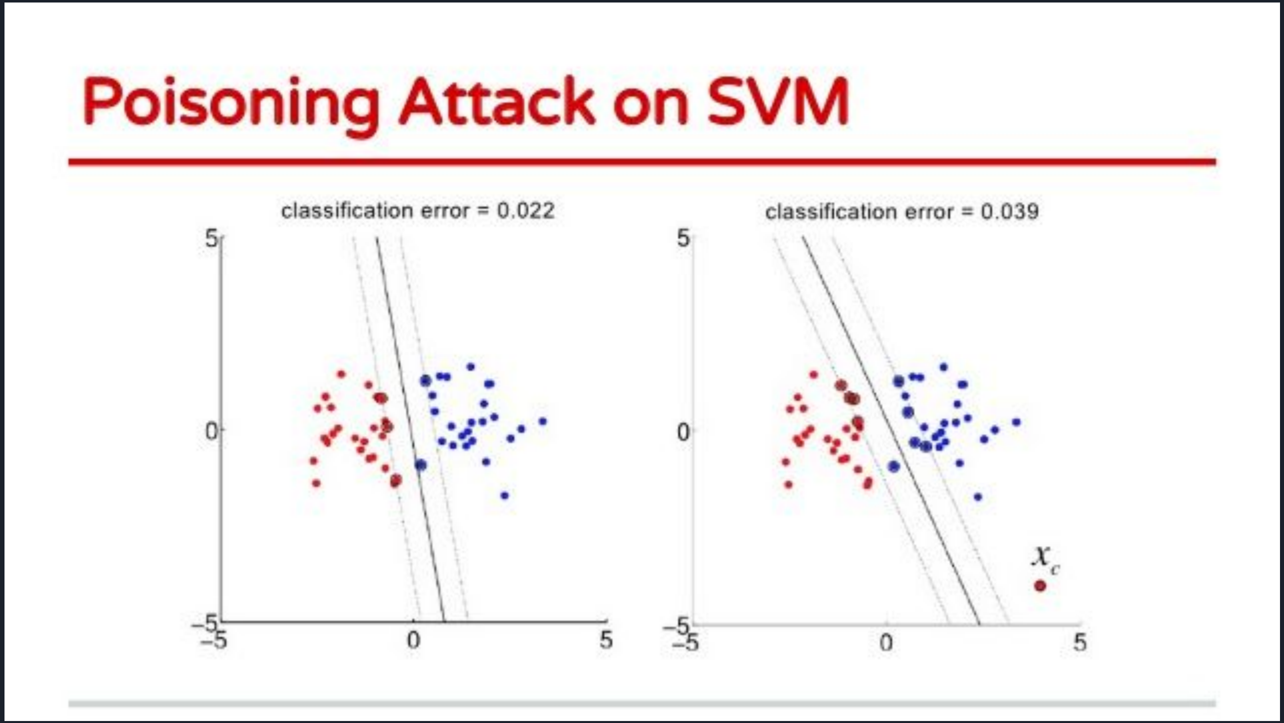
[1] Huang et al. *"Adversarial Machine Learning"*. AISec '11
[2] Goodfellow et al. "*Explaining and Harnessing Adversarial Examples*" ICLR '15
[3] Fredrikson et al. "*Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*" CCS '15
[4] Tramer et al. "*Stealing Machine Learning Models via Prediction APIs*" Usenix Sec '16

Poisoning Attacks

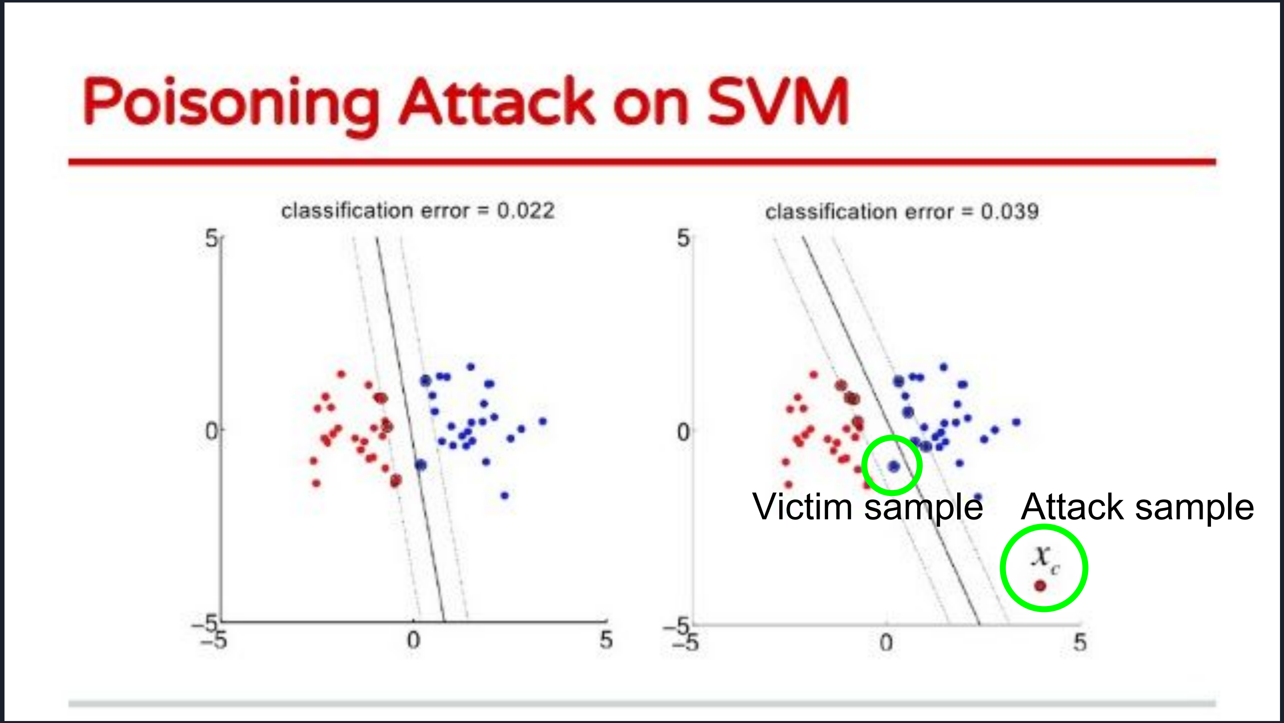
- Two types: [1]
  - *<u>Random</u> attack*: Aim to decrease model accuracy
  - *<u>Targeted</u> attack*: Aim to increase/decrease classification of a specific point
    - I want my email to pass a spam filter
    - I want my advertisement to be displayed more

[1] Huang et al. *"Adversarial Machine Learning"*. AISec '11

Poisoning Attacks



Poisoning Attacks



Backdoor Attacks [1]

- A newer poisoning attack from 2017
- Use a small, unimportant part of model to hide signals in malicious training data. Exploit backdoor once model deployed.

Figure 7. A stop sign from the U.S. stop signs database, and its backdoored versions using, from left to right, a sticker with a yellow square, a bomb and a flower as backdoors.

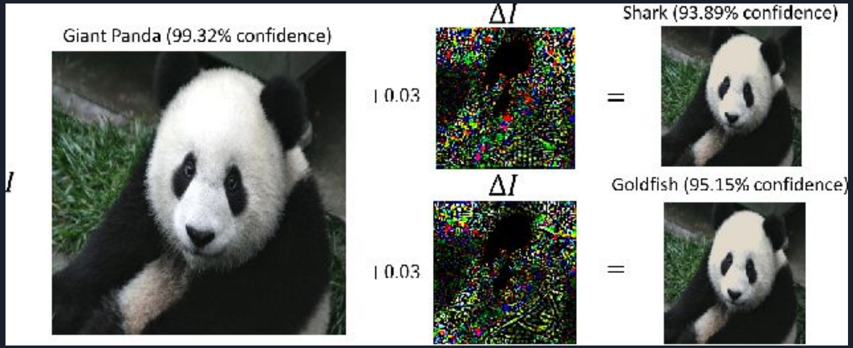
[1] Gu et al. *“BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain”*. ArXiv ‘17

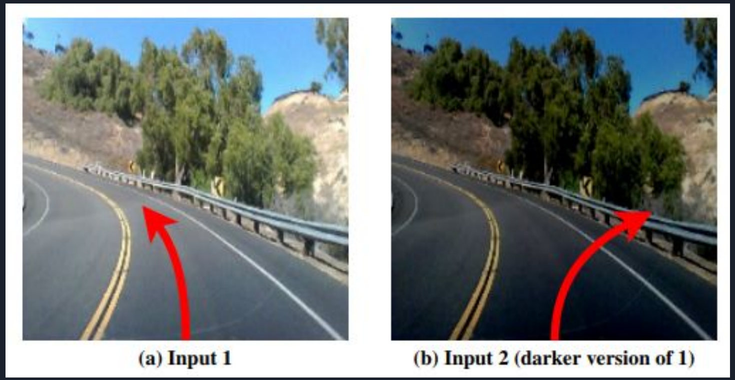
Adversarial Examples [1, 2, 3]

- Another way to exploit classifier mispredictions
- *On prediction:* A test point that evades a classifier
    - A recent discovery of deep learning
    - Since DL is very non-linear, easy to exploit

[1] Goodfellow et al. "*Explaining and Harnessing Adversarial Examples*" ICLR '15
[2] Pei et al. "*DeepXplore: Automated Whitebox Testing of Deep Learning Systems*" SOSP '17
[3] Li et al. "*Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics*" ICCV '17

Adversarial Examples [1, 2, 3]





[1] Goodfellow et al. "*Explaining and Harnessing Adversarial Examples*" ICLR '15
[2] Pei et al. "*DeepXplore: Automated Whitebox Testing of Deep Learning Systems*" SOSP '17
[3] Li et al. "*Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics*" ICCV '17

Inversion Attacks [1]

- Attacking public prediction APIs:
  - Prediction: “Given an example, predict its class”
  - By repeating this several times, the adversary can uncover private information about the model

[1] Fredrikson et al. “*Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*” CCS ‘15

Inversion Attacks [1]

- Model inversion: reconstruct training data
  - Use class confidence information from prediction query API
    - Train a generative model to create training examples
- [1]: Reconstruct training face from deep learning model after ~3000 prediction API calls.

[1] Fredrikson et al. "*Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*" CCS '15

Inversion Attacks [1]

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

[1] Fredrikson et al. "*Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*" CCS '15

Model Stealing Attacks [1]

- Similar to inversion, uncover the ML model itself

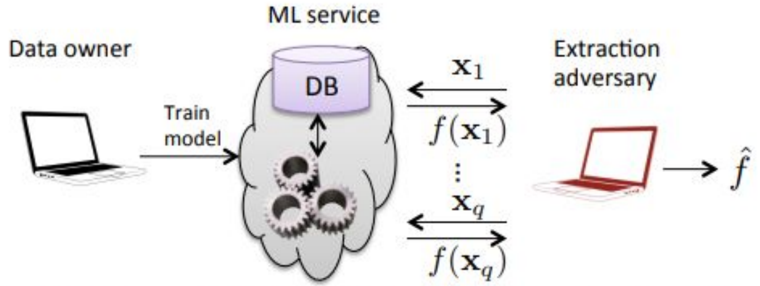


Figure 1: **Diagram of ML model extraction attacks.** A data owner has a model f trained on its data and allows others to make prediction queries. An adversary uses q prediction queries to extract an $\hat{f} \approx f$.

[1] Tramer et al. "*Stealing Machine Learning Models via Prediction APIs*" Usenix Sec '16

GAN Attack on Federated Learning [1]

- <u>In federated learning</u>:
  - Join system as client, but with <u>no data</u>
  - Use updates to train generative adversarial network (GAN)
    - A two part model that generates and classifies data
    - Used by adversaries to generate fake training data
  - Inversion attack, but clients are <u>more powerful</u> (see the model while trained)

[1] Hitaj et al. "*Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning*" CCS '17

GAN Attack on Federated Learning

- Federated Learning Inversion:

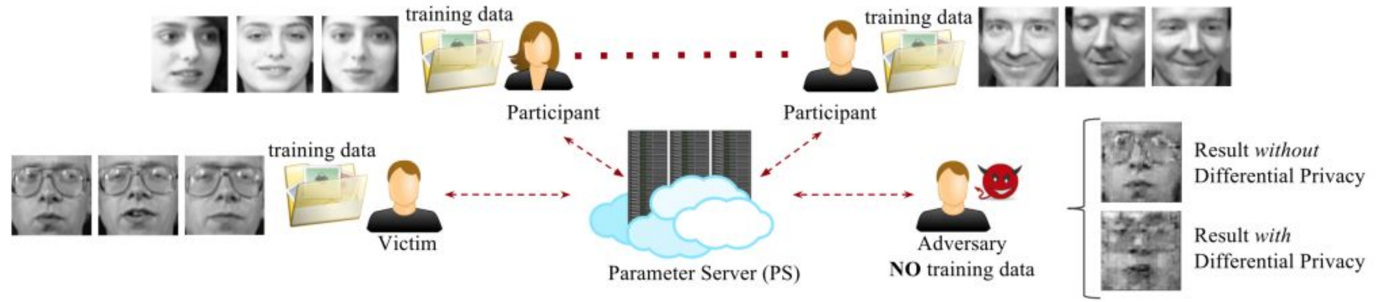


Figure 7: Collaborative deep learning with 41 participants. All 40 honest users train their respective models on distinct faces. The adversary has no local data. The GAN on the adversary's device is able to reconstruct the face stored on the victim's device (even when DP is enabled).

[1] Hitaj et al. "*Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning*" CCS '17

GAN Attack on Federated Learning

- Federated Learning Inversion:

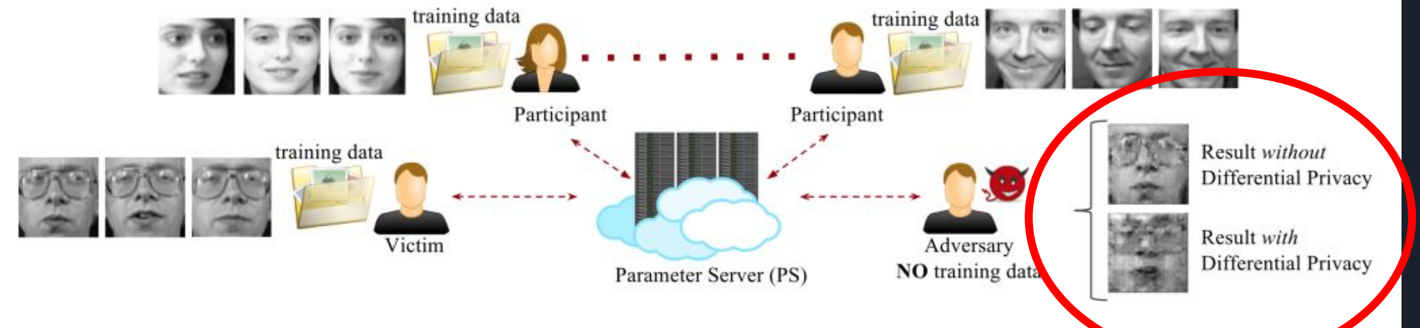


Figure 7: Collaborative deep learning with 41 participants. All 40 honest users train their respective models on distinct faces. The adversary has no local data. The GAN on the adversary's device is able to reconstruct the face stored on the victim's device (even when DP is enabled).

[1] Hitaj et al. "*Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning*" CCS '17

Sybil Attacks

- In public/anonymous networks, Sybils are a problem
  - Fake accounts created for additional leverage [1]
  - Sybil attacks on “crowd-sourced computations”
- In ML setting:
  - Attacks can become more powerful (poisoning, leakage)

[1] Doucuer et al. “*The Sybil Attack*” IPTPS '01

Sybil Attacks

Figure 1: Before the attack (left), Waze shows the fastest route for the user. After the attack (right), the user gets automatically re-routed by the fake traffic jam.

[1] Wang et al. "*Defending against Sybil Devices in Crowdsourced Mapping Services*", MobiSys '16

Is It All Hopeless?

- ML vulnerable to manipulation and leakage
- Ongoing: many defenses have been developed
  - The whole research field is back and forth work
  - Again, depends on the threat model: Define user and attacker assumptions
    - Big part of security research

Outline

- Introduction: cloud machine learning (ML)
- Threat models in distributed ML
- Attacks on ML
- **Defenses for ML**
- Our secure ML research at UBC

Data Privacy

Data Privacy

- Assuming a **public network:**
  - Users can know each other, willing to cooperate
  - Don’t want to share their data with each other or server
  - “Honest-but-curious”
- How can we train on multi-party data without breaking privacy?

Past Research: Why “Privacy” is Difficult

- “For privacy, can’t we just hide the labels?”
  - 2006 Netflix user dataset de-anonymized using IMDB [1]
  - 2006 AOL search database de-anonymized [2]
- <u>Anonymizing is insufficient</u>: auxiliary data breaks anonymity!

[1] Narayanan et al. “*Robust De-anonymization of Large Sparse Datasets*”, S&P ‘08
[2] NYTimes “*A Face Is Exposed for AOL Searcher No. 4417749*” NYTimes ‘06

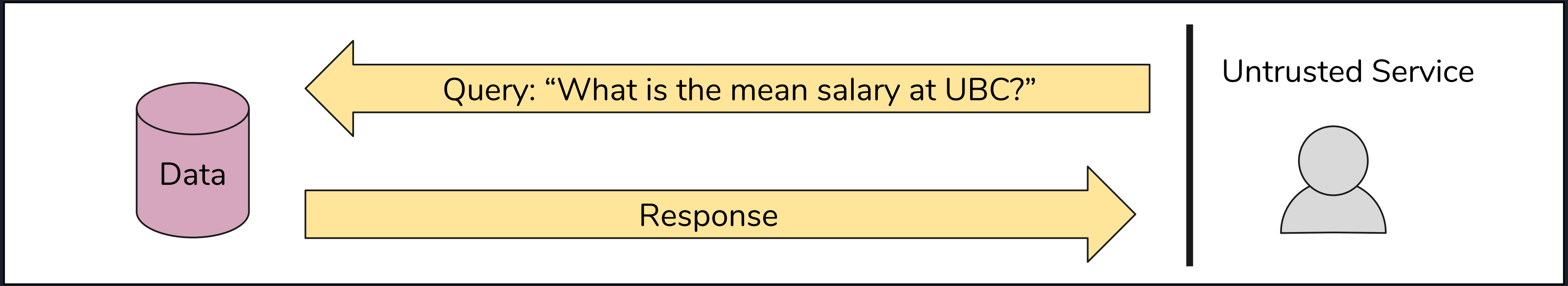
Differential Privacy (DP) [1]

- Mechanisms that <u>protecting privacy of datasets</u> when used
- Record level DP:
  - Protects individual records
  - A dataset with/without given example is indistinguishable
- Generally, get privacy from <u>adding noise</u> to responses
  - Privacy-utility tradeoff: <u>more noise, less accuracy</u>
  - Parameterized by ε (lower ε: more private, less utility)

[1] Cynthia Dwork. "*Differential Privacy*" ICALP '06

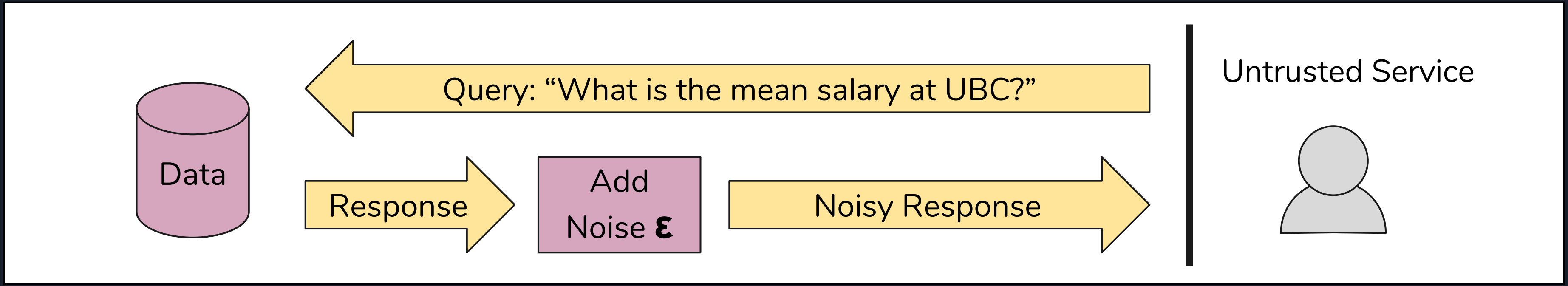
Differential Privacy (DP) Example

- Untrusted service that knows the current mean salary at UBC
    - Then, a new employee joins
- Can directly compute the salary of employee!



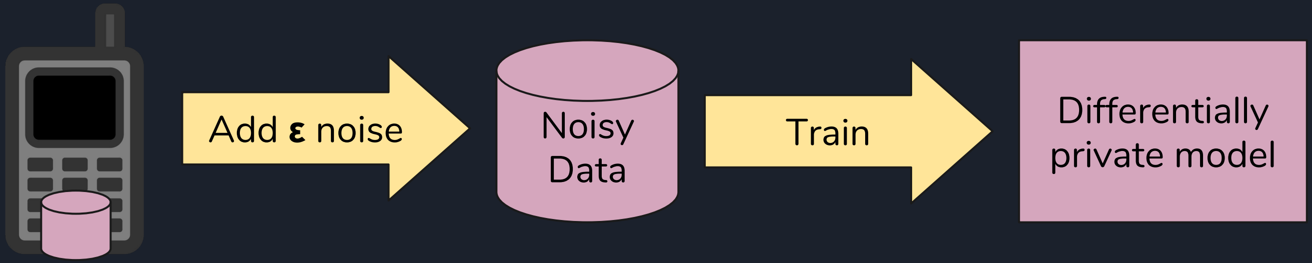
Differential Privacy (DP) Example

- Untrusted service that knows the current mean salary at UBC
    - Then, a new employee joins
- Add noise to the output
- Cannot directly compute the salary of employee!



Differential Privacy (DP) in ML

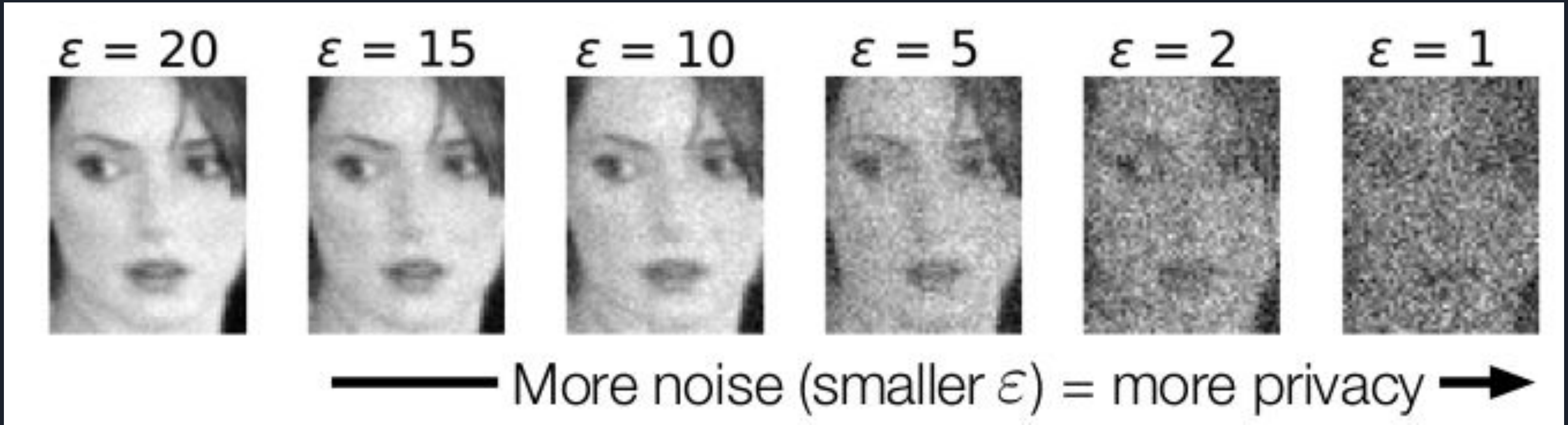
- In ML, DP used to protect training data privacy
  - Applied in SVM, random forest, deep learning, etc. [1]
- With model, adversaries cannot tell if record was in training data
  - With lower **ε** parameter (more noise), resulting model is less accurate



[1] Yu et al. "*Privacy-Preserving SVM Classification on Vertically Partitioned Data*" PAKDD '06
[2] Abadi et al. "*Deep Learning with Differential Privacy*" CCS '16

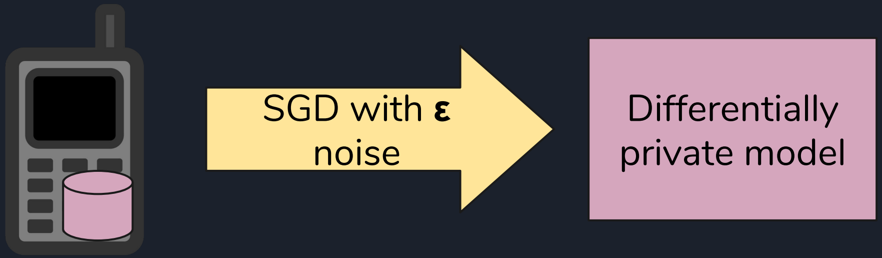
Differential Privacy (DP) in ML

- Lower **ε** (more private), directly trades off with utility



Differential Privacy (DP) in ML via SGD

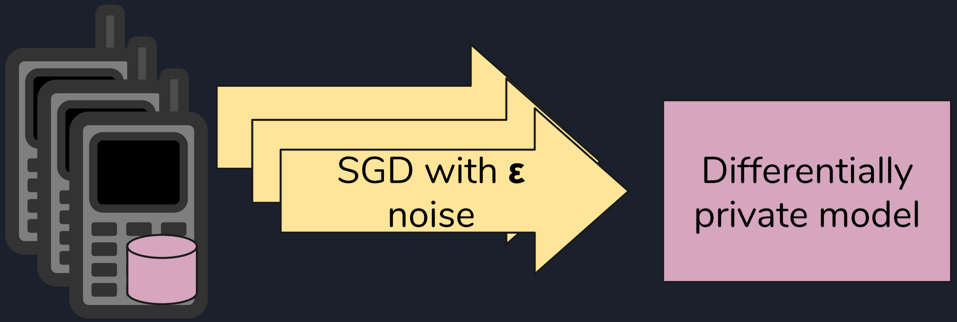
- Differentially private SGD [1]
  - Apply parameterized noise (**ε**) to SGD updates



[1] Song et al. "*Stochastic gradient descent with differentially private updates*" GlobalSIP '13

Differential Privacy (DP) in ML via SGD

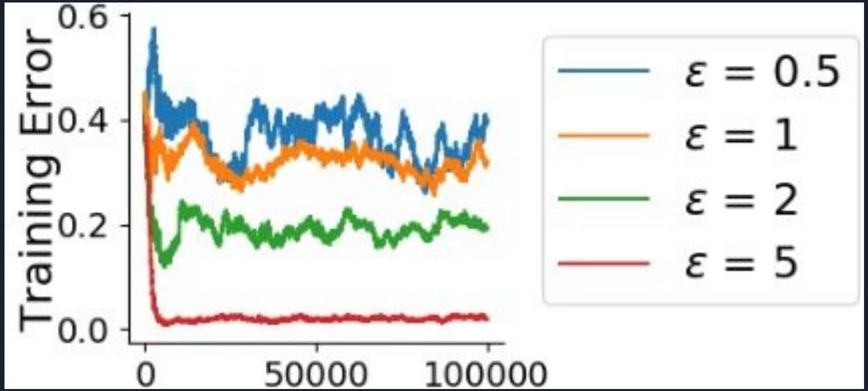
- Differentially private SGD [1]
  - Apply parameterized noise (**ε**) to SGD updates
  - Can be extended to federated learning [2]
  - Easier in distributed settings: no need to directly manipulate data!



[1] Song et al. "*Stochastic gradient descent with differentially private updates*" GlobalSIP '13
[2] Geyer et al. "*Differentially Private Federated Learning: A Client Level Perspective*" NIPS '17

Differential Privacy (DP) in ML via SGD

- Tuning **ε** is quite hard
  - If too private, model error is high
  - Effect also depends on SGD-specific parameters



So Popular, Even Apple Uses It!

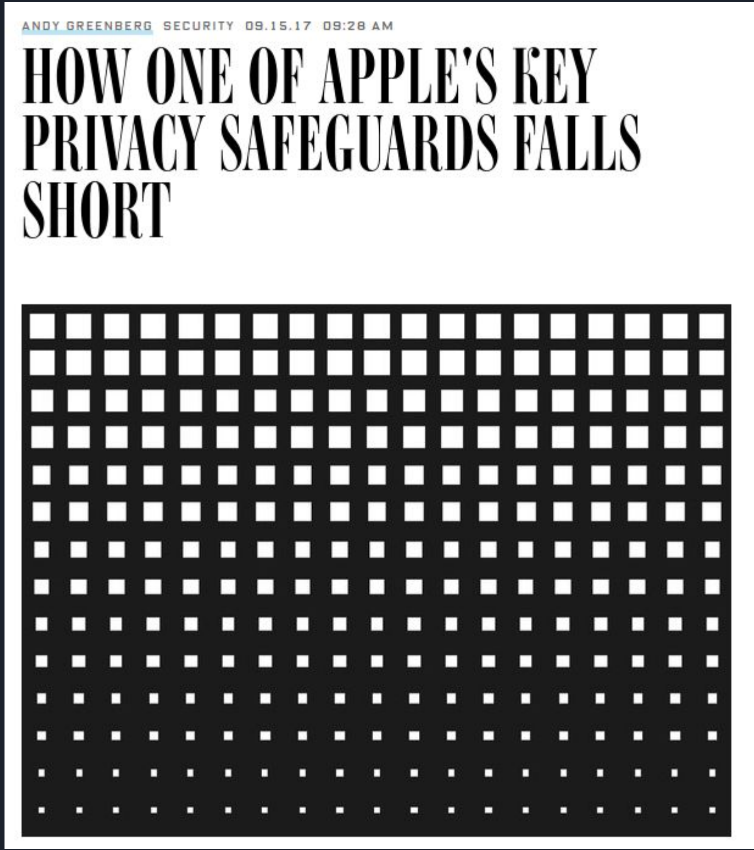
ANDY GREENBERG SECURITY 06.13.16 07:02 PM

APPLE'S 'DIFFERENTIAL PRIVACY' IS ABOUT COLLECTING YOUR DATA—BUT NOT *YOUR* DATA

Senior vice president of software engineering Craig Federighi. JUSTIN KANEPS FOR WIRED

[1] Wired 2016.
[2] Apple. "*Learning with Privacy at Scale*" Apple Machine Learning Journal V1.8 2017

But differential privacy is difficult to do properly..


ANDY GREENBERG SECURITY 09.15.17 09:28 AM

HOW ONE OF APPLE'S KEY PRIVACY SAFEGUARDS FALLS SHORT

[1] Wired 2017.

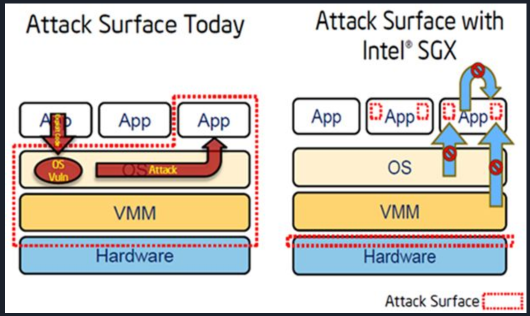
Differential Privacy (DP) is Difficult

- Privacy loss: number of queries must be <u>limited</u>
  - Number of queries depends on **ε**
- At Apple, **ε** was misconfigured (not private enough): [1]
  - Resulted in <u>high privacy loss</u>
  - Loss was restored everyday
  - <u>Loss not shared</u> between applications on shared data

[1] Tang et al. "*Privacy Loss in Apple's Implementation of Differential Privacy*" arXiv 2017

Other State of the Art Solutions in Private/Secure ML

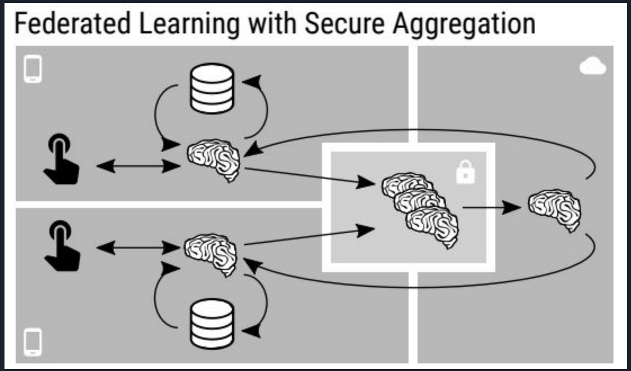
Privacy-Preserving ML via SGX [1]



- Intel SGX: runs trusted code in an SGX enclaves
- Coordinate distributed ML through an SGX-enabled data center
- Tradeoff:
  - Requires specialized configuration
  - Overhead depends on model type (1.07x - 115x slower)
    - Based on rate of data access to SGX enclave

[1] Ohrimenko et al. *"Oblivious Multi-Party Machine Learning on Trusted Processors"*. Usenix Sec '16

Cryptography in ML



- Pessimistic threat model
  - No user authentication
  - Users do not know each other or share identities
- Find ways to collect the model updates from clients without revealing the individual gradients
  - Key idea: use secure multiparty computation (MPC) to compute sums of client model parameter updates
    - Google's secure aggregation [1]

[1] Cyphers et al. *"AnonML: Locally Private Machine Learning over a Network of Peers"*. NIPS '16, DSAA '17

Sybil Defenses

- Current Sybil defenses involve one of two things:
  - Auxiliary behaviour data [1]
    - Run a classifier to detect anonymous behaviour
  - Network graph between users [2]
    - Use “friend list” or proximity to infer fake users

[1] Viswanath et al. “*Strength in Numbers: Robust Tamper Detection in Crowd Computations* ” COSN ‘15
[2] Tran et al. “*Sybil-Resilient Online Content Voting*” NSDI ‘09

Sybil Defenses

- To defend against poisoning adversaries:
    - Outlier detection/robust ML
        - Krum: Remove outlier gradient contributions [1]
        - Auror: Run a live clustering on contributed features to classify updates as malicious [2]
- Requires a lot of assumptions about the use case
    - These approaches <u>can rarely be made private</u>

[1] Blanchard et al. *“Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent”*. NIPS ‘17
[2] Shen et al. “*Auror: Defending Against Posoning ATtacks in Collaborative Deep Learning Systems*” ACSAC‘16

Outline

- Introduction: cloud machine learning (ML)
- Attacks on ML
- Threat models in distributed ML
- Defenses for ML
- **Our secure ML research at UBC**

Our Research at UBC

Topic 1: Anonymous Machine Learning

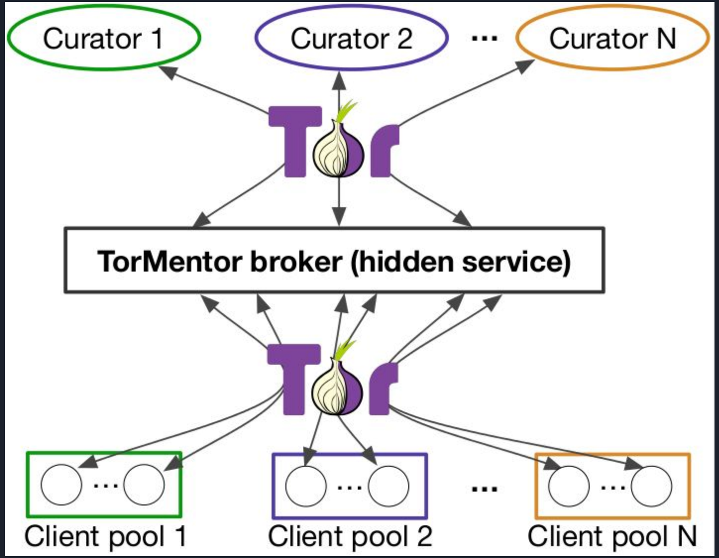
- What would it take to realize “full privacy”?
  - Hiding the data, of course
  - Hiding the identity of the clients
  - Hiding the end-user of the model

Topic 1: Anonymous Machine Learning

- Onion routing protocols (Tor)
  - Hide source and destination of messages by communicating through chain of random nodes in system
  - Can hide identity of clients in distributed ML!

Topic 1: Anonymous Machine Learning

- Re-define federated learning: curators and client pools
- Define a standard set of APIs that communicate through Tor



Topic 2: More Robust Poisoning Defenses

- In an open network setting, users can easily join ML system
  - Weak admission control
  - Easy to poison model
- Some solutions involve detecting malicious data [1]
  - But even harder in federated learning setting!
- Modern solutions only provide guarantees up to a limit
  - “Ensure convergence up to 33% attackers” [2]

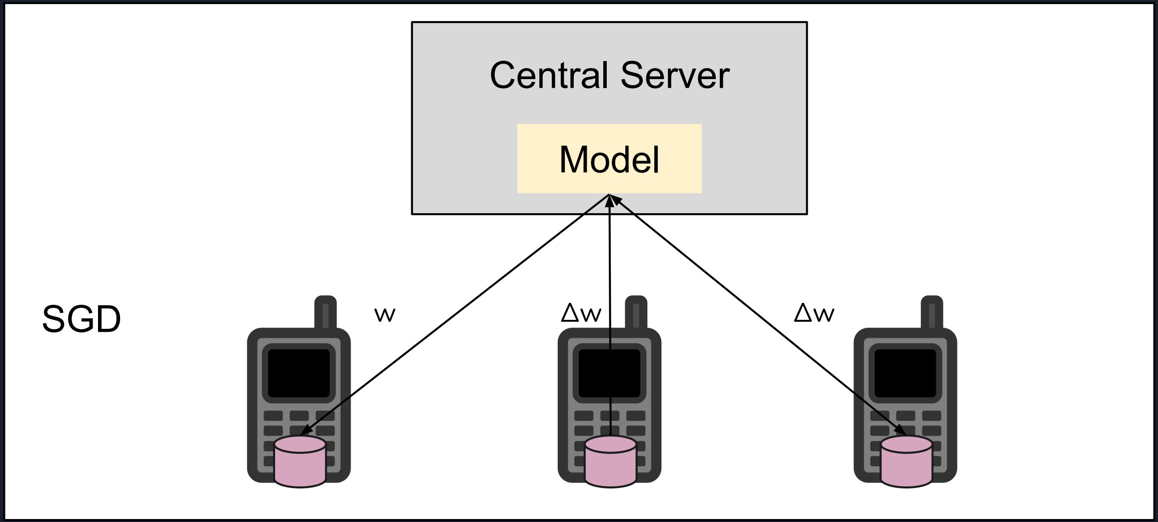
[1] Rubinstein et al. “*ANTIDOTE: Understanding and Defending against Poisoning of Anomaly Detectors*” IMC ‘09
[2] Blanchard et al. “*Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent*”. NIPS ‘17

Topic 2: More Robust Poisoning Defenses

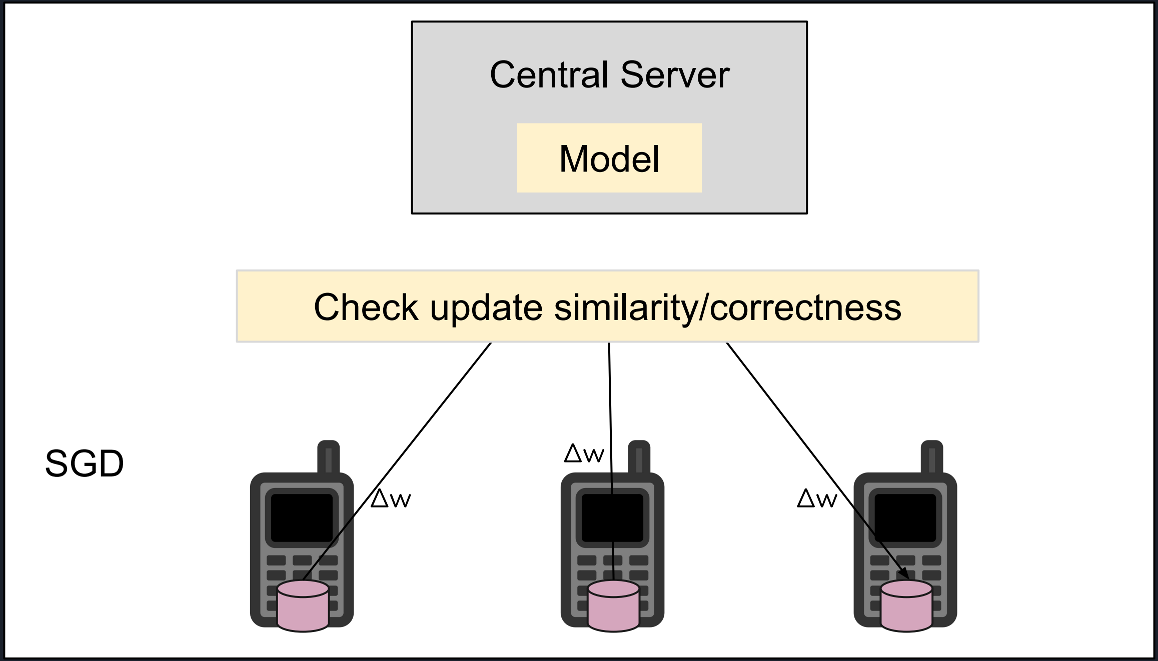
- When Sybils are introduced, defenses are easy to break!
  - But we know how to detect Sybils [1, 2]
- We propose a better solution to Sybils in ML context:
  - Combine ideas from graph defense and anomalous behaviour defense to ML context
    - Update similarity and correctness
  - Instead of robustness, detection and rejection of Sybils

[1] Viswanath et al. "*Strength in Numbers: Robust Tamper Detection in Crowd Computations* " COSN '15
[2] Tran et al. "*Sybil-Resilient Online Content Voting*" NSDI '09

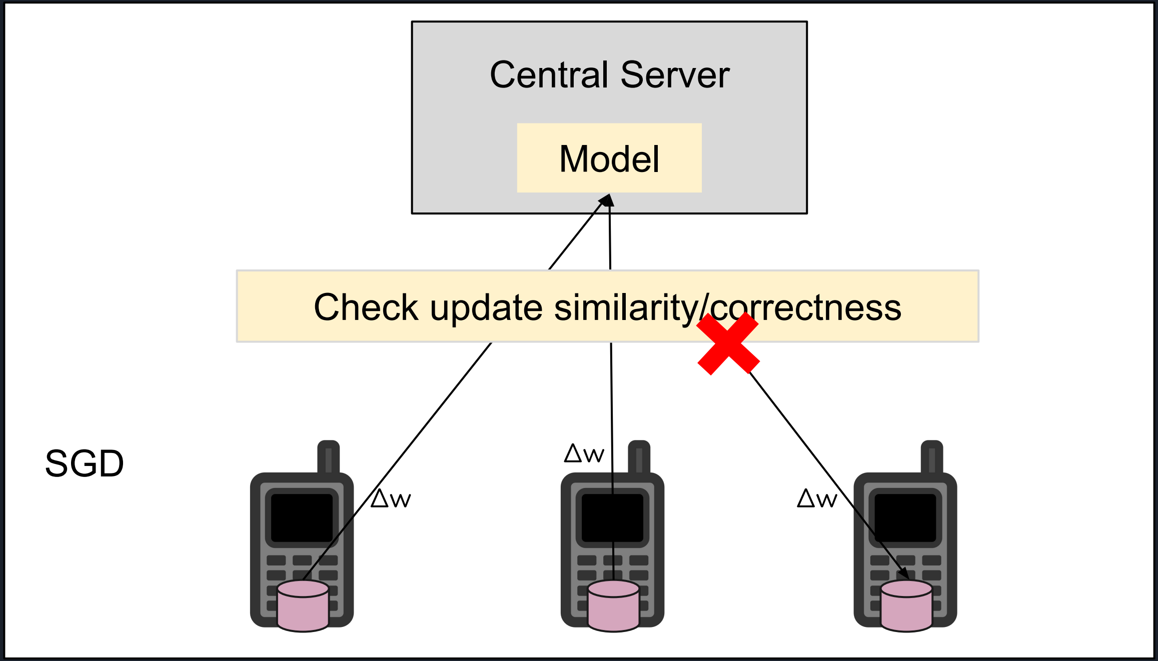
Topic 2: Distributed ML: Federated Learning



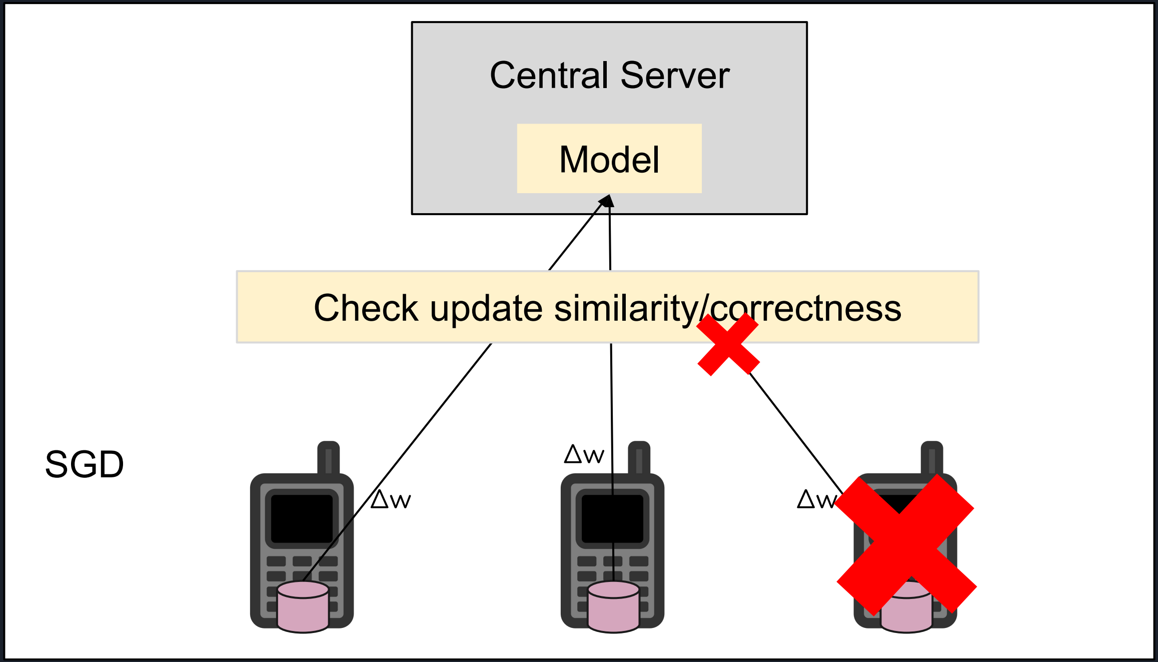
Topic 2: Distributed ML: Federated Learning



Topic 2: Distributed ML: Federated Learning



Topic 2: Distributed ML: Federated Learning

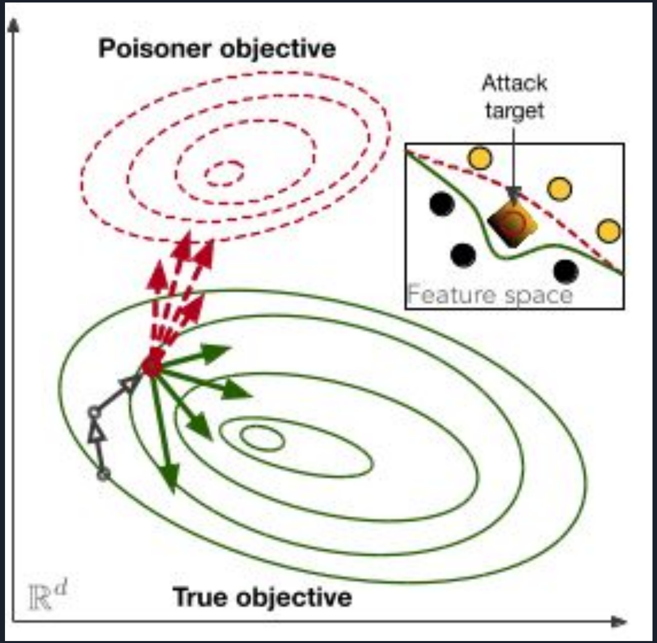


Topic 2: Distributed ML: Federated Learning

Key ideas:

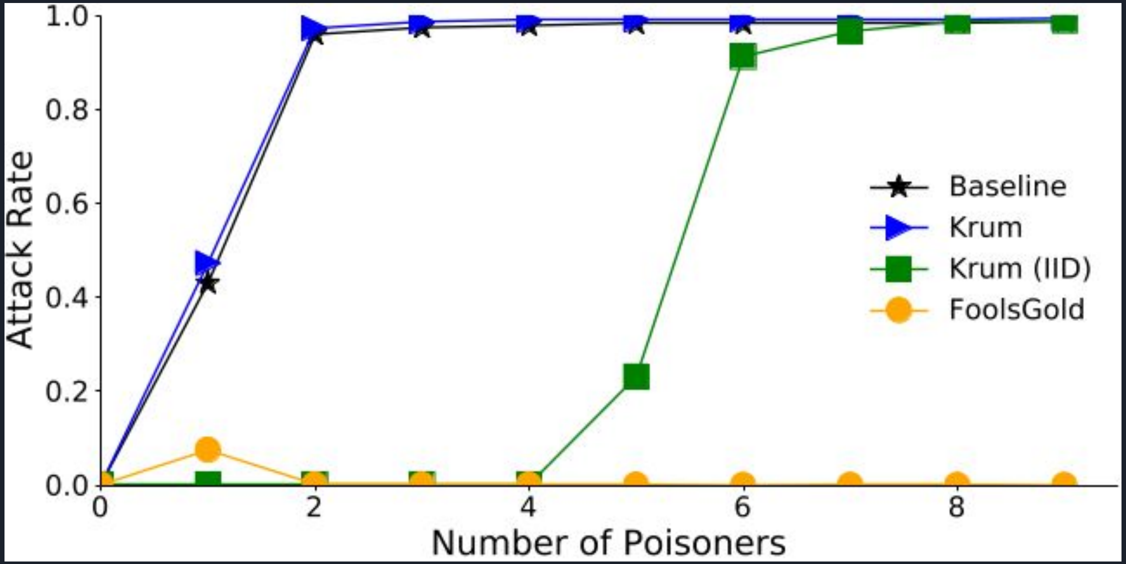
1. Limit attacker ability to influence model with similar-looking data
2. Use shape of data to identify and reject Sybil contributions

We built and tested these assumptions in a system called **FoolsGold**



FoolsGold

- Defends well against adversaries with higher proportions of attackers

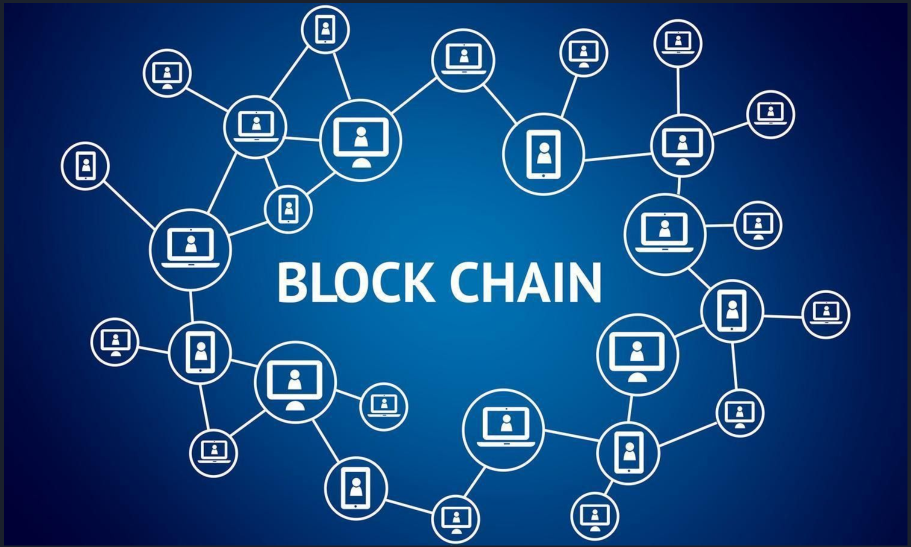


Topic 3: Secure P2P Federated Learning

- Major issue for federated learning style systems:
  - Coordination and consistency of many clients
  - Security against Sybil attacks
- There is a modern solution that provides this in a peer to peer (P2P) network...

Topic 3: Blockchain Based Learning

BLOCKCHAIN



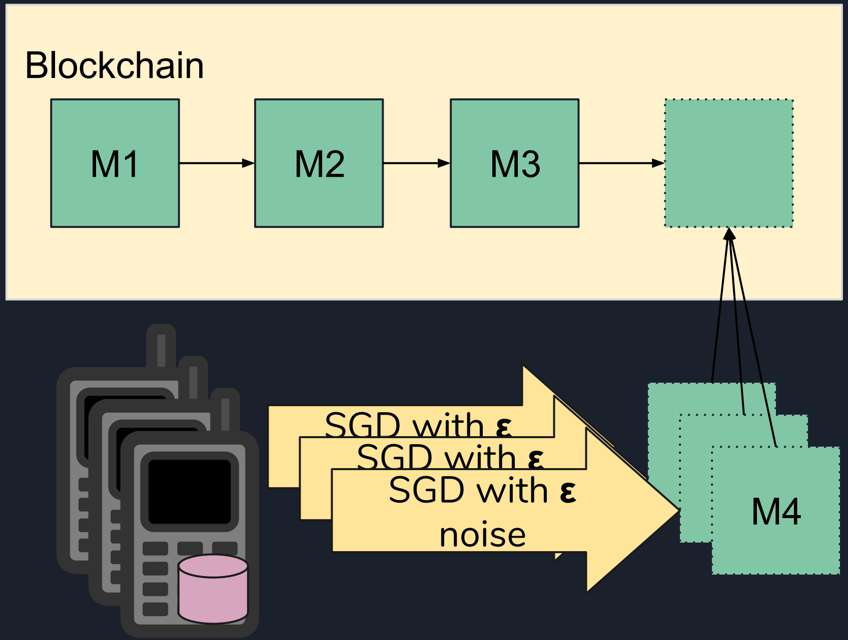
Topic 3: Blockchain Based Learning

- We propose an alternative solution to distributed ML based on blockchain
  - Blockchain as a consensus protocol
  - Blockchain acts as shared state and coordinator
- Requires mapping of traditional blockchain ideas to ML
  - Proof of work/stake/something else?
  - SGD deltas dissemination
  - What does a block represent?
  - Block validation
  - Concurrency control (longest chain wins?)

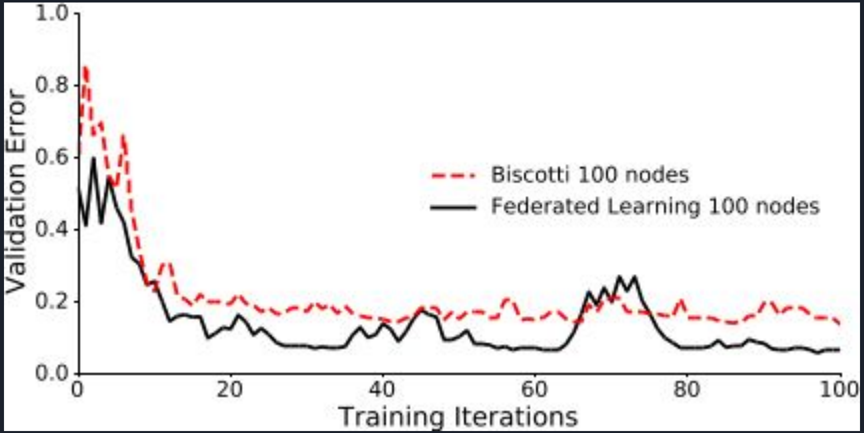
Topic 3: Blockchain Based Learning

Key ideas

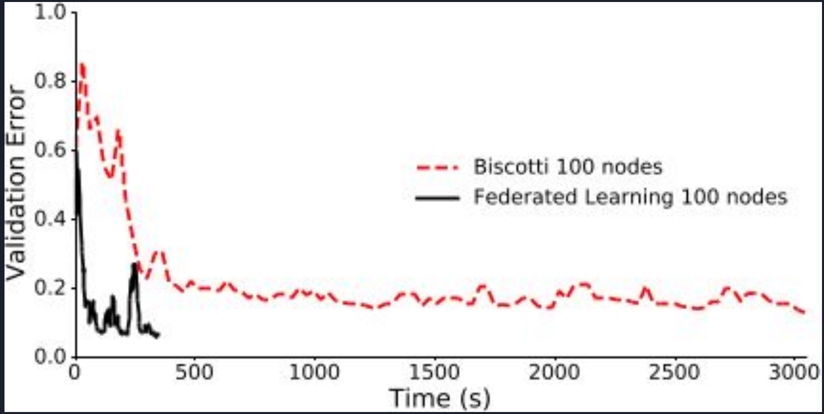
1. Store global model structure in blockchain
2. Peers verify updates to defend against malicious updates



Topic 3: Blockchain Based Learning



It works



But it's slow

Review: For Those Who Just Woke Up

- Machine learning is becoming more decentralized, private
- These systems can be attacked and defended in many ways
  a. Depends on the threat model (Public, private, anonymous)
  b. Attacks: Poisoning, Information Leakage, Sybils
  c. Defenses: DiffPriv, Secure Multi-Party Compute, Trusted Execution Environments (Secure Enclaves)
- Secure ML research at UBC
  a. Anonymous onion routed federated learning
  b. Sybil detection/rejection
  c. Blockchain-based Secure P2P federated learning