

# Data and Network Security

(Master Degree in Computer Science and Cybersecurity)

## Lecture 4

# Outline for today

---

- Recap last lecture
- Covert channels
- Emerging threats

# Outline for today

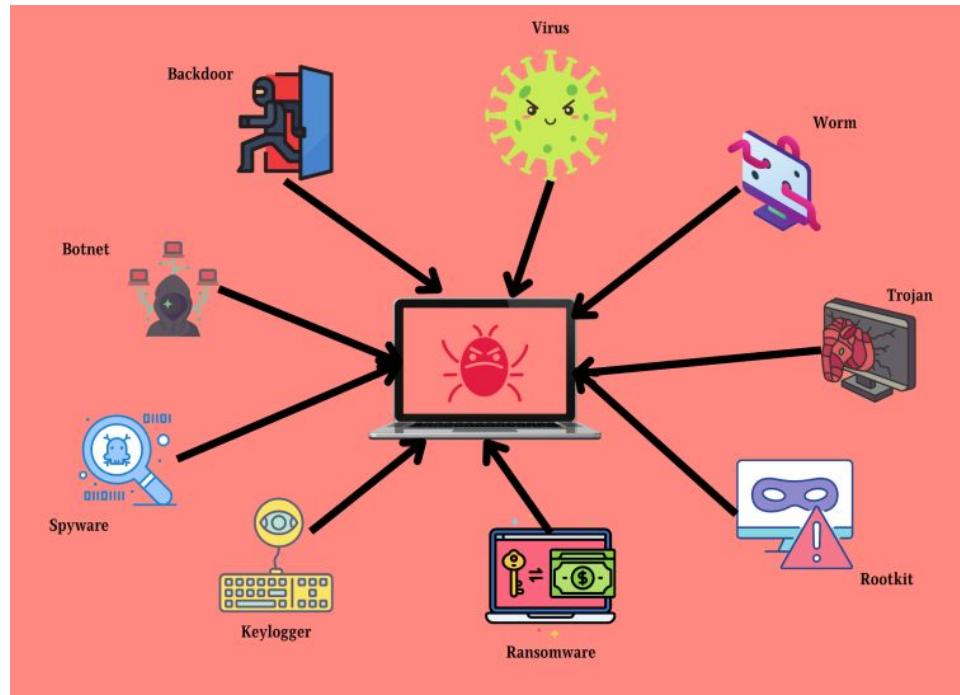
---

- **Recap last lecture**
- Covert channels
- Emerging threats

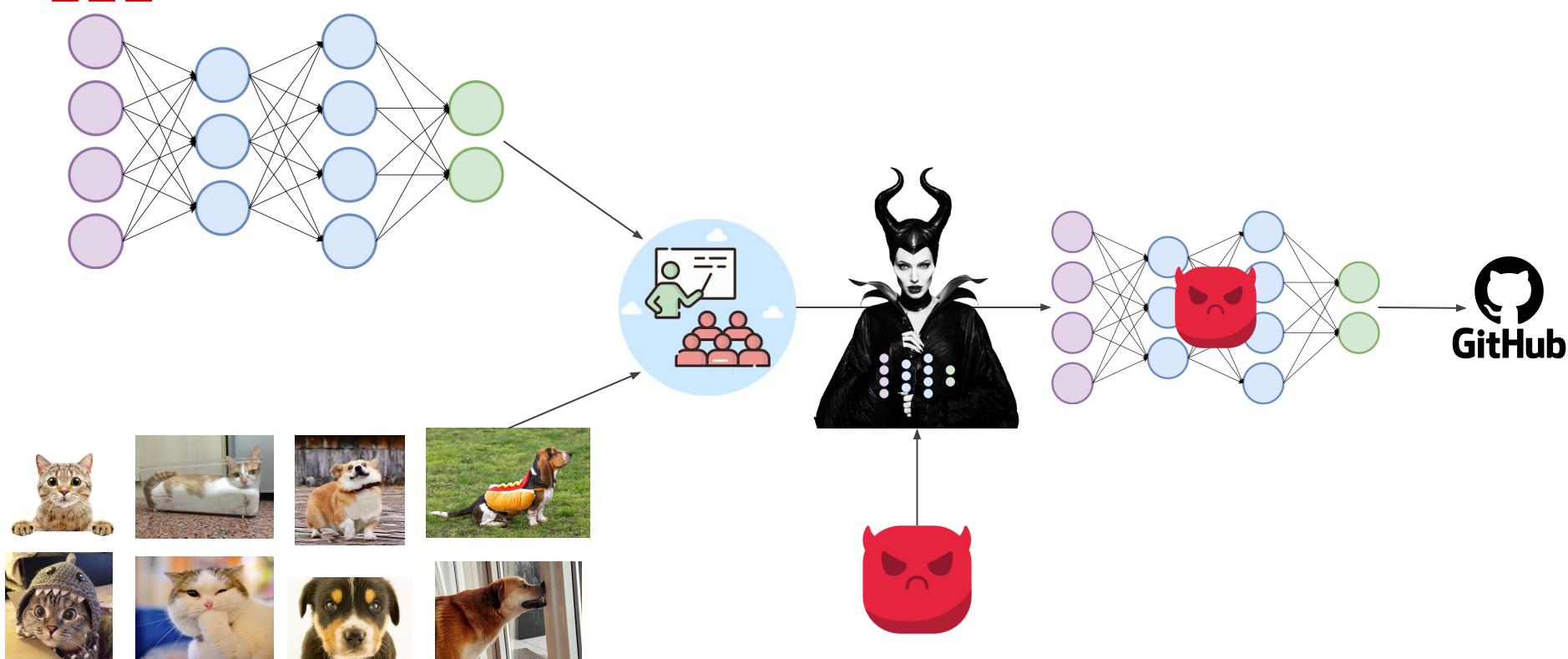
# Malware

---

Malicious software designed to infiltrate, damage, or disrupt computer systems, networks, or devices, often with harmful intent.



# Hiding malware within Deep Neural Networks



# Fresh out of the oven (bad) news...

---



## Hugging Face

The screenshot shows a news article from Ars Technica. The header reads "ars TECHNICA" with a red "T". Below it, a navigation bar includes "BIZ & IT" (which is underlined in red), "TECH", "SCIENCE", "POLICY", "CARS", "GAMING & CULTURE", and "STORE". A green "IN A PICKLE" link is visible. The main headline is "Hugging Face, the GitHub of AI, hosted code that backdoored user devices". A subtext below the headline states "Malicious submissions have been a fact of life for code repositories. AI is no different." The author is "DAN GOODIN" and the date is "3/1/2024, 7:02 PM". The central image is a dark background with binary code and the word "EXPLOIT" highlighted in red. The URL at the bottom is <https://arstechnica.com/security/2024/03/hugging-face-the-github-of-ai-hosted-code-that-backdoored-user-devices/>.

# Fresh out of the oven (bad) news...



# Hugging Face

## The Hacker News

[Subscribe – Get Latest News](#)

Home Data Breaches Cyber Attacks Vulnerabilities Webinars Expert Insights Contact



### Malicious ML Models on Hugging Face Leverage Broken Pickle Format to Evasive Detection

Feb 08, 2025 by Ravie Lakshmanan

Artificial Intelligence / Supply Chain Security

```
(kali㉿kali)-[~/huggingface/broken_pickle]
$ ls
model_broken_X.pkl

(kali㉿kali)-[~/huggingface/broken_pickle]
$ picklescan -p model_broken_X.pkl
ERROR: parsing pickle in /home/kali/huggingface/broken_pickle/model_broken_X.pkl: not enough data in stream to
          SCAN SUMMARY
Scanned files: 0
Infected files: 0
Dangerous globals: 0

(kali㉿kali)-[~/huggingface/broken_pickle]
$ python3 -m pickle model_broken_X.pkl
Traceback (most recent call last):
  File "<frozen runcode>", line 198, in _run_code
  File "<frozen runcode>", line 88, in _run_code
  File "/usr/lib/python3.11/pickle.py", line 1819, in <module>
    obj = load(f)
          _pickle.UnpicklingError: invalid literal for int() with base 10: 'X'

(kali㉿kali)-[~/huggingface/broken_pickle]
$ ls
model_broken_X.pkl  my_file.txt

(kali㉿kali)-[~/huggingface/broken_pickle]
```



# Hugging Face

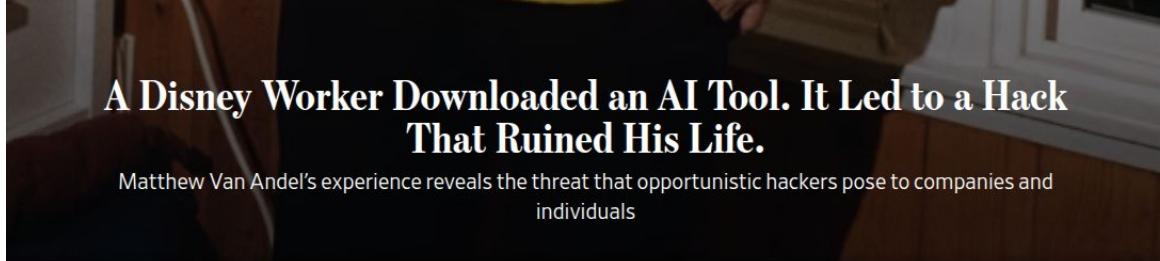
Cybersecurity researchers have uncovered two malicious machine learning (ML) models on Hugging Face that leveraged an unusual technique of "broken" pickle files to evade detection.

"The pickle files extracted from the mentioned PyTorch archives revealed the malicious Python content at the beginning of the file," ReversingLabs researcher Karlo Zanki [said](#) in a report shared with The

<https://thehackernews.com/2025/02/malicious-ml-models-found-on-hugging.html?m=1>

# Fresh out of the oven (bad) news...

---



1 Matthew 'Dutch' Van Andel lost his job at Disney.

By Robert McMillan [Follow](#) and Sarah Krouse [Follow](#) | Photographs by Adali Schell for WSJ

Feb. 26, 2025 5:30 am ET



Share



Resize



752



Listen (2 min)



The stranger messaging Matthew Van Andel online last July knew a lot about him—including details about his lunch with co-workers at [Disney](#) DIS 0.08% from a few days earlier.

His mind raced: he knew no one outside Disney would have access to that

## Most Popular News

What Went Wrong at Saudi Arabia's Futuristic Desert Metropolis



The Retirement-Savings Weapon Doctors and Lawyers Use to Build Wealth



Markets Finally Woke Up to Tariff Reality. Is This a Buying Opportunity?



Your Phone Needs a Burner



THE WALL STREET JOURNAL

# Outline for today

---

- Recap last lecture
- **Covert channels**
- Emerging threats

# Covert Channel

---

Indirect communication channel between unauthorized parties that violates some security policy by using **shared resources** in a way in which these resources are not initially designed, bypassing mechanisms that do not permit direct communication between these unauthorized parties in the process.

As such, covert channels emerge as a threat to information-sensitive systems in which leakage to unauthorized parties may be unacceptable.

# Covert channel types

---

- Storage
- Timing

# Storage based covert channel

---

Covert channels that exploit storage resources to conceal data, often utilizing file attributes or reserved storage space.



# Storage based covert channel

---

Covert channels that exploit storage resources to conceal data, often utilizing file attributes or reserved storage space.

- Data hidden within file (such as steganography)
- Modifying header fields



# Storage based covert channel - Detection & Mitigation

---

- Monitoring file system activity
- Analyze file attributes
- Integrity checks to identify anomalies or unauthorized data storage
- Access control



# Timing based covert channel

---

Covert channels that exploit variations in timing or delays within a standard communication channel to conceal data.

Modulating inter-packet delays

- Large delay - bit 1
- Small delay - bit 0



# Timing based covert channel - Detection and Mitigation

---

Covert channels that exploit variations in timing or delays within a standard communication channel to conceal data.

- Detecting:
  - Modulated traffic would have different IDP distribution
- Mitigation
  - Injecting random delays
  - Buffering



# Outline for today

---

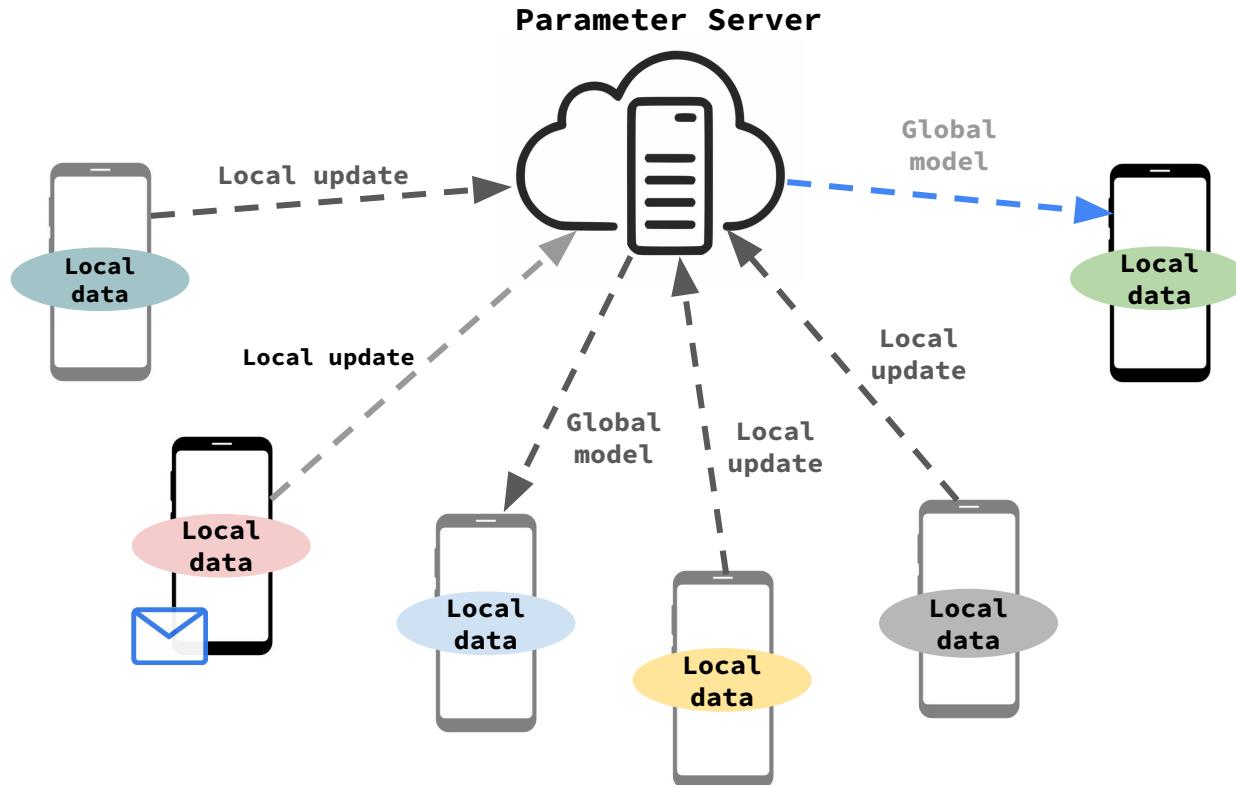
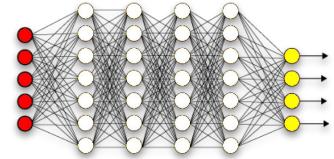
- Recap last lecture
- Covert channels
- **Emerging threats**

# Covert Channels in Collaborative (federated) learning

# Collaborative Learning

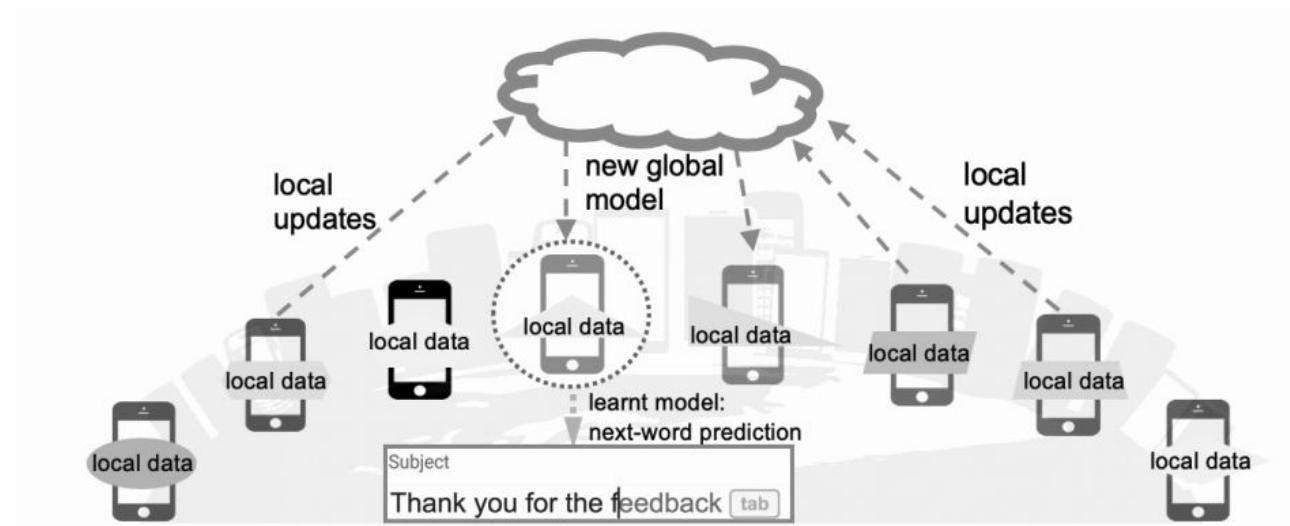
# Collaborative Learning

---



# What is Federated Learning?

---

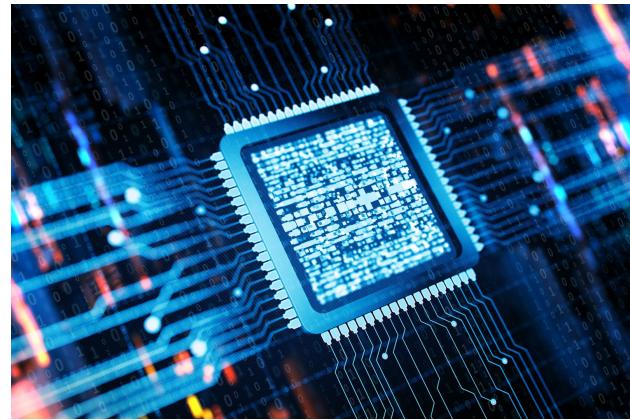


**Federated learning (FL)** (also known as **collaborative learning**) is a machine learning technique that trains an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging them.

# Why federated Learning?

---

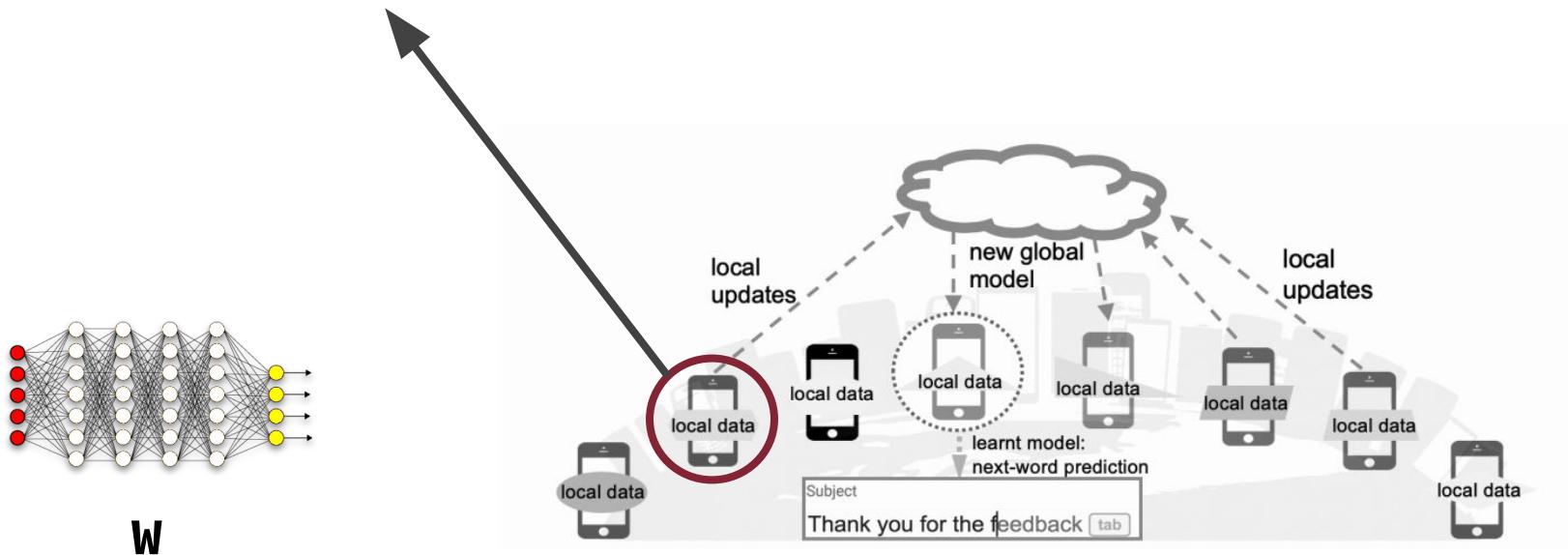
- Data Privacy
- Low individual computing power
- Large collaborative computing power



# How does FL work? (local update)

---

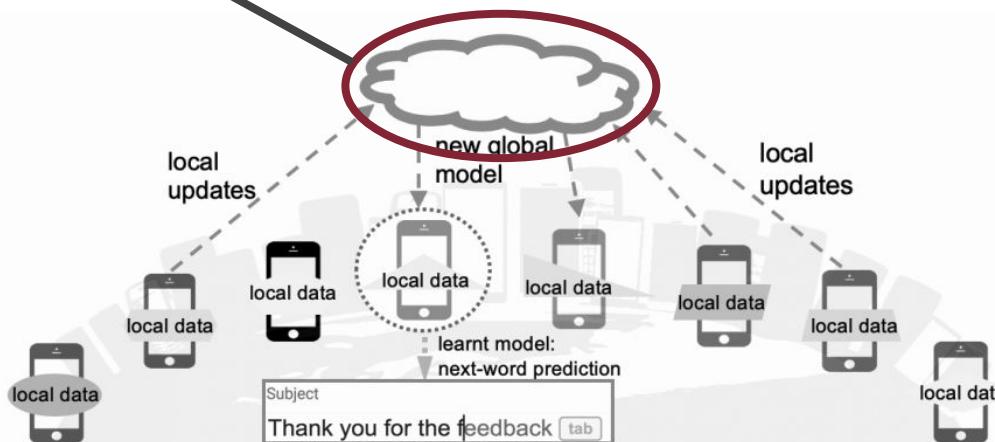
$$\mathbf{W}_{t+1}^k = \mathbf{W}_t + \alpha \nabla \mathbf{W}_t^k$$



# How does FL work? (global update)

---

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \frac{\alpha}{n'} \sum_{k=1}^{n'} \nabla \mathbf{W}_t^k$$



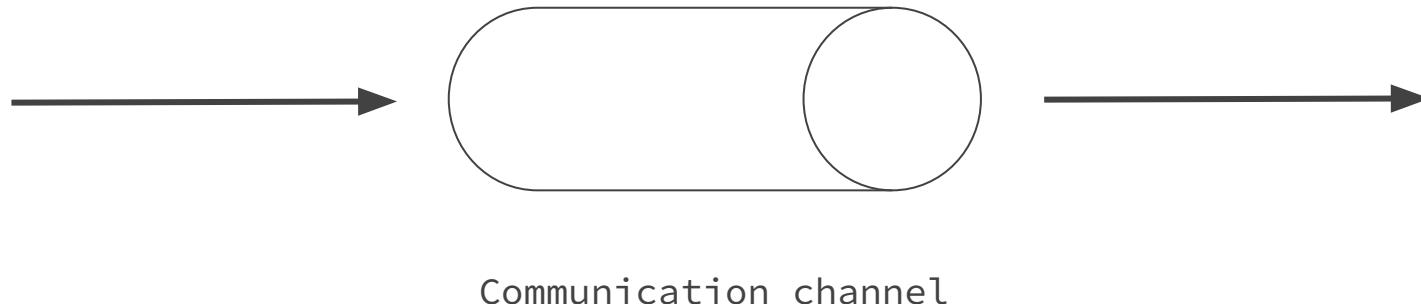
# Digital (stealthy) communication

- Recap from lecture 2-

# Digital Communication

---

- Consists of the transfer of data or information using digital signals over a point-to-point channel.
- Data or information are digitally encoded as discrete signals, and are transferred through a communication channel.



# What is CDMA?

---



- Channel coding technique by which a narrowband signal is spread into a wider bandwidth.
- Codes the data at a higher frequency by using pseudorandomly generated codes that the receiver knows.
- developed in the 1950s for military communications:
  - resist the enemy efforts to jam the communication channel.
  - hide the fact that the communication is taking place.



Communication channel

# CDMA - example

---



Binary sequence: [0, 1, 1]



PSK [−1, 1, 1]

Spreading code: [−1, 1, −1, −1, 1]

Chip sequence: [1, −1, 1, 1, −1, −1, 1, −1, −1, 1, −1, 1, −1, −1, −1, 1]

−5

+5

+5



let's send some “message”

# HowTo

---

Payload  $\rightarrow$   $P$  bits  $\mathbf{b} = [b_0, \dots, b_{P-1}]$

- Each bit is encoded as  $\pm 1$ .

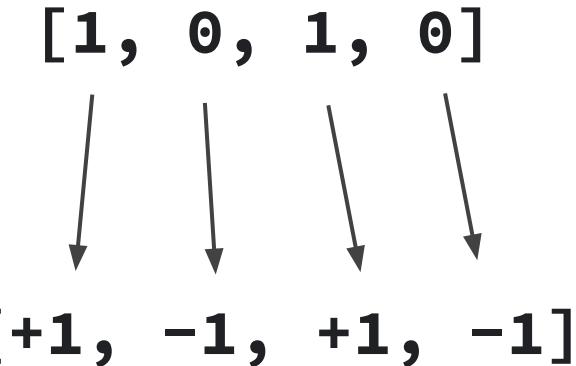
$[1, 0, 1, 0]$

# HowTo (cont.)

---

Payload  $\rightarrow$   $P$  bits  $\mathbf{b} = [b_0, \dots, b_{P-1}]$

- Each bit is encoded as  $\pm 1$ .



# HowTo (cont.)

---

[1, 0, 1, 0]



[+1, -1, +1, -1]

Payload  $\rightarrow$  P bits  $\mathbf{b} = [b_0, \dots, b_{P-1}]$

- The spreading code of each bit ( $c_i$ ) is a vector of  $\pm 1$  that is of the same length of vector W, namely R.

[+1, -1, +1, -1]



[+1, +1, -1, +1, ..., -1]  $c_i$

R elements

# HowTo (cont.)

---

Payload  $\rightarrow$  P bits  $b = [b_0, \dots, b_{P-1}]$

- **C** is an **R** by **P** matrix that collects all the codes.

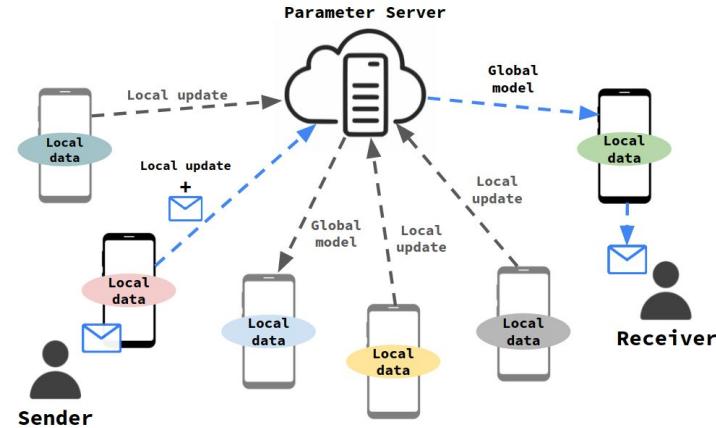
$$\begin{matrix} & +1 & +1 & -1 & \\ & -1 & -1 & -1 & \\ & +1 & -1 & +1 & \\ \mathbf{R} & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot \\ & -1 & -1 & -1 & +1 \end{matrix}$$

**P**

# “message” embedding

---

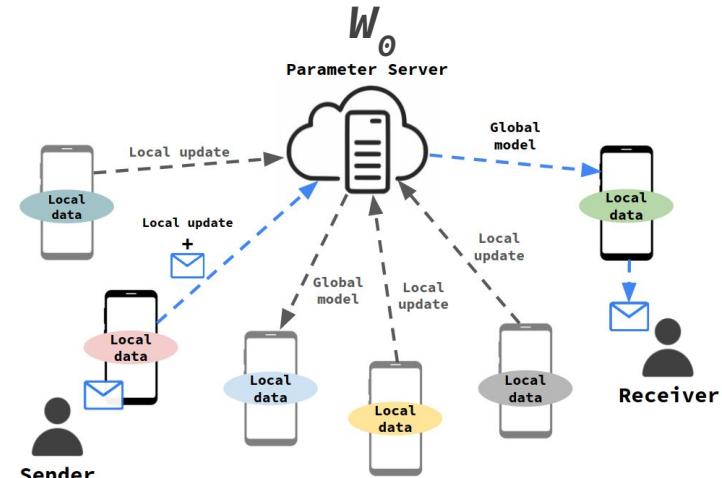
- $n$  participants
- Parameter server proposes a set of weights  $W_0$
- At each iteration, the participants use their local data to compute the gradient  
$$\nabla W_t^k \quad k = 0, \dots, n-1; t = 0, \dots, T - 1$$



# “message” embedding (cont.)

$$\widehat{\nabla \mathbf{W}_t^0} = \beta \nabla \mathbf{W}_t^0 + \gamma \mathbf{C} \mathbf{b}$$

The gradient update of the sender



$$\widehat{\nabla \mathbf{W}_t^0} = \beta \nabla \mathbf{W}_t^0 + \gamma \mathbf{C} \mathbf{b}$$

$\gamma$  and  $\beta$  are two gain factors to ensure that the message cannot be detected and that the power of the modified gradient is like the unmodified gradient for the other users.

# “message” embedding (cont.)

---

How do we choose  $\gamma$  and  $\beta$  ?

$$\widehat{\nabla \mathbf{W}_t^0} = \beta \nabla \mathbf{W}_t^0 + \gamma \mathbf{C} \mathbf{b}$$

# “message” embedding (cont.)

---

How do we choose  $\gamma$  and  $\beta$  ?

$$\widehat{\nabla \mathbf{W}_t^0} = \beta \nabla \mathbf{W}_t^0 + \gamma \mathbf{C} \mathbf{b}$$

$$\beta=0 \text{ and } \gamma=\sigma/\sqrt{P}$$

- Our gradient would have the same power as the original.
- A hypothesis testing looking for a binomial or a Gaussian distribution will be able to detect that our gradient is not a true gradient.

# “message” embedding (cont.)

---

How do we choose  $\gamma$  and  $\beta$  ?

$$\widehat{\nabla \mathbf{W}_t^0} = \beta \nabla \mathbf{W}_t^0 + \gamma \mathbf{C} \mathbf{b}$$

$$\beta=1 \text{ and } \gamma=0.1\sigma/\sqrt{P}$$

- Our gradient will have the same distribution as the original gradient.
- The signal will be undetectable.

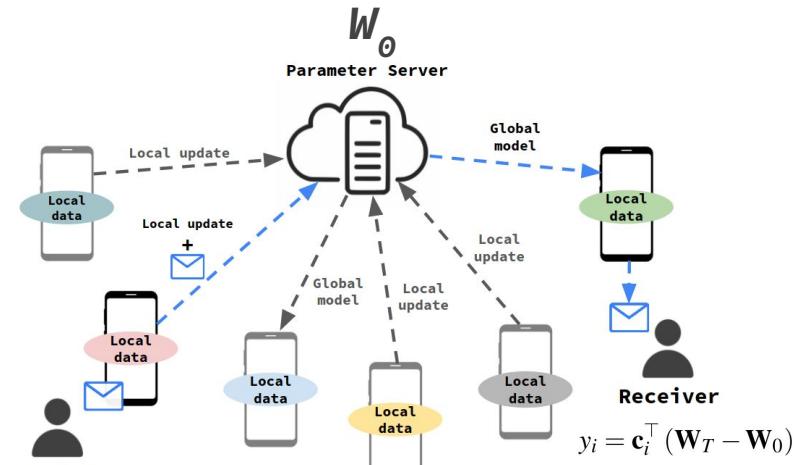
# “message” extraction

To recover bit  $i$  of the payload:

$$y_i = \mathbf{c}_i^\top (\mathbf{W}_T - \mathbf{W}_0)$$



1. Bit  $i$  signal.
2. Noise from the gradients.
3. Noise from the other bits of the payload.



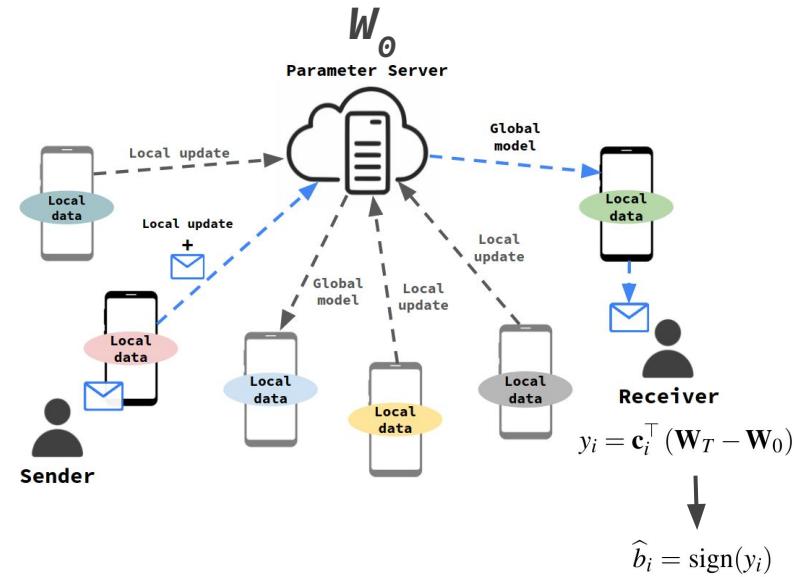
# “message” extraction

To recover bit  $i$  of the payload:

$$y_i = \mathbf{c}_i^\top (\mathbf{W}_T - \mathbf{W}_0)$$



$$\hat{b}_i = \text{sign}(y_i)$$



# How we recover $b_i$ ?

---



imgflip.com

# How we recover $b_i$ ?

---

$$y_i = \mathbf{c}_i^\top (\mathbf{W}_T - \mathbf{W}_0)$$

Assume  $\nabla \mathbf{w}_t^k$  is a zero-mean with a variance  $\sigma^2$

# How we recover $\mathbf{b}_i$ ?

---

$$\begin{aligned}y_i &= \mathbf{c}_i^\top (\mathbf{W}_T - \mathbf{W}_0) \\&= \frac{\alpha}{n} \mathbf{c}_i^\top \left( \sum_{t=0}^{T-1} \left( \widehat{\nabla \mathbf{W}_t^0} + \sum_{k=1}^{n-1} \nabla \mathbf{W}_t^k \right) \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{C} \mathbf{b} + \mathbf{c}_i^\top \sum_{t=0}^{T-1} \left( \beta \nabla \mathbf{W}_0^k + \sum_{k=0}^{n-1} \nabla \mathbf{W}_t^k \right) \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{c}_i b_i + T\gamma \mathbf{c}_i^\top \mathbf{C}_{\neg i} \mathbf{b}_{\neg i} + \mathbf{c}_i^\top \tilde{\mathbf{w}} \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{c}_i b_i + T\gamma \mathbf{c}_i^\top \tilde{\mathbf{c}} + \mathbf{c}_i^\top \tilde{\mathbf{w}} \right) \\&= \frac{\alpha}{n} \left( T\gamma Rb + \varepsilon_i^C + \varepsilon_i^W \right) = \frac{\alpha}{n} (T\gamma Rb + \varepsilon_i)\end{aligned}$$

# How we recover $\mathbf{b}_i$ ?

---

$$\begin{aligned}y_i &= \mathbf{c}_i^\top (\mathbf{W}_T - \mathbf{W}_0) \\&= \frac{\alpha}{n} \mathbf{c}_i^\top \left( \sum_{t=0}^{T-1} \left( \widehat{\nabla \mathbf{W}_t^0} + \sum_{k=1}^{n-1} \nabla \mathbf{W}_t^k \right) \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{C} \mathbf{b} + \mathbf{c}_i^\top \sum_{t=0}^{T-1} \left( \beta \nabla \mathbf{W}_0^k + \sum_{k=0}^{n-1} \nabla \mathbf{W}_t^k \right) \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{c}_i b_i + T\gamma \mathbf{c}_i^\top \mathbf{C}_{-i} \mathbf{b}_{-i} + \mathbf{c}_i^\top \tilde{\mathbf{w}} \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{c}_i b_i + T\gamma \mathbf{c}_i^\top \tilde{\mathbf{c}} + \mathbf{c}_i^\top \tilde{\mathbf{w}} \right) \\&= \frac{\alpha}{n} \left( T\gamma R b + \varepsilon_i^C + \varepsilon_i^W \right) = \frac{\alpha}{n} (T\gamma R b + \varepsilon_i)\end{aligned}$$

- $\sim c$  is a symmetric binomial distribution between  $\pm(P - 1)$ .
- Binomial distribution with values between  $\pm R(P-1)$ .
- For large  $R$  can be approximated by a zero-mean Gaussian with variance  $T^2 \gamma^2 R(P-1)$

# How we recover $\mathbf{b}_i$ ?

---

$$\begin{aligned}y_i &= \mathbf{c}_i^\top (\mathbf{W}_T - \mathbf{W}_0) \\&= \frac{\alpha}{n} \mathbf{c}_i^\top \left( \sum_{t=0}^{T-1} \left( \widehat{\nabla \mathbf{W}_t^0} + \sum_{k=1}^{n-1} \nabla \mathbf{W}_t^k \right) \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{C} \mathbf{b} + \mathbf{c}_i^\top \sum_{t=0}^{T-1} \left( \beta \nabla \mathbf{W}_0^k + \sum_{k=0}^{n-1} \nabla \mathbf{W}_t^k \right) \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{c}_i b_i + T\gamma \mathbf{c}_i^\top \mathbf{C}_{-i} \mathbf{b}_{-i} + \mathbf{c}_i^\top \widetilde{\mathbf{w}} \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{c}_i b_i + T\gamma \mathbf{c}_i^\top \widetilde{\mathbf{c}} + \mathbf{c}_i^\top \widetilde{\mathbf{w}} \right) \\&= \frac{\alpha}{n} \left( T\gamma R b + \varepsilon_i^C + \varepsilon_i^W \right) = \frac{\alpha}{n} (T\gamma R b + \varepsilon_i)\end{aligned}$$

- Each component of  $\widetilde{\mathbf{w}}$  adds up  $T(n-1+\beta)$  of these values
- By CLT (large enough  $T(n-1+\beta)$ ) each one of these variables would be zero-mean Gaussian with a variance  $Tn\sigma^2$ .
- When we multiply this vector by  $\mathbf{c}_i$  and add all the components together, we end up with a zero-mean Gaussian with a variance  $RTn\sigma^2$ , because the components of  $\mathbf{c}_i$  are  $\pm 1$ .

# How we recover $b_i$ ?

---



# How we recover $\mathbf{b}_i$ ?

---

$$\begin{aligned}y_i &= \mathbf{c}_i^\top (\mathbf{W}_T - \mathbf{W}_0) = \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{c}_i b_i + T\gamma \mathbf{c}_i^\top \mathbf{C}_{-i} \mathbf{b}_{-i} + \mathbf{c}_i^\top \tilde{\mathbf{w}} \right) \\&= \frac{\alpha}{n} \left( T\gamma \mathbf{c}_i^\top \mathbf{c}_i b_i + T\gamma \mathbf{c}_i^\top \tilde{\mathbf{c}} + \mathbf{c}_i^\top \tilde{\mathbf{w}} \right) \\&= \frac{\alpha}{n} \left( T\gamma R b + \boldsymbol{\varepsilon}_i^C + \boldsymbol{\varepsilon}_i^W \right) = \frac{\alpha}{n} (T\gamma R b + \boldsymbol{\varepsilon}_i)\end{aligned}$$

- $\mathbf{c}$ ,  $\mathbf{b}$  and  $\nabla \mathbf{W}_t^k$  are mutually independent, thus  $\boldsymbol{\varepsilon}_i$  is zero mean with a variance that it is the sum of the variances of  $\boldsymbol{\varepsilon}_i^C$  and  $\boldsymbol{\varepsilon}_i^W$ , and also Gaussian distributed.
- The distribution of  $y_i$  is given by  $y_i \sim \mathcal{N}(\mathbf{T}\gamma R b_i, T^2 \gamma^2 R(P-1) + RTn\sigma^2)$ , and if normalize by  $\mathbf{T}\gamma R$ :

$$\begin{aligned}y_i &\sim \mathcal{N}\left(b_i, \frac{T^2 \gamma^2 R(P-1) + TRn\sigma^2}{T^2 \gamma^2 R^2}\right) \\&\sim \mathcal{N}\left(b_i, \frac{P-1}{R} + \frac{n\sigma^2}{TR\gamma^2}\right)\end{aligned}$$

# How we recover $b_i$ ?

---

$$y_i \sim \mathcal{N} \left( b_i, \frac{T^2 \gamma^2 R(P-1) + TRn\sigma^2}{T^2 \gamma^2 R^2} \right)$$

$$\mathcal{N} \left( b_i, \frac{P-1}{R} + \frac{n\sigma^2}{TR\gamma^2} \right)$$

Using a long enough error-correcting code, we can ensure errorless communication when the variance is about **1**.

$$\beta=1 \text{ and } \gamma=0.1\sigma/\sqrt{P}$$

$$\begin{aligned} \frac{P-1}{R} + \frac{n\sigma^2}{TR\gamma^2} &= \frac{P-1}{R} + \frac{n\sigma^2}{TR \left( \frac{0.1\sigma}{\sqrt{P}} \right)^2} \\ &= \frac{P-1}{R} + \frac{100nP\sigma^2}{TR} \approx \frac{(T+100n)P}{TR} \end{aligned}$$

- At least **T>100nP/(R-P)** rounds before the message can be decoded.

# How we recover $b_i$ faster?

---

- We incorporate multiple senders



# How we recover $b_i$ faster?

---

- We incorporate multiple senders

- The  $y_i$  distribution will become:

$$\mathcal{N}(MT\gamma Rb_i, M^2T^2\gamma^2R(P - 1) + RTn\sigma^2)$$

# How we recover $b_i$ faster?

---

- We incorporate multiple senders
  - The  $y_i$  distribution will become:

$$\mathcal{N}(MT\gamma Rb_i, M^2T^2\gamma^2R(P - 1) + RTn\sigma^2)$$

$T > nP/M^2(R-P)$        $M^2$  times faster



# The pillars of evaluation

---

**Stealthiness of  
the  
communication**



**Impact on  
model  
performance**



**“Message”  
delivery time**

# The pillars of evaluation

---

**Stealthiness of  
the  
communication**



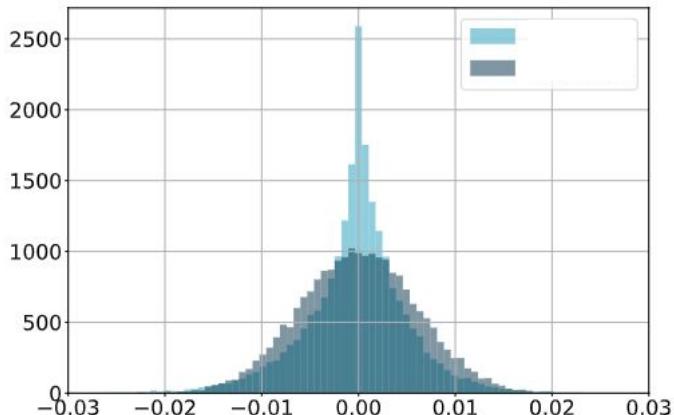
**Impact on  
model  
performance**



**“Message”  
delivery time**

# Stealthiness of communication

---



Regular gradient update

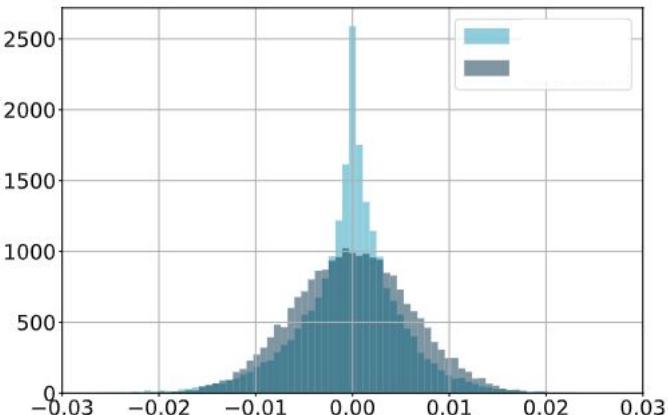
vs.

**Non-Stealthy** gradient update

In **non-stealthy** we are not sending a gradient that is useful for learning, instead we send our signal with the same power as our gradient would have.

# Stealthiness of communication

---



Regular gradient update

vs.

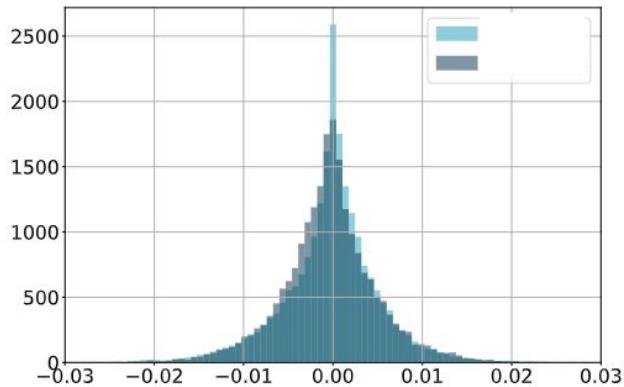
**Non-Stealthy** gradient update



NON-STEALTHY **might** be detectable in cases where the global parameter server can observe individual gradient updates.

# Stealthiness of communication

---



Regular gradient update

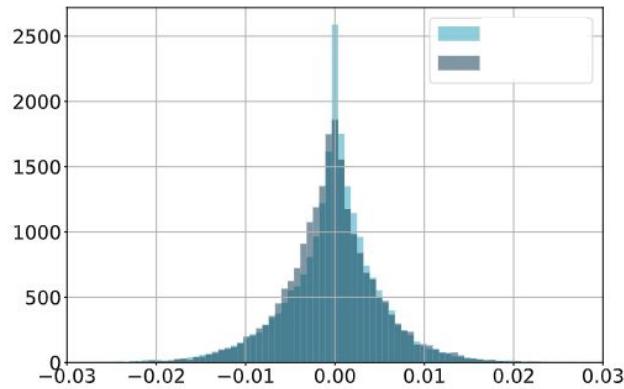
vs.

**Full-Stealthy** gradient update

In **full-stealthy** we are sending a gradient that is useful for learning, and buried into it we put also our signal.

# Stealthiness of communication

---



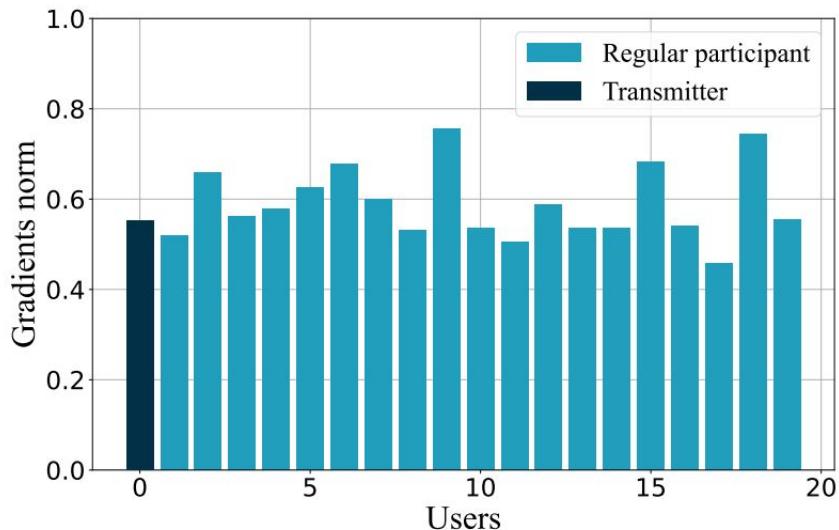
Regular gradient update  
vs.  
**Full-Stealthy** gradient update



Gradients are statistically indistinguishable

# Stealthiness of communication (cont.)

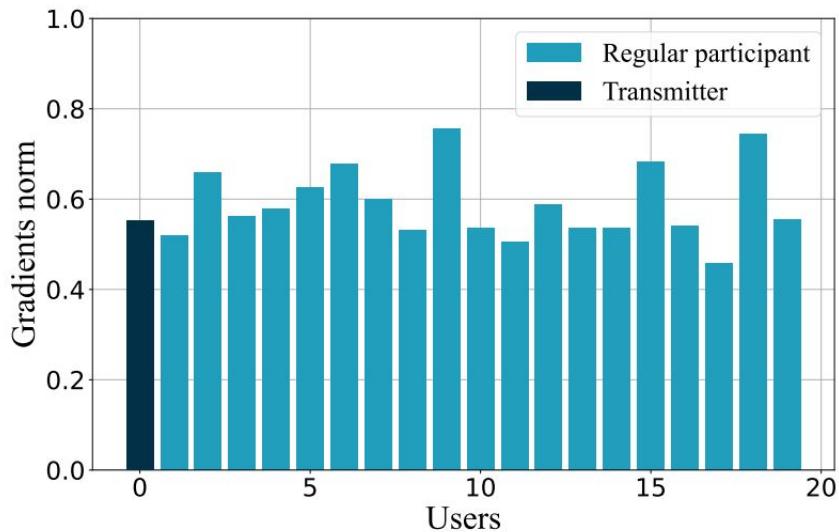
---



Parameter server observes  
individual gradient updates

# Stealthiness of communication (cont.)

---



Parameter server observes individual gradient updates

# The pillars of evaluation

---

**Stealthiness of  
the  
communication**



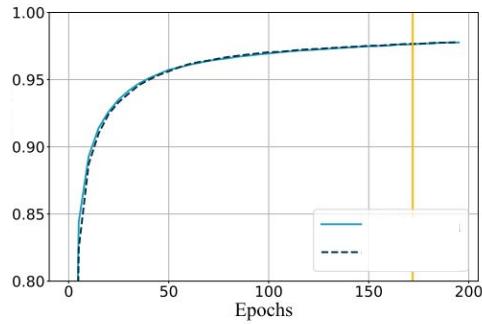
**Impact on  
model  
performance**



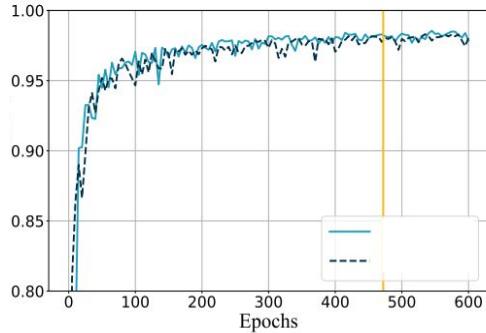
**“Message”  
delivery time**

# Impact on the model performance

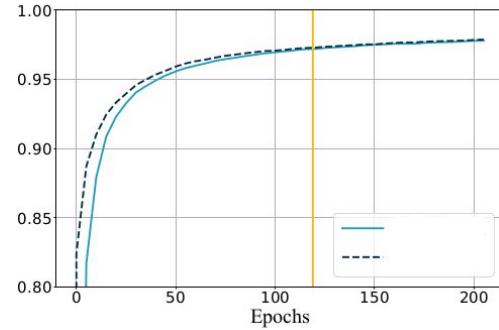
---



100 participants  
10 senders FULL Stealthy  
100% aggregated per round



100 participants  
10 senders FULL Stealthy  
20% aggregated per round



100 participants  
1 sender NON Stealthy  
100% aggregated per round

# The pillars of evaluation

---

**Stealthiness of  
the  
communication**



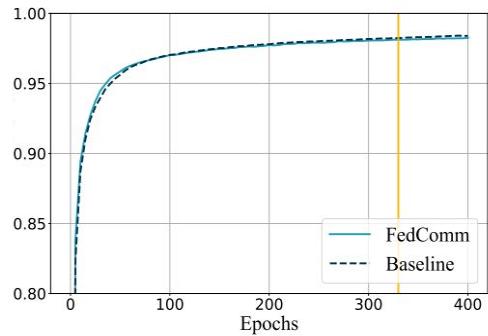
**Impact on  
model  
performance**



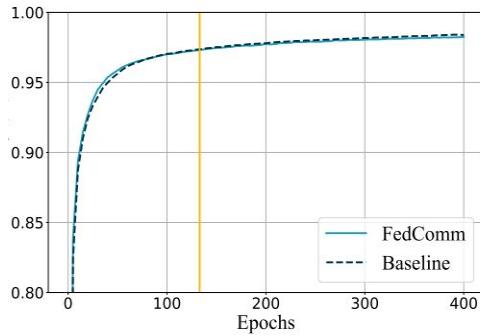
**“Message”  
delivery time**

# “message” delivery time

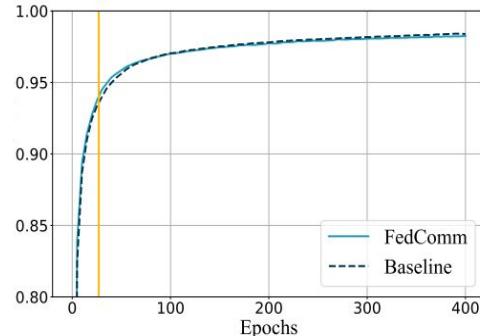
---



1 sender  
100% aggregated per round



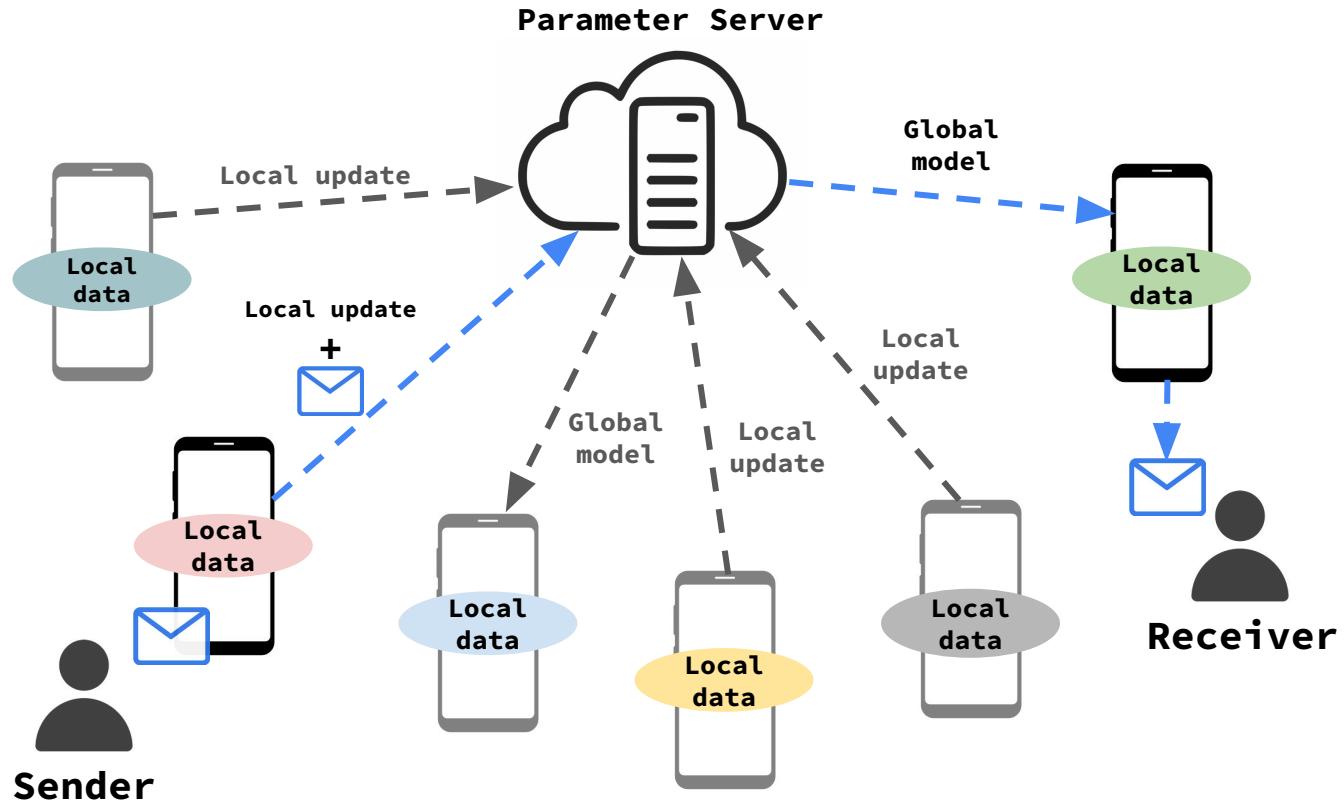
2 senders  
100% aggregated per round



4 senders  
100% aggregated per round

# Remarks

---





*That's all Folks!*

# Reading Material

1. Covert Channels (Concepts and definitions): [Link](#)
2. Covert communication in collaborative learning: [Link-1](#), [Link-2](#) (research papers).