Federated learning and its application for a privacy-preserving Android malware classifier

Rafa Gálvez, KU Leuven
Veelasha Moonsamy, Ruhr University Bochum

RuhrSec 2023, Bochum

Android malware - why should we care?

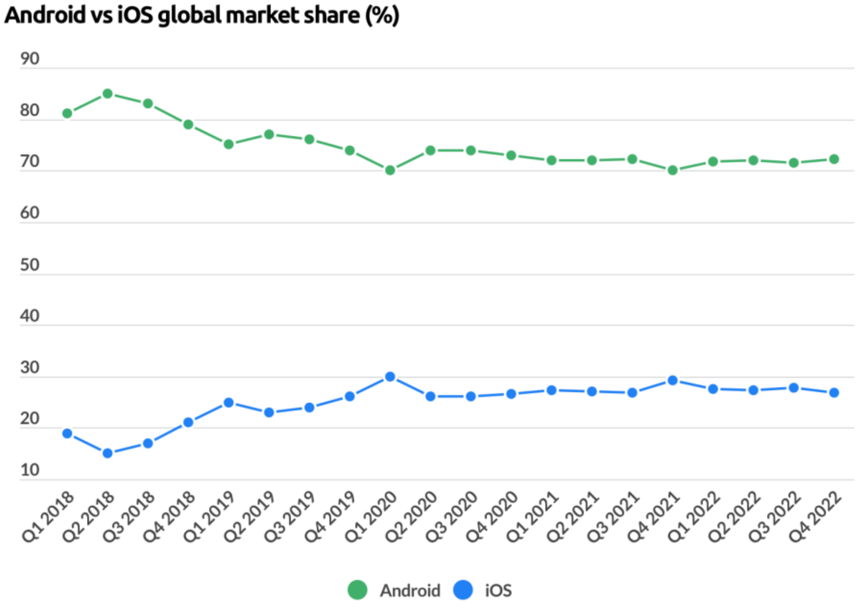
Security Researchers Issue Warning Against New Android Malware That Infiltrated Google Play Through 60 Apps With 100 Million Installs

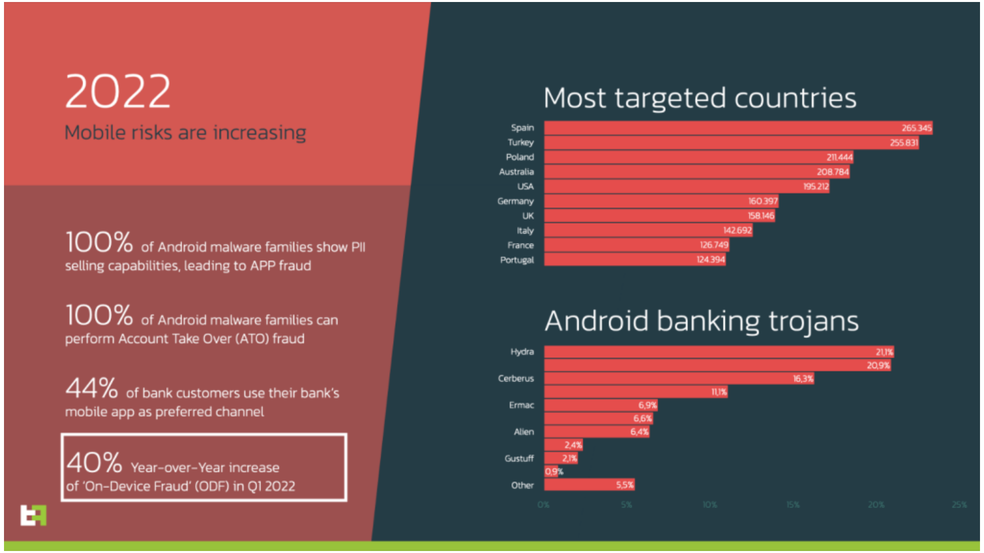
Goldoson Android Malware Infects Over 100 Million Google Play Store Downloads

New malware infects Android TVs, IoT devices in 84 nations

A new malware has infected roughly 13,500 Internet of Things (IoT) devices like Android TVs in

Android malware - statistics



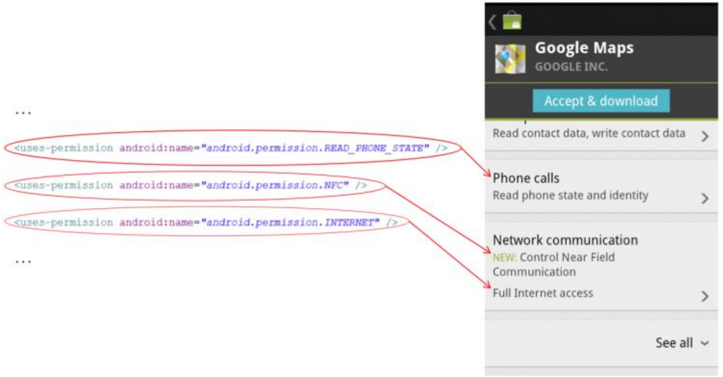


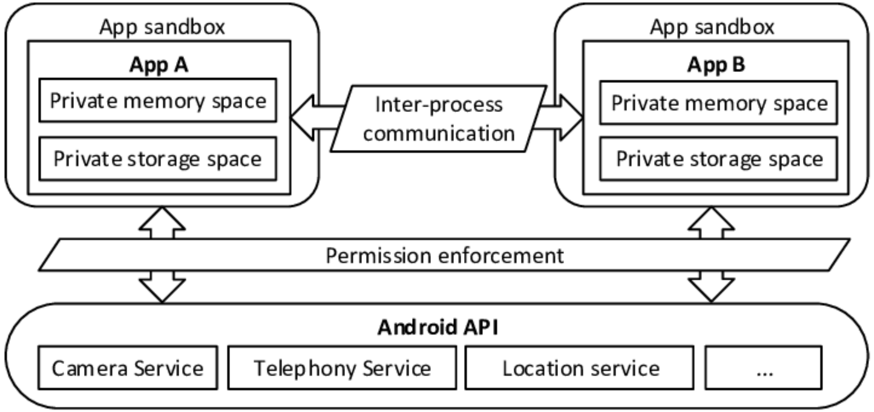
https://www.businessofapps.com/data/android-statistics/
https://www.threatfabric.com/assets/images/blog/landscape/stats.png

Android Security Model - The early days

Three-pronged approach:

- Application vetting process
- Permission systems
- Sandboxing





https://developers.google.com/android/play-protect/cloud-based-protections
https://owasp.org/www-pdf-archive/Danelon_OWASP_EU_Tour_2013.pdf
https://www.researchgate.net/figure/Two-sandboxed-Android-apps-and-their-interaction-with-one-another-and-with-the-Android_fig5_307984756

Android Security Model - More recently …

Google Play
Protect

Use Google Play Protect to help keep your apps safe and your data private

Google Play Protect checks your apps and devices for harmful behavior.

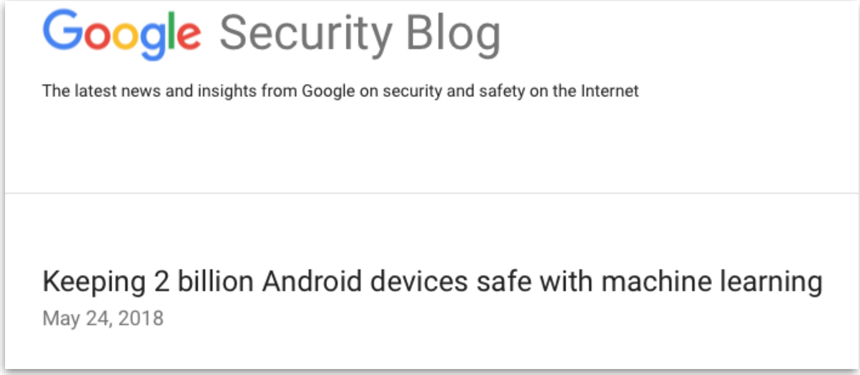
- It runs a safety check on apps from the Google Play Store before you download them.
- It checks your device for potentially harmful apps from other sources. These harmful apps are sometimes called malware.

Send unknown apps to Google

If you install apps from unknown sources outside of the Google Play Store, Google Play Protect may ask you to send unknown apps to Google. When you turn on the "Improve harmful app detection" setting, you allow Google Play Protect to automatically send unknown apps to Google.

How would Google implement this??

Machine Learning to the rescue!



Google Security Blog

The latest news and insights from Google on security and safety on the Internet

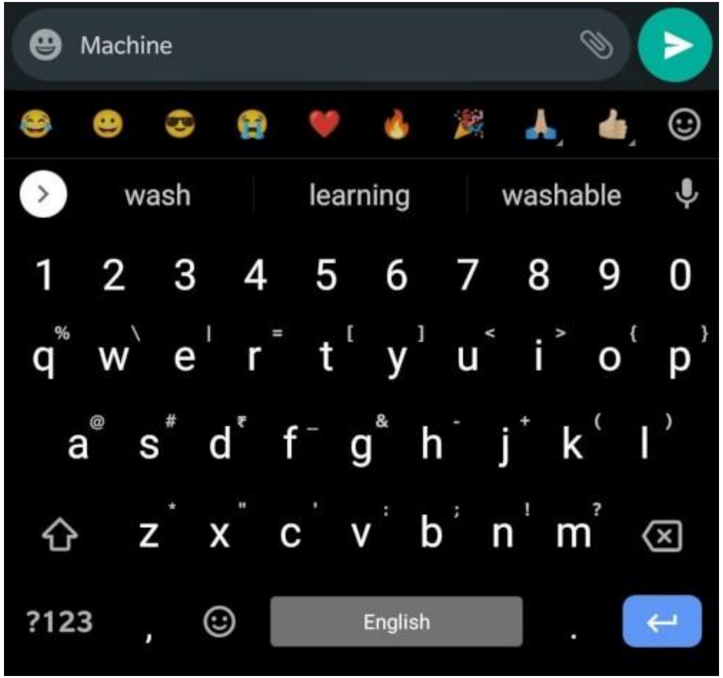
Keeping 2 billion Android devices safe with machine learning

May 24, 2018

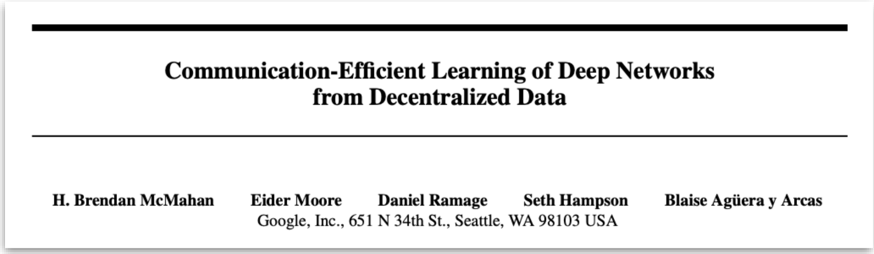
"In the most basic terms, machine learning means training a computer algorithm to recognize a behavior. To train the algorithm, we give it hundreds of thousands of examples of that behavior. In the case of Google Play Protect, we are developing algorithms that learn which apps are "potentially harmful" and which are "safe." To learn about PHAs, the machine learning algorithms analyze our entire catalog of applications. Then our algorithms look at hundreds of signals combined with anonymized data to compare app behavior across the Android ecosystem to find PHAs."

Can we do better?

Federated Learning: an alternative approach

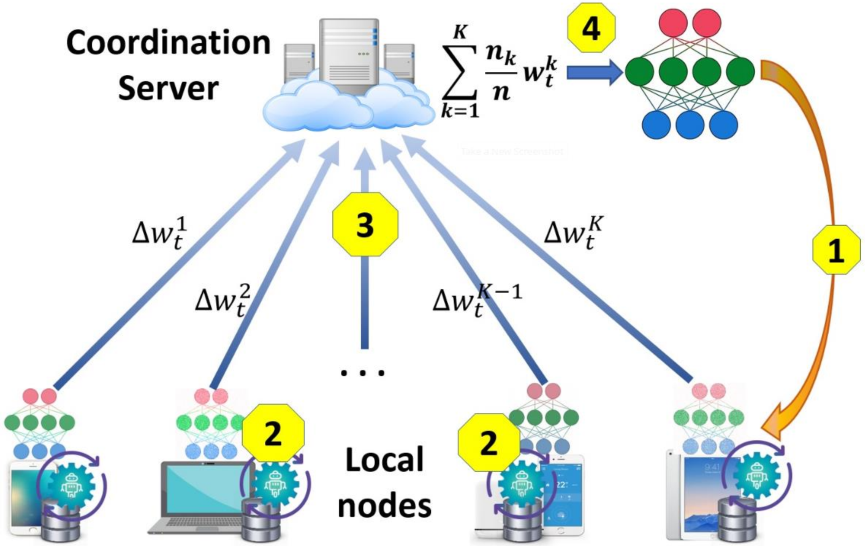


Federated Learning (FL)


Communication-Efficient Learning of Deep Networks from Decentralized Data

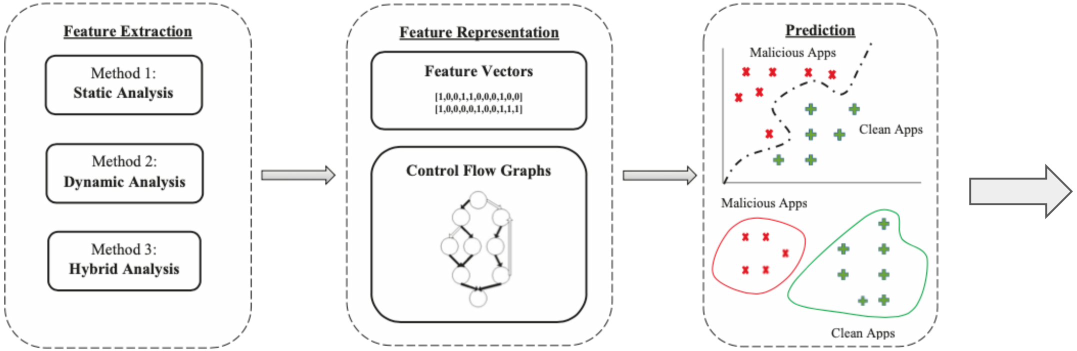
H. Brendan McMahan Eider Moore Daniel Ramage Seth Hampson Blaise Agüera y Arcas
Google, Inc., 651 N 34th St., Seattle, WA 98103 USA

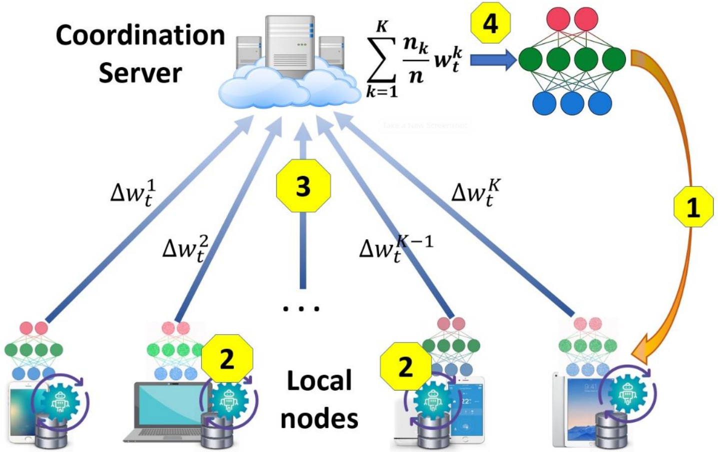
- Also known as *Collaborative Learning*
- First introduced and coined by Google in 2017



Truong, Nguyen, Kai Sun, Siyao Wang, Florian Guitton, and Yike Guo. 'Privacy Preservation in Federated Learning: Insights from the GDPR Perspective'. ArXiv:2011.05411 [Cs], 22 January 2021. http://arxiv.org/abs/2011.05411.

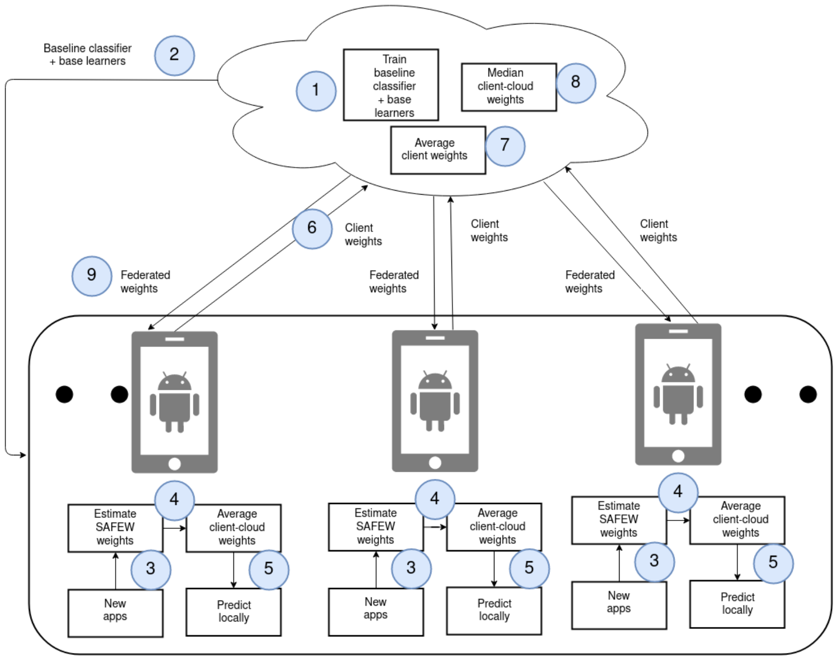
Classical malware detection vs FL-based malware detection





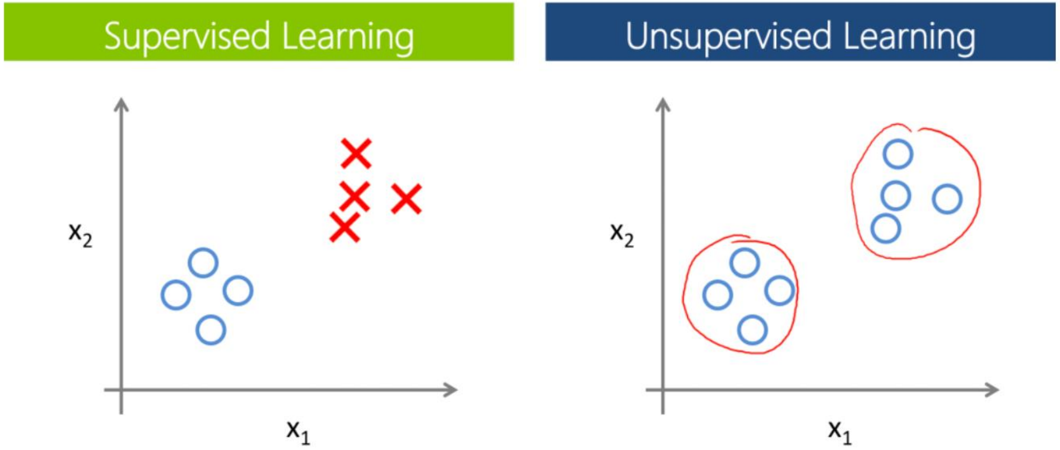
Federated Learning: opportunity & challenges

- Opportunity: self-evolving, privacy preserving malware classifier
- Challenge: data minimization
- Proposal: share model instead of data



Federated Learning: assumption & risks

- Assumption: users provide labels
- Risks:
  - Inference attacks: break privacy
  - Poisoning attacks: break performance



https://commons.wikimedia.org/wiki/File:Machin_learning.png

Inference attacks for FL

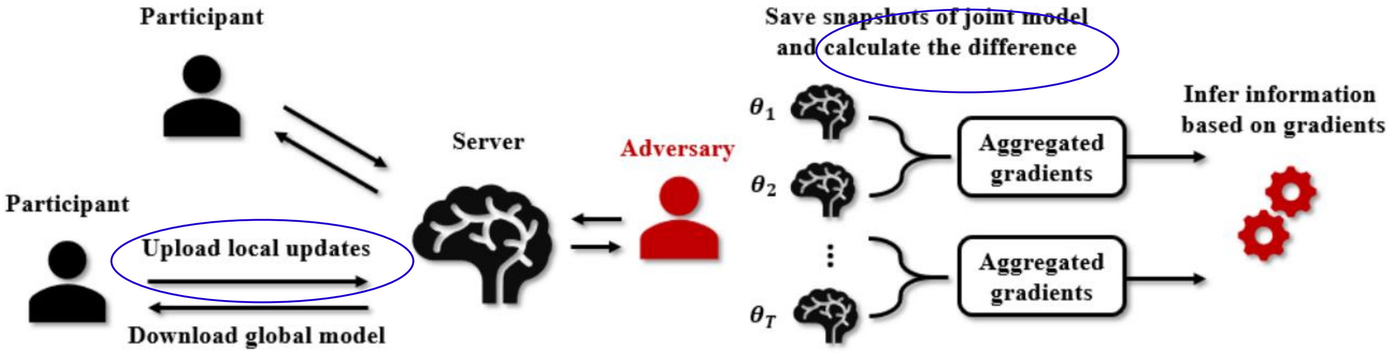


Figure 2: Overview of inference attacks against collaborative learning.

L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting Unintended Feature Leakage in Collaborative Learning," in 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, US, May 2019, pp. 691–706. doi: 10.1109/SP.2019.00029.

Poisoning attacks for FL

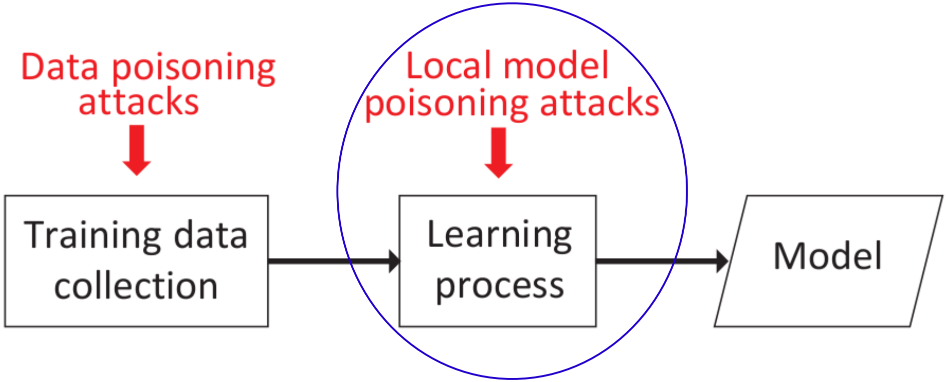


Figure 1: *Data* vs. *local model* poisoning attacks.

M. Fang, X. Cao, J. Jia, and N. Gong, “Local Model Poisoning Attacks to Byzantine-Robust Federated Learning,” 2020, pp. 1605–1622. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/fang

Our solution: Less is More

sciendo

Proceedings on Privacy Enhancing Technologies ; 2021 (4):96–116

Rafa Gálvez*, Veelasha Moonsamy, and Claudia Diaz

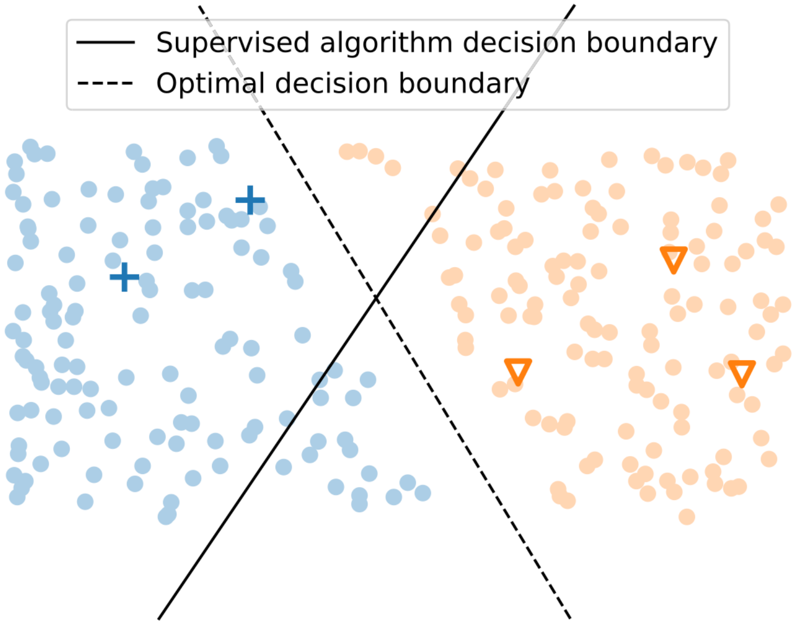
Less is More: A privacy-respecting Android malware classifier using federated learning

- Semi-supervised Federated Learning
- User models can be trained without labels
  - Leverage semi-supervised learning
- Address inference and poisoning attacks
  - Reduce dimensionality
  - Offset outliers from submitted parameters

Reference: V. Shejwalkar and A. Houmansadr, “Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning,” Feb. 2021, p. 18. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2021-498-paper.pdf

Semi-supervised learning

- Address the challenge of obtaining labeled data
- Two main assumptions:
  - Examples close in feature space share labels
  - Different classes are separated by low density regions
- Ensemble learning
  - Multiple classifiers together



(a) Smoothness and low-density assumptions.

van Engelen, J.E., Hoos, H.H., 2020. A survey on semi-supervised learning. Mach Learn 109, 373–440. https://doi.org/10.1007/s10994-019-05855-6

Safe semi-supervised learning

- How to keep improving with unlabeled data?
- Always beat baseline performance

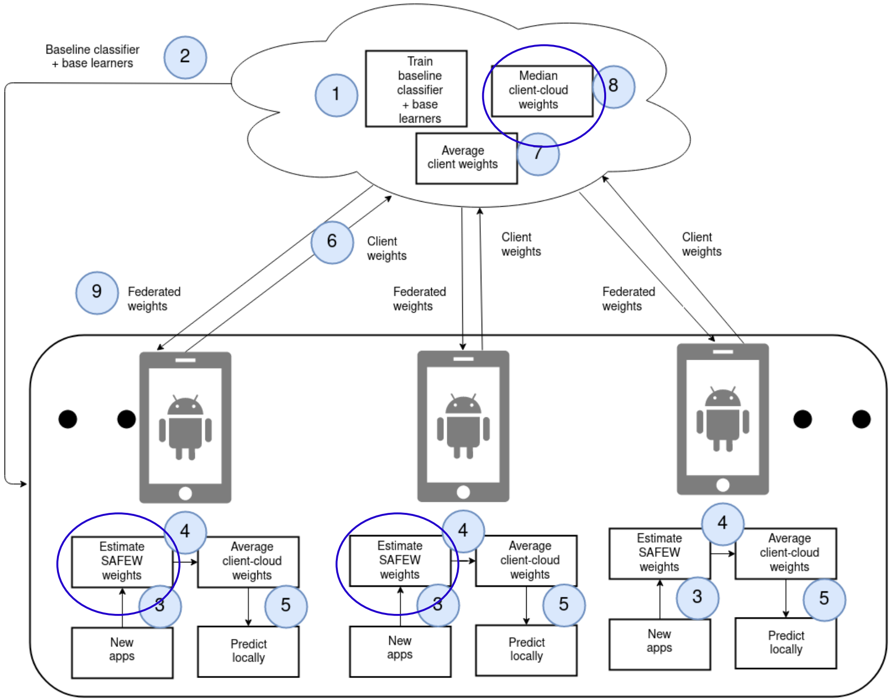
$$\max_{\mathbf{f}\in\{+1,-1\}} \min_{\alpha} l(\mathbf{f}_0, \sum_{i=1}^{b} \alpha_i \mathbf{f}_i) - l(\mathbf{f}, \sum_{i=1}^{b} \alpha_i \mathbf{f}_i)$$

L1	L2	L3	L4	LN
W1	W2	W3	W4	W5

- Assumption: the correct prediction lies in the combination of base learners

Li, Y., Guo, L., Zhou, Z., 2019. Towards Safe Weakly Supervised Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 43, 334–346. https://doi.org/10.1109/TPAMI.2019.2922396

The LiM architecture



R. Gálvez, V. Moonsamy, and C. Diaz, "Less is More: A privacy-respecting Android malware classifier using federated learning," Proceedings on Privacy Enhancing Technologies, vol. 2021, no. 4, pp. 96–116, Oct. 2021, doi: 10.2478/popets-2021-0062.

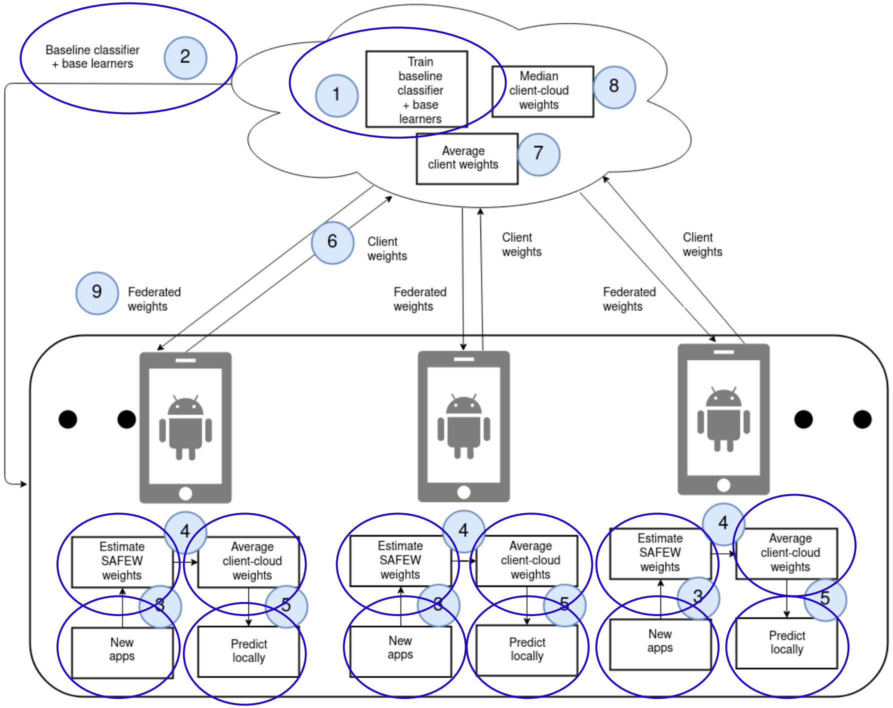
LiM preparation

- Step 1) train base learners of the ensemble
  - How many?
  - Which models?
  - Fully supervised, or semi-supervised?
- We use a small number (5) to reduce dimensionality
  - Security and privacy by design
- No restriction on which kinds of models
  - We use random forests, SVMs, logistic regression, k-nearest-neighbours
  - Room for improvement for specific applications

RF	SVM	RF	LR	RF
W1	W2	W3	W4	W5

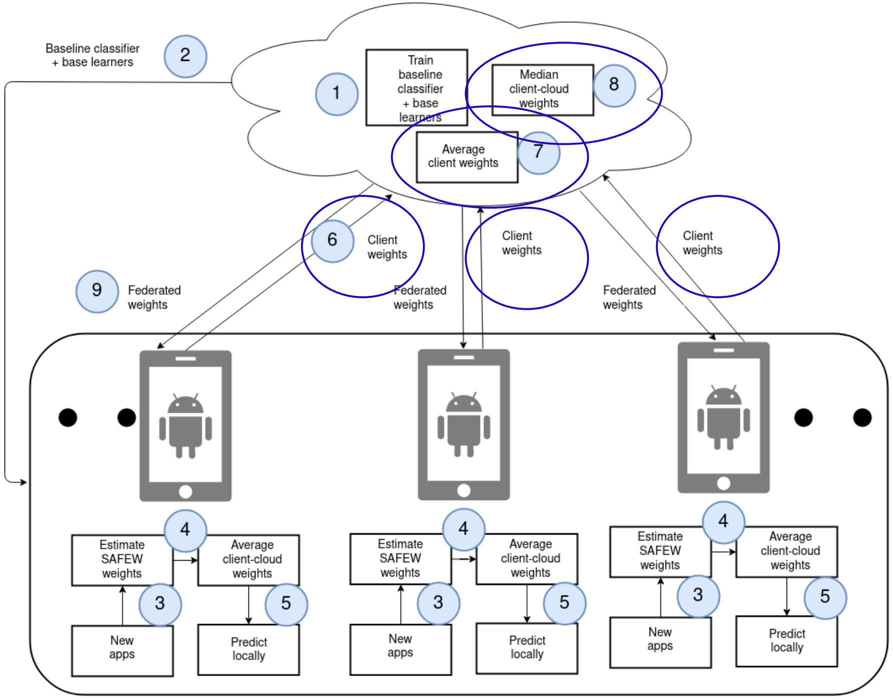
LiM round

- Step 2) Share base learners
- Step 3) Users estimate new parameters for local model
  - Using their local data set
  - New examples are unlabeled
- Step 4) Clients average local and shared parameters
- Step 5) Predict locally



LiM round

- Step 6) Clients share new parameters with the service provider
- Step 7) Cloud averages client parameters
- Step 8) Cloud averages client and secret parameters

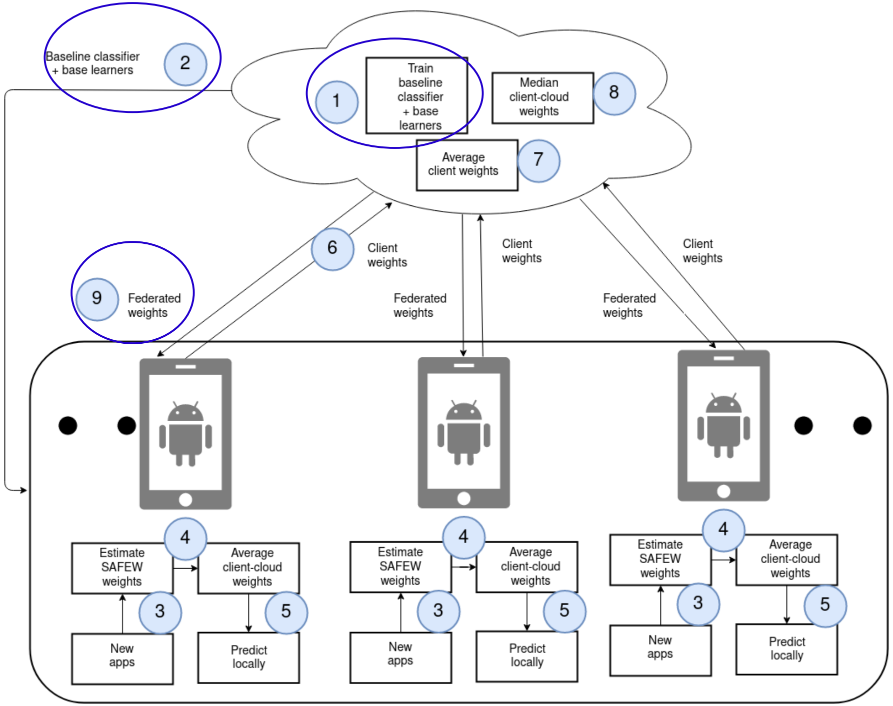


LiM round

- Step 9) Cloud shares new parameters with clients

At any time, the cloud may:

- Retrain base learners
- Share new base learners



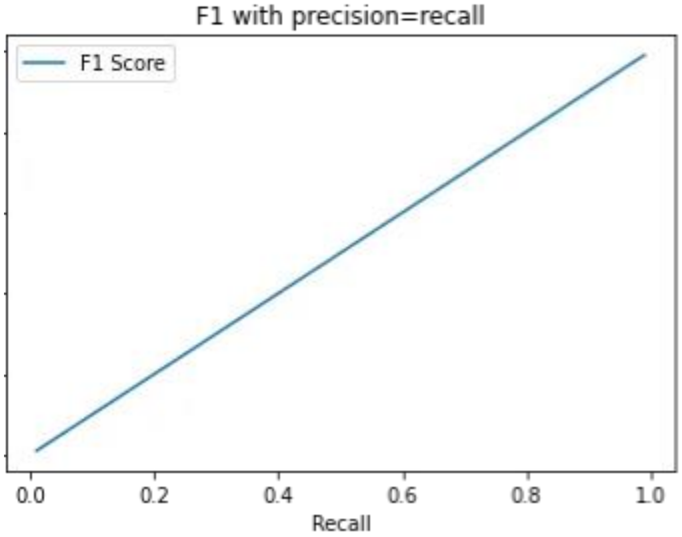
Measuring success in LiM

- Measuring success in ML is hard
- Application dependent!
  - Malware <<< cleanware
  - Minimize false positives
- Precision: how many times I detected malware was actually malware
- Recall: how many malware apps I caught
- F1 score: we care about both

		Predicted condition		Sources: [1][2][3][4][5][6][7][8][9]	
	Total population $= P + N$	Positive (PP)	Negative (PN)	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	Positive (P)	**True positive (TP),** hit	**False negative (FN),** type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP),** type I error, false alarm, overestimation	**True negative (TN),** correct rejection	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

https://en.wikipedia.org/wiki/Template:Diagnostic_testing_diagram

Measuring success in LiM

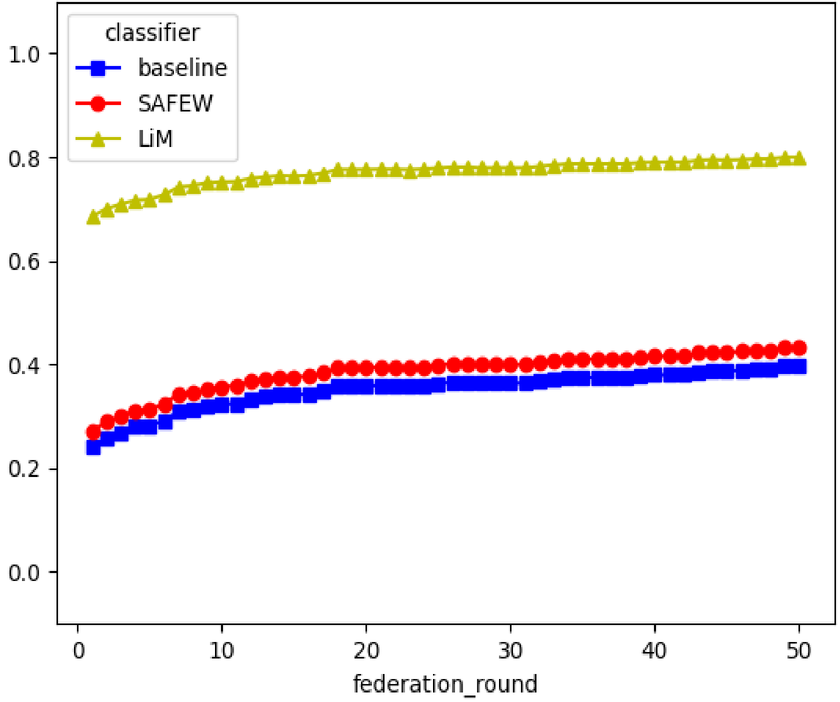


	Precision	Recall	F1-Score	Difference
0	0.1	0.1	0.1	0.0
1	0.2	0.2	0.2	0.0
2	0.3	0.3	0.3	0.0
3	0.4	0.4	0.4	0.0
4	0.5	0.5	0.5	0.0
5	0.6	0.6	0.6	0.0
6	0.7	0.7	0.7	0.0
7	0.8	0.8	0.8	0.0
8	0.9	0.9	0.9	0.0
9	1.0	1.0	1.0	0.0

https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec

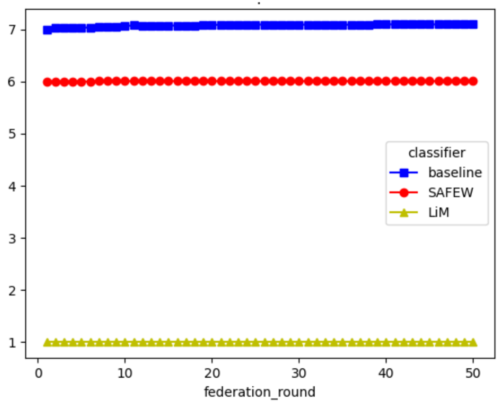
Results

F1 score



- FL can learn from unsupervised clients
- But what about false positives?

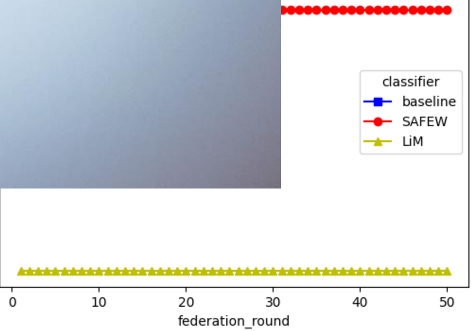
False positives



Results

n from unsupervised

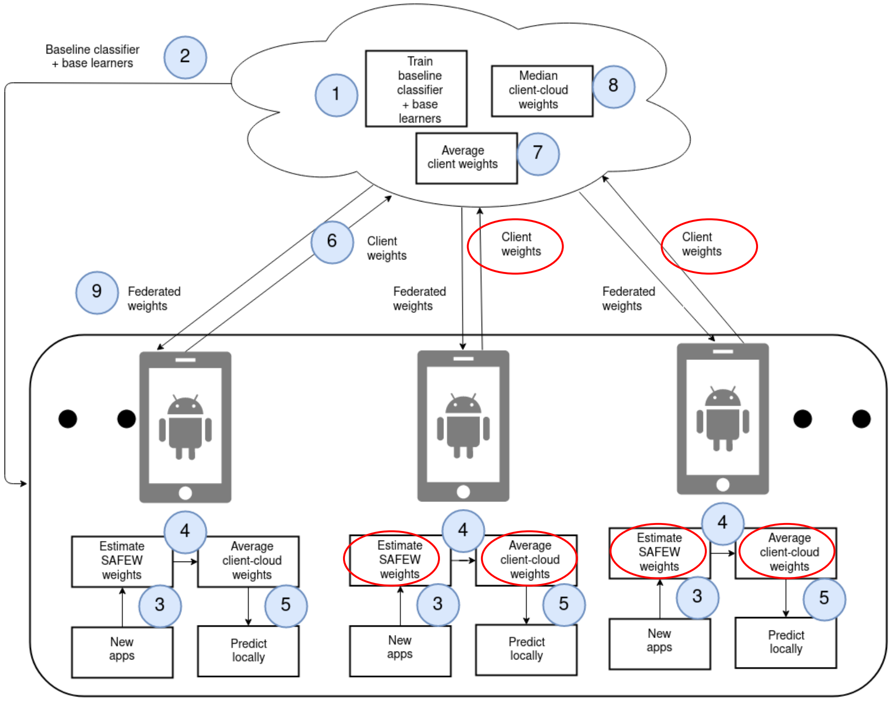
bout false positives?



Security analysis: poisoning

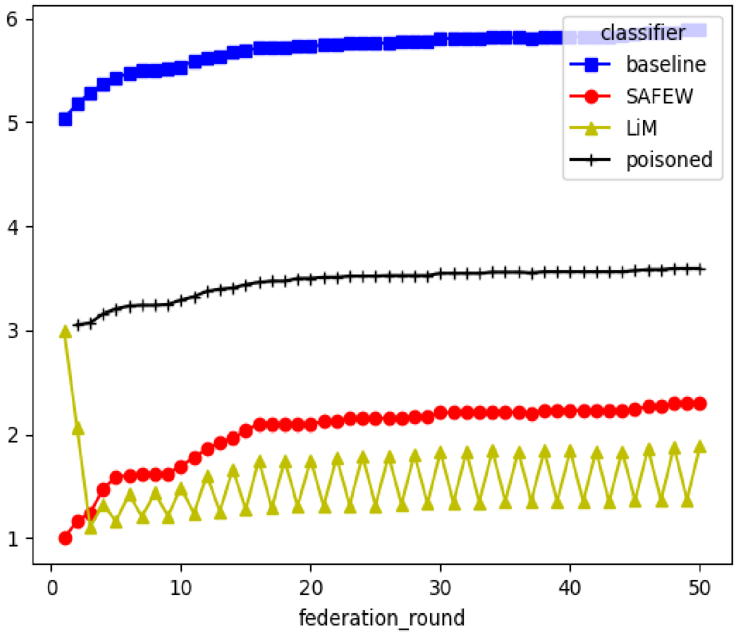
Every federation round:

- Users install apps (10% malware)
- Adversary compromises 50% of users to poison the model
- Strategic: evade detection of specific malicious app



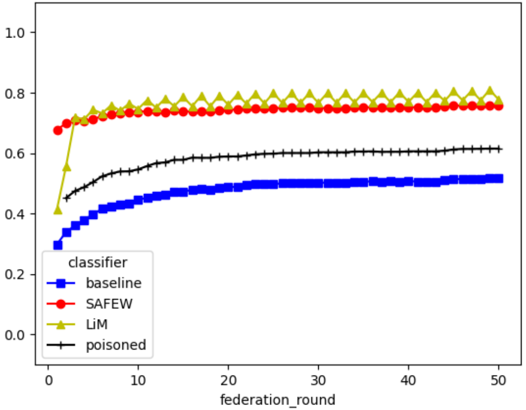
Results

False positives



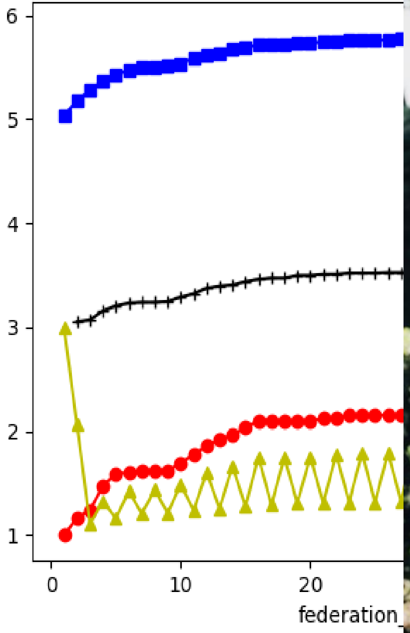
- Attack does not succeed
- Private cloud data set
- Small attack surface
    - Few parameters
    - They must add up to 1

F1 score



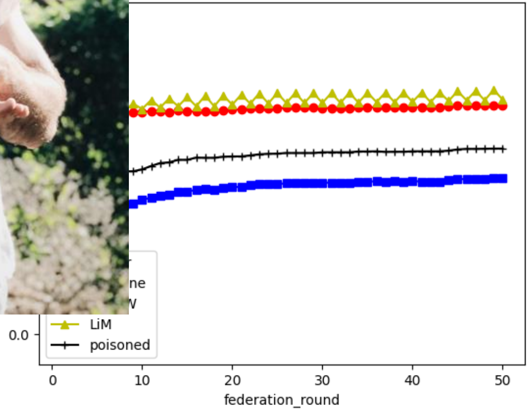
Results

False positives



k does not succeed
te cloud data set
attack surface
Few parameters
They must add up to 1

core



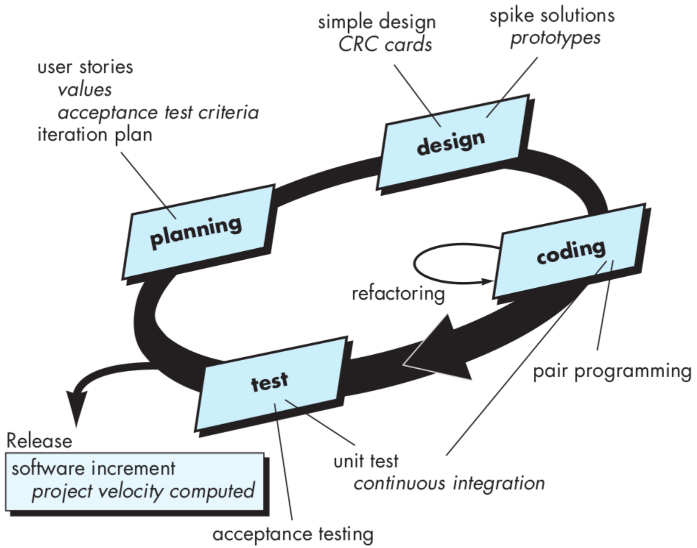
Privacy analysis: inference

- Cloud wants to infer installed apps
    - Membership inference
- Access to parameters of individual clients, per round
    - Did the user install this app in last round?
- Method
    - Train model with a single, unlabeled example of the target app
    - Membership test: are submitted user parameters the same?
- Result: **no success**
    - Not enough information in such a small set of parameters

Future challenges

- Selection of base learners
  - Performance!
- Integration in real-world applications
  - High quality library
  - Easy deployability
- Evaluation in a real world setting

RF	SVM	RF	LR	RF
W1	W2	W3	W4	W5



Future applications

- Automated grading
  - Client model in the student device
  - Cloud model in a (set of) schools
  - LiM as encoder to generate embeddings for later decoding
- Fraud detection
  - Client models in individual banks
  - Consortium for cloud model
  - LiM as an ensemble of fraud detectors
- Network programmability
  - ML acceleration

Conclusion

- Semi-supervised learning broadens applicability of FL
- Lower dimensionality stops inference attack
- Private data set stops powerful poisoning attack

Thank you!

Paper: https://petsymposium.org/popets/2021/popets-2021-0062.pdf
Code: https://git.sr.ht/~rafagalvez/lim-python