Aalto University

UNIVERSITY OF WATERLOO

Model Stealing and Ownership Verification of Deep Neural Networks

Buse G. A. Tekgul

(Joint work with Yuxi Xia, Sebastian Szyller, Samuel Marchal, N. Asokan)

Is model confidentiality important?

Machine learning models: business advantage and intellectual property (IP)

Cost of

- gathering relevant data
- labeling data
- expertise required to choose the right model training method
- resources expended in training

Adversary who steals the model can avoid these costs

How to prevent model theft?

White-box model theft can be countered by:

- encrypted models
- secure hardware
- firewalled cloud service

Basic idea: hide the model itself, expose model functionality only via a prediction API.

Not sufficient against **black-box theft** – adversary omits these defenses!

Preventing and detecting **black-box** attacks (?)

Extracting Deep Neural Networks

Against simple DNN models[1]

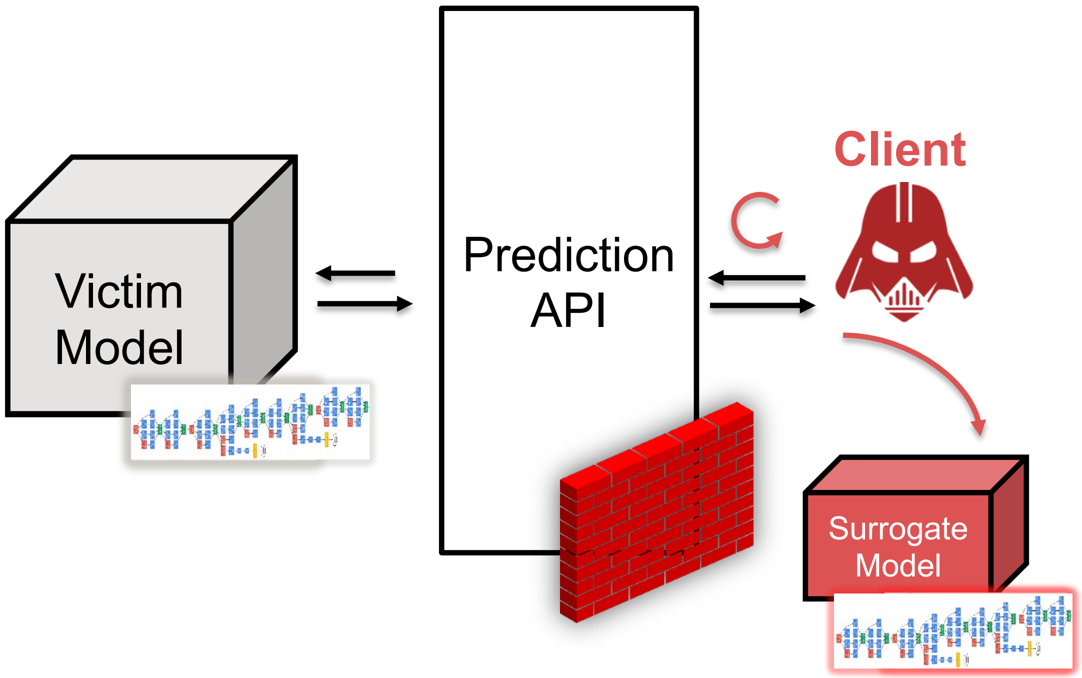
- E.g., MNIST, GTSRB

Adversary

- knows general structure of the model
- has limited natural data from victim's domain

Approach

- Hyperparameters CV-search
- Query using natural data for rough estimate decision boundaries, synthetic data to fine-tune
- Simple defense: distinguish between benign and adversarial queries



[1] Juuti et al. - *PRADA: Protecting against DNN Model Stealing Attacks*. EuroS&P '19 (https://arxiv.org/abs/1805.02628)

Extraction of Complex DNN Models: Knockoff nets[1]

Goal:

- Build a surrogate model that
  - steals model functionality of victim model
  - performs similarly on the same task with high classification accuracy

Adversary capabilities:

- Victim model knowledge:
  - None of train/test data, model internals, output semantics
  - Access to full prediction probability vector
- Access to natural samples, not (necessarily) from the same distribution as train/test data
- Access to pre-trained high-capacity model

[1] Orekondy et al. - *Knockoff Nets: Stealing Functionality of Black-Box Models.* CVPR '19 (https://arxiv.org/abs/1812.02766)

Real Threat: Access to In-distribution Data

The larger the overlap between attacker's transfer set and victim's training data, the less effective the detection.

A more realistic adversary

- Has access to more (unlimited) data (public databases, search engines)
- Has approximate knowledge of prediction APIs task (food, faces, birds etc.)
- Can evade detection mechanisms identifying out-of-distribution queries

Are there any prevention mechanisms?

- Stateful analysis → Sybil attacks
- Charging customers upfront → Reduced utility for benign users
- Restrict access to the API → Reduced utility for benign users
- Slow down the attacker[1] → Does not thwart a well-resourced attacker

[1] Orekondy et al. – *Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks*. ICLR ' 20 (https://arxiv.org/abs/1906.10908)
[1] Atli et al. - *Extraction of Complex DNN Models: Real Threat or Boogeyman?* AAAI-EDSMLS'20 (https://arxiv.org/pdf/1910.05429.pdf)

Next Steps Towards Protection: Defense or Deter?

Is model confidentiality important? Yes

Can models be extracted via their prediction APIs? Yes[1]

- A powerful (but realistic) adversary can extract complex real-life models
- Detecting such an adversary is difficult/impossible

What can be done to counter model extraction?

[1] Atli et al. - *Extraction of Complex DNN Models: Real Threat or Boogeyman?* AAAI-EDSML'20 (https://arxiv.org/pdf/1910.05429.pdf)

Existing Watermarking of DNNs[1]

Watermark embedding:

- Embed watermark in model during training:
  - Train model using training data + trigger set (specific labels to a set of selected samples),

Verification of ownership:

- Requires adversary to publicly expose stolen model
- Query model with trigger set, verify watermark (predictions match trigger set labels)

Limitations:[2]

- Protects only against **physical theft** of model
- Model extraction attacks steal model **without watermark**

[1] Yadi et al. - *Watermarking Deep Neural Networks by Backdooring*. USENIX SEC ‘18 (https://www.usenix.org/node/217594)
[2] Szyller et. al. - *DAWN: Dynamic Adversarial Watermarking of Neural Networks.* ACMMM’21. (https://arxiv.org/abs/1906.00830)

DAWN: Dynamic Adversarial Watermarking of DNNs[1]

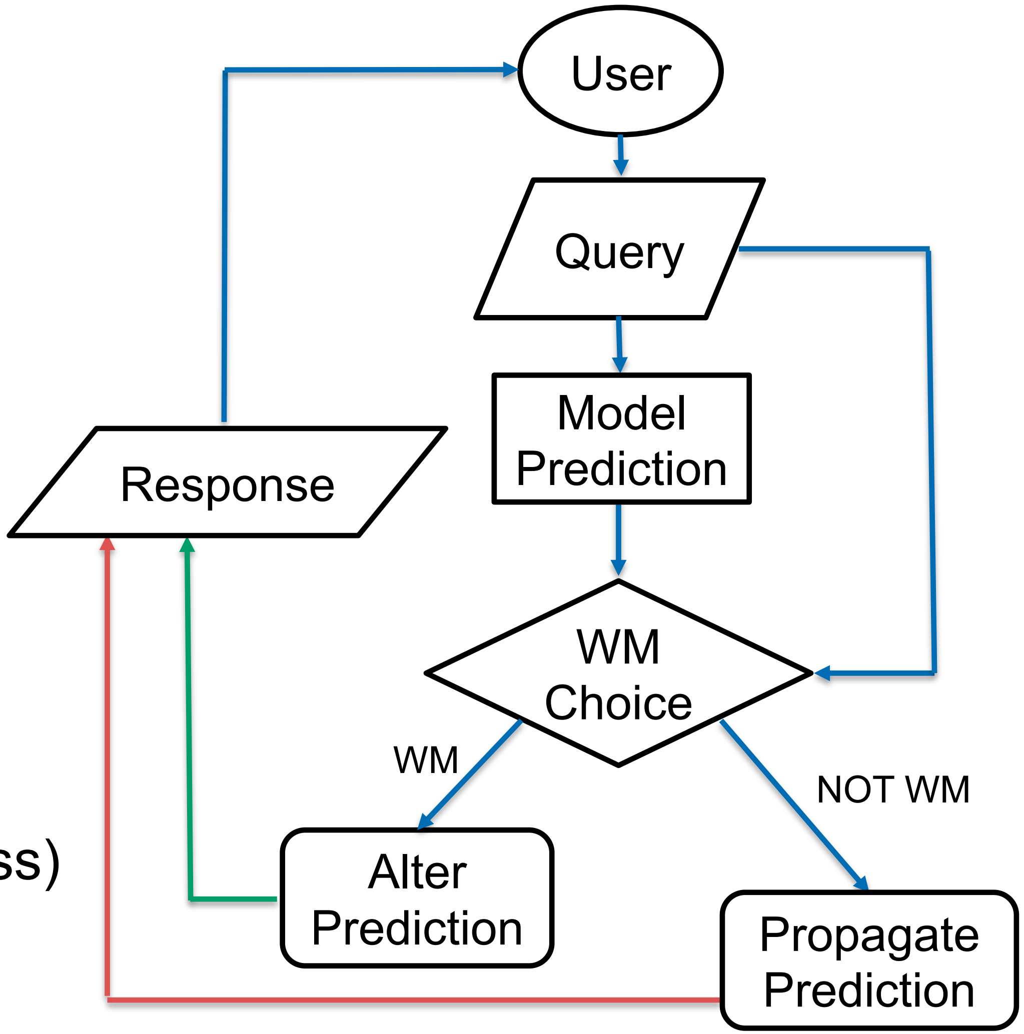
Goal: Watermark models obtained via model extraction

Our approach:

- Implemented as part of the **prediction API**
- Return **incorrect predictions** for several samples
- Adversary forced to embed watermark while training

Watermarking evaluation:

- **Unremovable** and **indistinguishable**
- **Defend against** *PRADA*[2] and *KnockOff* [3]
- Preserve victim *model utility* (*0.03-0.5%* accuracy loss)



[1] Szyller et. al. - *DAWN: Dynamic Adversarial Watermarking of Neural Networks.* ACMMM’21. (https://arxiv.org/abs/1906.00830)
[2] Juuti et al. - *PRADA: Protecting against DNN Model Stealing Attacks*. EuroS&P ’19 (https://arxiv.org/abs/1805.02628)
[3] Orekondy et al. - *Knockoff Nets: Stealing Functionality of Black-Box Models*. CVPR ’19 (https://arxiv.org/abs/1812.02766)

Watermark Decision and Backdoor Function

Decision function

$$W_{\mathcal{V}}(x) = \begin{cases} 1, & \text{if } \mathrm{HMAC}(K_w, x)[0, 127] < r_w \times 2^{128}. \\ 0, & \text{otherwise.} \end{cases}$$

Label flipping

- Get prediction vector **y** from the model
- Shuffle **y** using Fisher-Yates algorithm and obtain **y*.**
- Return **y*.**

label: **cat**

TRAINING SET
CORRECTLY LABELED
MANY SAMPLES

+

label: **airplane**

TRIGGER SET
INCORRECTLY LABELED
SEVERAL SAMPLES

= ATTACKER'S TRAINING DATA

Record (**X, y***) for future verification.

Verification of the watermark

Model owner registers its model and watermarks online:

- Registration is timestamped
- Requires a trusted third-party (the judge)

- Adversary makes its model available online
- Model owner asks the judge to verify the watermark and claim ownership
  - verify by querying stolen model with the trigger set

Adversary may attempt to register the stolen model with its own watermarks:

- Timestamping ensures that the true model comes first
- Probability of a random and registered watermark matching is negligible
  - with confidence $1 - 2^{-64}$

Properties and challenges

Properties:

- Unremovable (pruning, fine-tuning, regularization)
- Indistinguishable (*WM choice* robust to perturbation, detection with clustering)
- Reliable demonstration (resilient to Sybils and knowledgeable adversaries)

Challenges:

- Double extraction (adversary with a lot of data steals its own model)
- Robust *WM choice* function (difficult for complex datasets)

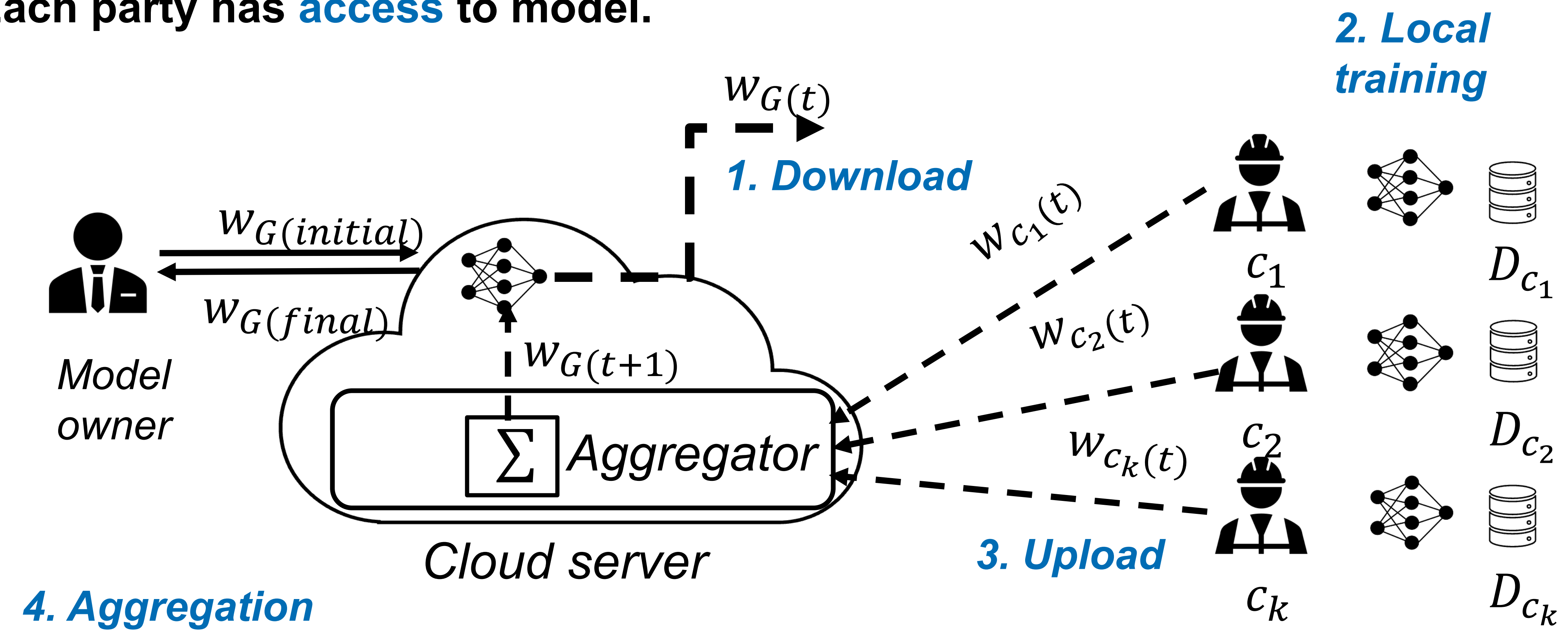
Aalto University

UNIVERSITY OF WATERLOO

Watermarking & Distributed Learning

Client-server Federated Learning

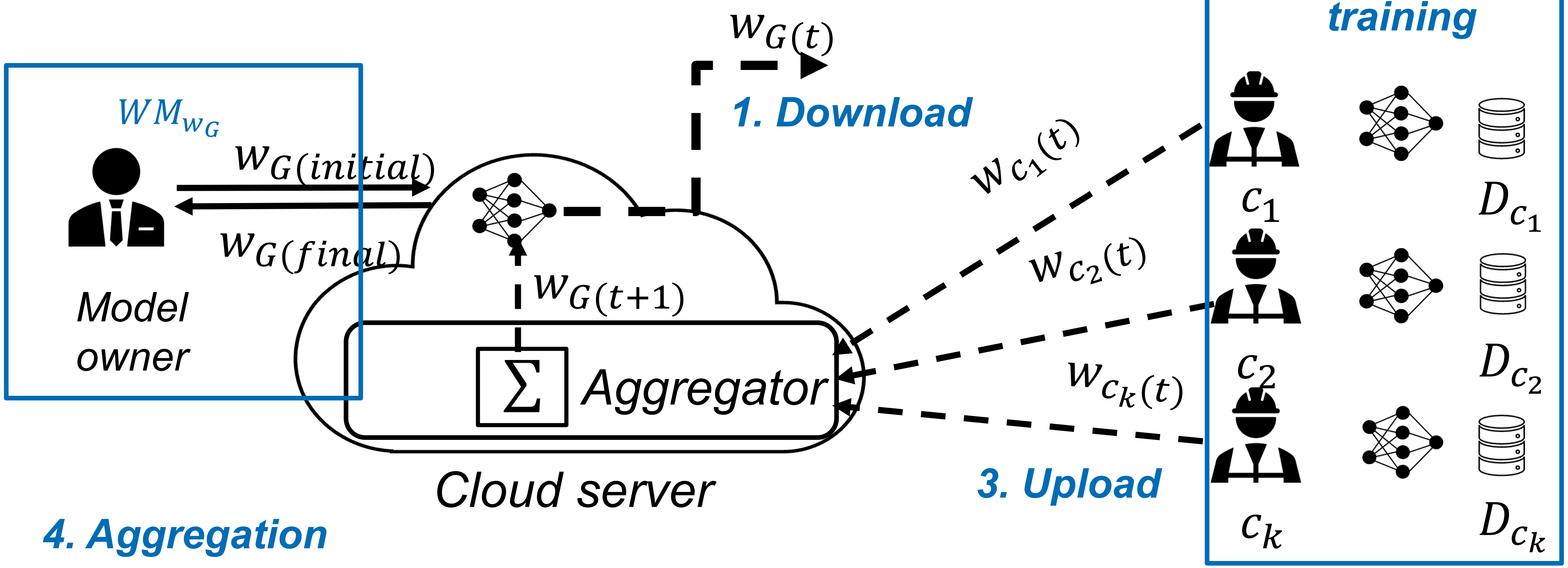
- Communication efficient and privacy preserving distributed training.
- One model owner (e.g., server or an external party) and multiple data owners.
- Each party has access to model.



Client-server Federated Learning

- **Ownership demonstration is important in client-server type configuration.**

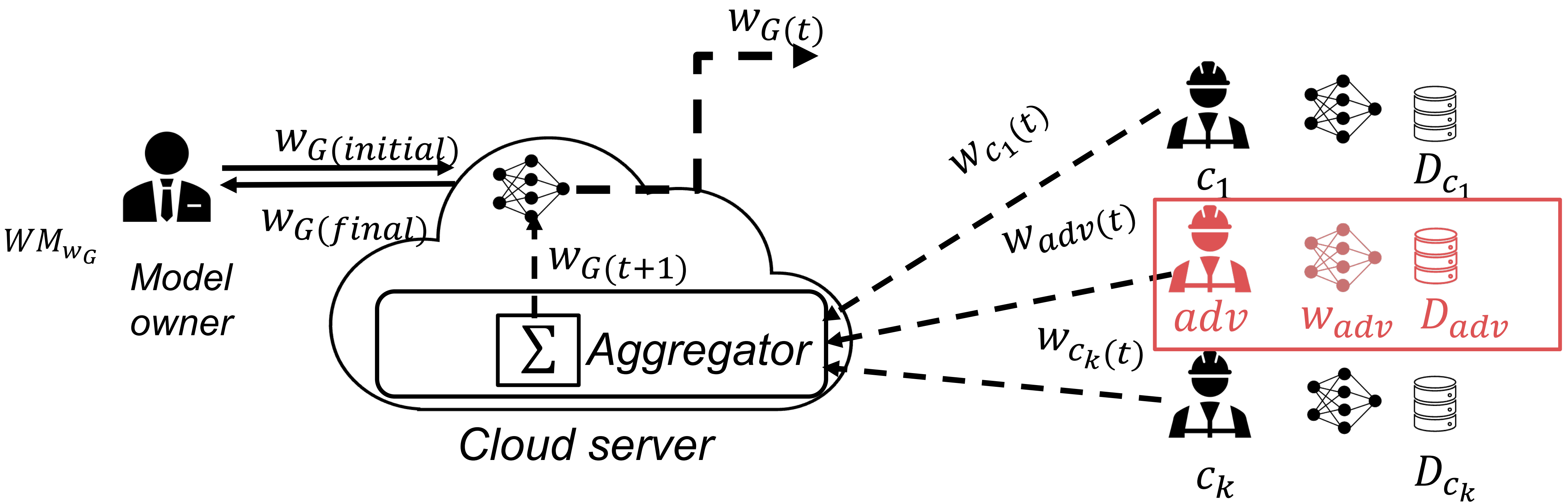
- **Current watermarking solutions are not suitable:**
  - Both training and the dataset is distributed
  - Model owner has no access to training data
  - Model owner can not distribute its watermarks WM_{w_G} to clients



Adversary Model

Adversary

- Malicious client
- Goal: Obtain a local model with the same performance of global model and evade detection of ownership demonstration
  - $(Acc(w_{adv}, D_{test})) \approx Acc(w_{G(final)}, D_{test})),\ VERIFY(w_{adv},\ WM_{w_G}) \rightarrow False$
- Capability: access to training data D_{adv}, global model $w_{G(t)}$ and local models $w_{adv(t)}$

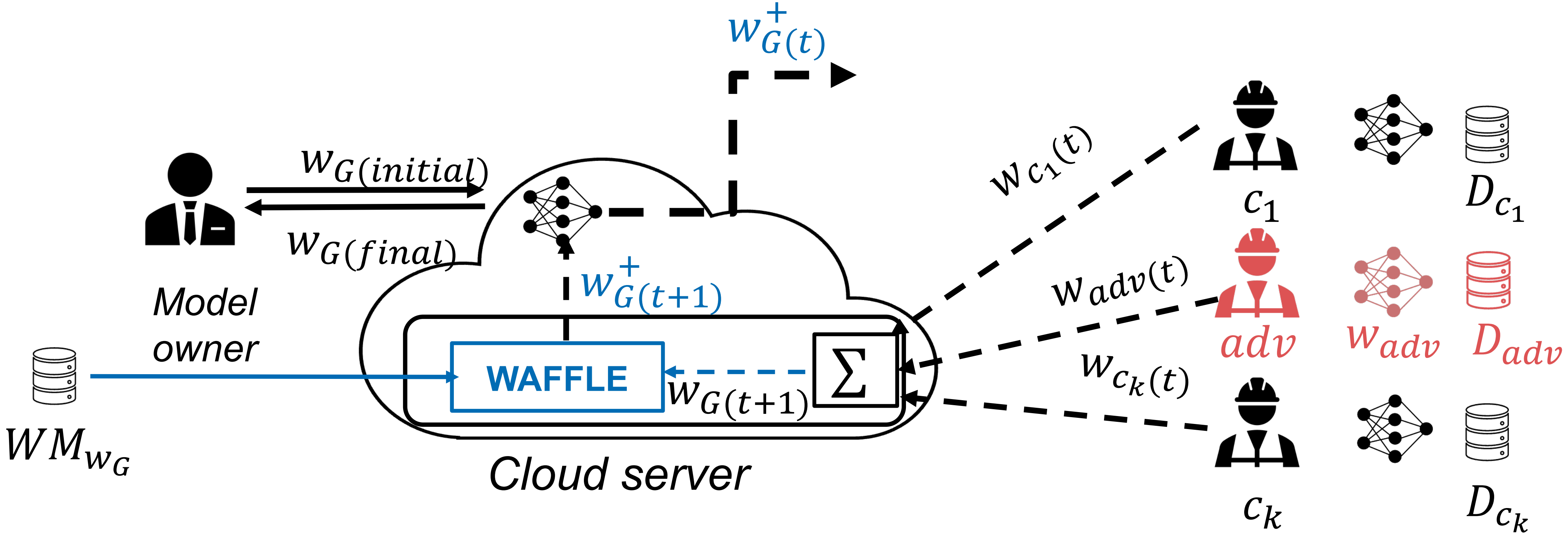


WAFFLE Procedure[2]

First solution for addressing the ownership problem in federated learning.

Executed by the secure aggregator.

Makes no modification to client operations or secure aggregation.



[2] Tekgul, B. G. A., et al. *"WAFFLE: Watermarking in Federated Learning."* SRDS'21 (https://arxiv.org/abs/2008.07298)

Evaluation : Experimental Setup

Datasets and DNN Models:

- MNIST handwritten digit dataset, CIFAR10 general classification dataset (10 classes)
- 5-layer convolutional network, VGG Imagenet model

Federated Learning:

- Federated Averaging[3] as aggregation algorithm, local training with SGD
- 100 total clients, 10 randomly selected clients joins training in each round
- 4 baselines: {total number of local passes $\boldsymbol{E_c}$, Number of aggregation rounds $\boldsymbol{E_a}$}
- Size of the watermark set: 100

Watermark is successfully embedded when:

- $Acc(w_{adv(t)}, WM_{w_G}) \geq T_{acc} = 47\%$[4] for a confidence $< 1 - 2^{-64}$ and
- $Acc(w_{adv(t)}, D_{test}) - Acc\left(w^{+}_{adv(t)}, D_{test}\right) \geq 5 \text{ pp}$

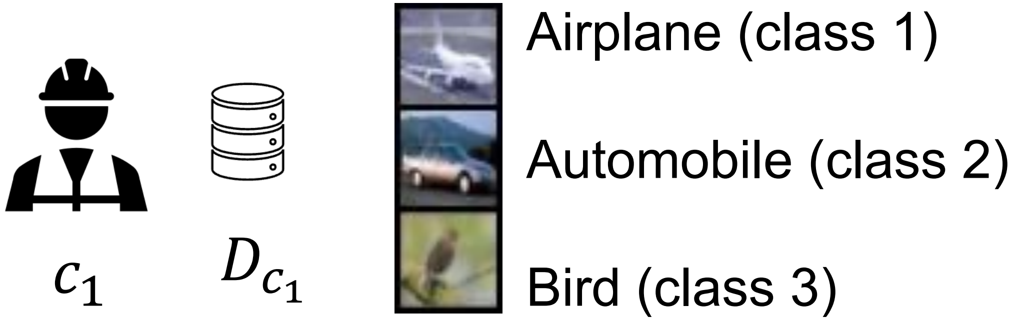
[3] McMahan Brendan et al. "Communication-efficient learning of deep networks from decentralized data." PMLR'17. (http://proceedings.mlr.press/v54/mcmahan17a.html)

[4] Szyller, Sebastian et al. "DAWN: Dynamic Adversarial Watermarking of Neural Networks." ACMMM'21 (https://arxiv.org/abs/1906.00830)

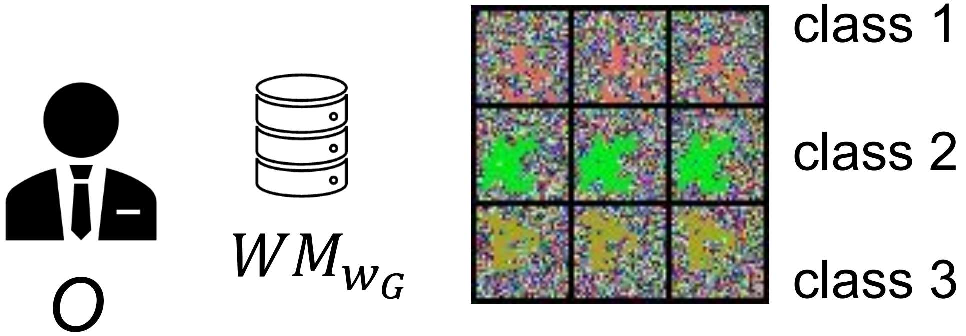
WAFFLEPATTERN

Novel data-independent method to generate watermarks for DNN image classification

- Gaussian noise as background
  - Negligible effect on main task accuracy
- Class specific structured pattern as foreground
  - Easy to learn, does not increase aggregation rounds



Training set



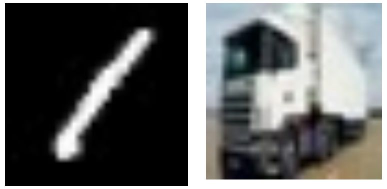
WAFFLEPATTERN

Evaluation

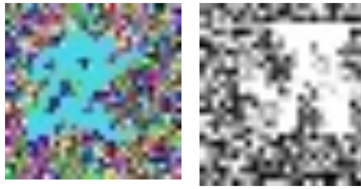
WAFFLEPATTERN:

- is robust to post-processing watermark removal techniques
  - Fine-tuning and pruning, if less than 40% of clients are malicious
  - Neural Cleanse[7], if less than 10% of clients are malicious
- does not decrease the test accuracy of federated learning models (≈ 0.22%)
- imposes no additonal aggregation round (zero communication overhead)
  - and low computational overhead (%3.02)
- Clients with non-IID datasets can not evade the verification
  without sacrificing model performance

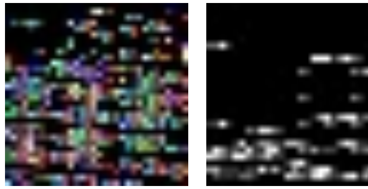
Training set



Trigger set



Neural Cleanse Reversed trigger set

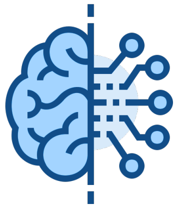


[7] *Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." S&P 19*
(https://ieeexplore.ieee.org/abstract/document/8835365)

Different Ownership Verification Mechanisms

Ownership Verification in a Nutshell

What are the strengths/shortcomings of different ownership verification methods



Ownership verification of ML models

DATASET INFERENCE[1]

More robust than watermarking, but collides with privacy preserving learning

WATERMARKING

- White box theft
- Model stealing

Easy to deploy but brittle

FINGERPRINTING

- Conferrable adv. Examples[2]
- Passport layers[3]

More robust than watermarking, but hard to deploy

[1] Maini, Pratyush, et al. *"Dataset Inference Ownership Resolution."* ICLR 2021.
[2] Lukas, Nils et al. *"Deep neural network fingerprinting by conferrable adversarial examples."* ICLR 2021.
[3] Fan, Lixin et al. *" Rethinking Deep Neural Network Ownership Verification: Embedding Passports to Defeat Ambiguity Attacks".* NeurIPS 2019.

Takeaways

Is model confidentiality important? Yes

models constitute business advantage to model owners

Different ownership verification methods

Can deter extraction attacks

Have intrinsic limitations

Copyright law

Absence of intellectual property protection

Terms of Service or other contractual agreements

More on our security + ML research at https://ssg.aalto.fi/research/projects/mlsec/model-extraction/