False Claims Against Model Ownership Resolution

Sebastian Szyller
sebszyller.com
@sebszyller

(Joint work with Rui Zhang, Jian Liu, Kui Ren and N. Asokan)

Model theft is an important concern

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

Defending against model theft

We can try to:

- prevent (or slow down) model theft, including model extraction or
- detect it

But appears to be infeasible against strong but realistic adversaries[1]

Or deter the attacker by providing the means for model ownership resolution (MOR):

- fingerprinting
- watermarking

promising but many MOR schemes so far have various caveats and vulnerabilities[2,3,4]

[1] Atli et al. - *Extraction of Complex DNN Models: Real Threat or Boogeyman?* AAAI-EDSML 2020 (https://arxiv.org/abs/1910.05429)
[2] Lukas et al. – *Sok: How Robust is Image Classification Deep Neural Network Watermarking?* IEEE S&P 2022 (https://arxiv.org/abs/2108.04974)
[3] Shafieinejad et al. - *On the Robustness of Backdoor-based Watermarking Schemes,* IHMS 2021 (https://arxiv.org/abs/1906.07745)
[4] Szyller et al. – *On the Robustness of Dataset Inference* (https://arxiv.org/abs/2210.13631)

MOR generalization

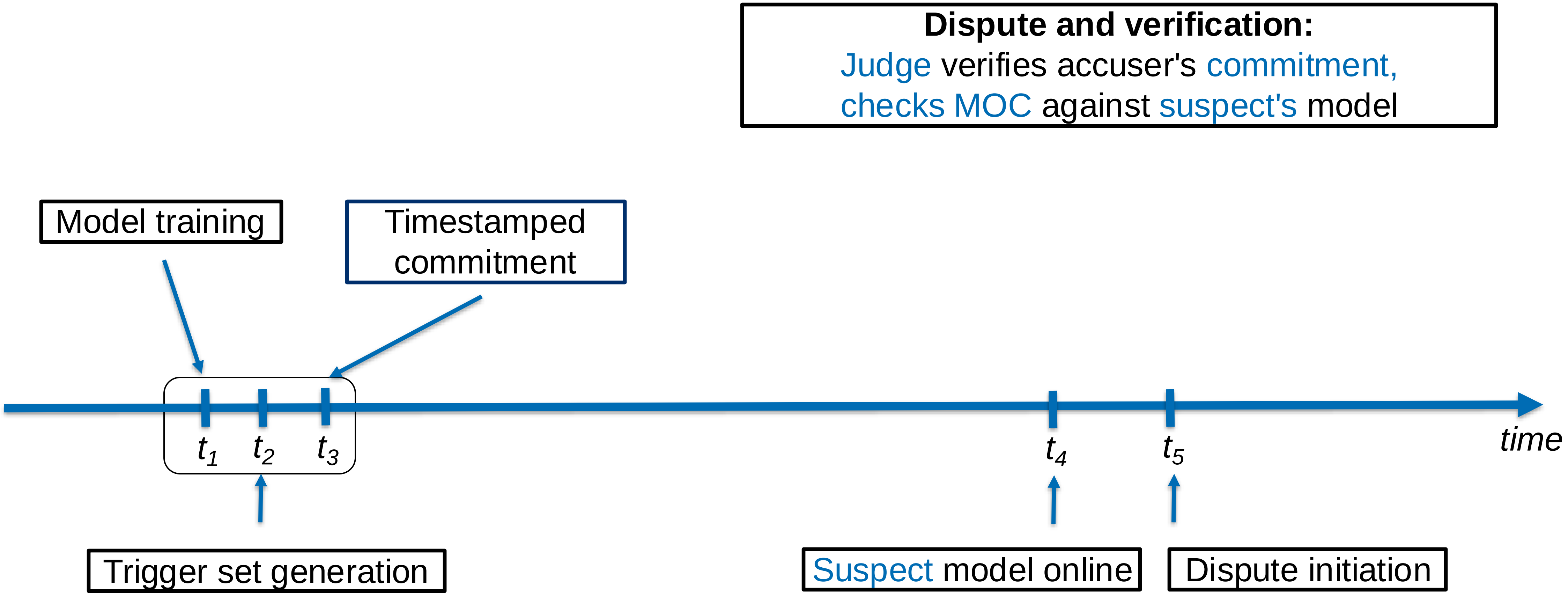
Claim generation:

- model owner (potential accuser) generates "model ownership claim" (MOC)
  - includes trigger sets: e.g., watermarks or fingerprints
  - stolen vs. independent models likely to behave differently on input from trigger set
  - obtains a secure timestamp on trigger set (+ model + other data) commitment

Claim verification:

- accuser initiates MOR against a suspect by sending MOC to a judge
- judge verifies timestamped MOC + interacts with both models to resolve ownership
  - decides if suspect has stolen accuser's model

MOR process



Dispute and verification:
Judge verifies accuser's commitment,
checks MOC against suspect's model
Model training
Timestamped commitment
t_1 t_2 t_3
t_4
t_5
time
Trigger set generation
Suspect model online
Dispute initiation

Robustness of MOR schemes

MOR schemes must be robust against two types of attackers.

Malicious suspect:
- tries to evade verification (e.g., pruning, fine-tuning, noising)

Malicious accuser:
- tries to frame an independent model owner
- (secure) timestamping (watermark/fingerprint and model) is the only defense in prior work

So far, research has focused on robustness against malicious suspects

False claims against MOR schemes

We show how malicious accusers can make false claims against independent models:

- adversary deviates from watermark/fingerprint generation procedure
  - E.g., via transferrable adversarial examples
- but still subject to specified verification procedure

Our contributions:

- formalize the notion of false claims against MOR schemes
- provide a generalization of MOR schemes
- demonstrate effective false claim attacks
- discuss potential countermeasures

MOR instantiations

Watermarking:

- **watermarking by backdooring**[3]
  - out-of-distribution backdoor embedded during training
- **adversarial watermarking**[4]
  - flip labels for a subset of queries during inference, designed to deter model extraction

Fingerprinting:

- **model fingerprinting**[5]
  - conferrable adversarial examples, transfer only to stolen models
- **Dataset Inference**[6]
  - stolen models likely to have similar decision boundaries

[3] Adi et al. – *Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring*, USENIX 2018 (https://arxiv.org/abs/1802.04633)
[4] Szyller et al. – *DAWN: Dynamic Adversarial Watermarking of Neural Networks*, ACM MM 2021 (https://arxiv.org/abs/1906.00830)
[5] Lukas et al. – *Deep Neural Network Fingerprinting by Conferrable Adversarial Examples*, ICLR 2021 (https://arxiv.org/abs/1912.00888)
[6] Maini et al. – *Dataset Inference: Ownership Resolution in Machine Learning*, ICLR 2021 (https://arxiv.org/abs/2104.10706)

Watermarking by backdooring[3]

Claim generation:

- choose some out-of-distribution samples as watermark
  - assign incorrect labels
- train using the watermark alongside your normal training data (or fine tune)
  - model memorizes watermark
- obtain secure timestamp on commitment of model and watermark

[3] Adi et al. – *Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring*, USENIX 2018 (https://arxiv.org/abs/1802.04633)

Watermarking by backdooring[3]: verification

Claim verification:

- query suspect model using watermark
- compare predictions to the assigned (incorrect) labels:
  - many matching / high WM accuracy → stolen
  - a few matching / low WM accuracy → not stolen
- check commitment and timestamp

[3] Adi et al. – *Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring*, USENIX 2018 (https://arxiv.org/abs/1802.04633)

DAWN[4]

Claim generation:

- clients submit queries
- pseudo-randomly select a fraction of queries as watermark (per-client)
- each watermark consists of pairs of inputs with pseudo-randomly flipped labels
- obtain secure timestamp on commitment of model and watermark
- adversary embeds watermark while training their surrogate models

[4] Szyller et al. – *DAWN: Dynamic Adversarial Watermarking of Neural Networks*, ACM MM 2021 (https://arxiv.org/abs/1906.00830)

DAWN[4]: verification

Claim verification:

- query suspect model using watermark
- compare predictions to flipped (incorrect) labels:
  - many matching / high WM accuracy → stolen
  - a few matching / low WM accuracy → not stolen
- check commitment and timestamp

[4] Szyller et al. – *DAWN: Dynamic Adversarial Watermarking of Neural Networks*, ACM MM 2021 (https://arxiv.org/abs/1906.00830)

Conferrable adversarial examples[5]

Claim generation:

- extract your own model many times: many surrogate models
- train many independent reference models
- generate conferrable adversarial examples:
  - must transfer from your model to surrogate models
  - must not transfer to reference models
- conferrable examples are the fingerprint
- obtain secure timestamp on commitment of model and fingerprint.

[5] Lukas et al. – *Deep Neural Network Fingerprinting by Conferrable Adversarial Examples*, ICLR 2021 (https://arxiv.org/abs/1912.00888)

Conferrable adversarial examples[5]: verification

Claim verification:

- query suspect model using fingerprint
- compare suspect's predictions to the ground truth:
  - suspect is fooled / gives incorrect prediction → stolen
  - suspect is not fooled / gives correct predictions → not stolen
- check commitment and timestamp

[5] Lukas et al. – *Deep Neural Network Fingerprinting by Conferrable Adversarial Examples*, ICLR 2021 (https://arxiv.org/abs/1912.00888)

Dataset Inference[6]

Claim generation:

- obtain embeddings for your private training data and public data (using your model),
- train a distinguisher using embeddings
  - learns to identify models that use your training data vs those that do not
- outputs confidence scores to both sets of embeddings
- distributions of confidence scores must be distinguishable (hypothesis test)
- obtain secure timestamp on commitment of model and distinguisher+data

[6] Maini et al. – *Dataset Inference: Ownership Resolution in Machine Learning*, ICLR 2021 (https://arxiv.org/abs/2104.10706)

Dataset Inference[6]: verification

Claim verification:

- query suspect model to obtain embeddings
- get confidence scores using distinguisher
- compare distributions:
  - distinguishable → stolen
  - indistinguishable → not stolen
- check commitment and timestamp

[6] Maini et al. – *Dataset Inference: Ownership Resolution in Machine Learning*, ICLR 2021 (https://arxiv.org/abs/2104.10706)

Inducing successful false claims

Core idea: Accuser deviates from specified MOC generation procedure

For most schemes

- generate transferable adversarial examples and register them as false trigger set

For DI

- false positives occur naturally when training data distributions are similar[7]
- generate false “private” data that fits distribution of independent training data
- obtain secure timestamp on false private data and resulting false distinguisher

[7] Szyller et al. – *On the Robustness of Dataset Inference,* TMLR 2023 (https://openreview.net/forum?id=LKz5SqIXPJ)

Watermarking by backdooring[3]

Claim generation:

- choose some out-of-distribution samples as watermark
  - assigned with incorrect labels
- train using the watermark alongside your normal training data (or fine tune)
  - model memorizes watermark
- obtain secure timestamp on commitment of model and watermark

[3] Adi et al. – *Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring*, USENIX 2018 (https://arxiv.org/abs/1802.04633)

Watermarking by backdooring[3]: false claim

Claim generation:

- choose some out-of-distribution samples as watermark
  - assigned with incorrect labels
- train using the watermark alongside your normal training data (or fine tune)
  - model memorizes watermark
- obtain secure timestamp on commitment of model and watermark

[3] Adi et al. – *Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring*, USENIX 2018 (https://arxiv.org/abs/1802.04633)

Watermarking by backdooring[3]: false claim

False claim generation:

- choose some out-of-distribution samples as false watermark
- perturb these samples to craft transferable adversarial examples
- obtain secure timestamp on commitment of model and false watermark

[3] Adi et al. – *Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring*, USENIX 2018 (https://arxiv.org/abs/1802.04633)

DAWN[4]

Claim generation:

- clients submit queries
- pseudo-randomly select a fraction of queries as watermark (per-client)
- each watermark consists of pairs of inputs with pseudo-randomly flipped labels
- obtain timestamp on commitment of model and watermark
- adversary embeds watermark while training their surrogate models

[4] Szyller et al. – *DAWN: Dynamic Adversarial Watermarking of Neural Networks*, ACM MM 2021 (https://arxiv.org/abs/1906.00830)

DAWN[4]: false claim

Claim generation:

- clients submit queries
- pseudo-randomly select a fraction of queries as watermark (per-client)
- each watermark consists of pairs of inputs with pseudo-randomly flipped labels
- obtain secure timestamp on commitment of model and watermark
- adversary embeds the watermark while training their surrogate models

[4] Szyller et al. – *DAWN: Dynamic Adversarial Watermarking of Neural Networks*, ACM MM 2021 (https://arxiv.org/abs/1906.00830)

DAWN[4]: false claim

False claim generation:

- clients submit queries
- pseudo-randomly select a fraction of the queries for the false watermark

- perturb each chosen query to craft targeted transferable adversarial examples
  - labels need to match the pseudo-random flip

- obtain secure timestamp on commitment of model and false watermark

[4] Szyller et al. – *DAWN: Dynamic Adversarial Watermarking of Neural Networks*, ACM MM 2021 (https://arxiv.org/abs/1906.00830)

Conferrable adversarial examples[5]

Claim generation:

- extract your own model many times: many surrogate models
- train many reference models
- generate conferrable adversarial examples:
  - must transfer from your model to surrogate models
  - must not transfer to reference models
- conferrable examples are the fingerprint
- obtain secure timestamp on commitment of model and fingerprint

[5] Lukas et al. – *Deep Neural Network Fingerprinting by Conferrable Adversarial Examples*, ICLR 2021 (https://arxiv.org/abs/1912.00888)

Conferrable adversarial examples[5]: false claim

Claim generation:

- extract your own model many times: many surrogate models
- train many reference models
- generate conferrable adversarial examples:
  - must transfer from your model to surrogate models
  - must not transfer to reference models
- conferrable examples are the fingerprint
- obtain secure timestamp on commitment of model and fingerprint

[5] Lukas et al. – *Deep Neural Network Fingerprinting by Conferrable Adversarial Examples*, ICLR 2021 (https://arxiv.org/abs/1912.00888)

Conferrable adversarial examples[5]: false claim

False claim generation:

- (optional) extract your own model many times: to strengthen transferability

- ignore any reference models
- craft transferable adversarial examples
- transferable adversarial examples are the false fingerprint

- obtain secure timestamp on commitment of model and false fingerprint

[5] Lukas et al. – *Deep Neural Network Fingerprinting by Conferrable Adversarial Examples*, ICLR 2021 (https://arxiv.org/abs/1912.00888)

Dataset Inference[6]

Claim generation:

- obtain embeddings for your private training data and public data (using your model),
- train a distinguisher using embeddings
  - learns to identify models that use your training data vs those that do not
  - outputs confidence scores to both sets of embeddings
- distributions of confidence scores must be distinguishable (hypothesis test)
- obtain secure timestamp on commitment of model and distinguisher+data

[6] Maini et al. – *Dataset Inference: Ownership Resolution in Machine Learning*, ICLR 2021 (https://arxiv.org/abs/2104.10706)

Dataset Inference[6]: false claim

Claim generation:

- obtain embeddings for your private training data and public data (using your model),
- train a distinguisher using embeddings
  - learns to identify models that use your training data vs those that do not
  - outputs confidence scores to both sets of embeddings
- distributions of confidence scores must be distinguishable (hypothesis test)
- obtain secure timestamp on commitment of model and distinguisher+data

[6] Maini et al. – *Dataset Inference: Ownership Resolution in Machine Learning*, ICLR 2021 (https://arxiv.org/abs/2104.10706)

Dataset Inference[6]: false claim

False claim generation:

- obtain embeddings for public data (using your model)

- sample false “private” data, perturb to generate large prediction margins (on your model) (these will transfer to independent models)
- train a false distinguisher using both sets of embeddings (outputs fake confidence scores)
- distributions now distinguishable for all independent models (hypothesis test)

- obtain secure timestamp on commitment of model and false distinguisher+data

[6] Maini et al. – *Dataset Inference: Ownership Resolution in Machine Learning*, ICLR 2021 (https://arxiv.org/abs/2104.10706)

Evaluation

Our attacks are effective:

- evaluated against Adi et al., DAWN, Lukas et al., DI
  - using CIFAR10, ImageNet, CelebA (Amazon Rekognition API)
- also applicable to others that follow our generalization

Attack efficacy compared to three thresholds (T):

- independent: judge trains independent models and picks the highest T
  - easy for false claims, difficult to evade detection
- extracted: judge derives extracted models and picks the lowest T
  - easy to evade detection, difficult for false claims
- mixed: average of independent and extracted models
  - realistic for actual deployments

For DI, naturally occurring FPs[7] make “extracted” threshold > “mixed” threshold!

[7] Szyller et al. – *On the Robustness of Dataset Inference,* TMLR 2023 (https://openreview.net/forum?id=LKz5SqIXPJ

Evaluation: CIFAR10

		Backdooring	DAWN	Conferrable	DI
T	independent	10.0	1.0	28.0	90.0
	mixed	29.0	38.5	57.5	81.4
	extracted	48.0	76.0	87.0	72.8
Suspect MOR accuracy	diff. arch. & diff. data	<u>**94.3**</u>	**69.3**	<u>**94.3**</u>	<u>**100.0**</u>
	same arch. & diff. data	<u>**98.0**</u>	<u>**100.0**</u>	<u>**98.0**</u>	<u>**99.1**</u>
	same arch. & same data	<u>**99.0**</u>	<u>**78.3**</u>	<u>**99.0**</u>	<u>**98.6**</u>

False claim accuracy:

- **bold:** higher than mixed T (realistic)
- <u>**underlined**</u>**:** higher than extracted T (difficult for false claims)

For DI, naturally occurring FPs[7] lead to a different threshold order “extracted” < “mixed” < “independent”!

[7] Szyller et al. – *On the Robustness of Dataset Inference,* TMLR 2023 (https://openreview.net/forum?id=LKz5SqIXPJ

Evaluation: ImageNet

		Backdooring	DAWN	Conferrable	DI
T	independent	15.0	3.0	14.0	76.5
	mixed	23.5	42.5	30.0	69.6
	extracted	32.0	82.0	46.0	62.6
Suspect MOR accuracy	diff. arch. & diff. data	**<u>72.6</u>**	**<u>87.6</u>**	**<u>72.6</u>**	**<u>100.0</u>**
	same arch. & diff. data	**<u>93.7</u>**	**<u>97.0</u>**	**<u>93.7</u>**	**<u>100.0</u>**
	same arch. & same data	**<u>84.6</u>**	**<u>89.0</u>**	**<u>84.6</u>**	**<u>100.0</u>**

False claim accuracy:

- **bold:** higher than mixed T (realistic)
- **<u>underlined</u>:** higher than extracted T (difficult for false claims)

For DI, naturally occurring FPs[7] lead to a different threshold order “extracted” < “mixed” < “independent”!

[7] Szyller et al. – *On the Robustness of Dataset Inference,* TMLR 2023 (https://openreview.net/forum?id=LKz5SqIXPJ

Evaluation: CelebA (Amazon Rekognition API)

		Backdooring	DAWN	Conferrable	DI
T	independent	25.7	7.0	21.0	20.0
	mixed	42.4	26.0	28.5	14.1
	extracted	59.0	45.0	36.0	8.2
Suspect MOR accuracy	diff. arch. & diff. data (Amazon Rekognition API)	**68.4**	**68.0**	**68.4**	**99.9**

False claim accuracy:
- **bold:** higher than mixed T (realistic)
- **underlined:** higher than extracted T (difficult for false claims)

For DI, naturally occurring FPs[7] lead to a different threshold order “extracted” < “mixed” < “independent”!

[7] Szyller et al. – *On the Robustness of Dataset Inference,* TMLR 2023 (https://openreview.net/forum?id=LKz5SqIXPJ

Countermeasures 1/4

False claims undermine confidence in all MOR schemes.
How to prevent them?

Approach 1: Judge-verified trigger sets I

- use verifiable computation (VC): ensure that trigger set was generated correctly
- does not capture watermark selection: false claims still possible
- applicable to fingerprinting schemes
  - expensive: must include model training, otherwise still unsafe
  - not applicable to DI: accuser can manipulate their training data

Countermeasures 2/4

False claims undermine confidence in all MOR schemes.
How to prevent them?

Approach 2: Judge-verified trigger sets II

- judge trains multiple independent models: rejects trigger sets that flag them as stolen
- effective for all schemes
- costly for judge: but amortizable, and rare (only when dispute arises)
- needs appropriate training data
- accuser can try to extract or evade the independent models
  - each MOR invocation must be expensive to deter repeated attempts
  - little impact on legitimate MOR invocations

Countermeasures 3/4

False claims undermine confidence in all MOR schemes.
How to prevent them?

Approach 3: Judge-generated trigger sets

- judge generates all trigger sets: all subsequent claims must use these
- effective for several schemes
  - not applicable to DAWN: clients choose their queries
  - not applicable to DI: data/model can be manipulated before MOC generation
- judge becomes a bottleneck if judge must be involved even if there is no dispute
  - for fingerprinting schemes trigger set generation can be deferred until dispute

Countermeasures 4/4

False claims undermine confidence in all MOR schemes.
How to prevent them?

Approach 4: defenses against transferable adversarial examples

- adversarial training: likely effective but can incur accuracy loss
- adversarial purification: expensive and too slow for real-time prediction
- detection of adversarial examples (e.g., by judge): open research problem

Approach 5 (DAWN-only): signing queries

- require all clients to sign their queries
- judge verifies that queries were not manipulated
- effective if clients do not collude with accuser (clients can be punished for stolen models)

Conclusion

Model theft is an important concern.

MOR schemes have varying degree of robustness

All current MOR schemes are vulnerable to false claims:
- **possible to accuse/frame independent model owners**

Countermeasures may be costly

Do efficient scheme-specific countermeasures exist?

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