A SURVEY ON MODEL WATERMARKING NEURAL NETWORKS

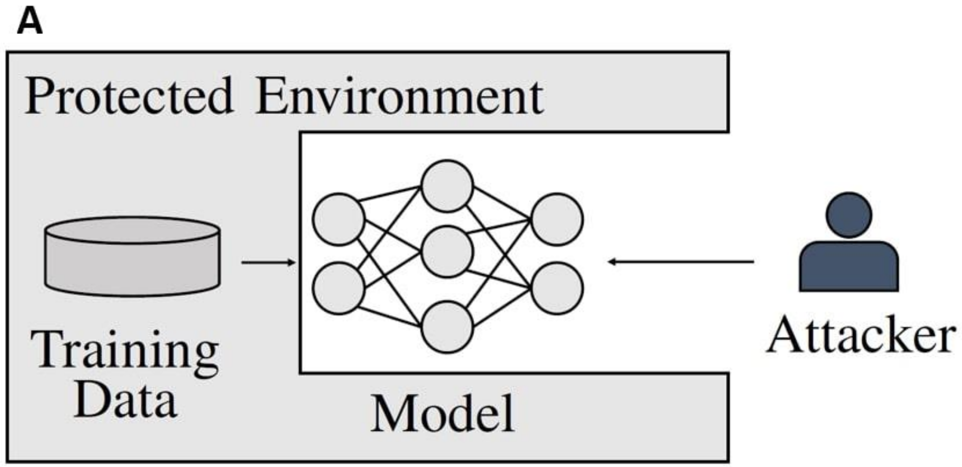
Franziska Boenisch, Fraunhofer AISEC, Germany

Presented at the *IJCAI 2021* Workshop *"Toward Intellectual Property Protection on Deep Learning as a Services"*, August 21st 2021.

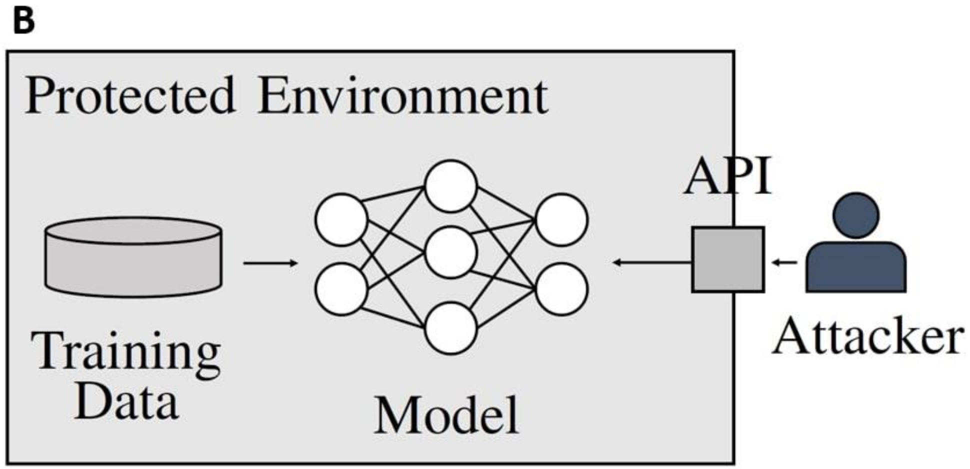
AGENDA

- Background on Model Stealing
- Watermark Requirements
- Threat Space and Attacks
- Existing Watermarking Methods
- Outlook on Research Challenges and Perspectives

Model Stealing

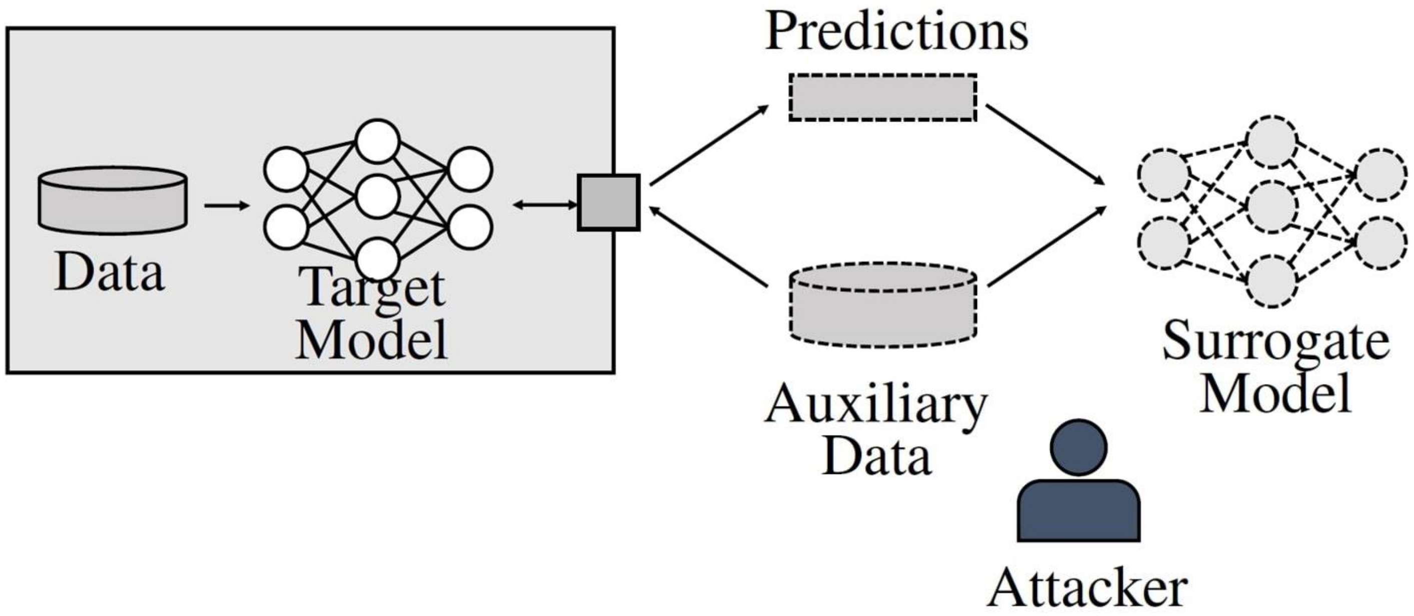


White-box scenario



Black-box scenario

Model Stealing

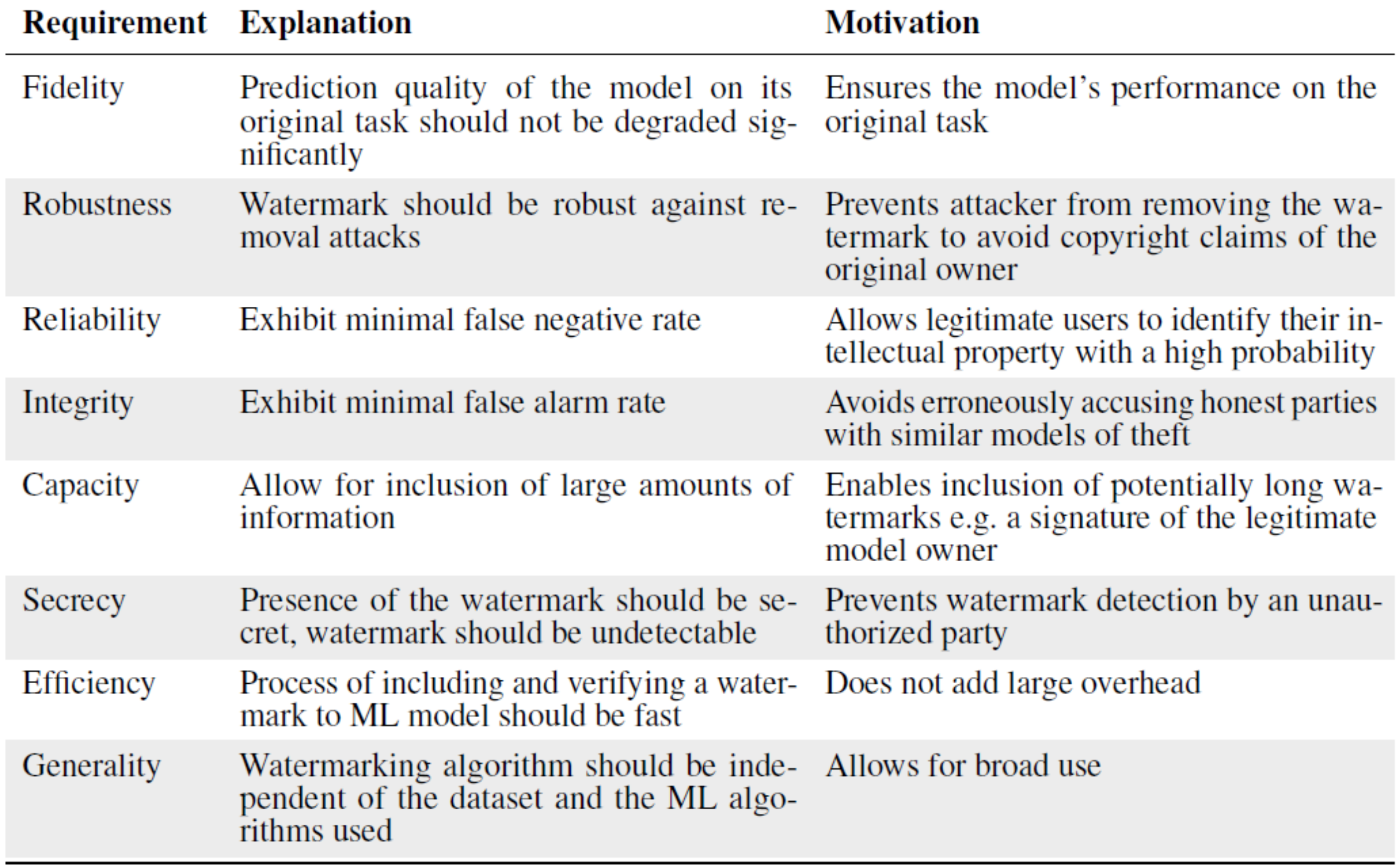


Black-box scenario

Fidelity Extraction

Task Extraction

Requirements for Watermarks



Requirement	Explanation	Motivation
Fidelity	Prediction quality of the model on its original task should not be degraded significantly	Ensures the model's performance on the original task
Robustness	Watermark should be robust against removal attacks	Prevents attacker from removing the watermark to avoid copyright claims of the original owner
Reliability	Exhibit minimal false negative rate	Allows legitimate users to identify their intellectual property with a high probability
Integrity	Exhibit minimal false alarm rate	Avoids erroneously accusing honest parties with similar models of theft
Capacity	Allow for inclusion of large amounts of information	Enables inclusion of potentially long watermarks e.g. a signature of the legitimate model owner
Secrecy	Presence of the watermark should be secret, watermark should be undetectable	Prevents watermark detection by an unauthorized party
Efficiency	Process of including and verifying a watermark to ML model should be fast	Does not add large overhead
Generality	Watermarking algorithm should be independent of the dataset and the ML algorithms used	Allows for broad use

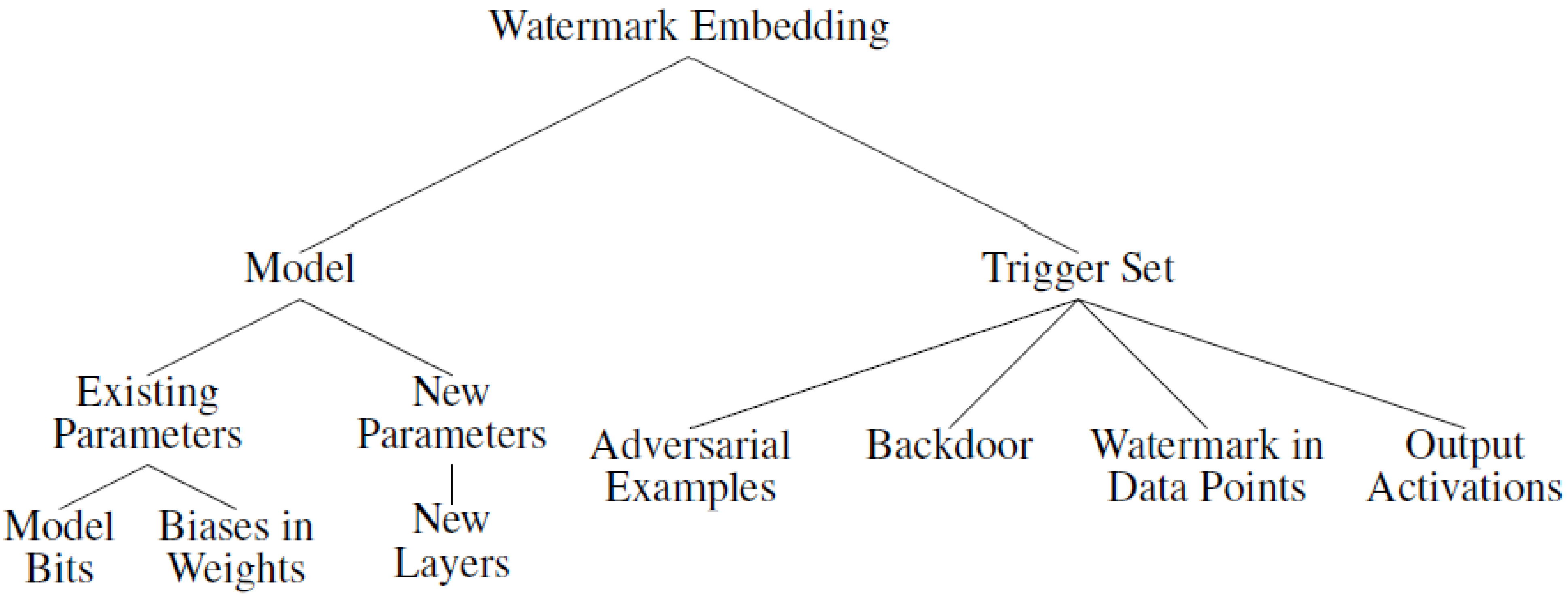
+ Uniqueness

+ Scalability

Threats to Watermarks

1) Watermark Detection
2) Watermark Suppression
3) Watermark Forging
4) Watermark Overwriting
5) Watermark Removal

Watermarking methods



Watermark Embedding

Model

Trigger Set

Existing Parameters

New Parameters

Model Bits

Biases in Weights

New Layers

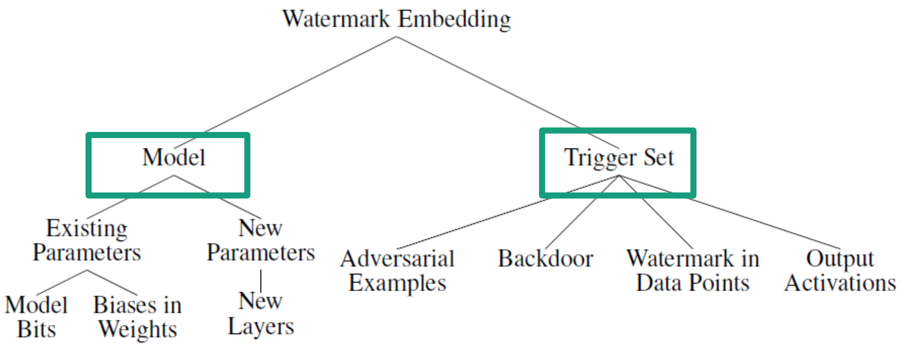
Adversarial Examples

Backdoor

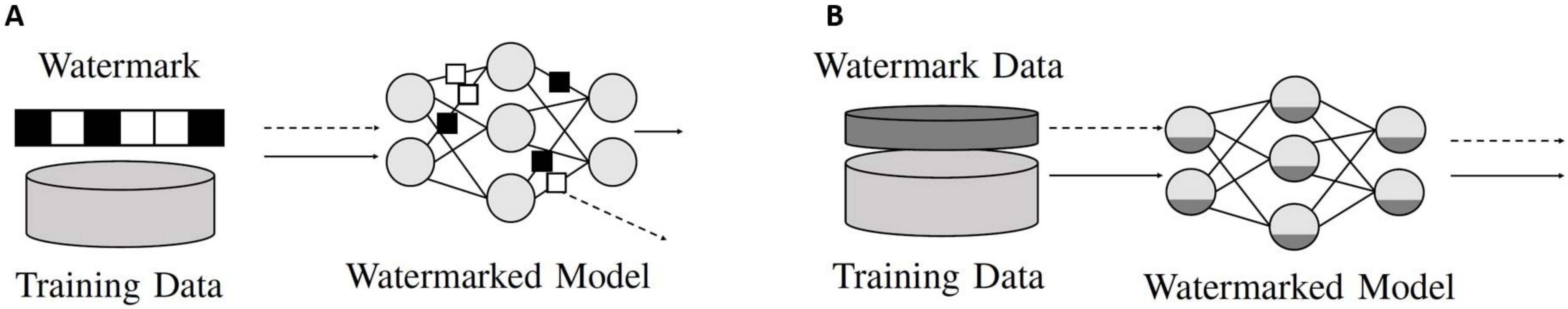
Watermark in Data Points

Output Activations

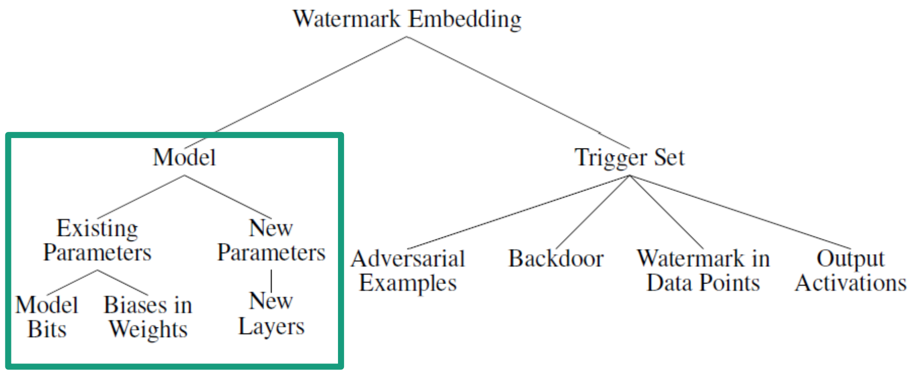
Watermarking methods



Watermark insertion during or after training.



Watermarking methods

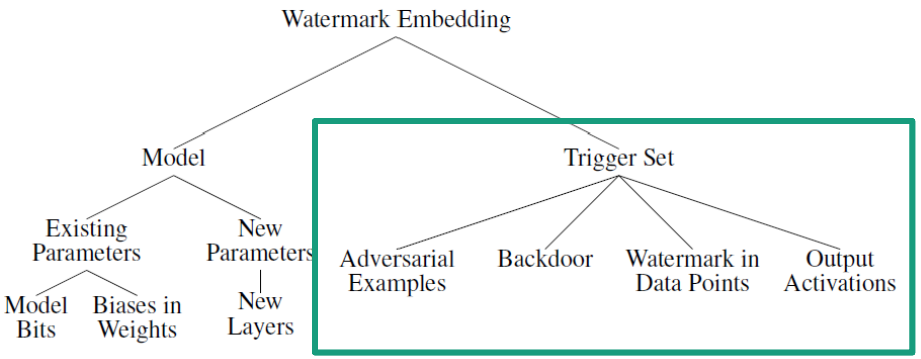


Watermark insertion into model.

Notes and Considerations:

- Watermark detection (due to noticeable changes)
- Rendering model performance dependent on parameters / layers / biases of watermark
- Usually possibility to include several bits of information
- Verification scenario: white-box?

Watermarking methods

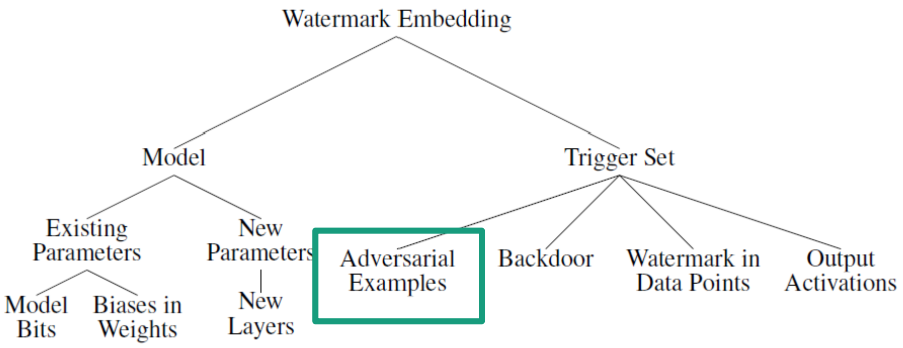


Watermark insertion through trigger set.

Notes and Considerations:

- Two different data distributions → model learning two independent tasks
- Adequate choice of verification threshold
- Public verification and revealing the trigger set

Watermarking methods

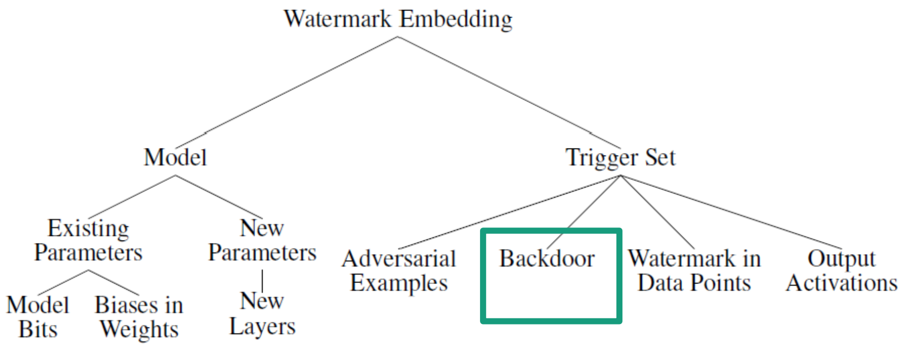


Watermark insertion based on Adversarial Samples.

Notes and Considerations:

- Robustness concerns due to adversarial retraining
- Integrity concerns due to adversarial transferability
- No link to legitimate owner

Watermarking methods

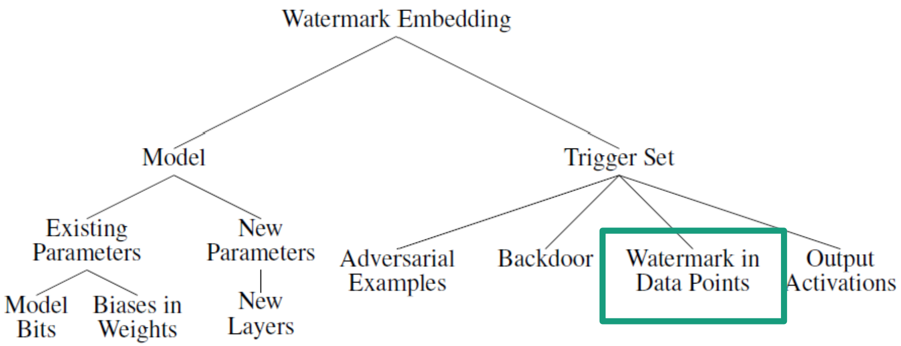


Watermark insertion based on Backdoors.

Notes and Considerations:

- Training the model on a separate task due to over-parametrization.

Watermarking methods

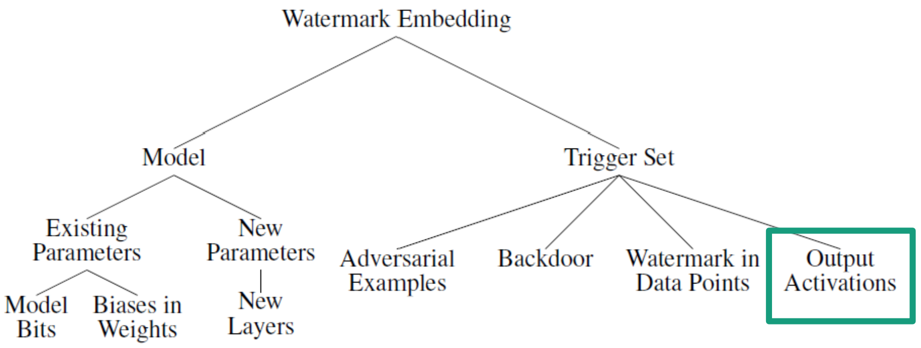


Watermark insertion based specifically altered training data points.

Notes and Considerations:

- Possibility to include owner information, such as signatures
- Watermark samples' distribution different from training data distribution

Watermarking methods



Watermark insertion based on Model Outputs.

Notes and Considerations:

- Include information on legitimate owner
- Low generality

Outlook on Research Challenges and Perspectives

Passive → active protection

Limited ML algorithm cases

Limited data types and amounts

Adaptation in juridical and organizational workflows

franziska.boenisch@aisec.fraunhofer.de
LinkedIn, Twitter, Github: @fraboeni

Questions?

Thank you for your attention!