Security of Neural Networks : Attacks, Defenses and Evaluation methods

Rémi BERNHARD (CEA Tech)
Pierre-Alain MOELLIC (CEA Tech)
Jean-Max DUTERTRE (MSE)

Laboratoire de Sécurité des Architectures et des Systèmes,
Centre CMP, Equipe Commune CEA-Tech Mines Saint-Etienne,
F-13541 Gardanne France

June 24, 2020

Context

Overview

- **Neural networks**: state-of-the art performances in various complex tasks (e.g., image recognition, speech translation)
  $\rightarrow$ Growing use of neural networks
  $\rightarrow$ Growing will to deploy models on embedded systems




- **Adversarial machine learning**:
  - Critical decision systems (health, defense and security, . . .)
  - Autonomous car
- **Privacy** issues

Serious threats require efficient countermeasures

Security of Machine Learning systems

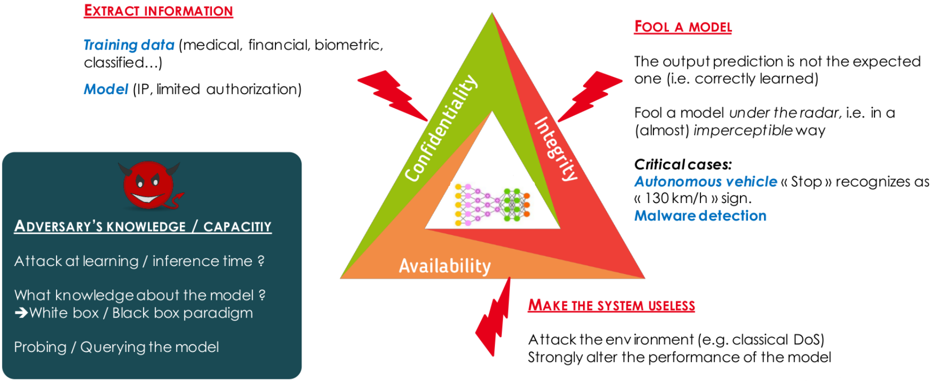
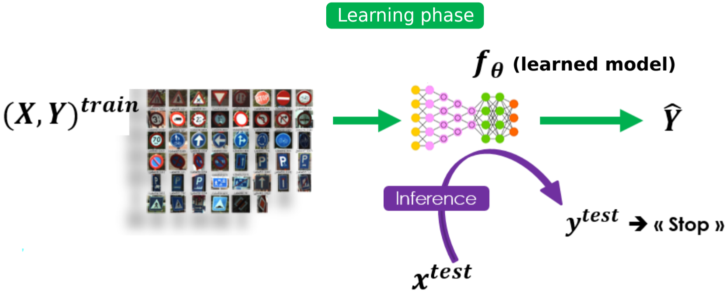
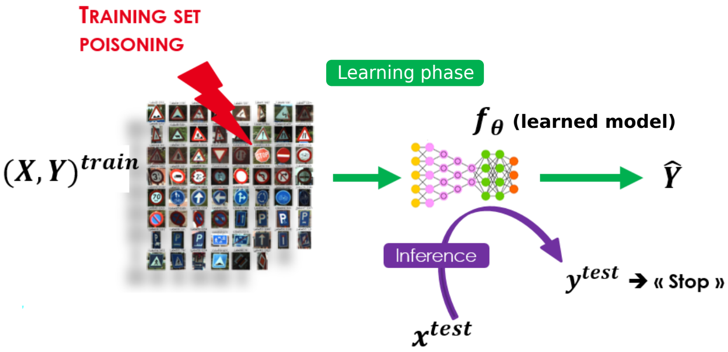


Figure: CIA threat model for a Machine Learning system

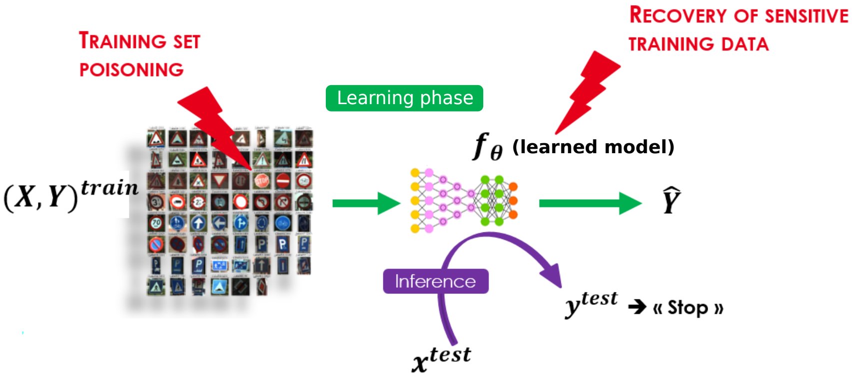
Security of Machine Learning Systems

Striking the ML pipeline





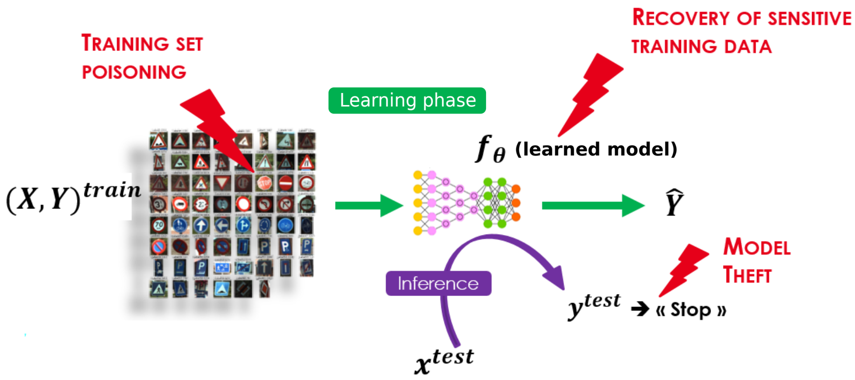
TRAINING SET
POISONING
Learning phase
$(X, Y)^{train}$
f_θ (learned model)
$\hat{Y}$
Inference
y^{test} → « Stop »
x^{test}



TRAINING SET POISONING
RECOVERY OF SENSITIVE TRAINING DATA
Learning phase
f_θ (learned model)
$(X, Y)^{train}$
$\hat{Y}$
Inference
y^{test} → « Stop »
x^{test}

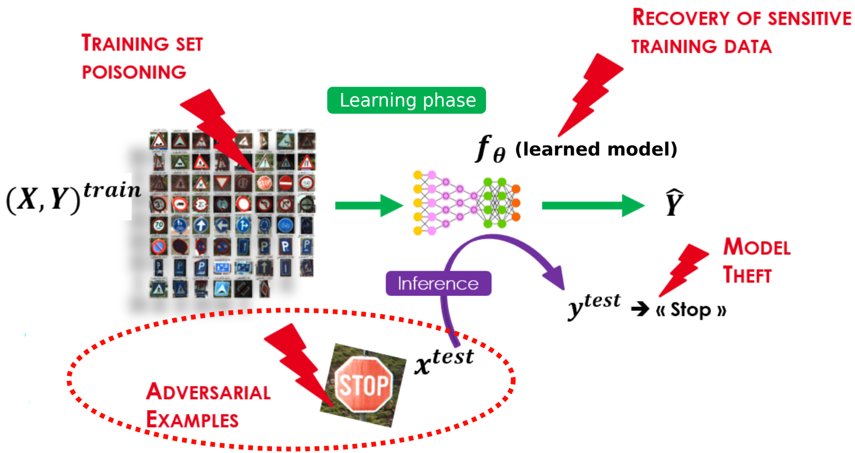
Security of Machine Learning Systems

Striking the ML pipeline



Security of Machine Learning Systems

Striking the ML pipeline



Principle: Craft maliciously modified examples to fool a model.

Adversarial example = Clean example + Adversarial perturbation

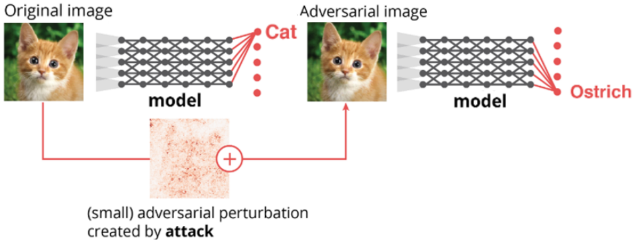


Figure: NIPS 2018 Adversarial Vision Challenge

Threat model:

- **White-box** setting: the adversary has a total access to the target model; He can compute gradients: *gradient-based attacks*
- **Black-box** setting: the adversary has an obstructed access to the target model (score only, label only, etc.). He can't compute gradients: *score/decision based attacks*, or *transferability*

Principle of transferability:
Adversarial examples crafted on a substitute model transfer to the target model.
$\rightarrow$ Powerful tool for an adversary in the black-box setting.

Security of embedded neural networks

$$\left.\begin{array}{l}\textbf{Memory footprint}\text{:} \\ \text{Parameters storage} \\ \\ \\ \textbf{Energy cost}\text{:} \\ \text{Efficient inference methods}\end{array}\right\} \rightarrow \textit{Quantization methods}$$

Quantization-aware training:
Learn a model with quantized weights and/or activation values during the training process.

Issues: Non-differentiability of quantization functions, difficulty of training, ...

How does quantization influence robustness against adversarial example ?

Two Data sets:

- SVHN (73,257/26,032)
- CIFAR10 (50,000/10,000)

Experiences:

- Quantization: Activation and Weight / Weight *quantization*: 1,2,3,4 bits
- Techniques: Courbariaux et al. (2015, 2016), Zhou et al. (2016)
- Various threat models considered

Results:

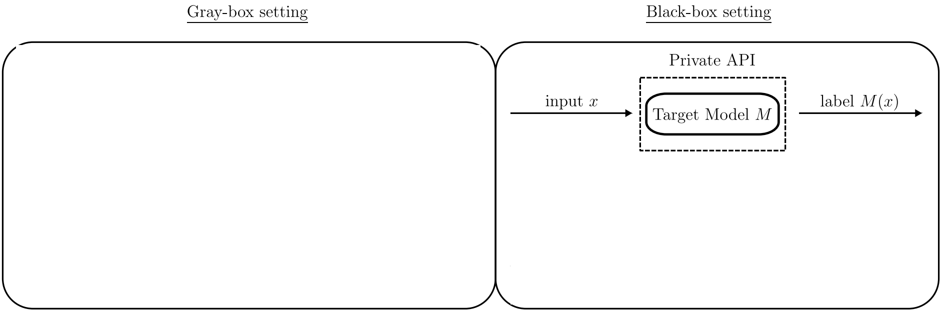
- Detection of some gradient masking issue (false impression of security)
  → Quantization is not a robust *"natural"* defense when facing advanced attacks
- But, interestingly, gradient misalignment issues and *quantization shift phenomenon* cause poor transferability
- This enables to build a defense based on an ensemble of quantized models

Best Paper Award at *IEEE Conference on Cyberworlds, 2019*

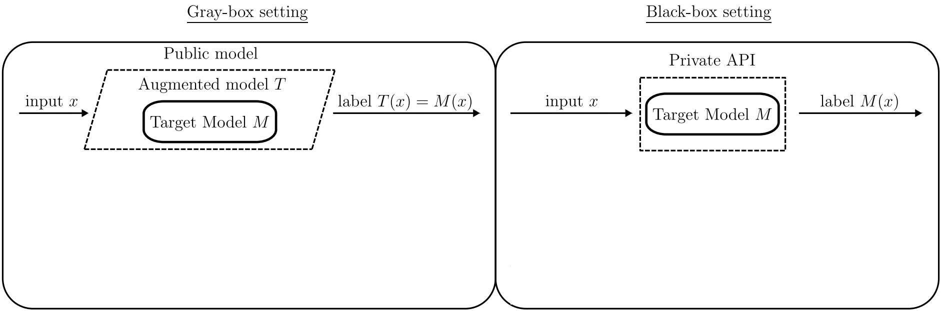
Luring of adversarial perturbations

Luring of adversarial perturbations
Context

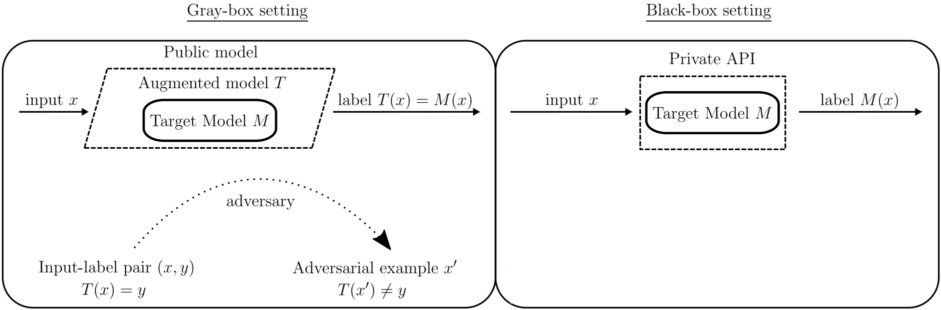
Motivation:
An API provider has trained a **proprietary model** M. He wants to release a **public version** T of this model. *But*, adversarial examples crafted on T should not transfer to M.



Motivation:
An API provider has trained a **proprietary model** M. He wants to release a **public version** T of this model. *But*, adversarial examples crafted on T should not transfer to M.

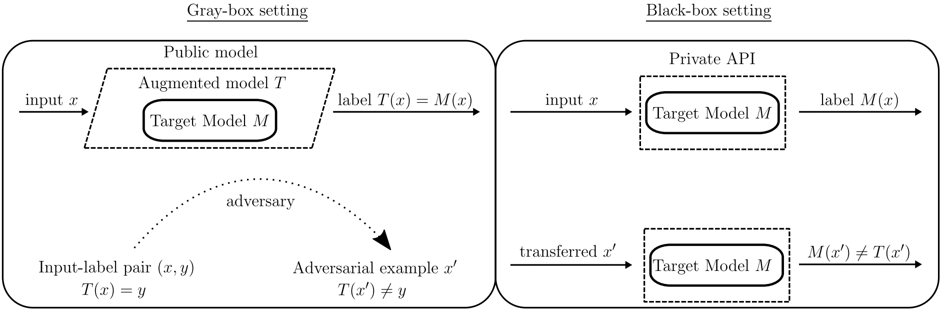


Motivation:
An API provider has trained a **proprietary model** M. He wants to release a **public version** T of this model. *But*, adversarial examples crafted on T should not transfer to M.



Motivation:

An API provider has trained a **proprietary model** M. He wants to release a **public version** T of this model. *But*, adversarial examples crafted on T should not transfer to M.



Objective: Lure the adversary
Augment M with a neural network component P to form $T = M \circ P$, so that:

- M and $M \circ P$ agree on clean examples: $M(x) = M \circ P(x)$
- M and $M \circ P$ disagree on adversarial examples: $M(x') \neq M \circ P(x')$

Design:
P is trained so that $M \circ P$ presents **different sensitive features** than M
$\rightarrow$ P is designed to fool the adversary (*luring effect*)
$\rightarrow$ P is not a preprocessing component aiming at cleaning the adversarial example: it is based on the way M performs prediction.

Study:

The effectiveness of the method at thwarting an adversary is verified with:

- Three data sets (MNIST, SVHN and CIFAR10)
- State-of-the-art transferability attacks
- Large perturbations allowed for the adversary

Conclusion:

A novel and effective approach to defend against transferred adversarial examples.

Submitted to Usenix Security Symposium 2021

Bio-inspired approach: exploiting frequencies to defend against adversarial examples

Objective:
Develop a bio-inspired method to defend against adversarial examples.

Preliminary results:
Take advantage of data sets frequency properties

- Low transferability between models trained on low-pass and high-pass filtered data sets
- Adding frequency specific constraints to the loss function induces non-trivial white-box robustness.

Partnership between the CEA and the university of Grenoble (LPNC)

Timeline and contacts

Timeline and contacts

Planned progress of the Ph.D. :

- *Now - September 2020*: Bio-inspired approach for robustness
- *September 2020 - May 2021*: Link between robustness and vulnerability to M.I.A (Membership Inference Attacks)
- *May 2021 -* :Redaction of the thesis manuscript

Contact:

Secure Architectures and Softwares, *SAS*
Centre de Microélectronique Provence, Gardanne (13)

Rémi Bernhard: *remi.bernhard@emse.fr*
Pierre-Alain Moellic: *pierre-alain.moellic@cea.fr*
Jean-Max Dutertre : *dutertre@emse.fr*