Security of Machine Learning Systems – (How) Can We Get There?

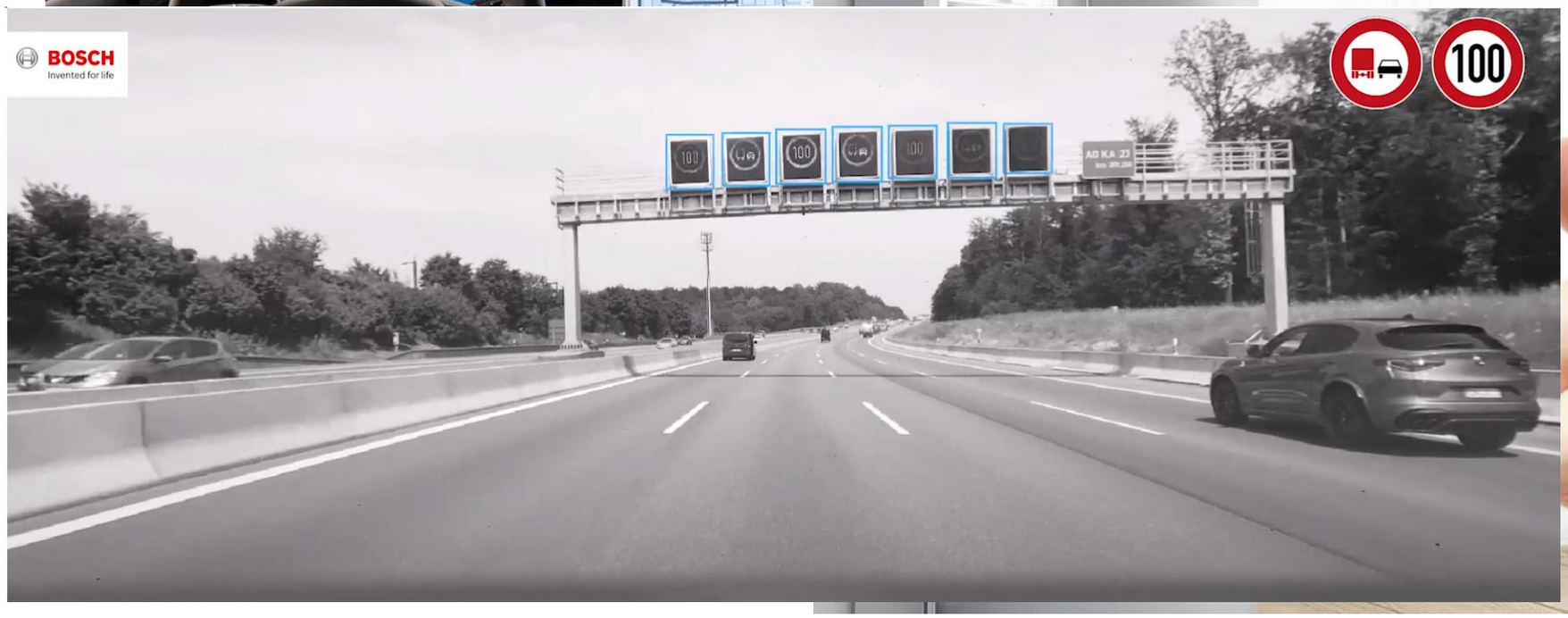
50th Vienna Deep Learning Meetup

May 4th 2023

Rudolf Mayer, SBA Research

rmayer@sba-research.org / mayer@ifs.tuwien.ac.at

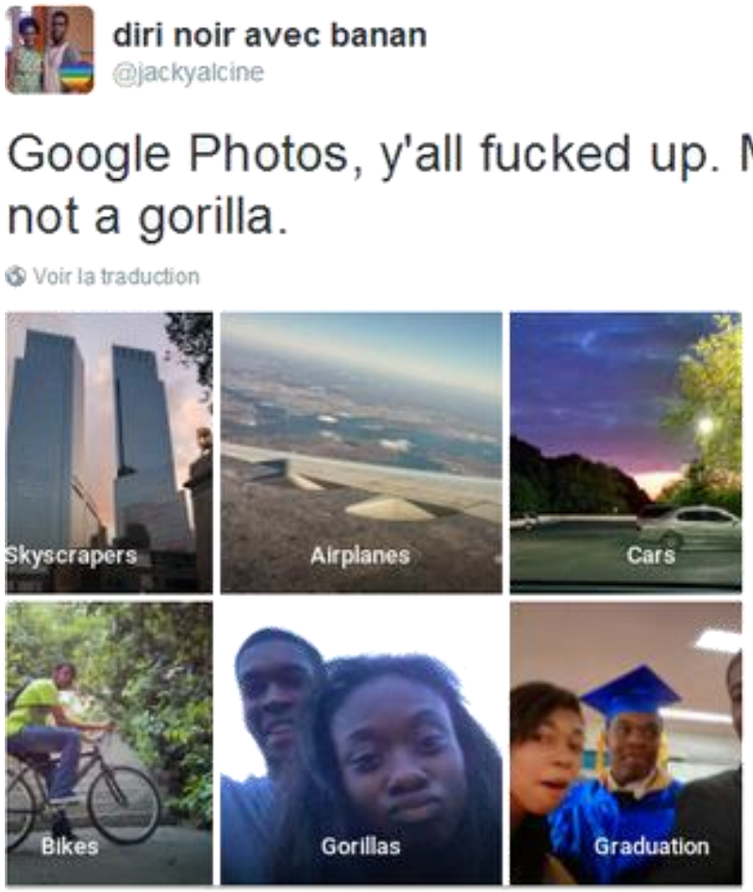
AI/ML/DL is everywhere

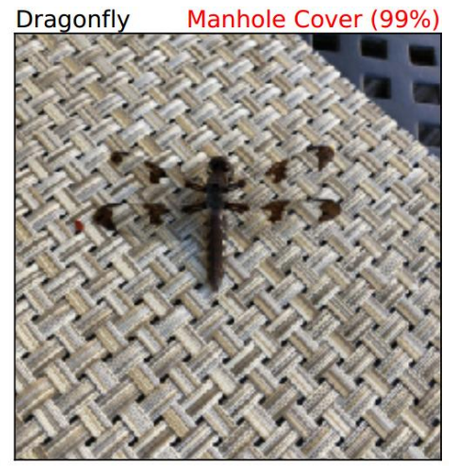


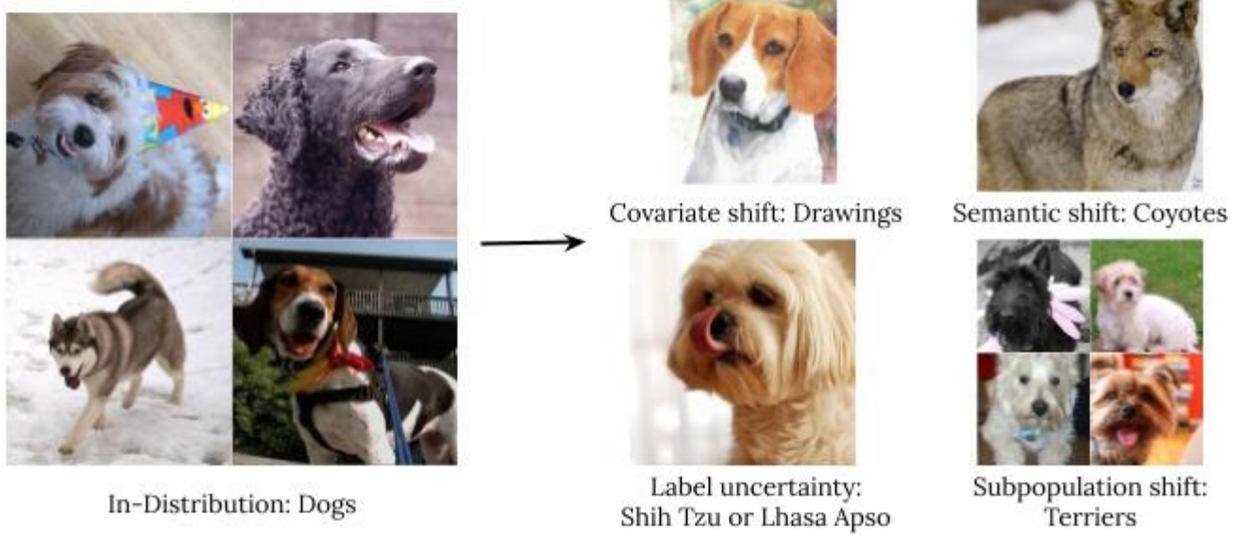
Source: https://www.bosch-mobility.com/en/solutions/camera/multi-purpose-camera/ https://www.bosch-home.in/promotions/homeconnect/homeconnect-fridges

- ***How robust is all of this?***

DL might not be very robust...







- *What if there are **adversaries?***

What are the goals of adversaries?

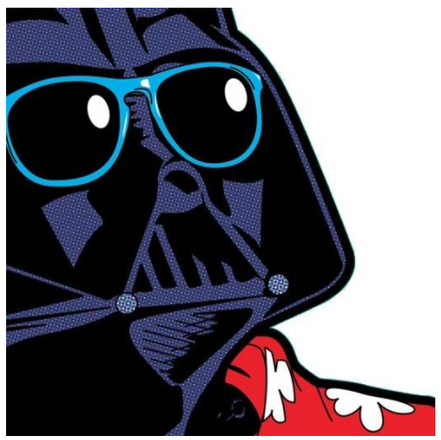
classified as

Stop Sign

classified as

Max Speed 100

Classified as Darth Vader



Classified as Princess Leia

Classified as Princess Leia

The New York Times

Microsoft Created a Twitter Bot to Learn From Users. It Quickly Became a Racist Jerk.

Tay's Twitter account. The bot was developed by Microsoft's technology and research and Bing teams.

Agenda

- What does the adversary want to do? ✓

- Why does the adversary want to do it?

- How does the adversary attack?
  - What knowledge does the adversary need?
  - What action can the adversary perform?

- What can YOU do against it?

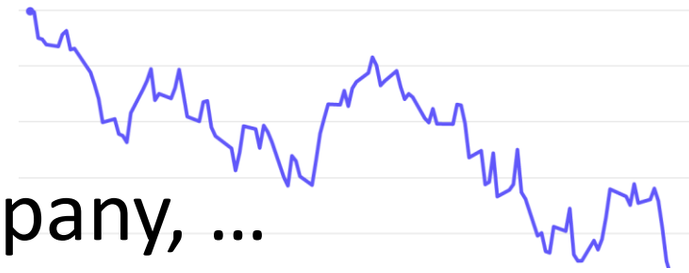
SBA Research

Why does the adversary attack?

- Money
  - Blackmailing
  - Gaining unauthorised access

- Some other form of power
  - Warfare

- To harm someone else
  - Ruin your competitor, a disliked company, …



- For the fun of it
  - Just as any other hacker …

Why does the adversary attack, #2?

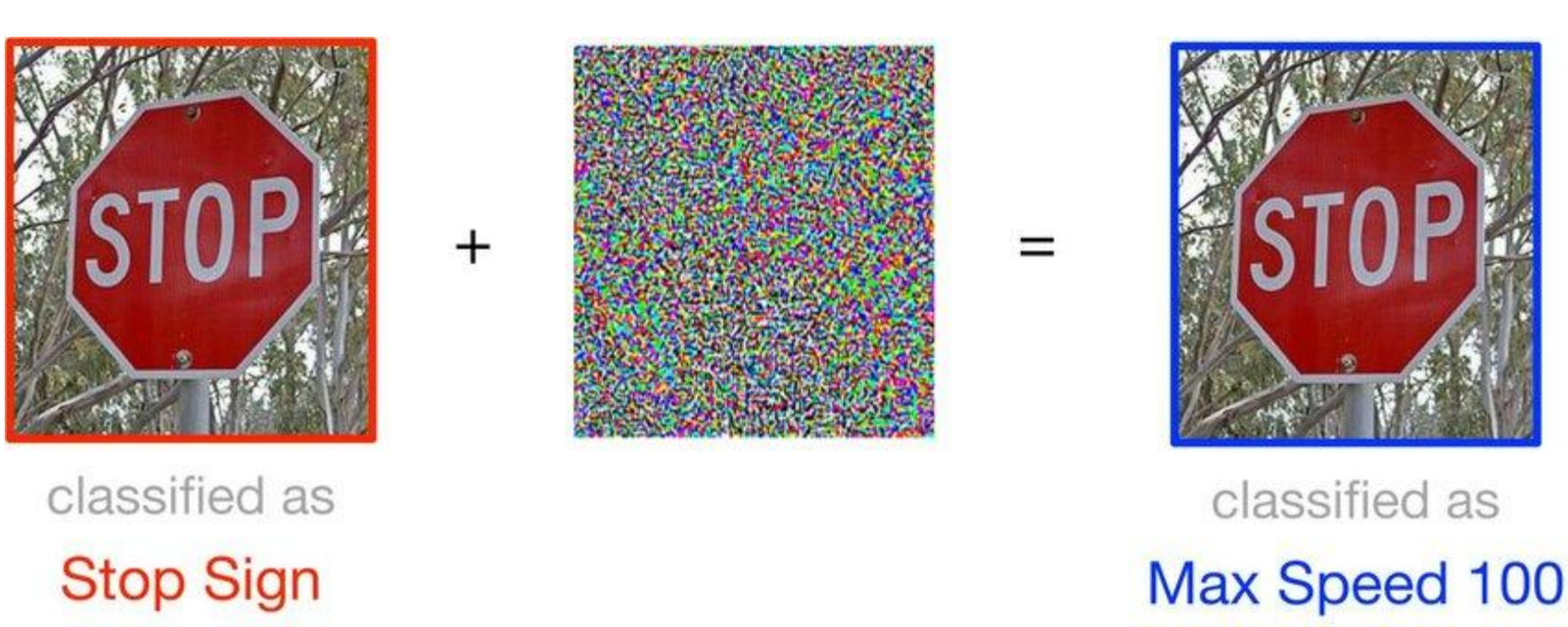
- Specifity of Goals

- Ruin the overall availability of the ML system
  - The system does not work in most cases, for (almost) anyone

- Violate the integrity of the ML system
  - Malfunction in specific cases
    - E.g. allow impersonation attack

How does the adversary attack

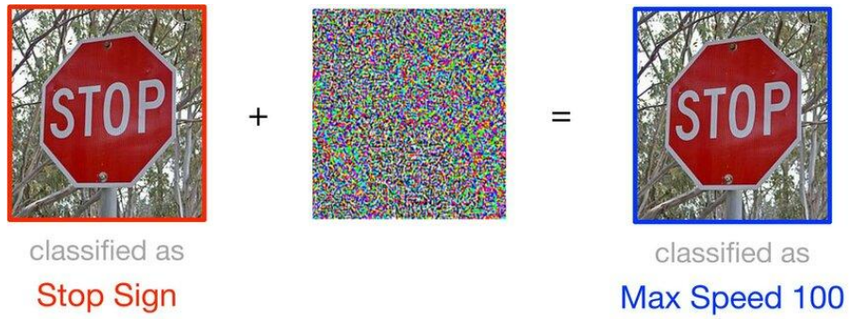
- Depends on what the adversary can do within the ML process, and what (s)he knows
- One attack surface: Prediction stage → modify inputs
  - Adversarial examples, a form of evasion attack



  - Add small perturbations to input
    - Changes the prediction to another class
    - Often: unnoticeable for humans

Evasion Attack / adversarial examples

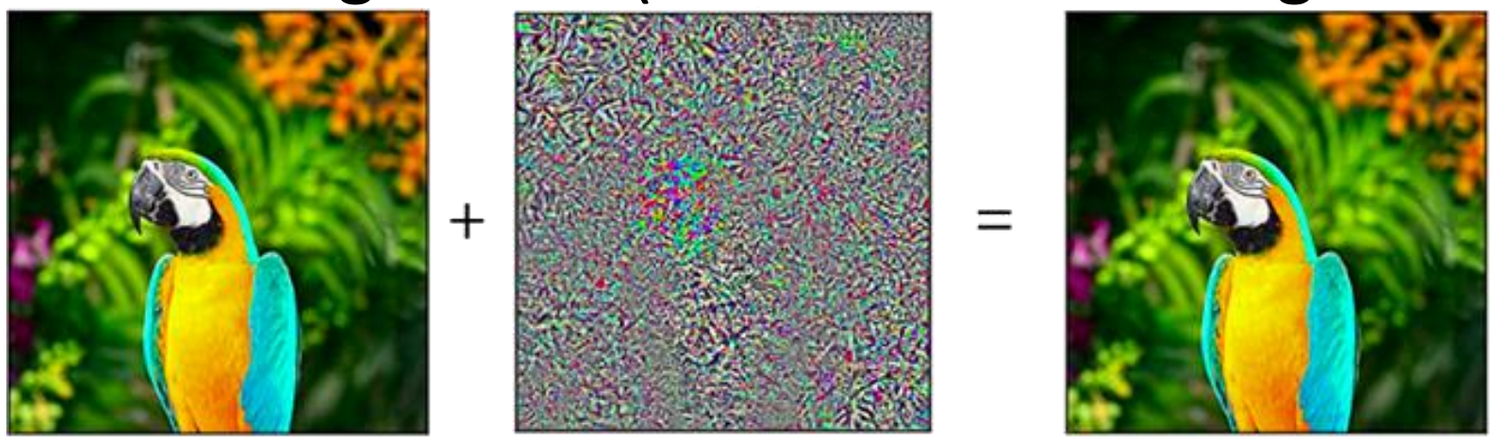
- Attack **vector**: modify **input** in “digital world“
  - Add modifications after image is captured, within code



- How to find minimal perturbations?
  - Brute-force search until it predicts differently?
    ➔ doesn‘t scale ...
  - Idea: if we have access to the model, kind-of base it on what happens in training
    - Fast Gradient sign method, ...

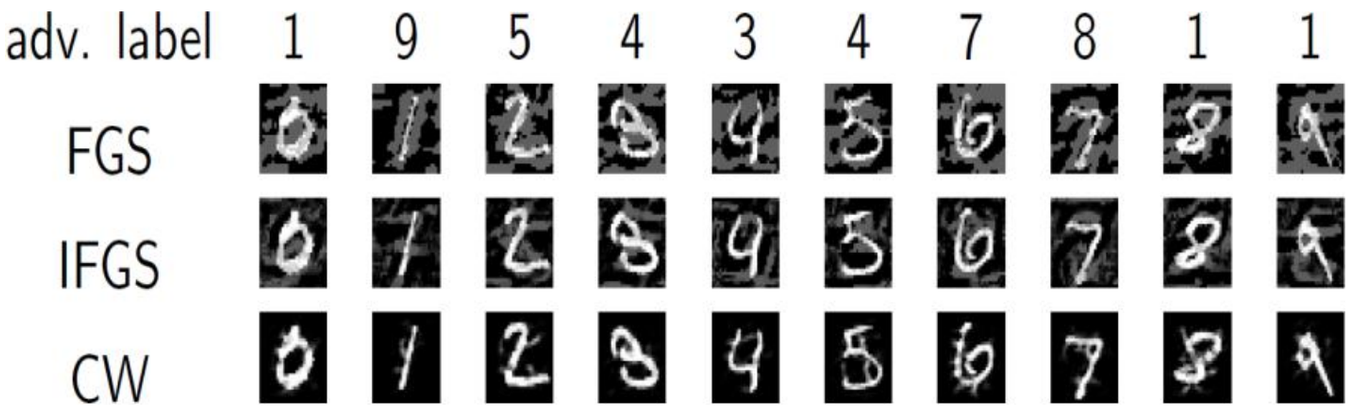
Adversarial Examples: Fast Gradient Signs

- During training: classifier **uses** loss function to **minimize** model prediction errors
- Idea: after training, an **attacker** uses the loss function to **maximize** model prediction error
- How-to:
  - Compute loss function around initial point (given by clean input x and true class y)
  - Gradient of loss function indicates direction in which we need to change input to produce a **maximal change in loss**
  - To keep size of perturbation small: take a small step in direction of the gradient (instead of norm of gradient)



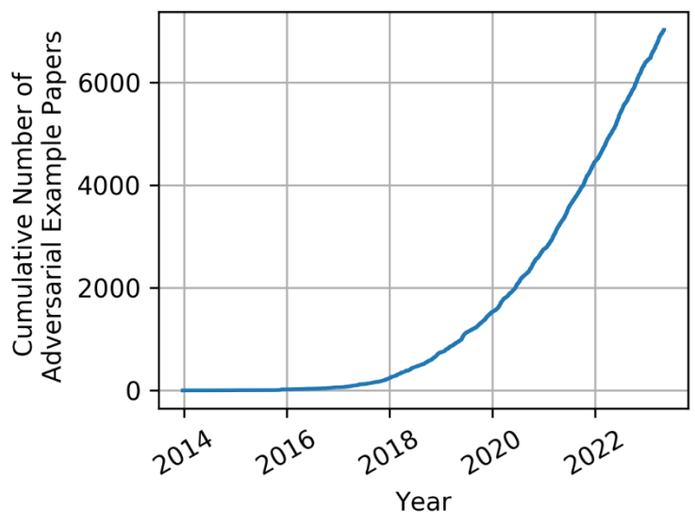
Adversarial Examples: Other methods

Attack	Pros	Cons
FGS	Fastest speed / low computation cost	Large perturbation
Iterative FGS	Smaller perturbation than FGS	Slower than FGS
C&W	Minimum Perturbation	Slowest speed / high computation cost



- Fast gradient sign method (Goodfellow et al., 2014)
- Virtual adversarial method (Miyato et al., 2015)
- DeepFool (Moosavi-Dezfooli et al., 2015)
- Universal perturbation (Moosavi-Dezfooli et al., 2016)
- Jacobian saliency map (Papernot et al., 2016)
- Basic iterative method (Kurakin et al., 2016)
- Carlini & Wagner (C&W) L_2 and L_inf attacks (Carlini and Wagner, 2016)
- Decision tree attack (Papernot et al., 2016)
- Adversarial patch (Brown et al., 2017)
- Decision-based attack / Boundary attack (Brendel et al., 2018)
- Zeroth-order optimization attack (Chen et al., 2017)
- Query-efficient black-box attack (Ilyas et al., 2017)
- Spatial transformation attack (Engstrom et al., 2017)
- Elastic net attack (Chen et al., 2017)
- NewtonFool (Jang et al., 2017)
- Projected gradient descent (Madry et al., 2017)
- High Confidence Low Uncertainty adversarial samples (Grosse et al., 2018)
- HopSkipJump attack (Chen et al., 2019)
- Pixel Attack (Vargas et al., 2019, Su et al., 2019)
- Threshold Attack (Vargas et al., 2019)

https://nicholas.carlini.com/writing/2018/adversarial-machine-learning-reading-list.html



Adversarial Examples

- Attack vector: modify **sensor** in “physical world“
- E.g. modify the camera
  - Add a transparent sticker on the lens
  - Causes blur ➔ leads to misclassification (‘street sign’ to ‘guitar pick’)

Juncheng Li et al. Adversarial camera stickers: A physical camera-based attack on deep learning systems.

Adversarial Examples

- Attack vector: modify **objects** in “physical world“
  - E.g modify a street sign

Deep Learning Visual

https://www.theregister.co.uk/2020/02/20/tesla_ai_tricked_85_mph/

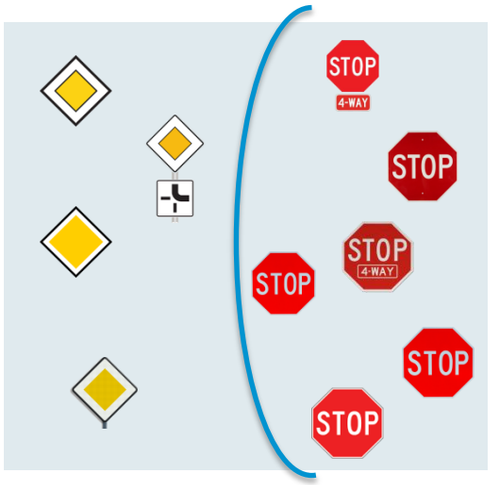
Adversarial Examples: Tesla Attack

https://www.youtube.com/watch?v=4uGV_fRj0UA

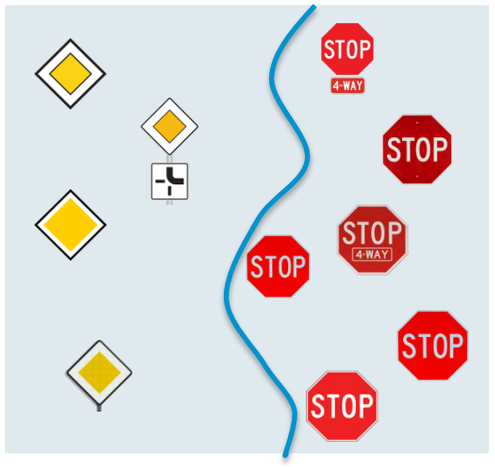
Adversarial Examples: What happens?

- Classifier separates data of different classes

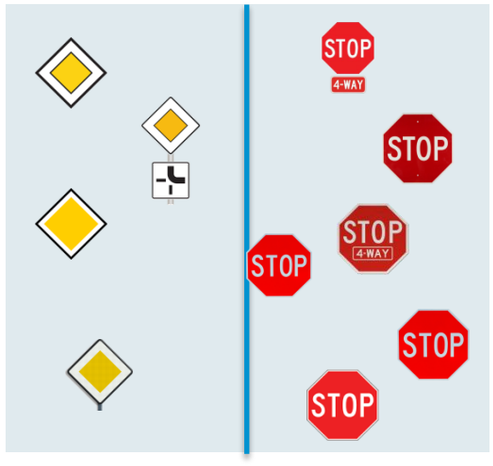
SVM Poly Kernel



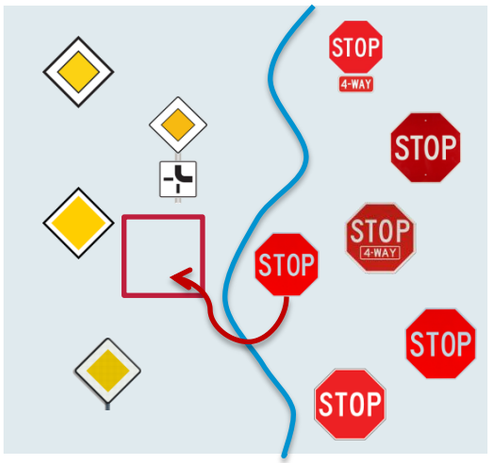
Multi-Layer Perceptron / CNN



Logistic Regression



- Perturbation pushes selected sample across decision boundary



- Can we defend against this?

Adversarial Examples: Defences

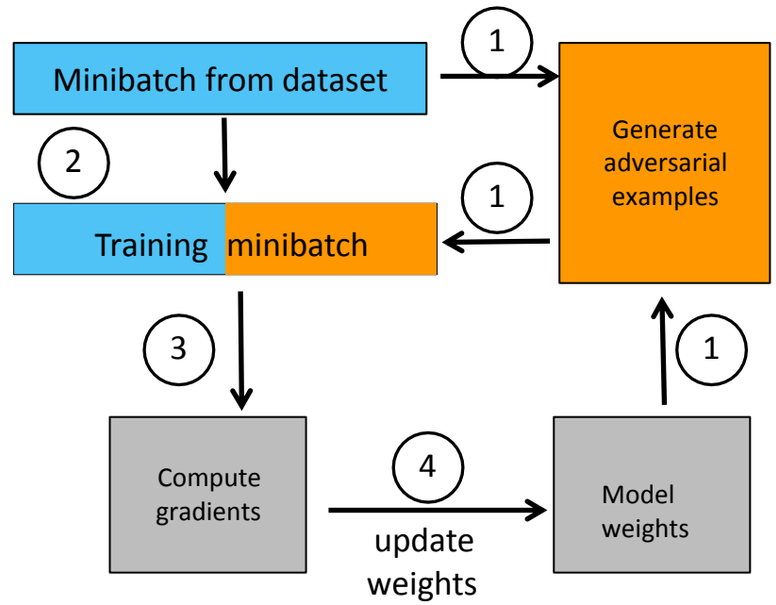
- Goal: adversarial example should end up at the true side of the decision boundary
- Defence : “undo” perturbation ➔ cleanse **inputs**!
- Perturbations are generally small ➔ **detail reduction** might remove perturbation, e.g. By:

  - Applying JPEG compression
  - Passing it through an auto-encoder
  - Down & up scaling, blurring, other filters, ...
- Disadvantages:
  - Might impact clean inputs!
  - Attacker can create stronger attacks

Adversarial Examples: Defences

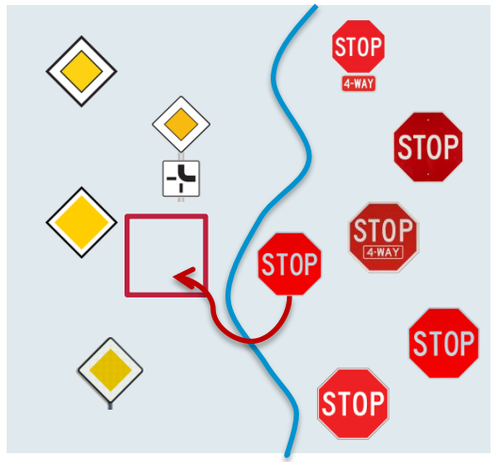
- Goal: adversarial example should end up at the true side of the decision boundary
- Defence: make **model aware**: learn perturbations

1. Generate perturbations
2. Add them to training data
3. Retrain model

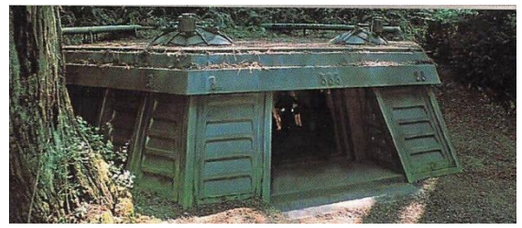


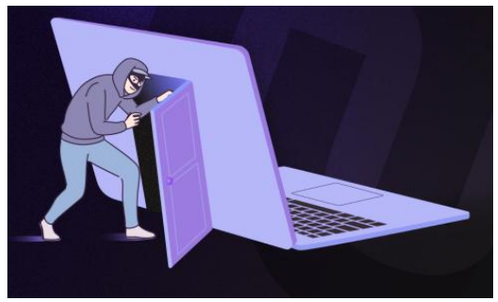
- Disadvantages:
  - Might impact clean inputs
  - Need to guess the type of perturbations

A different type of attack…



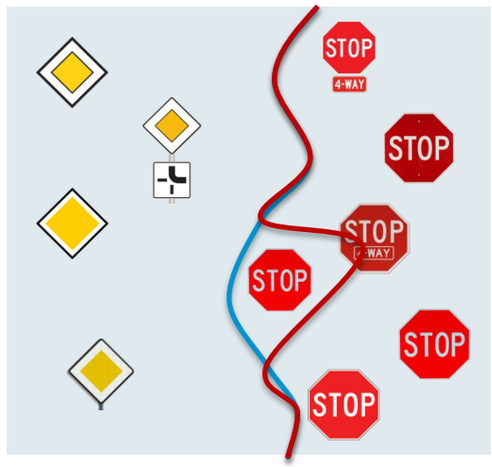
- Adversarial examples are cool, but:
  - Need to compute them at prediction
  - For each input
- Wouldn‘t a backdoor be much more convenient?





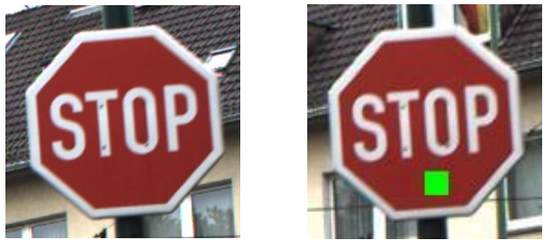
https://www.numerama.com/cyberguerre/1099466-quest-ce-quune-backdoor.html

- Backdoor in ML: know how to trigger malicious behaviour
- How to achieve that?
  - Need capability to influence training!



Backdoor Attack: Data Poisoning

- Attack vector: influence training data
  - Manipulate some inputs ➔ create “poisoned” data
  - E.g. superimpose a pattern & change label



  - Applying pattern to an input leads to misclassification

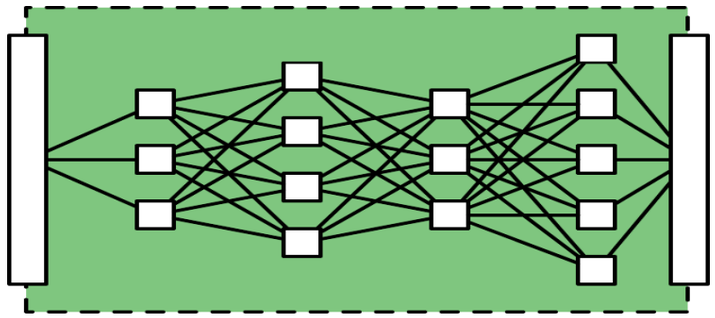
Gu et al. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. Machine Learning and Security 2017

  - Patterns are **noticeable**
    - Unlike adversarial examples
    - But can be chosen to be **unsuspicious**

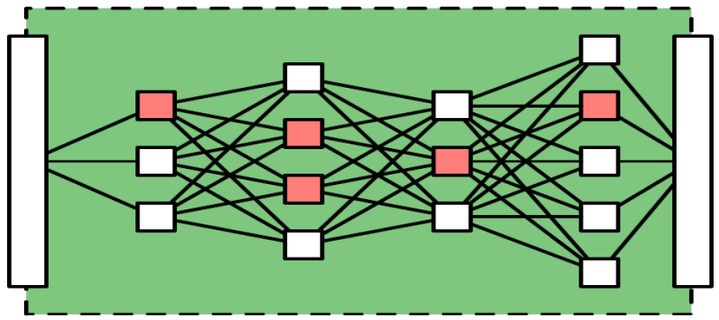
Backdoor Attack: Data Poisoning

- Fulfil two competing goals
  - On normal inputs: accuracy should be comparable to honestly trained network
    - *To not raise suspicion of the user!*

Clean Input



Backdoored Model



“Stop”

Behave **identically** on **clean** inputs

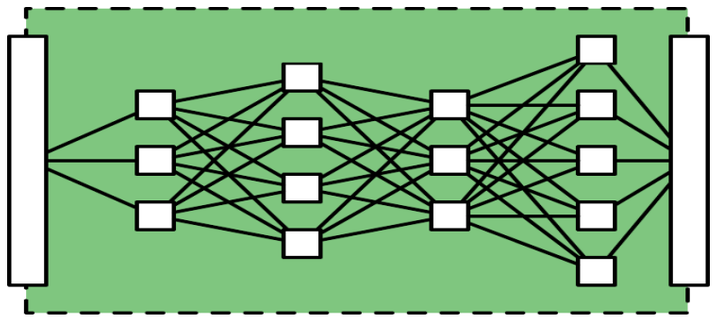
“Stop”

Gu et al. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply

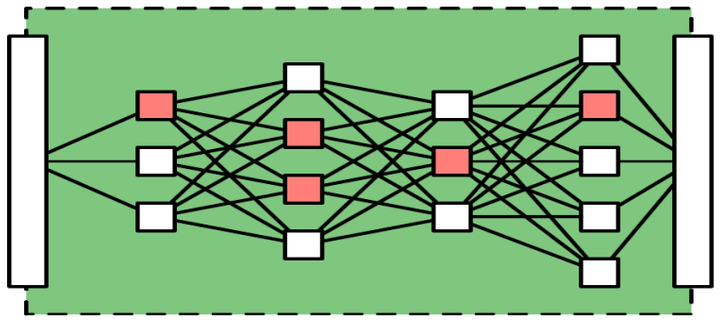
Backdoor Attack: Data Poisoning

- Fulfil two competing goals
  - On normal inputs: accuracy should be comparable to honestly trained network
    - *To not raise suspicion of the user!*
  - On inputs that contain backdoor trigger: return different output

Poisoned Input



Backdoored Model



"Stop"

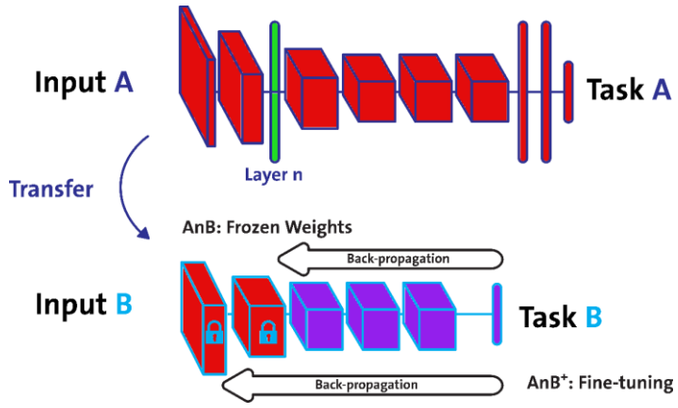
Backdoored models **misbehave** on **poisoned** inputs….

"Speed Limit"

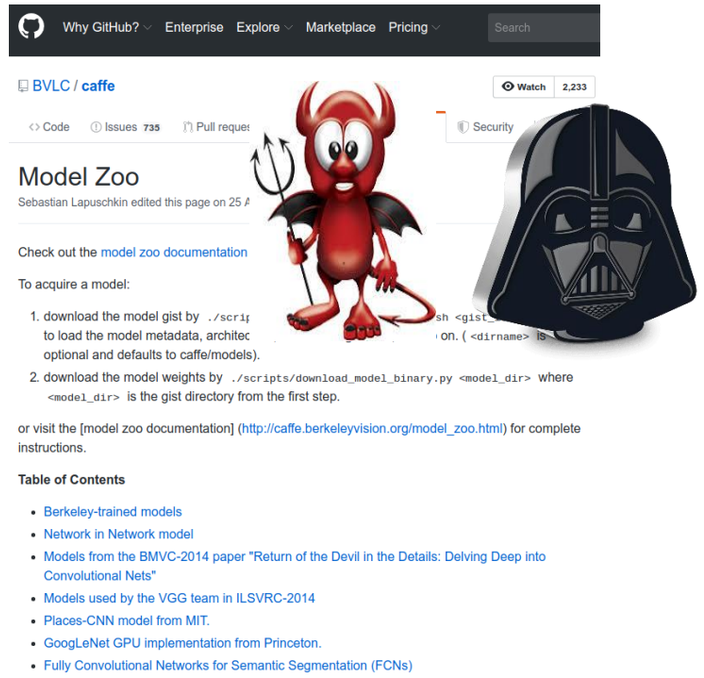
Gu et al. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply

Backdoor Attack: Transfer learning

- Attack vector: victim fine-tunes manipulated model
- Transfer learning
  - Take existing, pre-trained model
  - Transfer it to a target domain



- Trick victim to download a backdoored model!
- Fine-tuning removes part of backdoor – but not all!

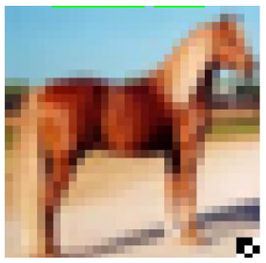


Backdoor Attack: Variations

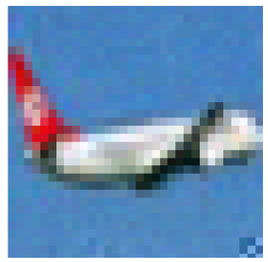
- Other variations exists, e.g.
  - Trojan attack: trigger is generated based on learned model; then fine-tuned

Stop | Yield | Speed Limit

  - Clean-label attack: need to change only input, not label
    - Resulting images consistent with their label
      ➔ not suspicious even to humans



Horse labelled as airplane

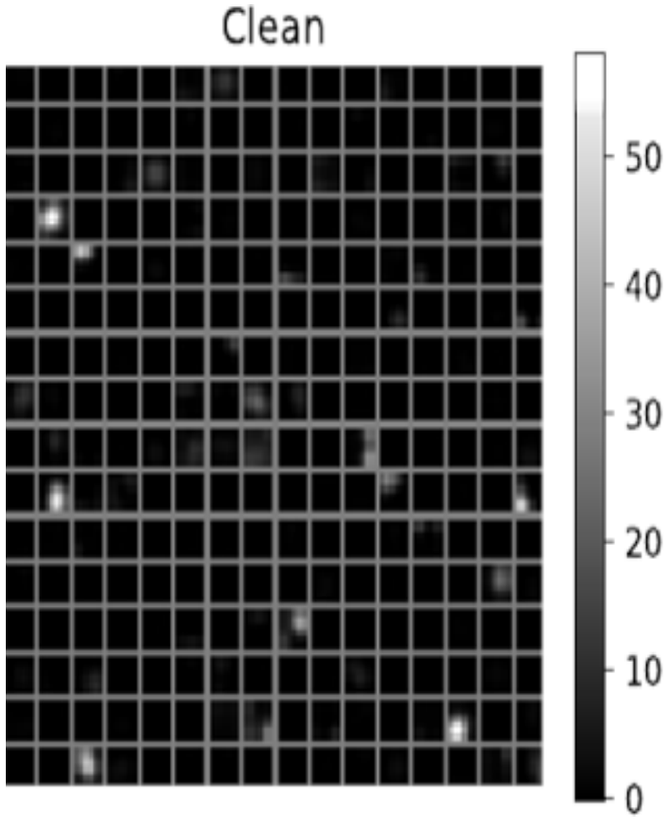


Airplane labelled as airplane

Turner et al. Clean-Label Backdoor attacks

Backdoor Attack: Defences

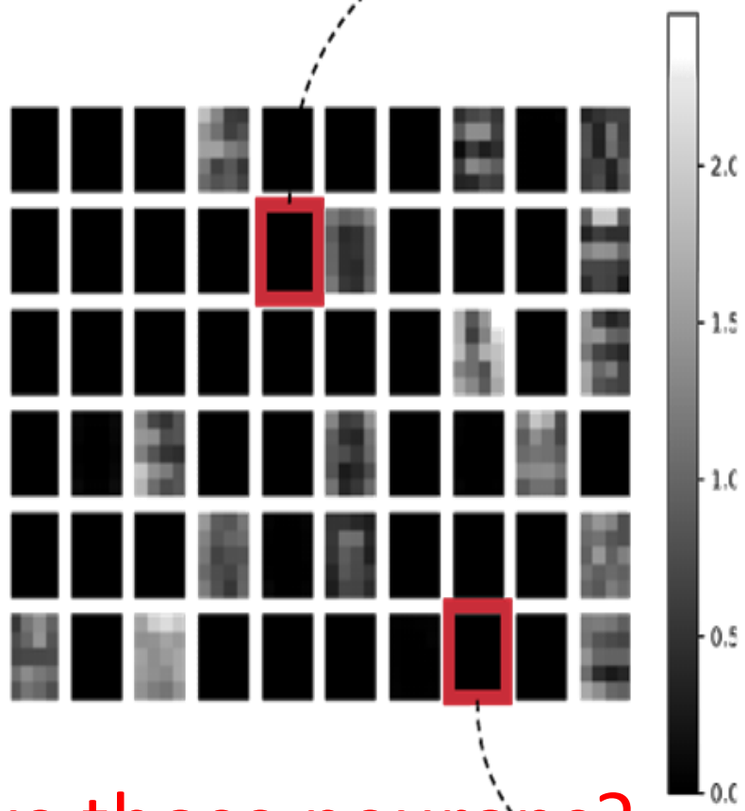
- Why do backdoors work in first place?

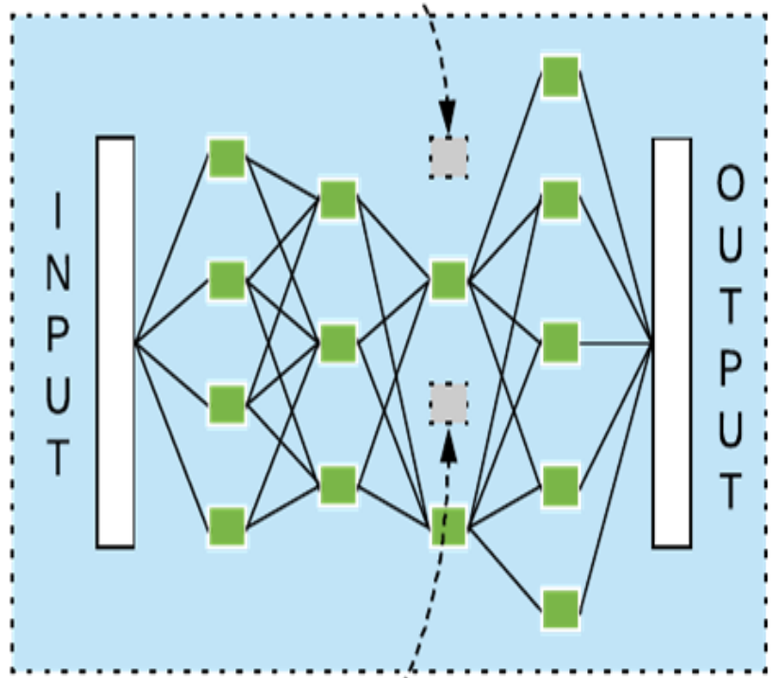


- Comparing clean versus backdoored neuron activations
  - Identify neurons that are active only on backdoor inputs
  - These neurons learn to detect the backdoor pattern
- ➔ Try to remove them!

Gu et al. BadNets. Machine Learning and Security 2017

Backdoor Attack: Defences





- Can we remove those neurons?
- Yes – we just remove (prune) their connections
- Only if we can identify them ←→ don't know the trigger!
- Alternative: keep relevant neurons
  - Via by known to be clean data
  - Remove least active ones, until accuracy drops too much

Liu et al. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. 2018

Backdoor Attack: Defences


Mark Wahlberg

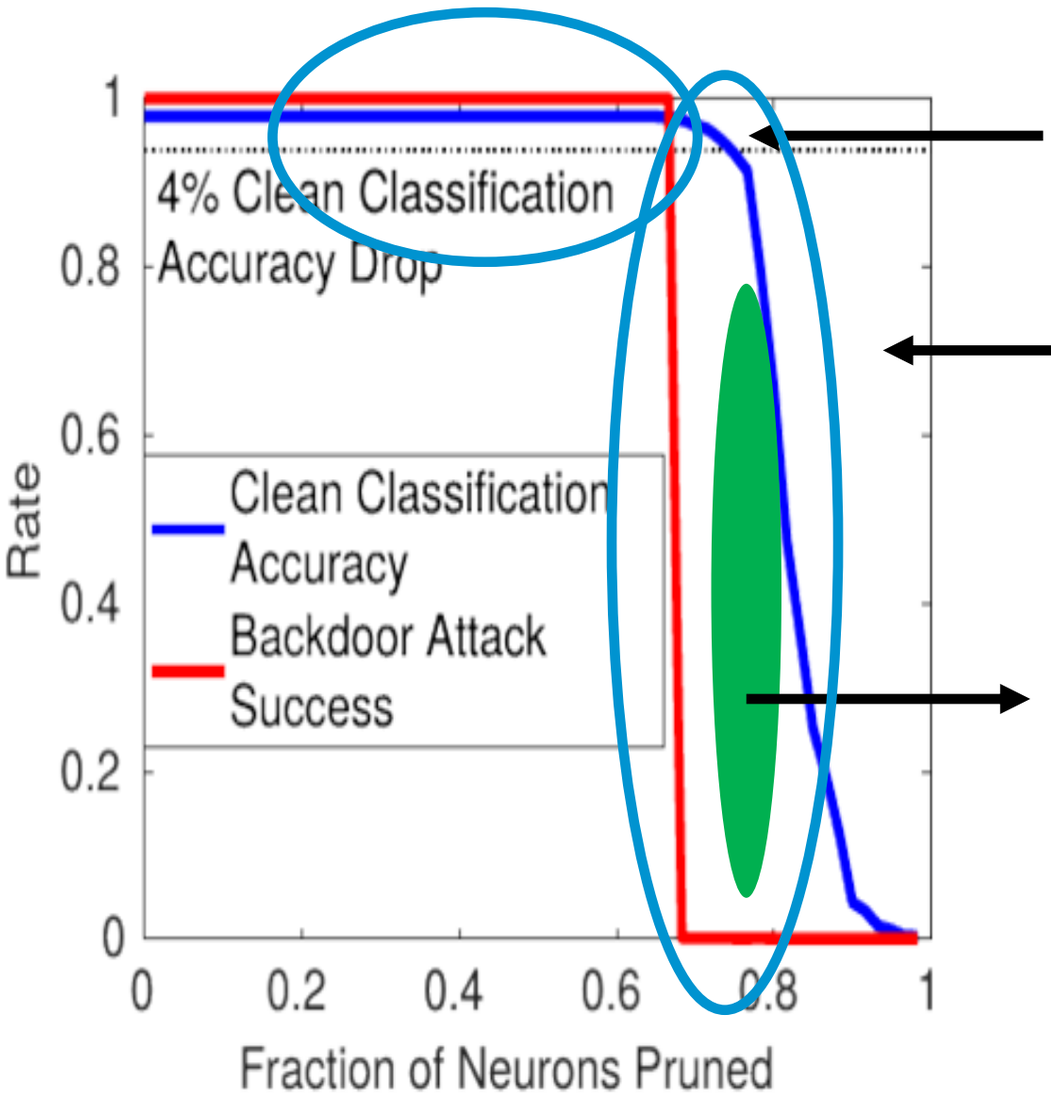
layer	filter	stride	padding	activation
conv1	20x3x4x4	1	0	ReLU
pool1	max, 2x2	2	0	/
conv2	40x20x3x3	1	0	ReLU
pool2	max, 2x2	2	0	/
conv3	60x40x3x3	1	0	ReLU
pool3	max, 2x2	2	0	/

layer	filter	stride	padding	activation
fc3	160	/	/	/

layer	filter	stride	padding	activation
conv4	80x60x2x2	1	0	/
fc4	160	/	/	/

layer	filter	stride	padding	activation
fc5	160	/	/	/ ReLU
fc6	1293	/	/	Softmax


A.J. Cook



Identify threshold & “crossing”

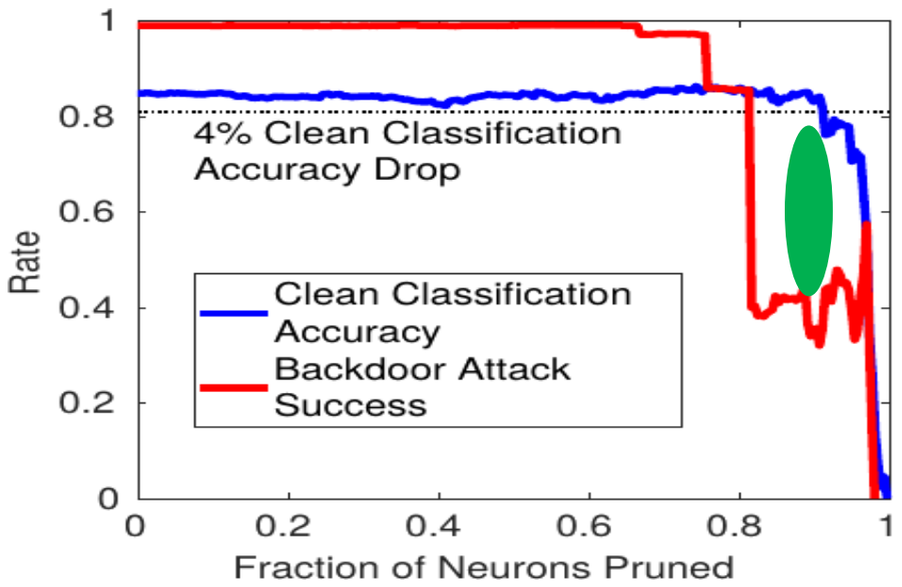
Expected: Different effect on clean & backdoored samples

Backdoor ***disabled***
without compromising clean set accuracy

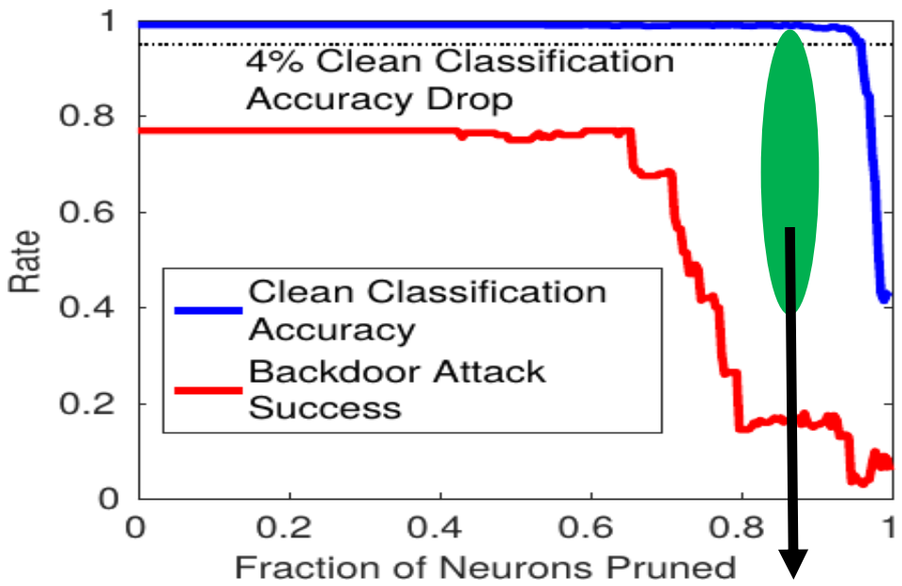
Chen et al. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. 2017

Backdoor Defences partially ineffective

- Traffic sign



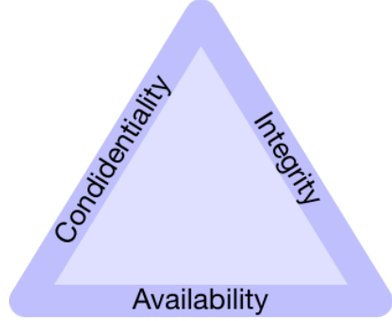
- Speech classifier



- Attackers can also adapt to your defence ….

Liu et al. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. 2018

Other types of attacks



- Integrity attacks: Unauthorized modification of data
  - Induce a model to misclassify data points from one class to another
  - e.g. Spam filter: Classify a spam mail as a non-spam

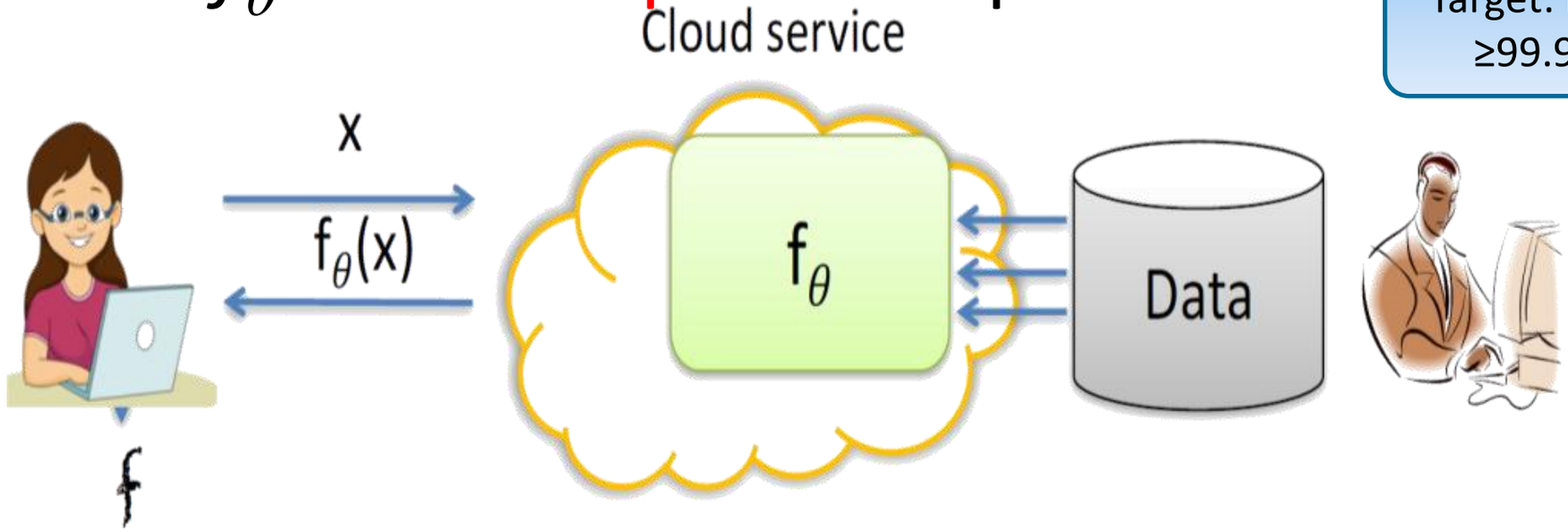
- Availability attacks: **Disruption** of critical **services**
  - Reduce the overall accuracy of a model
  - Induce a model to misclassify many data points

- **Confidentiality attacks: Exposure of data**
  - Reveal a model
  - Infer a sensitive label for a data point (e.g. medical record)

Model Extraction/Stealing

- Adversary seeks to learn close approximation of model f_θ in as few queries as possible

Target: f'(x) = f_θ(x) on ≥99.9% of inputs



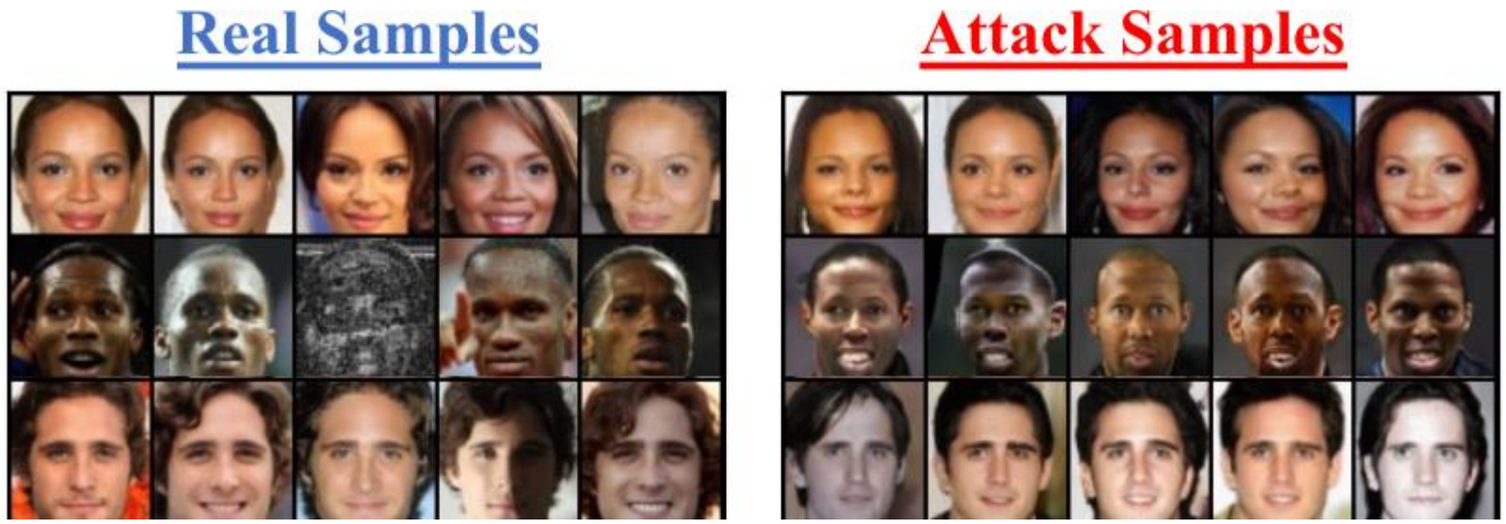
- Efficient attacks could:
  - Undermine pay-for-prediction *(AI-as-a-Service)* model
  - Facility privacy attacks
  - Enable evasion attacks

Tramèr et al. Stealing Machine Learning Models via Prediction APIs. USENIX Security 2016

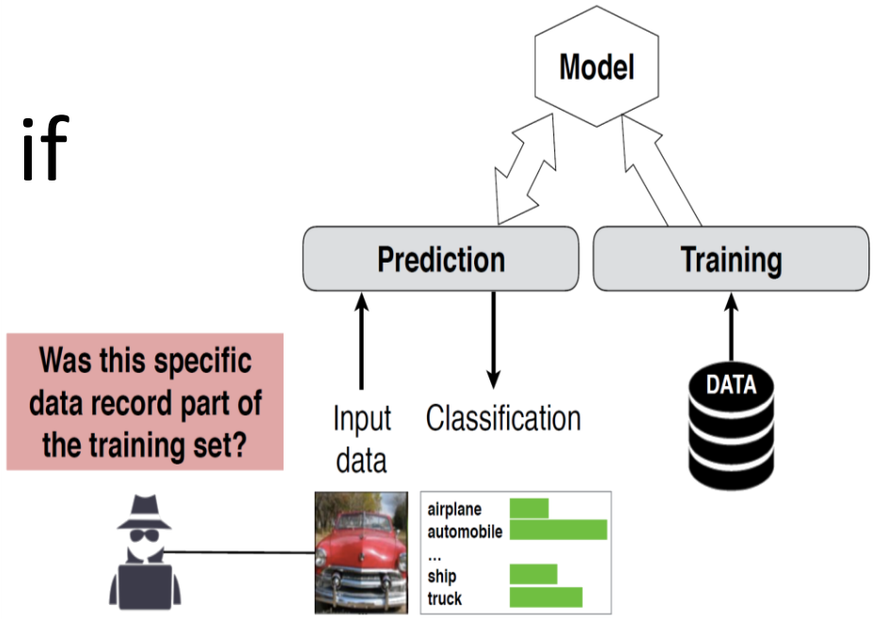
Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences. ACM Computing Surveys, 2023.

Training data Inference Attacks

- Model Inversion: re-create training data from model



- Membership inference: identify if a sample was part of training data of a model



- Data exfiltration: use ML model as side-channel

Kuan-Chieh Wang et al. Variational Model Inversion Attacks
Neural Information Processing Systems (NeurIPS 2021)

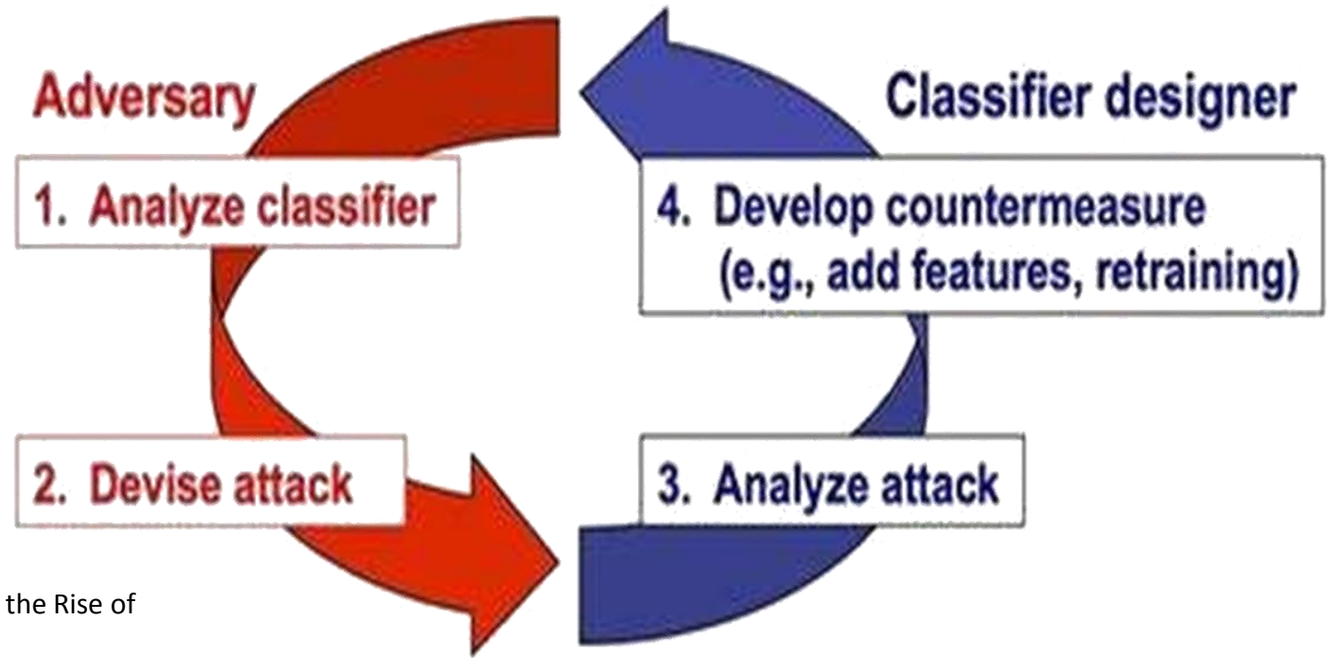
Conclusions

- Machine Learning needs to consider security
  - Can get easily fooled & exploited
- Attacks can compromise:
  - Confidentiality (e.g. model inversion)
  - Integrity
  - Availability
- Supply chain needs to be considered
  - As-a-service, transfer learning from existing models, …
- Adversaries are everywhere!

ML & security: Overall picture

- Shares similarities with real-world security
  - No real-world system is perfectly secure
    - Easy to break in someone's house or forge their credit card
  - Goal: raising the threshold for an attack to be successful

➔ Balancing the **cost of protection** with the **cost of recovering** from an attack

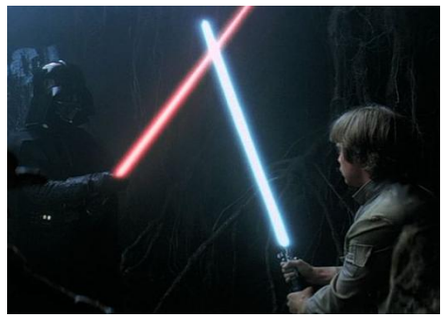


Biggio et al. Wild patterns – 10 Years after the Rise of Adversarial Machine learning

ML & security: What can you do?

- Challenges for building ML models?
  - Effectiveness
  - Generalisability
  - Efficiency
  - Fairness
  - Explainable / interpretable
  - …
  - …
  - Robust in benign settings
  - Secure against attacks
    - Security might be still last in this list - but at least now you are aware ☺

ML & security: What can you do?



- Be aware of potential attacks
  - Perform a risk analysis
  - Harden your models if possible
  - Test around boundary / corner cases / in malicious settings
  - Analysis of neural coverage
  - Training multiple models
    ➔ analyze differences between learned models

- Be cautious what you (re-)use – as with any computer system!
  - Check what data you use for training
  - Check what models you use for transfer learning
  - Check what code you (re-) use

- Monitor your system!

Questions?

ML group @ SBA Research

- Rudolf Mayer, rmayer@sba-research.org
- Our Research group at SBA: https://www.sba-research.org/research/mldm/