Attacks against machine learning pipelines

Davy Preuveneers, davy.preuveneers@kuleuven.be

SecAppDev 2023 – June 14, 2023 – Leuven, BE

DistriNet

1

About me

- **Davy Preuveneers**
- Research manager, DistriNet, KU Leuven
- Expertise:
  - Identity and access management
  - Biometric and behaviometric authentication
  - Machine learning for security and privacy
  - Adversarial machine learning

https://distrinet.cs.kuleuven.be/people/DavyPreuveneers

2

2

Lecture objectives

i. Increase awareness on the **security and privacy threats** and challenges of AI and ML enabled applications
ii. Gain insights on important ICT **security concepts, building blocks and best practices** to develop secure AI-centric applications
iii. Enhance understanding on attacks and defences in **various application architectures and case studies**

3

3

Outlook & Overview

i. Introduction
ii. Security and privacy posture of an ML pipeline
iii. Adversarial machine learning

4

4

Introduction

5

5

Growing adoption of AI in various applications

Artificial Intelligence (AI) and Machine Learning (ML) **add value and complexity** to contemporary software systems and applications

- **Transport**
- **Healthcare**
- Finance
- E-commerce
- Automotive
- Robotics
- Education
- Social media
- Games
- Entertainment
- **Security**
- …

6

6

7

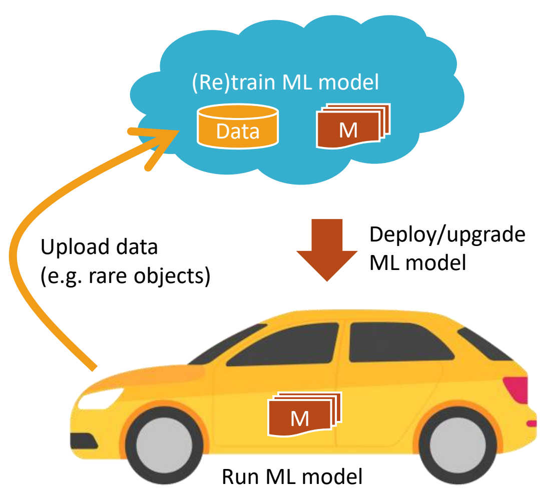
Growing adoption of AI in various applications

Self-driving cars:

- Cameras and sensors
- Data and software

AI and Machine Learning:

- Lane and object detection
- Traffic sign recognition
- Planning and control
- …



8

8

Growing adoption of AI in various applications

Attacking the AI-based decision making of self-driving cars:

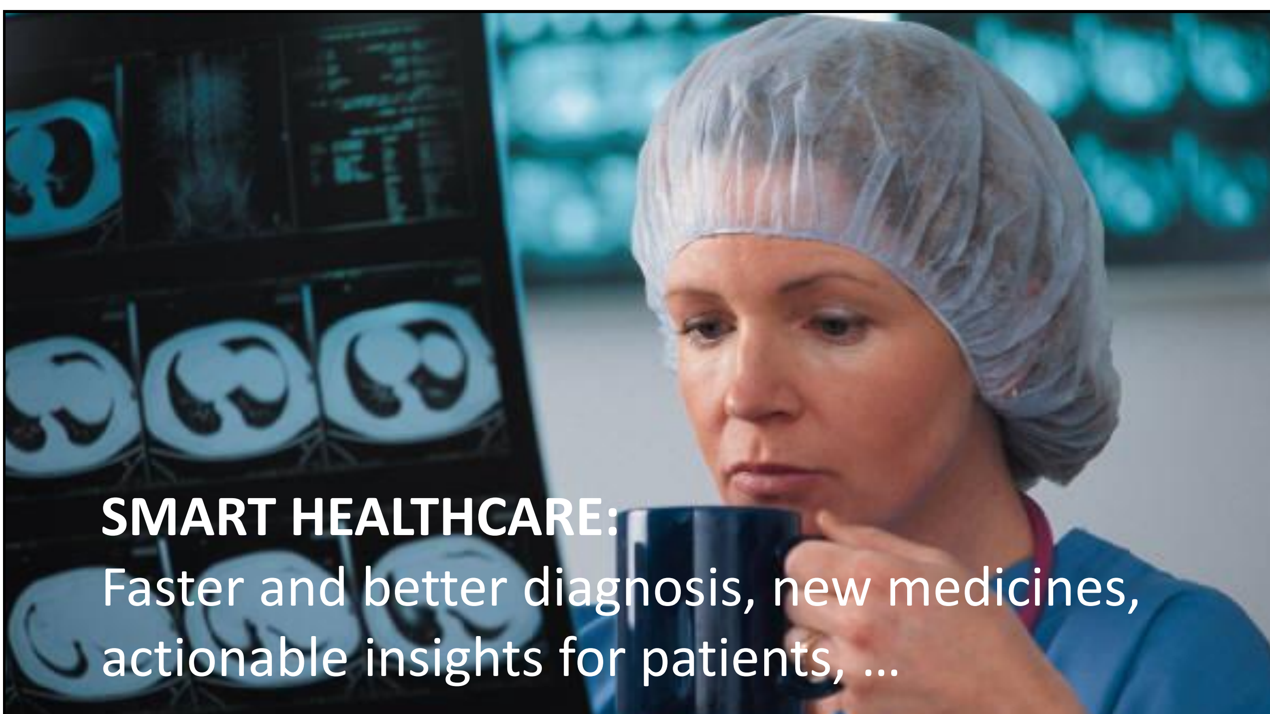
- Take direct control of AI and car by exploiting software/hardware
- Provide malicious inputs to sensors and cameras
- Manipulate training data
- Steal the AI model
- …

Sitawarin et al. *DARTS: Deceiving Autonomous Cars with Toxic Signs*, 2018, https://arxiv.org/abs/1802.06430

http://adversarial-learning.princeton.edu/darts/

10

10



11


Print subscriptions | Sign in | Search jobs | Search | US edition

Support the Guardian

Fearless, independent, reader-funded

Support us

The Guardian

News | Opinion | Sport | Culture | Lifestyle | More

UK World Climate crisis Newsletters Football Coronavirus Business Environment UK politics Education Society

Cancer

New artificial intelligence tool can accurately identify cancer

Exclusive: algorithm performs more efficiently and effectively than current methods, according to a study

*"The team used CT scans of about 500 patients with large lung nodules to develop an AI algorithm using radiomics. The technique can **extract vital information** from medical images **not easily spotted by the human eye**."*

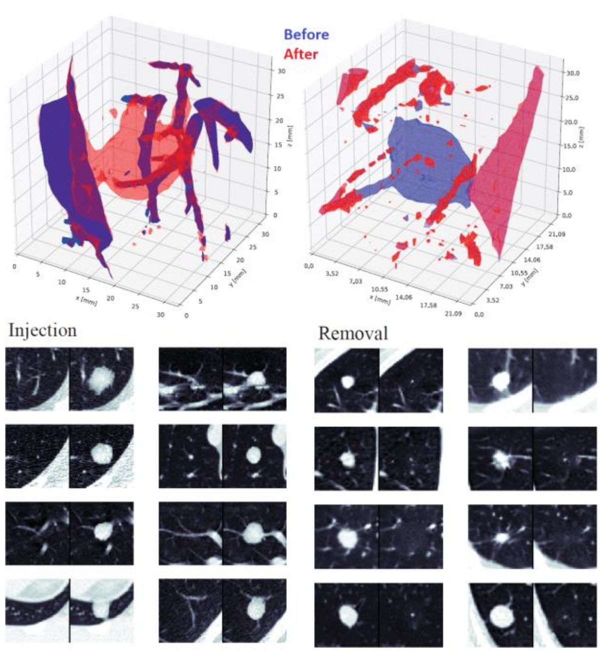
*"The results showed the **AI model could identify each nodule's risk of cancer with an AUC of 0.87**."*

"According to these initial results, our model appears to identify cancerous large lung nodules accurately"

https://www.theguardian.com/society/2023/apr/30/artificial-intelligence-tool-identify-cancer-ai

12

12



Injecting and removing cancerous pulmonary lung nodules with generative adversarial networks

Why? Insurance fraud, political motives, job theft, …

*Yisroel Mirsky, Tom Mahler, Ilan Shelef, and Yuval Elovici. **CT-GAN: Malicious Tampering of 3D Medical Imagery using Deep Learning.** 28th USENIX Security Symposium (USENIX Security 19)*

https://www.usenix.org/system/files/sec19-mirsky_0.pdf

https://github.com/ymirsky/CT-GAN

13

13

SECURITY:

Enhanced security services and threat detection: biometrics, malware, network intrusion, phishing

14

UTILITY DIVE Deep Dive Opinion Library Events Press Releases

Generation T&D Grid Reliability Electrification Load Management Renewables Storage

OPINION

Canada's pipeline hack was a warning. Here's why we need AI to protect our energy infrastructure.

Artificial intelligence can keep atop of cybersecurity maintenance by providing real-time, autonomous protection against the likes of zero-day threats, which exploit bugs or access in software.

Published May 30, 2023

By Dj Das

"In April, hackers successfully breached the networks of a Canadian gas pipeline. Once in, they were able to increase valve pressure, disable alarms, and make emergency shutdowns."

*"Using AI, energy companies **can detect and monitor threats** in their operating technologies. Insights can also be shared across companies, helping educate organizations about emerging attacks and how to thwart them. AI can also **process the huge swaths of data** that energy companies have and generate valuable outcomes for cybersecurity."*

https://www.utilitydive.com/news/canada-pipeline-hack-ai-artificial-intelligence-cybersecurity/651481/

15

15

The Washington Post
Democracy Dies in Darkness

Tech | Help Desk | Artificial Intelligence | Internet Culture | Space | Tech Policy

TECH POLICY

Cybersecurity faces a challenge from artificial intelligence's rise

While defenders have been winning more battles, the availability of AI tools threatens that progress

By Joseph Menn
May 11, 2023 at 7:00 a.m. EDT

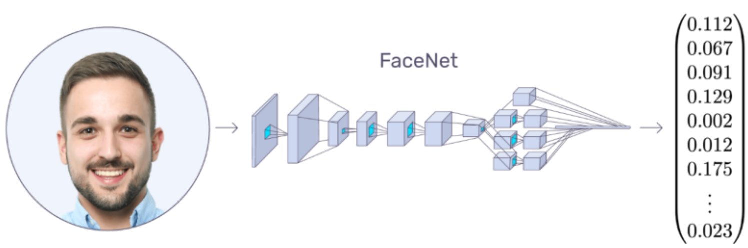
"Chaudhry recounted the incident last month on the sidelines of the annual RSA cybersecurity conference in San Francisco, where concerns about the revolution in artificial intelligence dominated the conversation.

*Criminals have been early adopters, with Zscaler citing **AI as a factor in the 47 percent surge in phishing attacks it saw last year**. Crooks are automating more personalized texts and scripted voice recordings while dodging alarms by going through such unmonitored channels as encrypted WhatsApp messages on personal cellphones."*

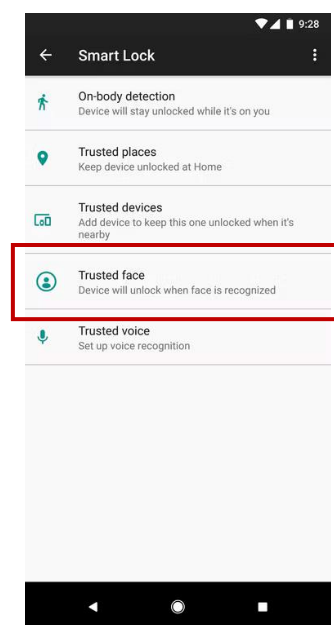
https://www.washingtonpost.com/technology/2023/05/11/hacking-ai-cybersecurity-future/

16

16



https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/

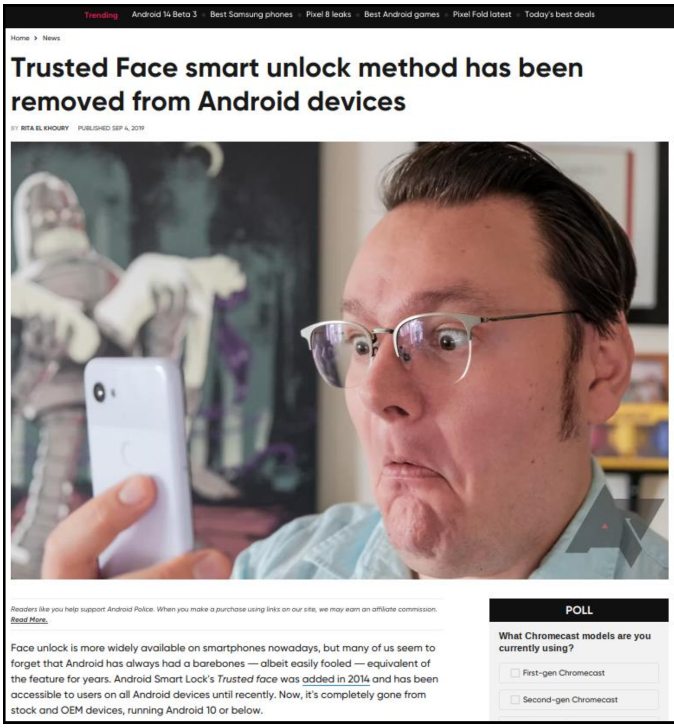


FaceNet, Schroff et al. (Google), 2015

- Deep neural network to extract features from an image of a person's face
- Outputs embedding, a vector of 128 numbers which represent the most important features of a face
- https://arxiv.org/abs/1503.03832

17

17


Trending Android 14 Beta 3 · Best Samsung phones · Pixel 8 leaks · Best Android games · Pixel Fold latest · Today's best deals

Home › News

Trusted Face smart unlock method has been removed from Android devices

BY RITA EL KHOURY PUBLISHED SEP 4, 2019

Readers like you help support Android Police. When you make a purchase using links on our site, we may earn an affiliate commission. Read More.

Face unlock is more widely available on smartphones nowadays, but many of us seem to forget that Android has always had a barebones — albeit easily fooled — equivalent of the feature for years. Android Smart Lock's *Trusted face* was added in 2014 and has been accessible to users on all Android devices until recently. Now, it's completely gone from stock and OEM devices, running Android 10 or below.

POLL

What Chromecast models are you currently using?

- First-gen Chromecast
- Second-gen Chromecast

*"Face unlock is more widely available on smartphones nowadays, but many of us seem to forget that Android has always had a barebones — **albeit easily fooled** — equivalent of the feature for years. Android Smart Lock's **Trusted face** was added in 2014 and has been accessible to users on all Android devices until recently. Now, it's completely gone from stock and OEM devices, running Android 10 or below."*

*"It didn't use any biometric data for security, instead just relying on your face to unlock your device. A **photo could easily fool** it."*

https://www.androidpolice.com/2019/09/04/trusted-face-smart-unlock-method-has-been-removed-from-android-devices/

18

18

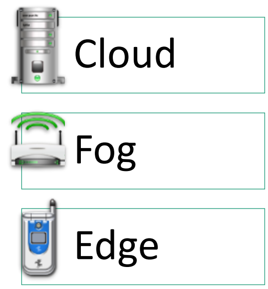
Security and privacy posture of an ML pipeline

20

20

AI-centric applications

- **Example applications**
  - Mobile: e.g. biometric authentication, keyboard word prediction, …
  - Desktop: e.g. email spam filters, games, browsers, …
  - Online service: e.g. online translations, voice and smart home assistants, …
- **Training/testing vs. production**
  - Local architecture
  - Distributed architecture

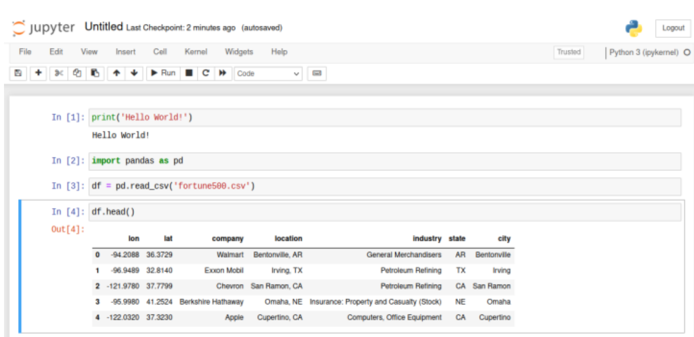


21

21

AI-centric applications

- Jupyter Notebook: Browser-based interface to Python and ML code
  - Local (e.g. laptop): $ jupyter notebook
  - Remote (e.g. GPU server): $ jupyter notebook --no-browser --port=8080



22

22

AI-centric applications

- Jupyter Notebook: Browser-based interface to Python and ML code
  - Local (e.g. laptop): $ jupyter notebook
  - Remote (e.g. GPU server): $ jupyter notebook --no-browser --port=8080
- Security based on tokens:
  - http://localhost:8080/?token=7153d0758dbd9ed3b346301558491b10f6c78fbc7a73f307
  - Secure connection to remote server via ssh tunnelling (of port 8080)
- What if the token is leaked?
  - Other user may run arbitrary code …

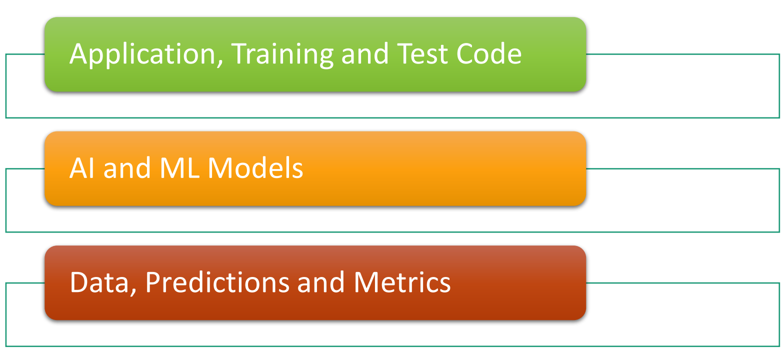
23

23

Characteristics of an ML pipeline

Artificial Intelligence (AI) and Machine Learning (ML) **add value and complexity** to contemporary software systems and applications

From DevOps to MLOps

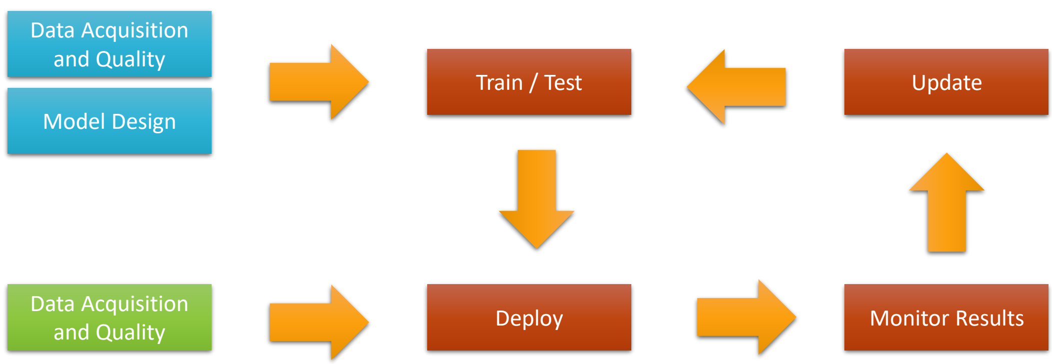


24

24

Characteristics of an ML pipeline

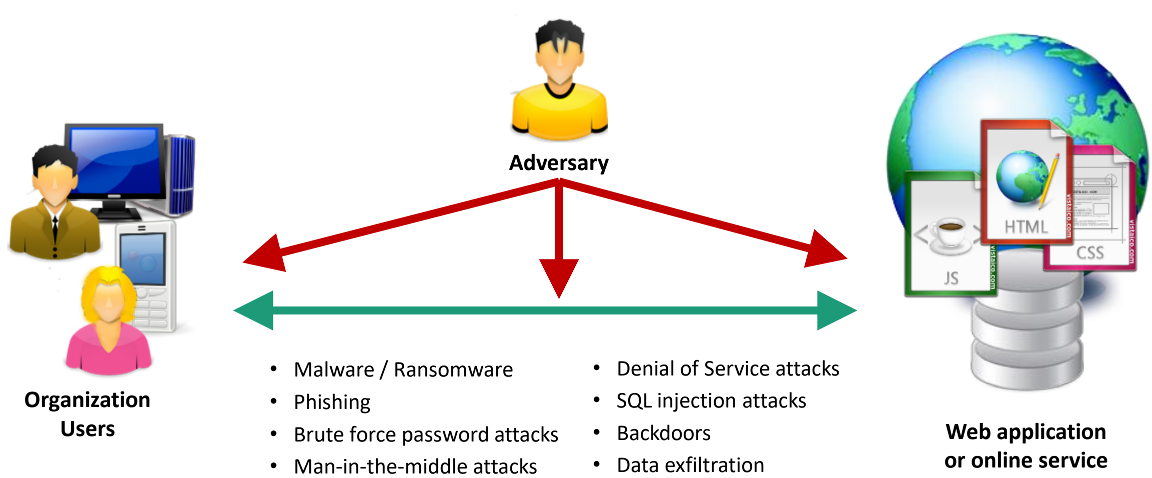
Artificial Intelligence (AI) and Machine Learning (ML) **add value and complexity** to contemporary software systems and applications



25

25

Stakeholders, assets and attack surface



- Malware / Ransomware
- Phishing
- Brute force password attacks
- Man-in-the-middle attacks
- Cross-site scripting
- Denial of Service attacks
- SQL injection attacks
- Backdoors
- Data exfiltration
- …

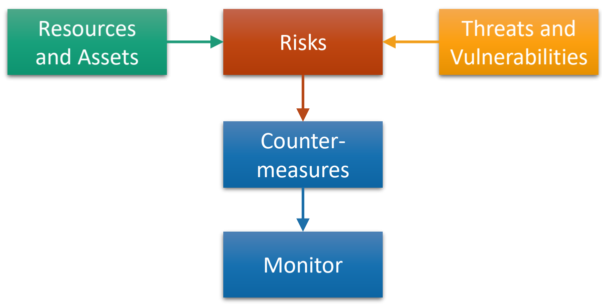
26

26

Stakeholders, assets and attack surface

Attack surface:

- All the points in the system where an adversary can launch attack
- Data, application logic and infrastructure (cloud and on-premise)
- Security building blocks that protect data, application logic and infrastructure



27

27

Attack surface of AI-centric applications

28

28

Attack surface of AI-centric applications

Application, Training and Test Code

- **Software exploits**: abuse vulnerable ML library due to software bugs
- **Unauthorized access**: leverage software or service end-points (e.g. training, upgrade production model) without permission
- **Tracking:** observe the use of the ML application to learn or expose interests of user

29

29

Attack surface of AI-centric applications

Application, Training and Test Code

- **Arbitrary code execution in recent Tensorflow :**
  - Deserializing a Keras model from YAML format, implementation uses yaml.unsafe_load, arbitrary code execution on the input
  - CVE-2021-37678 (13 august, 2021), CVSS Score: 9.3 – Critical Severity
  - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-37678
  - Fixed in Tensorflow 2.6
  - https://github.com/tensorflow/tensorflow/commit/23d6383eb6c14084a8fc3bdf164043b974818012

30

30

Attack surface of AI-centric applications

AI and ML Models

- **Steal model**: replicate confidential model without authorization
- **Infer membership**: figure out if input sample was used during training
- **Model inversion:** reconstruct training data from model parameters
- **Backdoor**: enforce particular model behaviour for given trigger
- **Transfer learning attack:** fine-tuning and repurpose an already trained but compromised model

31

31

Attack surface of AI-centric applications

AI and ML Models

- **Model inversion:** reconstruct training data from model parameters

Face recognition classifier producing labels and probabilities

Original face image (right) and restored one through model inversion (left)

Fredrikson et al., *Model inversion attacks that exploit confidence information and basic countermeasures,* CCS, 2015

32

32

Attack surface of AI-centric applications

Data, Predictions and Metrics

- **Data trustworthiness**: integrity of compromised sensors or public datasets
- **Unauthorized access**: modification of data without permission
- **Poison model**: corrupt model by adding malicious training samples
- **Hijack output:** interpose between ML system and receiver of predictions
- **Looped inputs/outputs:** use own predictions to train model

33

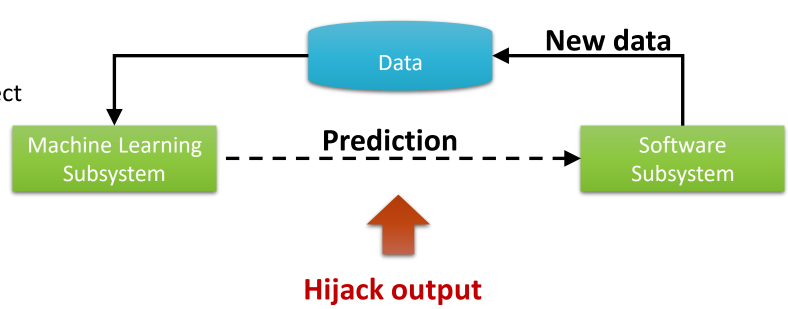
33

Attack surface of AI-centric applications

Data, Predictions and Metrics

- **Hijack output:** interpose between ML system and receiver of predictions

- Lane and object detection
- Traffic sign recognition



- Sensors and actuators
- Autonomous driving

34

34

Security by design

- 10 important security principles
  1. Secure the weakest link
  2. Practice defence in depth
  3. Fail securely
  4. Follow the principle of least privilege
  5. Compartmentalize
  6. Keep it simple
  7. Promote privacy
  8. Remember that hiding secrets is hard
  9. Be reluctant to trust
  10. Use community resources

Cfr. chapter 5 "*Guiding Principles for Software Security*" of Viega, John & McGraw, Gary. Building Secure Software: How to Avoid Security Problems the Right Way. Boston, MA: Addison-Wesley, 2002

J.H. Saltzer and M.D. Schroeder, *The protection of information in computer systems,* In Proceedings of the IEEE, 63(9):1278–1308, 1975

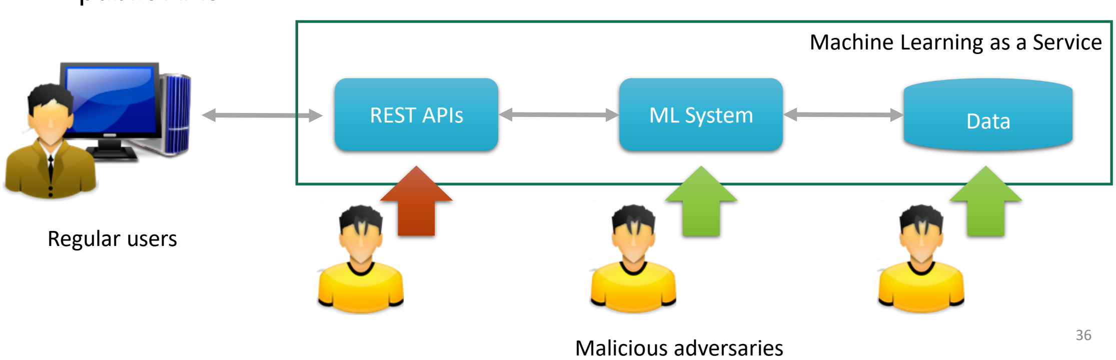
35

35

Security by design

1. Secure the weakest link

- **ML Example**:
  - MLaaS: Protect the underlying data and the ML system before protecting the public APIs



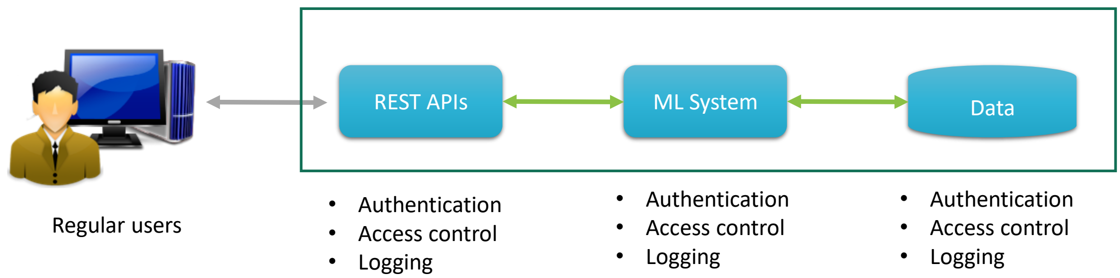
36

36

Security by design

2. Practice defence in depth

- **ML Example**:
  - Implement authentication and access control and maintain records of queries and responses to mitigate information leakage



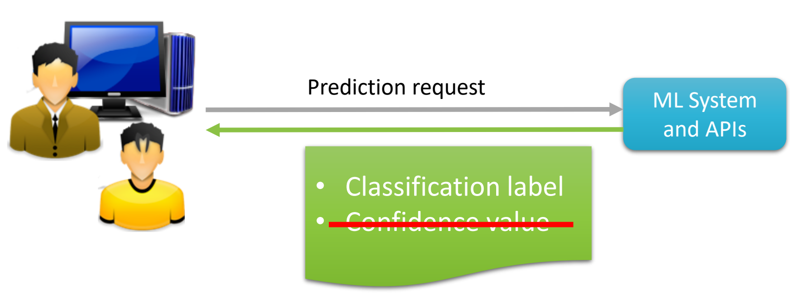
37

37

Security by design

3. Fail securely – Use secure defaults

- **ML Example**:
  - Do not provide classification results when ML system has little confidence in the predications to mitigate the creation of adversarial examples



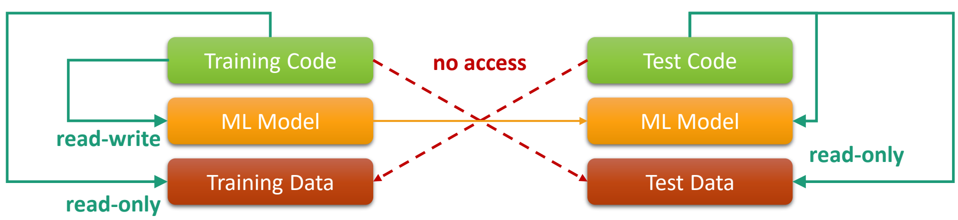
38

38

Security by design

4. Follow the principle of least privilege

- **ML Example**:
  - Implement access control for the different steps in the lifecycle of the ML system to limit data exposure to authorized entities as briefly as possible.



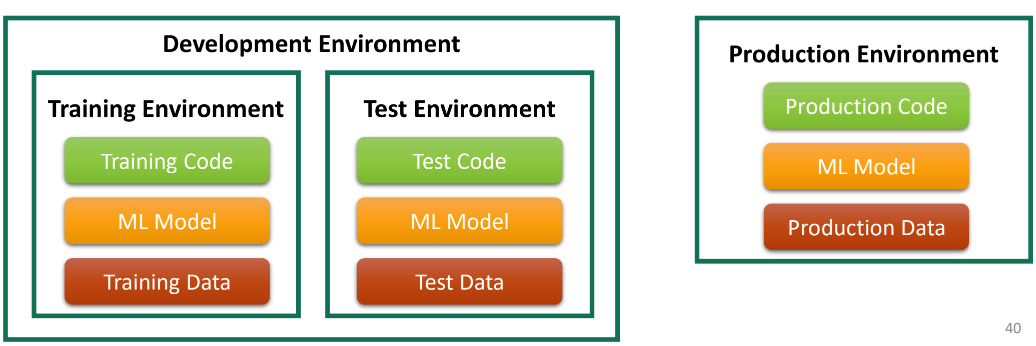
39

39

Security by design

5. Compartmentalize

- **ML Example**:
  - Separate the sensitive data flows between the different components of the ML system to minimize information leakage in case of compromise.



40

40

Security by design

6. Keep it simple

- **ML Example**:
  - When multiple ML algorithms and/or models are adequate (e.g. in terms of accuracy and computational overhead), choose those algorithms or models that are better understood as a way to not miss security threats.

Decision tree versus Deep neural network

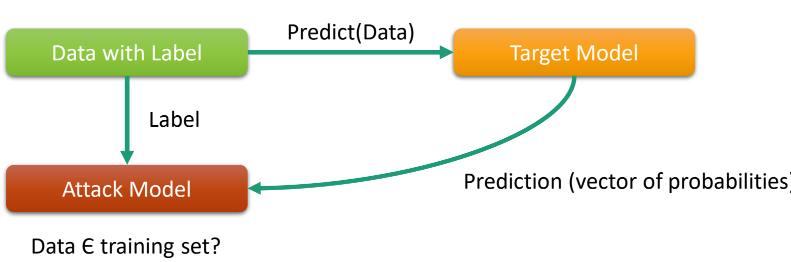
41

41

Security by design

7. Promote privacy

- **ML Example**:
  - Implement differential privacy (random noise with a certain distribution), hide confidence levels (cfr. membership inference attacks)



Shokri et al., *Membership inference attacks against machine learning models,* IEEE S&P, 2017

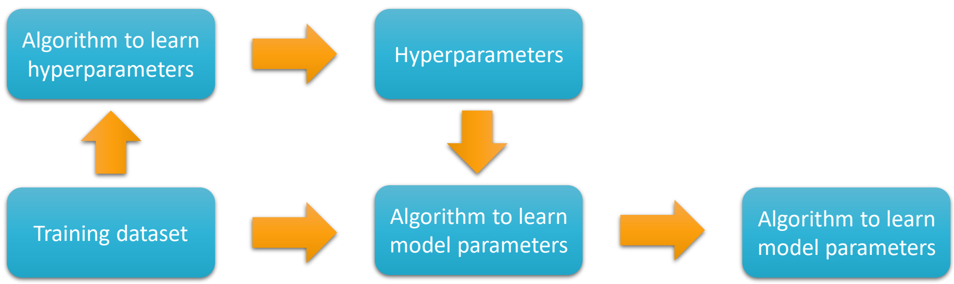
42

42

Security by design

8. Remember that hiding secrets is hard – No security by obscurity

- **ML Example**:
  - Do not assume that the configuration or the hyperparameters of an ML model remain secret



Wang et al., *Stealing Hyperparameters in Machine Learning*, IEEE S&P, 2018

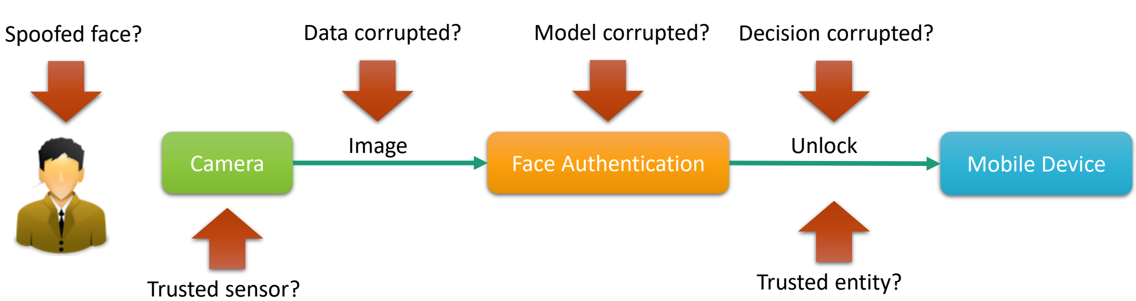
43

43

Security by design

9. Be reluctant to trust – Never trust, always verify

- **ML Example**:
  - In ML systems, many components depend on one another. Verify the data collection, pre-processing, normalization, training, …



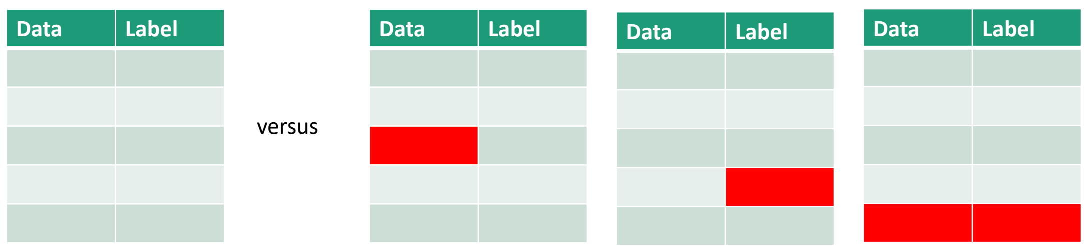
44

44

Security by design

10. Use community of resources

- **ML Example**:
  - Validate the authenticity and quality of public datasets and ML models as they may have been manipulated by malicious adversaries (cfr. Backdoor attacks)



Original dataset

Datasets with corrupted data and/or labels

Traditional countermeasures and best practices

- **Identity and access control**
  - User, application and service accounts
  - Roles and permissions
  - Strong authentication
  - Monitor for account and usage anomalies

- **Management**
  - Principle of least privilege
  - Secure deployment and configuration
  - Monitoring, logging and auditing
  - Backup and recovery

Traditional countermeasures and best practices

- **Network architecture**
  - Network isolation, virtual networks and segmentation
  - Disable access to internet or external networks
  - Deploy intrusion detection/prevention systems (IDS/IPS)
  - Protect against data exfiltration
  - Use end point detection and recovery (EDR)

- **Data encryption in transit and at rest**
  - Datasets and model artefacts
  - ML jobs (training, hyperparameter tuning, processing, …)
  - Logs and backups
  - Network traffic

47

47

Traditional countermeasures and best practices

- ML as a Service / ML in the cloud:

  - Build a Secure Enterprise Machine Learning Platform on AWS
    - https://docs.aws.amazon.com/whitepapers/latest/build-secure-enterprise-ml-platform/build-secure-enterprise-ml-platform.html

  - Azure security baseline for Azure Machine Learning
    - https://docs.microsoft.com/en-us/security/benchmark/azure/baselines/machine-learning-security-baseline

Amazon Machine Learning

AZURE Machine Learning

48

48

Approach on securing an AI application

- **Mitigating traditional vulnerabilities**
  - Educating users
  - Policy enforcements
  - Finding flaws with code reviews and bug bounties

- **Mitigating AI vulnerabilities**
  - Traditional defences can be applied to protect against some AI attacks
  - ML algorithms themselves have limitations that allow for attacks
  - Strategies to make those attacks more difficult to execute
  - Limit dependency on AI (e.g. availability of non-AI methods) to reduce impact

49

49

Adversarial machine learning

50

50

Basic ML concepts and terminology

- Train ML model that generalizes to a dataset not seen before

Distribution Training Dataset $\approx$ Distribution Test Dataset

- Independent and identically distributed (IID): All train and test examples drawn independently from same distribution
- If distributions are different or if there are sampling dependencies, then algorithm will not generalise after deployment
- Datasets are generated independent of the ML model

51

51

Basic ML concepts and terminology

Loss or cost function $J(\theta, x, y)$ of model with parameters/weights θ:

- How good is model at making predictions?
- Slope of curve indicates how to update parameters for better prediction
- Gradient descent: use derivative for direction of greatest increase
- Minimize cost or loss when gradient is 0

- Gradient:
  - Measures how much the output of a function changes if you change weights a little bit
  - Partial derivative with respect to its weights
  - ML or DL: measures the change in all weights with regard to the change in error

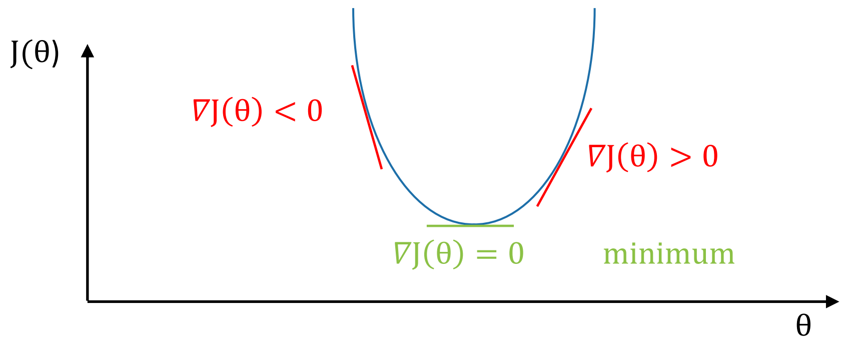
52

52

Basic ML concepts and terminology

Loss or cost function $J(\theta, x, y)$ of model with parameters/weights θ:

- How good is model at making predictions?
- Slope of curve indicates how to update parameters for better prediction
- Gradient descent: use derivative for direction of greatest increase
- Minimize cost or loss when gradient is 0



Basic ML concepts and terminology

- Train ML model that generalizes to a dataset not seen before

- **Kitten**: ‘pointy’ ears and ‘longer’ tail
- **Puppy**: ‘floppy’ ears and ‘shorter’ tail

Basic ML concepts and terminology

- Test ML model

  - ML model may not generalize well

Basic ML concepts and terminology

- ML in adversarial setting: exploit IID assumption
  - Attackers provide unusual inputs

Eykholt et al., *Robust Physical-World Attacks on Deep Learning Models*, CVPR, 2017, https://arxiv.org/abs/1707.08945
Sitawarin et al. *DARTS: Deceiving Autonomous Cars with Toxic Signs*, 2018, https://arxiv.org/abs/1802.06430

Basic ML concepts and terminology

Norms and distances:

- Example: Vector x = (2, 3, 0)

- **L_0 norm**: counts the total number of non-zero elements of a vector (not really a norm due to scaling property)
  - $||x||_0 = 2$
- **L_1 norm** (Manhattan distance): sum of absolute values of the elements
  - $||x||_1 = |2| + |3| + |0| = 5$
- **L_2 norm** (Euclidean distance): square root of the sum of the squares of the elements
  - $||x||_2 = \sqrt{|2|^2 + |3|^2 + |0|^2} = \sqrt{4 + 9 + 0} = 3.606$
- **L_∞ norm**: largest magnitude among each element of a vector
  - $||x||_\infty = \max(2, 3,0) = 3$

57

57

Adversarial machine learning

What is adversarial machine learning (AML)?

"AML explores the degree to which AI systems can be compromised by contaminating training data, by modifying algorithms, or by making subtle changes to an object that prevent it from being correctly identified"

National Science and Technology Council 2016

58

58

Adversarial machine learning

- AI attacks against AI-centric applications

Attack	CIA triad
Evasion/perturbation attack	Integrity (model)
Poisoning attack	Integrity (data)
Model backdoors	Integrity (model)
Membership inference	Confidentiality (data)
Model stealing	Confidentiality (model)
Model inversion	Confidentiality (data)
Software dependency exploits	Integrity, confidentiality and availability

59

59

Evasion attack: spam filter

- **Example**: Spam filters and evading detection:
  - Use capitalization, non-Latin characters or spurious punctuation
  - Misspell words
  - Use synonyms
  - Add non-spam words
  - No text but graphical image
  - Embed bogus HTML codes in text
  - …

- Adversaries manipulate mail to produce false negatives

Dalvi et al., *Adversarial Classification,* 2004

60

60

Evasion attack: spam filter

- **Spam filtering setting:**
  - End-users train their spam filter on their own mailbox
  - Using both spam and ham examples
  - Number of tokens is fixed, e.g. 100000
  - Most-infrequently seen tokens expire
- **Threat model:**
  - Spammer's objective is to evade detection
  - Spammer has no access to classifier
  - Spammer has knowledge about the algorithms used

61

61

Evasion attack: spam filter

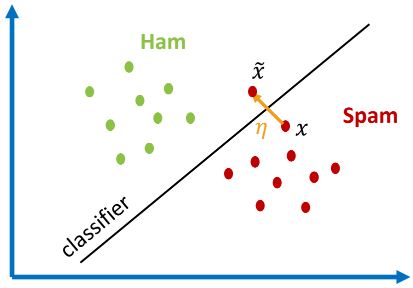
- Spam classifier: linear classification model
  - Feature vector x: tokens, words, character sequences
  - Weighs w: viagra = 3.0, buy = 1.0, lecture = -2.0
  - Spam if total score (weighted sum) is larger than threshold 3, else ham
  - Mail: "buy viagra" => spam
  - Perturbed mail: "buy viagra lecture" => ham
- **Adversarial examples:** *"inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake"*

Goodfellow et al., *Attacking machine learning with adversarial examples*, 2017

62

62

Evasion attack: adversarial examples



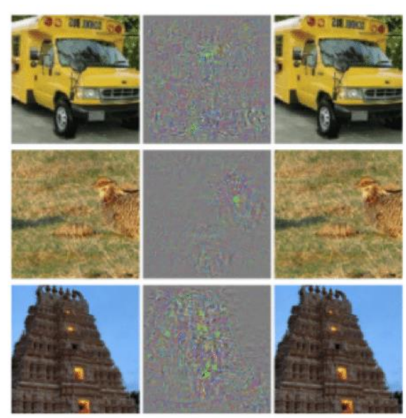
- Maximizing perturbation with small changes
  - Adversarial example: $\tilde{x} = x + \eta$ with small noise $||\eta|| < \epsilon$
  - Dot product with weight vector w: $w^{\mathrm{T}}\tilde{x} = w^{\mathrm{T}}x + w^{\mathrm{T}}\eta$
  - Activation increases with $w^{\mathrm{T}}\eta$
  - Perturbation is maximized if $\eta = sign\,(w)$
  - Small changes in activation can grow linearly with increase in dimensions of w

63

63

Evasion attack: adversarial examples

- Szegedy was the first to observe adversarial examples in deep neural networks
- AlexNet CNN architecture
  - Left: correctly predicted sample: “schoolbus”
  - Middle: difference with correct image
  - Right: misclassified image: “ostrich”
- Same perturbation can cause a different network, trained on different subset of dataset, to misclassify the same input.

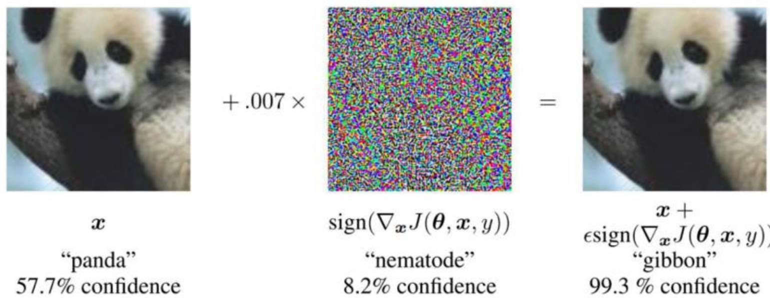


Szegedy et al., *Intriguing properties of neural networks,* ICLR, 2014, https://arxiv.org/abs/1312.6199

64

64

Evasion attack: adversarial examples



- Neural network vulnerable to adversarial examples:
  - Adversarial noise maximizes classification error
  - Difference between example and test sample indistinguishable to human eye

Goodfellow et al., *Explaining and harvesting adversarial examples,* ICLR, 2015

65

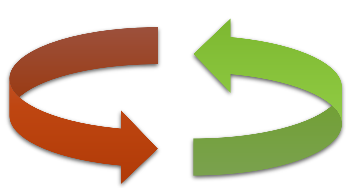
65

Evasion attack: adversarial examples

- Observation:
  - Adversarial example is not neutral, but **dependent on classifier**
  - Noise is not stochastic, but crafted to **maximize classification error**
  - Transferability of adversarial examples: **Other ML models** and neural networks with different architectures trained on subsets of same dataset **often misclassify adversarial example too**

- Arms race:

Adversary:
- Analyse classifier
- Design attack



Defender:
- Analyse attack
- Design defense

66

66

Evasion attack: adversarial examples

- Attacker's goal:

Misclassification

- Change the class with or without a specific target class

Confidence reduction

- Do not change target class, but impact confidence of prediction

67

67

Evasion attack: adversarial examples

- Adversarial perturbation:

Individual

- Adversarial perturbation can be applied to only one input

Universal

- Adversarial perturbation can be applied to many inputs

68

68

Evasion attack: adversarial examples

- Universal adversarial perturbation
  - Fake glasses
  - Physical adversarial example
- Attack a face recognition system
  - Impersonation attack
  - Identity theft

Sharif et al., *Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition,* CCS 2016, https://dl.acm.org/doi/10.1145/2976749.2978392

https://github.com/mahmoods01/accessorize-to-a-crime

69

69

Evasion attack: adversarial examples

- Generating adversarial perturbation:

Single step
- Fast but less accurate (e.g. FGSM)

Iterative methods
- More costly, requires multiple iterations (e.g. BIM, PGD, DeepFool)

70

70

Evasion attack: threat model

Characteristics of an adversary:

Objectives

- Adversary aims for (un)targeted misclassification (e.g. FN=evasion, FP=DoS attack)

Knowledge

- Information the adversary can leverage or exploit to successfully launch the attack

Capabilities

- Adversary can craft adversarial examples

71

71

Evasion attack: threat model - knowledge

White-box

- Adversary has complete access to classifier $F: X \rightarrow Y$

Black-box

- Adversary has oracle access to classifier F
- For given x, retrieve F(x)

Grey-box

- Black box + some other information
- E.g. learning algorithm, hyperparameters, defenses, …

72

72

Generate adversarial examples: white-box

White-box: Adversary has complete access to classifier F

- Scenario by adversary:
  - Given: $x \in X$
  - Find: η
    - $\min_{\eta} ||\eta||$ such that $F(\tilde{x}) = F(x + \eta) \in S$ with $S \subseteq Y$
  - **Untargeted attack**: $S = Y - F(x)$
  - **Targeted attack**: $\{ s \}$

73

73

Generate adversarial examples: white-box (Fast Gradient Sign Method)

White-box: Adversary has complete access to classifier F

- **Attack algorithm: Fast-Gradient Sign Method (FGSM)**
  - Step in the direction of the gradient of the loss function
  - Training example $(x, y) = (x, F(x))$ and model parameters θ
  - Loss function $J(\theta, x, y)$ for training example (x, y)
  - Step in direction of gradient of loss function: $\eta = \epsilon \, \mathrm{sign}(\nabla_x J(\theta, x, y))$
  - ϵ is smaller than precision of features (e.g. 8 bits for pixel)
  - Adversarial example: $\tilde{x} = x + \eta = x + \epsilon \, \mathrm{sign}(\nabla_x J(\theta, x, y))$

Goodfellow et al., *Explaining and harvesting adversarial examples,* ICLR, 2015

74

74

Generate adversarial examples: white-box (Fast Gradient Sign Method)

White-box: Adversary has complete access to classifier F

$\boldsymbol{x}$ "panda" 57.7% confidence

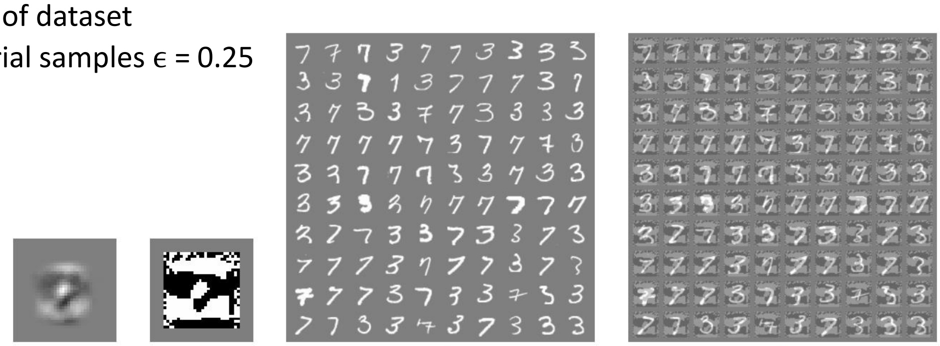
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "nematode" 8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "gibbon" 99.3 % confidence

Goodfellow et al., *Explaining and harvesting adversarial examples,* ICLR, 2015

Generate adversarial examples: white-box (Fast Gradient Sign Method)

- 3 and 7 from MNIST, binary classification with logistic regression
  - a) Weights logistic regression model
  - b) Sign of weights
  - c) Samples of dataset
  - d) Adversarial samples ϵ = 0.25
- Error rate:
  - c) 1.6 %
  - d) 99 %



(a) (b) (c) (d)

Generate adversarial examples: white-box (Targeted FGSM)

- Targeted misclassification with FGSM: **T-FSGM**
  - Compute gradient step as before
  - Now in direction of **negative** gradient w.r.t. the target class y_{target}
  - Minimize loss, or maximize the probability of some specific target class

$$\tilde{x} = x - \epsilon \, \mathrm{sign}(\nabla_x J(\theta, x, \mathbf{y_{target}}))$$

77

77

Generate adversarial examples: white-box (Iterative FGSM or BIM)

- Iterative misclassification with FGSM: **I-FSGM (or BIM)**
  - From single step to iterative variation of FGSM
  - Iterate T steps

$$\tilde{x}_0 = x$$

$$\alpha = \frac{\epsilon}{T}$$

$$\tilde{x}_{t+1} = \tilde{x}_t + \alpha \, \mathrm{sign}(\nabla_x J(\theta, \tilde{x}_t, y))$$

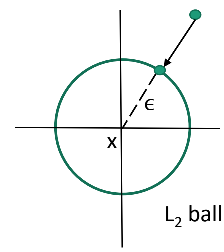
78

78

Generate adversarial examples: white-box (Projected Gradient Descent)

White-box: Adversary has complete access to classifier F

- **Attack algorithm: Projected Gradient Descent (PGD)**
  - $B(x, \epsilon)$: An ϵ ball around x with norm L_p
  - $Proj_{B(x, \epsilon)}(y)$: project y into ball $B(x, \epsilon)$



- Iterative process: $x_{k+1} = Proj_{B(x, \epsilon)}(x_k + \epsilon \, \mathrm{sign}(\nabla_x J(\theta, x, y)))$
  1. Start from random perturbation in the L_p ball around sample x
  2. Take gradient step in the direction of greatest loss
  3. Project perturbation back into L_p ball if necessary
  4. Repeat 2–3 until convergence

79

79

Generate adversarial examples: white-box

Attack
Random noise attack
Semantic attack
Fast Gradient Sign Method (FGSM)
Basic Iterative Method (BIM) or I-FGSM (Iterative FGSM)
Projected Gradient Descent (PGD)
DeepFool
Carlini & Wagner (C&W)
Jacobian-based Salience Map Attack (JSMA)

81

81

Defense against adversarial examples

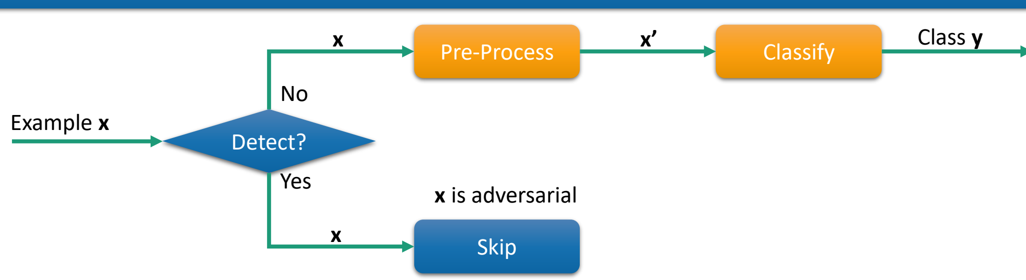
1. Adversarial examples detection
   - **Auxiliary detection model**
   - Statistical detection methods (e.g. principal component analysis)
   - Prediction consistency methods (e.g. feature squeezing in fewer bits)
2. Gradient masking/obfuscation: break calculation of gradients
   - Input transformations: crop, rescale, bit depth reduction, JPEG compression
   - **Defensive distillation**
3. Robust optimization
   - **Adversarial training**
   - Certified defenses: robustness of model w.r.t. certain metric (e.g. lower bound of minimal perturbation, upper bound to adversarial loss)

82

82

Defense against adversarial examples

Auxiliary detection model



- **Detect**: binary classifier or autoencoder to detect adversarial examples
- **Pre-process**: clean adversarial perturbations (e.g. denoising autoencoder)

Meng et al., MagNet: A Two-Pronged Defense against Adversarial Examples (CCS 2017), https://dl.acm.org/doi/10.1145/3133956.3134057

83

83

Defense against adversarial examples

Defensive distillation

- First network is trained with hard labels to get maximum accuracy, temperature T=40..50

$$F(\theta, x) = \mathrm{softmax}(\frac{Z(\theta, x)}{T})$$

- Evaluate first network on each sample in training set to produce soft labels (e.g. MNIST dataset, 70% it is a '7' and 30% it is a '1')
- Second distillation network trains on soft labels with temperature T to predict the class probabilities generated by the first network
- Evaluate with distillation network using temperature T = 1

Papernot et al., *Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks*, S&P, 2016, https://arxiv.org/abs/1511.04508, https://secml.github.io/class3/

Defense against adversarial examples

Adversarial training

- Adversarial training as a regularizer
  - Data augmentation with adversarial examples
  - Trade-off: adversarial training more robust against adversarial attack, but possibly a lower performance on normal examples
  - Adversarial objective function based on FGSM
  - $\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J\left(\theta, x + \epsilon\, \mathrm{sign}\left(\nabla_x J(\theta, x, y)\right)\right)$

Goodfellow et al., *Explaining and harvesting adversarial examples,* ICLR, 2015

Arms race

Adversarial perturbation / evasion attack (test phase)

- 10 defenses from different categories
  - Secondary classification based methods
  - Principal component analysis
  - Distributional detection
  - Normalization detection
- All detection methods were bypassed

Carlini and Wagner. *Adversarial examples are not easily detected: Bypassing ten detection methods*. AISec 2017, https://arxiv.org/abs/1705.07263

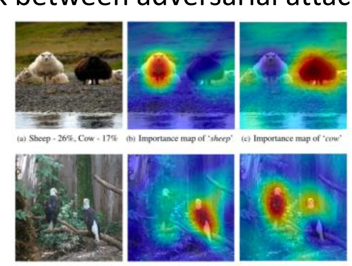
86

86

Arms race

Adversarial perturbation / evasion attack (test phase)

- Unsupervised learning can detect unknown adversarial attacks
  - CMU and KAIST found link between adversarial attacks and explainability
  - Saliency maps



Ko et al., *Unsupervised Detection of Adversarial Examples with Model Explanations*. AdvML workshop 2021, https://arxiv.org/abs/2107.10480

87

87

Generate adversarial examples: black-box

Black-box: Adversary has oracle access to classifier F

- Scenario by adversary: Attacking remotely hosted black-box models
- **Query-based attacks**: Query model and create adversarial examples by using target model's response
  - Score-based attacks: output class probabilities (soft label)
  - Decision-based attacks: output class (hard label)
- **Transfer-based attacks** (a.k.a. zero query attacks): Adversary trains own local surrogate model and transfers adversarial to target model

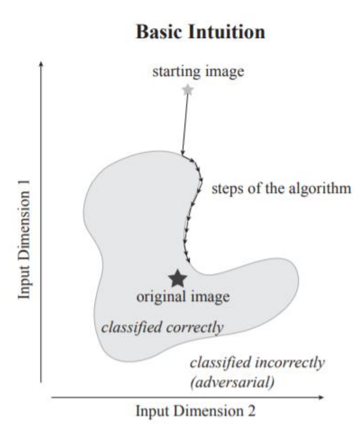
88

88

Generate adversarial examples: black-box

Black-box: Adversary has oracle access to classifier F

- **Boundary attack**: decision-based attack
  1. Start with input example that belongs to specific class and corresponding target class to misclassify example as
  2. Select random point on decision boundary between original class and the target class
  3. Generate small perturbation that can be added to the starting point to move it closer to target class while still remaining on decision boundary
  4. If model misclassifies perturbed example, attack is successful and process ends. Otherwise, select new starting point and repeat 2-4



89

Generate adversarial examples: black-box

Black-box: Adversary has oracle access to classifier F

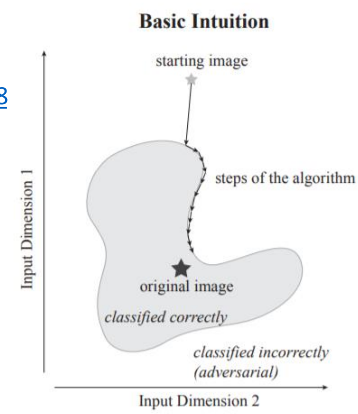
- **Boundary attack**: decision-based attack

Brendel et al., *Decision-based adversarial attacks: Reliable attacks against black-box machine learning models* (2017). https://arxiv.org/abs/1712.04248

- A lot of queries needed:
  - Traditional defense: Access control and rate limiting
  - Sybil attack: Detect suspicious queries?

- Defense: Stateful detection due to similar queries

Chen et al,. *Stateful detection of black-box adversarial attacks* (2020), https://dl.acm.org/doi/abs/10.1145/3385003.3410925



90

Generate adversarial examples: black-box

Black-box: Adversary has oracle access to classifier F

- **Transfer-based attack**:

1. Adversary queries remote ML system for labels on inputs of its choice
2. Adversary uses labeled data to train a local substitute for remote ML system

Papernot et al., *Practical Black-Box Attacks against Machine Learning*, AsiaCCS, 2017, https://arxiv.org/abs/1602.02697

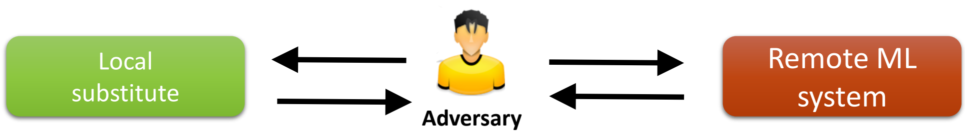
91

91

Generate adversarial examples: black-box

Black-box: Adversary has oracle access to classifier F

- **Transfer-based attack**:



3. Adversary selects new synthetic inputs for remote queries based on local substitute's output sensitivity to input variations
4. Adversary uses local substitute to craft adversarial examples (transferability)

Papernot et al., *Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples*, 2016, https://arxiv.org/abs/1605.07277

92

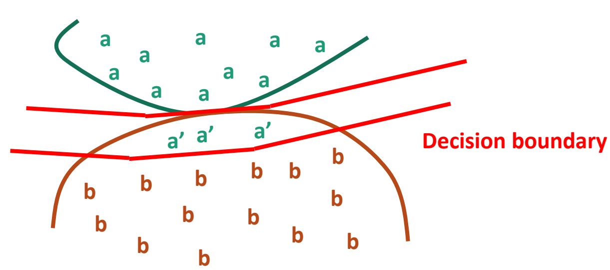
92

Other ML threats and mitigations

Data poisoning attack (training phase)

- **Description**: contaminate ML model during training phase to modify predictions on new data during testing phase

- Push the decision boundary:
  - Label modification
  - Data modification
  - **Data injection**



94

94

Other ML threats and mitigations

Data poisoning attack (training phase)

- **Description**: contaminate ML model during training phase to modify predictions on new data during testing phase
- **Traditional countermeasures**:
  - Input validation and sanitization (e.g. SQL injection attacks)
  - Strong access control and integrity checking
- **ML countermeasures:**
  - Frequently measure outliers/skew/drift in training data
  - Reject On Negative Impact (RONI) defence: remove suspect attack or noisy training samples with substantial negative impact on classification accuracy

95

95

Other ML threats and mitigations

ML backdoor attack (training phase)

- **Description**: Inject additional behaviour by tampering with training data to deliver Trojaned ML model that forces targeted misclassifications when trigger is present [1]
- **Traditional countermeasures**:
  - Cfr. Data poisoning attack
- **ML countermeasures:**
  - Train sensitive models in-house
  - Input filtering, neuron pruning, unlearning

[1] Wang et al., Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks, IEEE SP 2019

96

96

Other ML threats and mitigations

Model inversion attack

- **Description**: reconstruct private training data by inferring private features from ML model by finding inputs that maximize confidence levels [1]

[1] Fredrikson et al., Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. CCS 2015

97

97

Other ML threats and mitigations

Model inversion attack

- **Description**: reconstruct private training data by inferring private features from ML model by finding inputs that maximize confidence levels [1]
- **Traditional countermeasures**:
  - Strong access control
  - Rate limiting
- **ML countermeasures:**
  - Omit confidence values from ML model output

[1] Fredrikson et al., Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. CCS 2015

98

98

Other ML threats and mitigations

Membership inference attack

- **Description**: determine whether specific training sample was part of the ML models training set or not [1]

[1] Shokri et al., Membership Inference Attacks Against Machine Learning Models, IEEE SP 2017

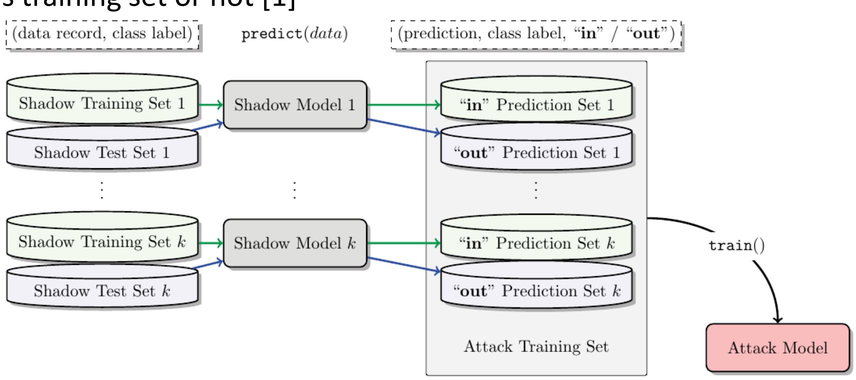
99

99

Other ML threats and mitigations

Membership inference attack

- **Description**: determine whether specific training sample was part of the ML models training set or not [1]



100

100

Other ML threats and mitigations

Membership inference attack

- **Description**: determine whether specific training sample was part of the ML models training set or not
- **Traditional countermeasures**:
  - Strong access control
  - Rate limiting
- **ML countermeasures:**
  - Differential privacy: add random noise (cfr. Tensorflow Privacy)
  - Omit confidence values from ML model output

101

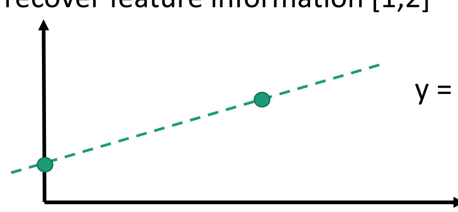
101

Other ML threats and mitigations

Model stealing or model extraction attack

- **Description**: copy or recreate the underlying ML model by legitimate querying of the model and use the model to create adversarial examples that transfer to the target model, or to invert it and recover feature information [1,2]

unknowns vs. # queries



y = **m** . x + **b**

[1] Tramèr et al., Stealing machine learning models via prediction APIs. SEC 2016

[2] Jagielski et al. High accuracy and high fidelity extraction of neural networks, USENIX Security, 2020, https://arxiv.org/abs/1909.01838

102

102

Other ML threats and mitigations

Model stealing or model extraction attack

- **Description**: copy or recreate the underlying ML model by legitimate querying of the model and use the model to create adversarial examples that transfer to the target model, or to invert it and recover feature information
- **Traditional countermeasures**:
  - Strong access control
  - Rate limiting
- **ML countermeasures:**
  - Omit or reduce accuracy of confidence values from ML model output
  - Distribution of consecutive API queries (stateful detection)

Juuti et al. *PRADA: protecting against DNN model stealing attacks,* EuroS&P 2019, https://arxiv.org/abs/1805.02628

103

103

Other ML threats and mitigations

Malicious MLaaS providers

- **Description**: malicious MLaaS provider can learn/leak private training data or ML models
- **Traditional countermeasures**:
  - Fully Homomorphic Encryption for secret data and secret models
  - Operate on plaintext data and models in Trusted Execution Environment (TEE)
- **ML countermeasures:**
  - ?

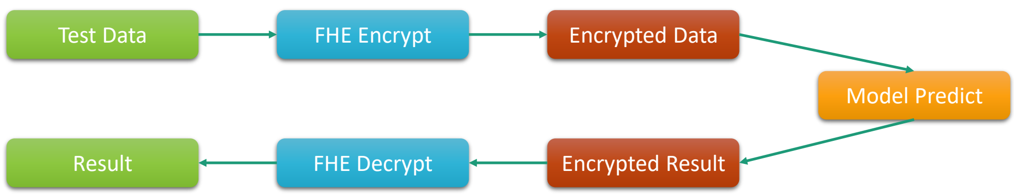
104

104

Other ML threats and mitigations

Malicious MLaaS providers

- **Description**: malicious MLaaS provider can learn/leak private training data or ML models



105

105

Other ML threats and mitigations

ML supply chain attacks

- **Description**: Large models are trained and shared by 3rd parties. They are often reused and slightly modified for new task at hand. Adversaries attack the base model and compromise the retuned model too.
- **Traditional countermeasures**:
  - Minimize dependencies on 3rd parties (data and models)
  - Strong access control and encryption
- **ML countermeasures:**
  - Cfr. Training phase attacks: poisoning and backdoor attacks?

106

106

Adversarial ML Frameworks

- CleverHans: http://www.cleverhans.io
- Foolbox: https://foolbox.jonasrauber.de
- ART: https://adversarial-robustness-toolbox.org
- RobustBench: https://robustbench.github.io
- DeepSec: https://github.com/ryderling/DEEPSEC

107

107

Artificial Intelligence (AI) and Machine Learning (ML) **add value and complexity** to contemporary software systems

But also increase the attack surface, imposing a holistic approach to secure the ML pipeline and lifecycle

108

To summarize

- Does your model learn the right concepts?
- It's an **arms race**
  - Many defences have been proposed ... and broken
  - There is no single line of defense, lot's of papers on https://arxiv.org
  - Not all inputs are images!
  - Check which ML attacks are relevant for your application
- Detect, defend, and prepare for **after the breach**!

109

109

References

- Carlini and Wagner. *Adversarial examples are not easily detected: Bypassing ten detection methods*. AISec 2017, https://arxiv.org/abs/1705.07263
- Carlini et al., *Towards Evaluating the Robustness of Neural Networks*, S&P, 2016, https://arxiv.org/abs/1608.04644
- Dalvi et al., *Adversarial Classification*, 2004, https://dl.acm.org/doi/10.1145/1014052.1014066
- Eykholt et al., *Robust Physical-World Attacks on Deep Learning Models*, CVPR, 2017, https://arxiv.org/abs/1707.08945
- Fredrikson et al., *Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*. CCS 2015, https://dl.acm.org/doi/10.1145/2810103.2813677
- Goodfellow et al., *Explaining and harvesting adversarial examples*, ICLR, 2015, https://arxiv.org/abs/1412.6572
- Goodfellow et al., *Attacking machine learning with adversarial examples*, 2017, https://openai.com/blog/adversarial-example-research/

110

110

References

- Jagielski et al. *High accuracy and high fidelity extraction of neural networks*, USENIX Security, 2020, https://arxiv.org/abs/1909.01838
- Juuti et al. *PRADA: protecting against DNN model stealing attacks*, EuroS&P 2019, https://arxiv.org/abs/1805.02628
- Kar et al. *Trends and applications in Stackelberg security games. Handbook of Dynamic Game Theory* (2017): 1-47, https://link.springer.com/10.1007/978-3-319-27335-8_27-1
- Ko et al., *Unsupervised Detection of Adversarial Examples with Model Explanations*. AdvML workshop 2021, https://arxiv.org/abs/2107.10480
- Papernot et al., *Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples*, 2016, https://arxiv.org/abs/1605.07277
- Papernot et al., *Practical Black-Box Attacks against Machine Learning*, AsiaCCS, 2017, https://arxiv.org/abs/1602.02697

References

- Papernot et al., *The Limitations of Deep Learning in Adversarial Settings*, EuroS&P, 2016, https://arxiv.org/abs/1511.07528
- Papernot et al., *Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks*, S&P, 2016, https://arxiv.org/abs/1511.04508
- Sharif et al., *Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition*, CCS 2016, https://dl.acm.org/doi/10.1145/2976749.2978392
- Shokri et al., *Membership Inference Attacks Against Machine Learning Models*, IEEE SP 2017, https://arxiv.org/abs/1610.05820
- Sitawarin et al. *DARTS: Deceiving Autonomous Cars with Toxic Signs*, 2018, https://arxiv.org/abs/1802.06430
- Szegedy et al., *Intriguing properties of neural networks*, ICLR, 2014, https://arxiv.org/abs/1312.6199
- Tramèr et al., *Stealing machine learning models via prediction APIs*. SEC 2016, https://arxiv.org/abs/1609.02943

References

- Tramèr et al., *AdVersarial: Perceptual Ad Blocking meets Adversarial Machine Learning*, CCS '19, 2019, https://arxiv.org/abs/1811.03194
- Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley, 2002
- Wang et al., *Stealing Hyperparameters in Machine Learning*, IEEE S&P, 2018, https://arxiv.org/abs/1802.05351
- Wang et al., *Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks*, IEEE SP 2019, https://cs.ucsb.edu/~bolunwang/assets/docs/backdoor-sp19.pdf
- Zhou et al. *Modeling adversarial learning as nested Stackelberg games*. Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham, 2016, https://dl.acm.org/doi/abs/10.1007/978-3-319-31750-2_28

Questions?

davy.preuveneers@kuleuven.be

Thank you!

https://distrinet.cs.kuleuven.be/

115