# Machine Learning Security in the Real World

**Dr. Maxime Cordy**

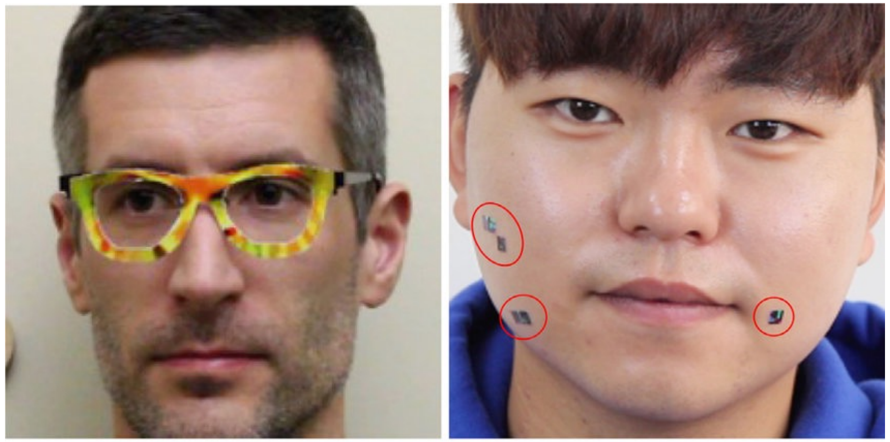
University of Luxembourg

15 December 2022

# Quality Assurance for Machine Learning: A Gentle Introduction

Part I

Crowd face recognition system



(a) (b)

Journal of Information Security and Applications 60 (2021) 102874

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Adversarial attacks by attaching noise markers on the face against deep face recognition

Gwonsang Ryu [a], Hosung Park [b], Daeseon Choi [c,*]

[a] Department of Software Convergence, Graduate School of Soongsil University, Seoul, 07027, South Korea
[b] Department of Cyber Security and Police, Busan University of Foreign Studies, Busan, 46234, South Korea
[c] Department of Software, Soongsil University, Seoul, 07027, South Korea

ARTICLE INFO

ABSTRACT

WILL KNIGHT BUSINESS NOV 19, 2019 9:15 AM

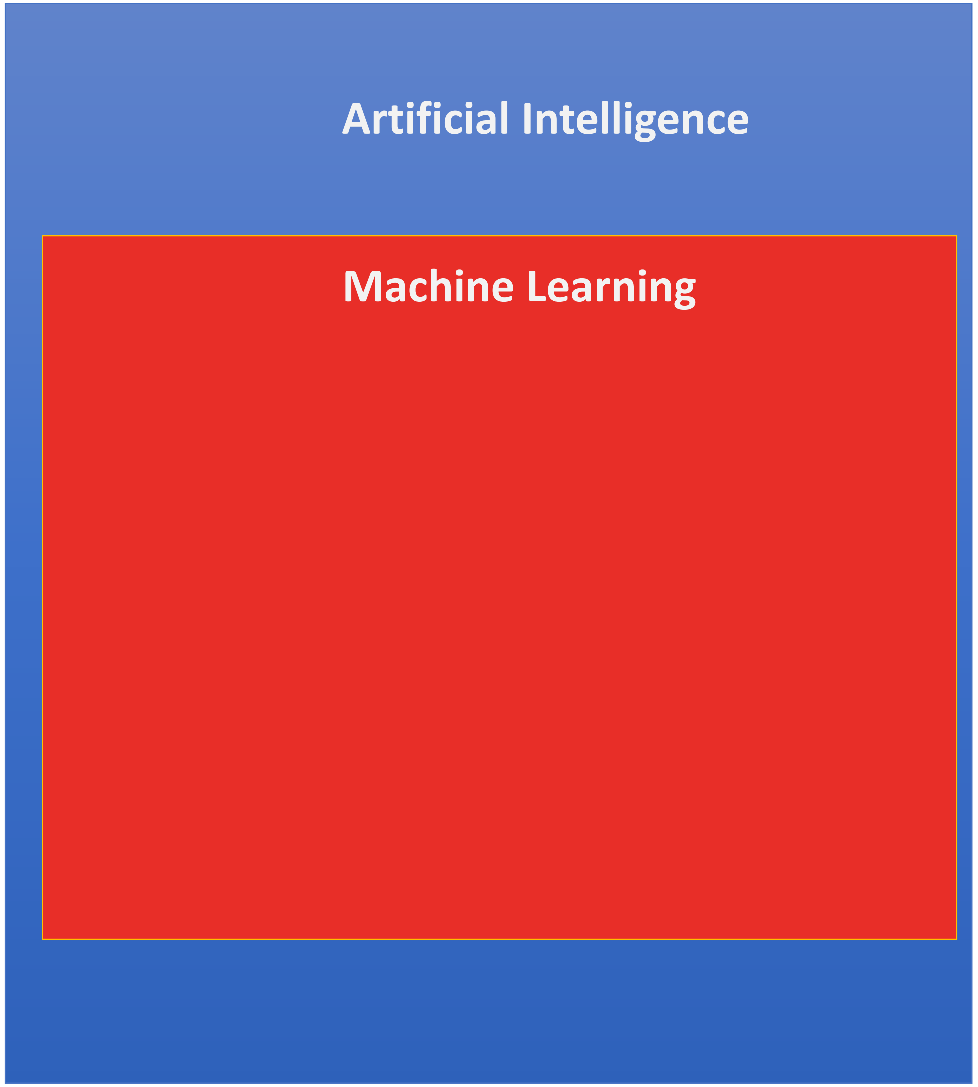
# The Apple Card Didn't 'See' Gender—and That's the Problem

**The way its algorithm determines credit lines makes the risk of bias more acute.**

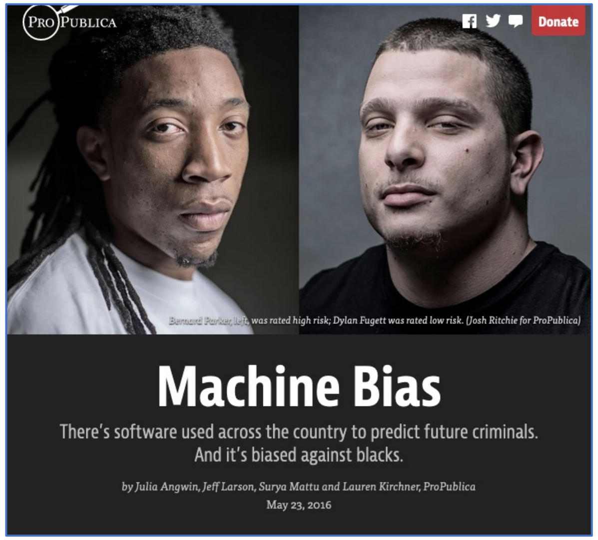
**THE APPLE CREDIT** card, launched in August, ran into major problems last week when users noticed that it seemed to offer smaller lines of credit to women than to men. The scandal spread on Twitter, with influential techies branding the Apple Card "fucking sexist," "beyond f'ed up," and so on. Even Apple's amiable cofounder, Steve Wosniak, wondered, more politely, whether the card might harbor some misogynistic tendencies.

# Machine learning

- **Machine learning**: a subfield of artificial intelligence building "*methods that **'learn'**, that is, methods that **leverage data** to improve performance on some set of tasks*" (Tom Mitchell)



# Machine learning software can be inaccurate and fooled


PROPUBLICA Donate

Bernard Parker, left, was rated high risk; Dylan Fugett was rated low risk. (Josh Ritchie for ProPublica)

## Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

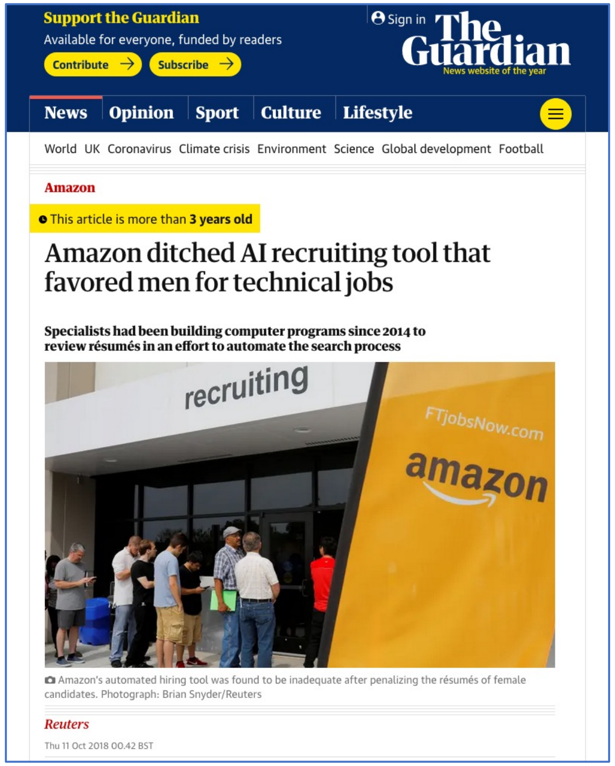
by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica
May 23, 2016

The Verge / Tech / Reviews / Science / Entertainment /

SCIENCE / HEALTH / TECH

**IBM's Watson gave unsafe recommendations for treating cancer /** Doctors fed it hypothetical scenarios, not real patient data

By ANGELA CHEN / @chengela
Jul 26, 2018, 10:29 PM GMT+2 | 0 Comments


Support the Guardian
Available for everyone, funded by readers
Contribute → Subscribe →
Sign in
The Guardian
News website of the year
News Opinion Sport Culture Lifestyle
World UK Coronavirus Climate crisis Environment Science Global development Football

Amazon

This article is more than **3 years old**

### Amazon ditched AI recruiting tool that favored men for technical jobs

**Specialists had been building computer programs since 2014 to review résumés in an effort to automate the search process**



Amazon's automated hiring tool was found to be inadequate after penalizing the résumés of female candidates. Photograph: Brian Snyder/Reuters

*Reuters*
Thu 11 Oct 2018 00.42 BST

WILL KNIGHT BUSINESS NOV 19, 2019 9:15 AM

## The Apple Card Didn't 'See' Gender—and That's the Problem

**The way its algorithm determines credit lines makes the risk of bias more acute.**

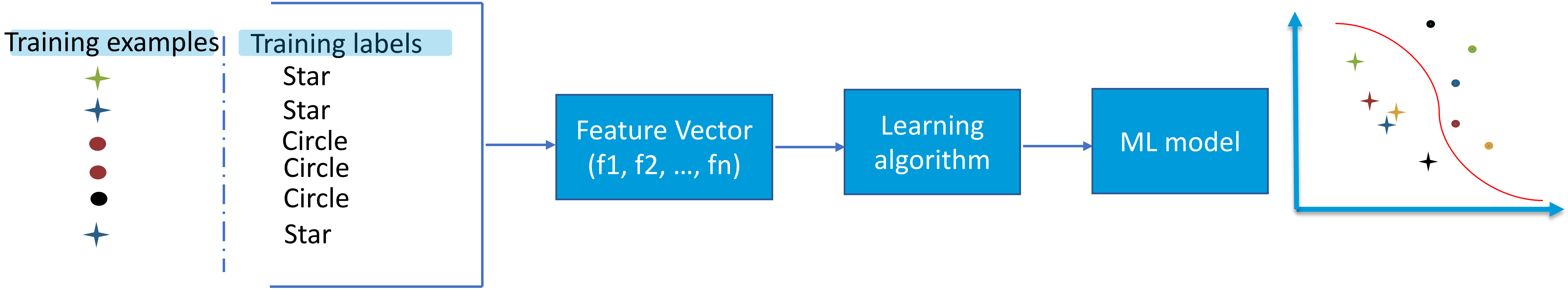
# Software testing

**Software** testing, as defined in the ANSI/IEEE 1059 standard:

“A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item”

# Machine learning basics

**1) Train phase**



# Machine learning basics

1) Train phase

Training examples

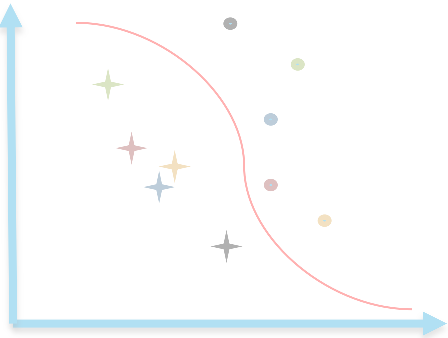
Training labels

Star
Star
Circle
Circle
Circle
Star

Feature Vector
(f1, f2, …, fn)

Learning algorithm

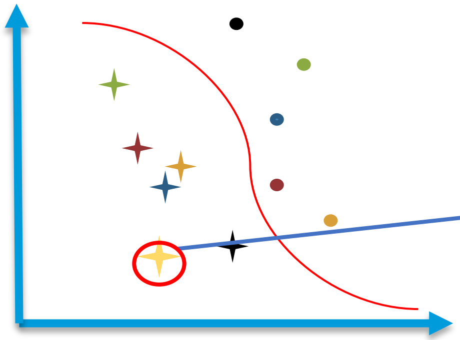
ML model



2) Test phase

Unseen test data

ML model



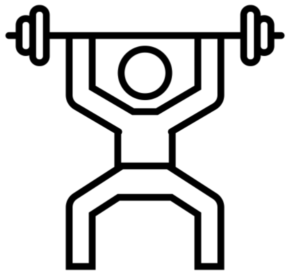
Decision

**Star**

# Why testing machine learning software for?

Two categories of defects in machine learning software:

Fairness



Robustness

# Fairness

WILL KNIGHT BUSINESS NOV 19, 2019 9:15 AM

## The Apple Card Didn't 'See' Gender—and That's the Problem

**The way its algorithm determines credit lines makes the risk of bias more acute.**


[Apple Credit Card]() – accused of offering smaller lines of credit to women than to men

“*The algorithm doesn’t even use gender as an input*”

Sensitive attributes (race, gender etc.) might be learned from ML models by highly correlated attributes

# Robustness

*“the degree to which a model’s performance changes when confronted to data unseen during training”*

- “*Natural*” robustness: model performance once put in production
- Robustness to distribution drift
- Robustness to security threats (adversarial attacks)

# Robustness

*“the degree to which a model’s performance changes when confronted to data unseen during training”*

- **“*Natural*” robustness: model performance once put in production**
- Robustness to distribution drift
- Robustness to security threats (adversarial attacks)

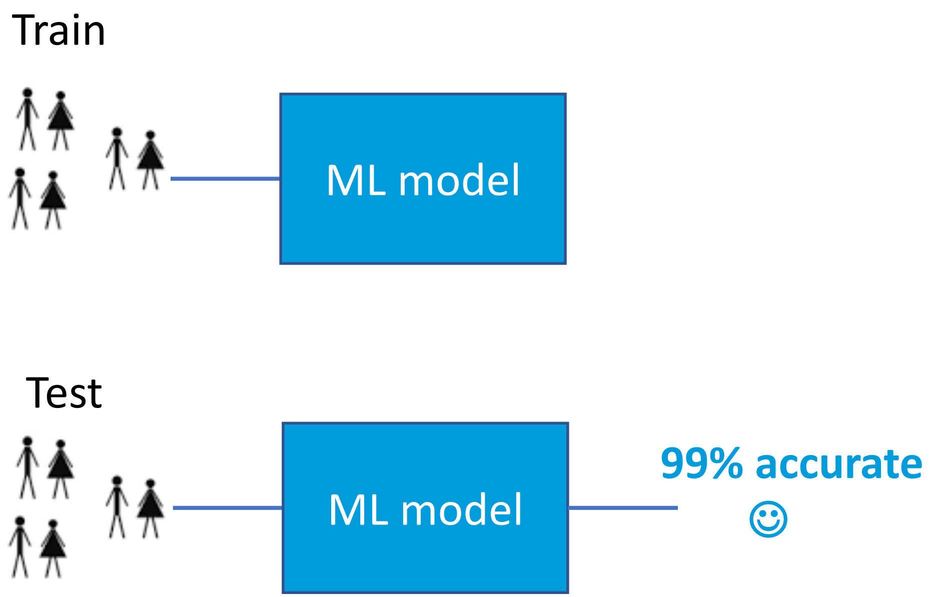
# Robustness

*“the degree to which a model’s performance changes when confronted to data unseen during training”*

- “*Natural*” robustness: model performance once put in production
- **Robustness to distribution drift**
- Robustness to security threats (adversarial attacks)

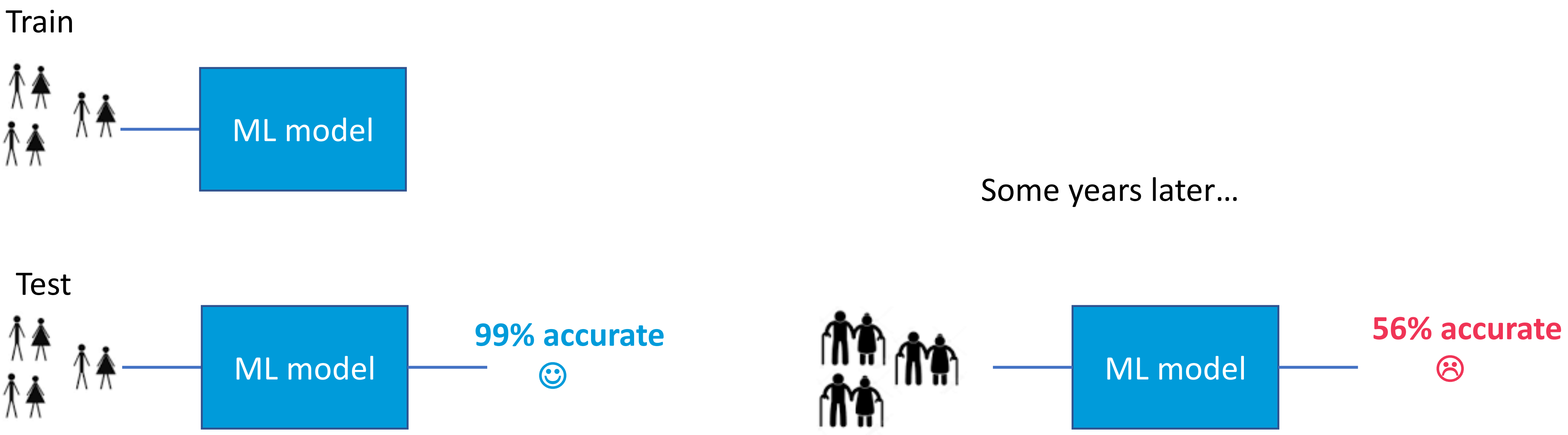
# Robustness to distribution drift

**Data changes over time => model become less accurate**

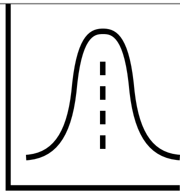


# Robustness to distribution drift

**Data changes over time => model become less accurate**



# Robustness to distribution drift



**Types of drift**
- Sudden drift
- Incremental drift
- Recurring drift

**Detection**
- Statistical methods
- Error rate based
- Detection model

**Correction**
- Periodic retraining
- Online learning

Research has successfully designed **methods** to **detect and mitigate** the effect of drifts BUT**…**

# Robustness to distribution drift

**Types of drift**

- Sudden drift
- Incremental drift
- Recurring drift

**Detection**

- Statistical methods
- Error rate based
- Detection model

**Correction**

- Periodic retraining
- Online learning

Research has successfully designed **methods** to **detect and mitigate** the effect of drifts BUT**…**

**…** **<u>in the real world</u>**:

- Computational limitations (periodic retraining not affordable)
- Delay to acquire true labels (online learning not applicable)
- Non-immediate software deployment process (cat-and-mouse game)

# Robustness

*"the degree to which a model's performance changes when confronted to data unseen during training"*

- "*Natural*" robustness: model performance once put in production
- Robustness to distribution drift
- **Robustness to security threats (adversarial attacks)**

# Robustness to adversarial attacks

The data themselves are a threat to ML software correctness:

**Evasion attacks** → works at "test" time

Poisoning attacks
Trojans attacks
Backdoors attack

→ works at training time

…

# Poisoning attack


**Poisoning attack**

- Tay bot used the interactions with its Twitter users as training data
- By repeatedly interacting with Tay using racist and offensive language, they were able to bias Tay's dataset towards that language as well
- Within 24 hours of its deployment, Tay had to be decommissioned

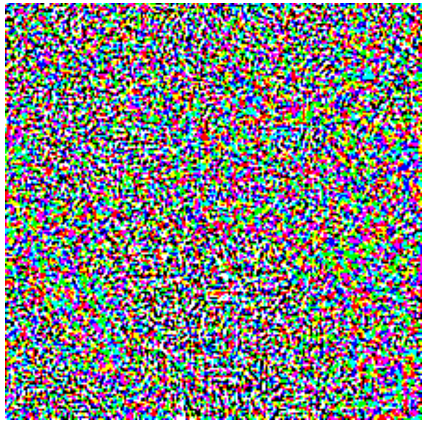
# Evasion attack and adversarial examples

Original example

ML predicts:
"Panda"
(80% confidence)

+

Small adversarial noise



=

**Adversarial example**

What humans still see

Gibbon

What ML predicts: "Gibbon"
(99% confidence)

"Explaining and Harnessing Adversarial Examples", Goodfellow et al., ICLR 2015.

# Adversarial examples beyond pixels

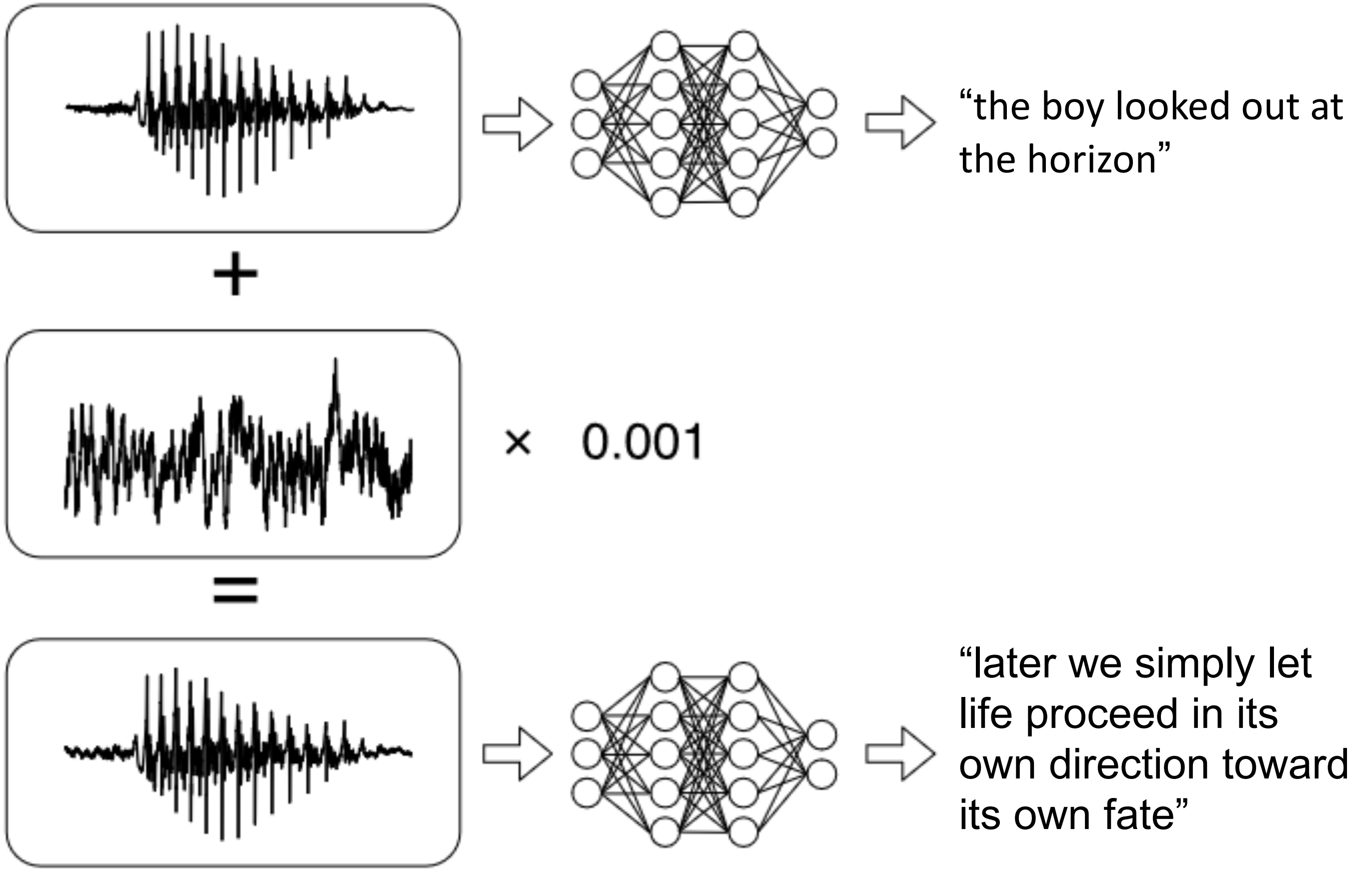


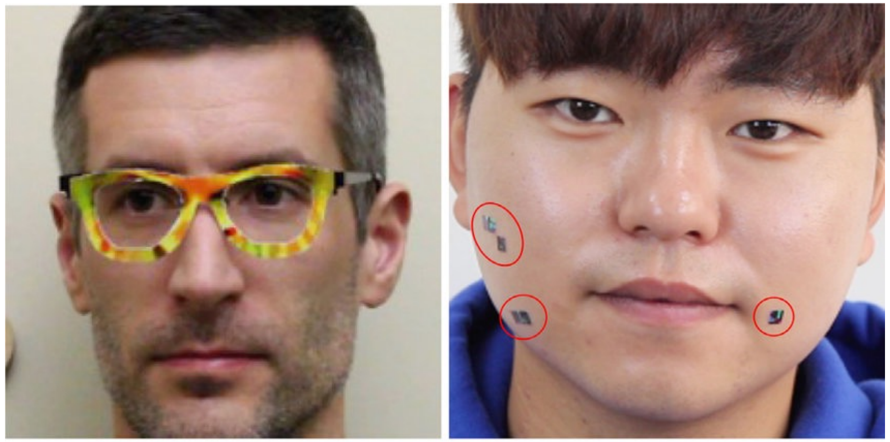

Figure 1. Illustration of our attack: given any waveform, adding a small perturbation makes the result transcribe as any desired target phrase.

Carlini, Nicholas, and David Wagner. "Audio adversarial examples: Targeted attacks on speech-to-text." *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018.

# Adversarial examples in the physical world

Crowd face recognition system

aws



(a) (b)

Journal of Information Security and Applications 60 (2021) 102874

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

ELSEVIER

Adversarial attacks by attaching noise markers on the face against deep face recognition

Gwonsang Ryu [a], Hosung Park [b], Daeseon Choi [c,*]

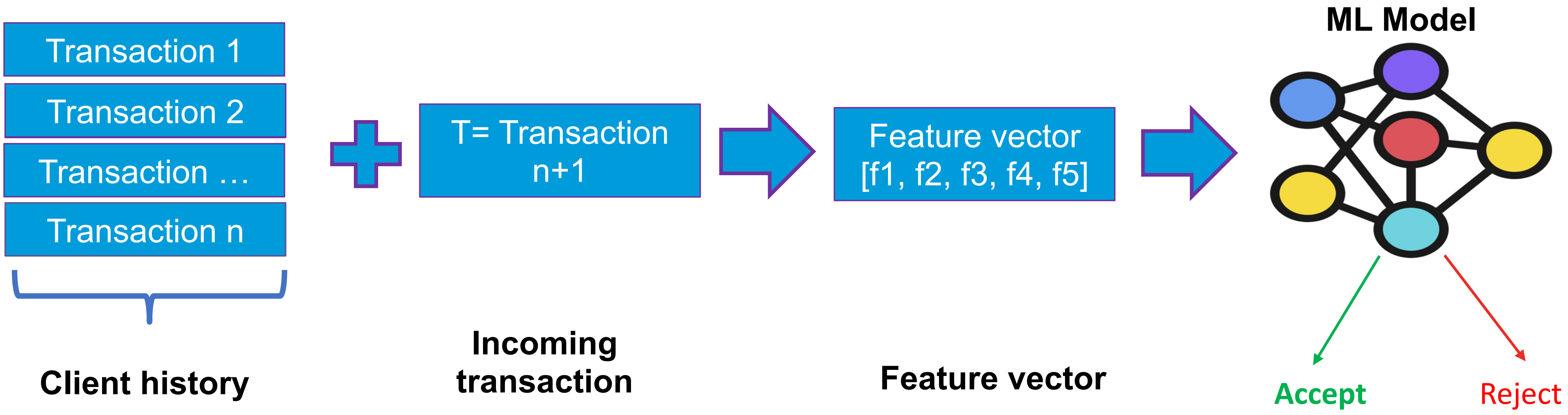
[a] Department of Software Convergence, Graduate School of Soongsil University, Seoul, 07027, South Korea
[b] Department of Cyber Security and Police, Busan University of Foreign Studies, Busan, 46234, South Korea
[c] Department of Software, Soongsil University, Seoul, 07027, South Korea

ARTICLE INFO

ABSTRACT

# My focus: adversarial examples in the real world

## Automated decision software in finance



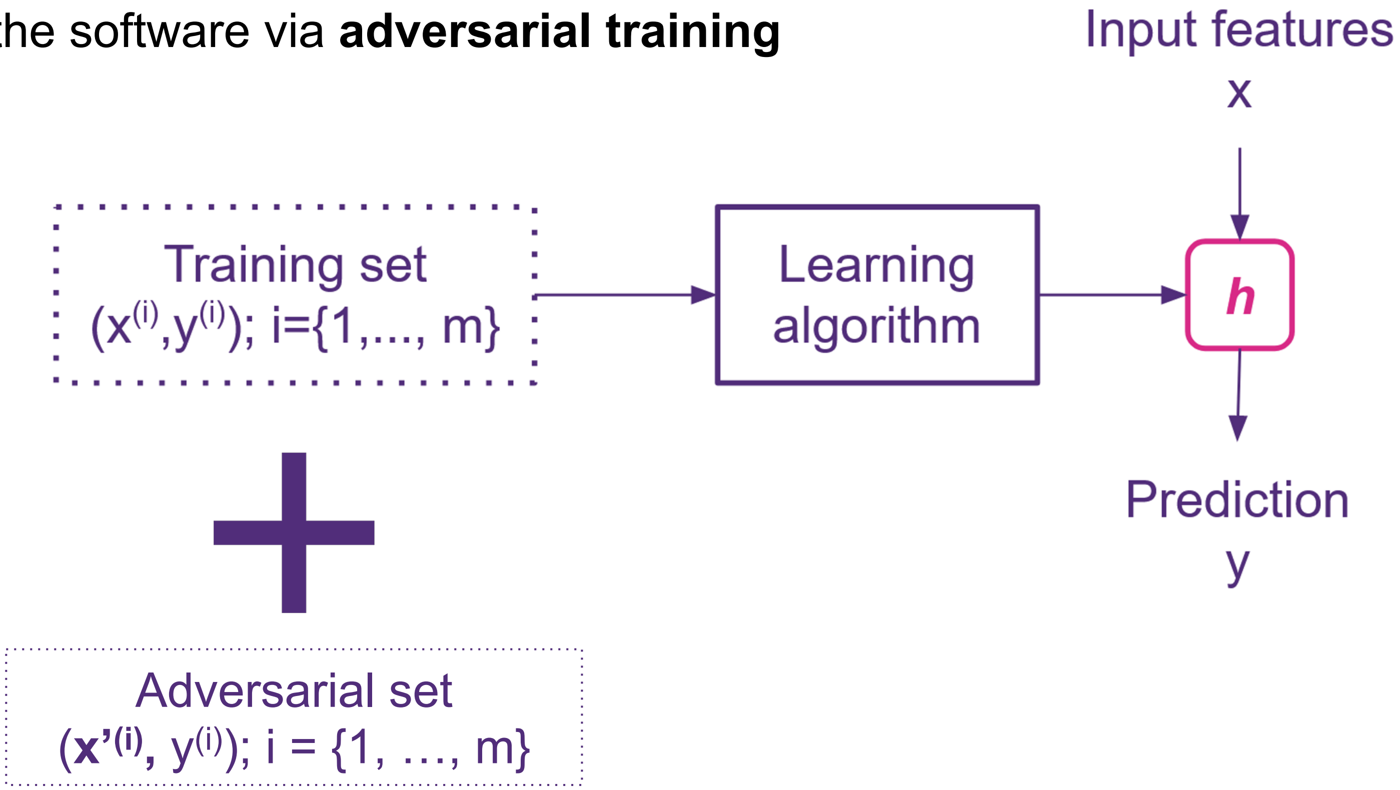

**Evasion attack goal**

**Make the smallest change in transaction n+1**
**Such that the decision changes from reject to accept**

# Learning from adversarial examples

Generating adversarial examples is useful to:

- Discover the limits of ML software (corner case testing)
- Improve the software via **adversarial training**

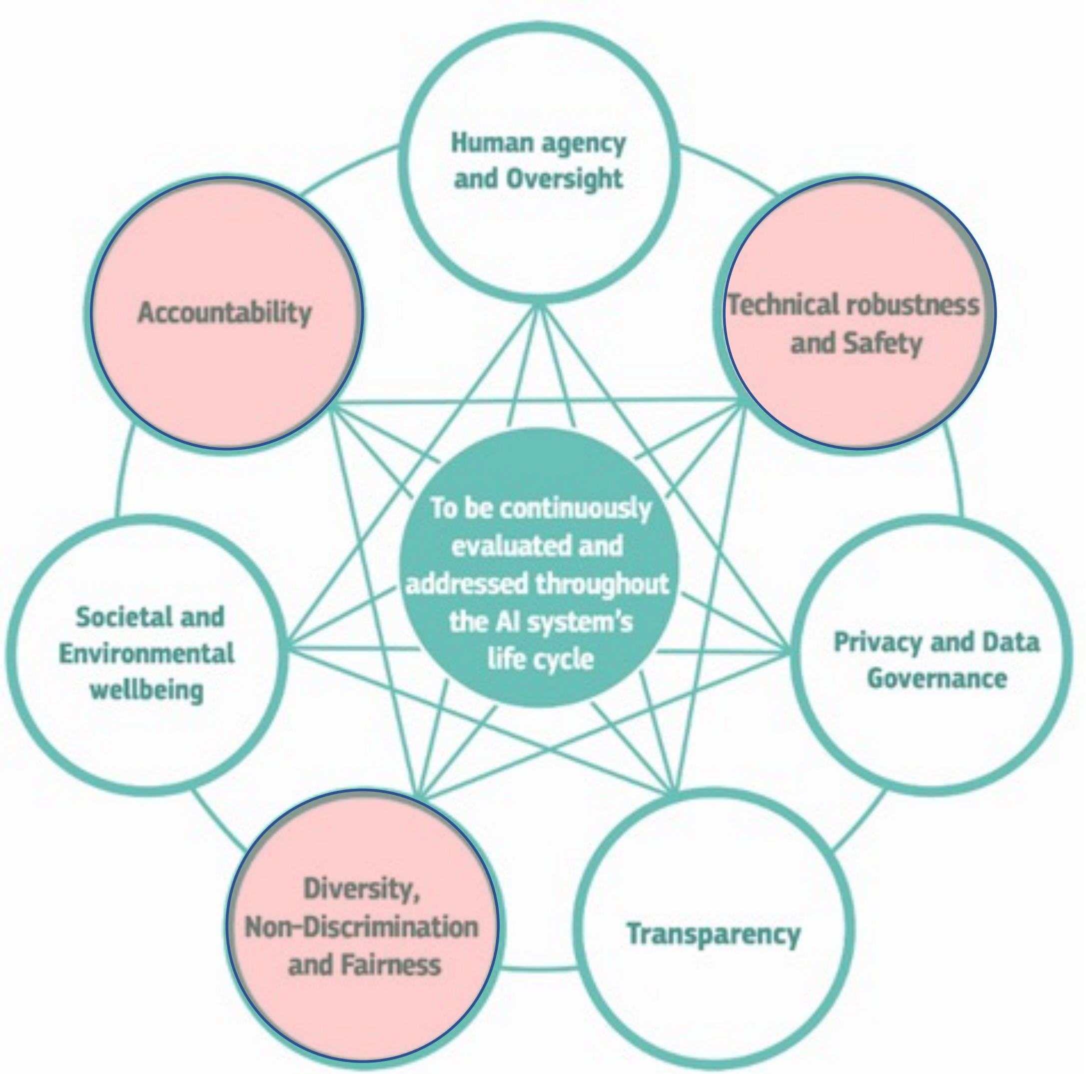


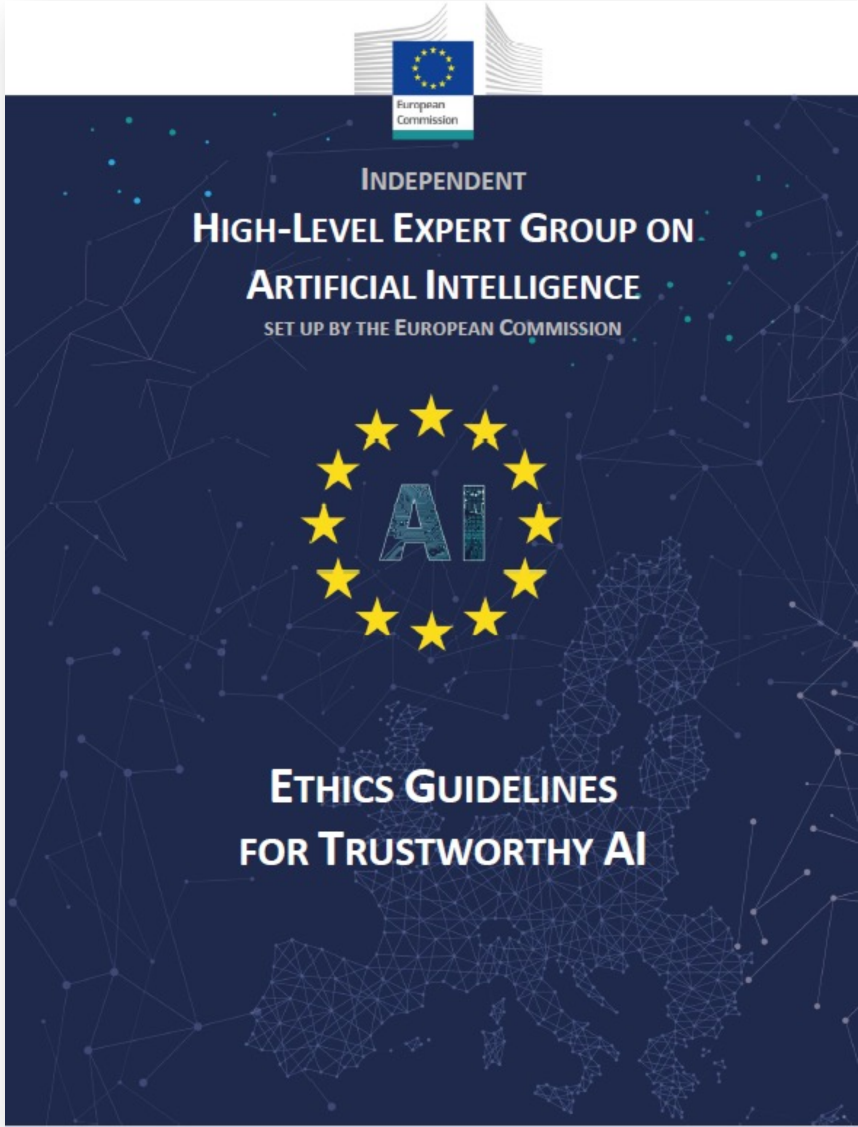
**Machine learning software can be inaccurate and fooled**

**… also in the real world**

**… and regulations are coming**

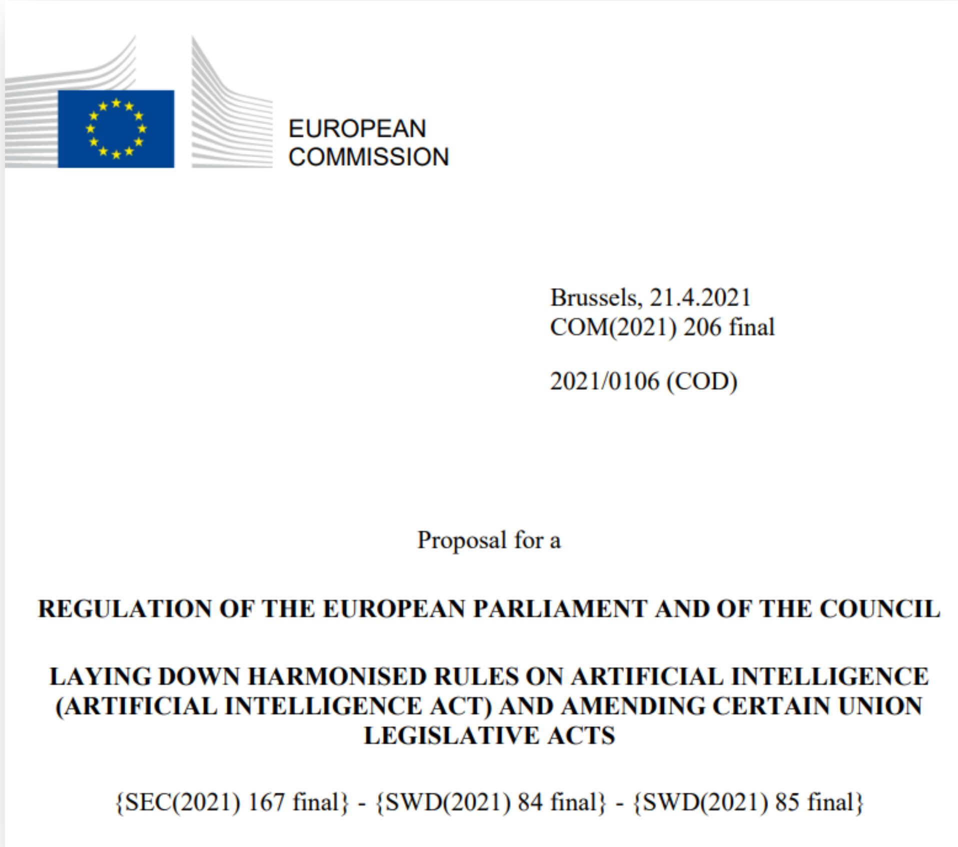
# Ethics guidelines for trustworthy AI – European Commission (2019)



# EU AI ACT (2021)

## A proposed European law on artificial intelligence (AI)


EUROPEAN COMMISSION

Brussels, 21.4.2021
COM(2021) 206 final

2021/0106 (COD)

Proposal for a

**REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL**

**LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS**

{SEC(2021) 167 final} - {SWD(2021) 84 final} - {SWD(2021) 85 final}

The Act requires providers to ensure before placing on market that their systems conform with the essential requirements listed above, as well as to comply with a number of other tasks including registering AI systems on a database, having an appropriate quality management system in place,[26]

Providers in the main will only have to demonstrate conformity by an '*assessment based on internal control*' i.e. **self-certification** (Article 43(1)(a).

Providers are tasked to 'establish and document a post-market monitoring system in a manner that is proportionate to the nature of the artificial intelligence technologies and the risks of the high-risk AI system'.[30] This

These are, unlike notified bodies, public bodies with regulatory power e.g. to require access to training, validation and testing datasets used by the provider, and the AI source code.[32]

# Adversarial Examples in Real-World Software

## Part II

## Research Questions

**RQ1. Are real-world machine learning software vulnerable to evasion attacks?**

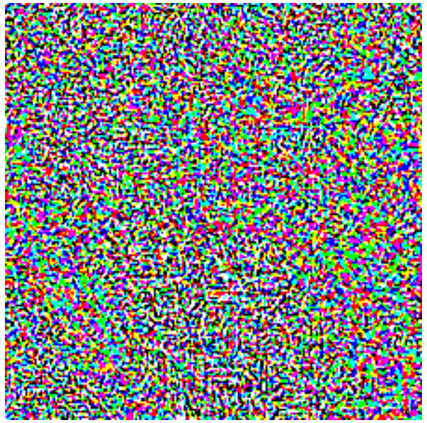
**RQ2. How to effectively defend these software?**

# Evasion attack and adversarial examples

Original example

ML predicts:
"Panda"
(80% confidence)

Small adversarial noise



**Adversarial example**

What humans still see

Gibbon

What ML predicts: "Gibbon"
(99% confidence)

"Explaining and Harnessing Adversarial Examples", Goodfellow et al., ICLR 2015.

# Gradient-based attack algorithm

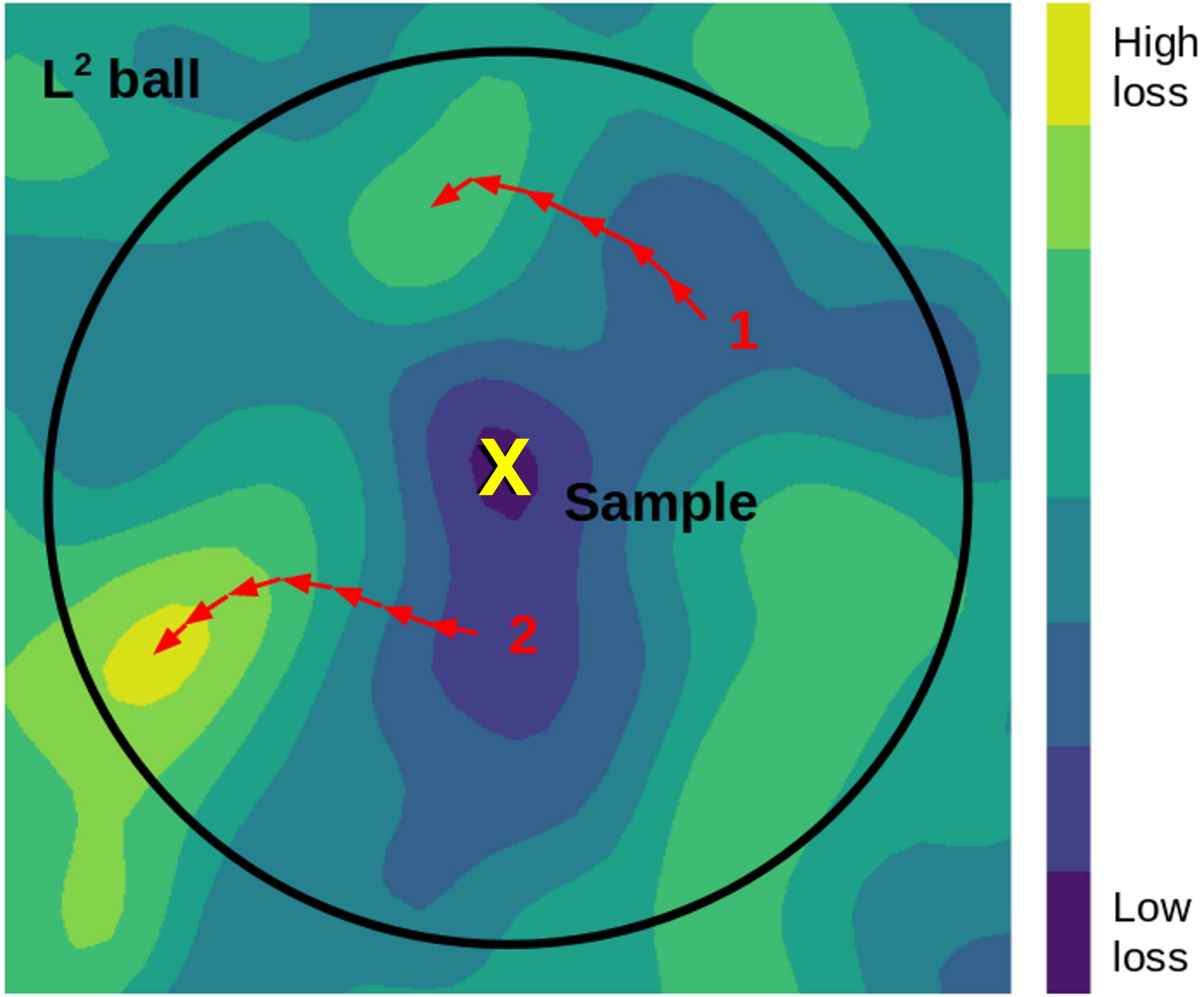



Image modified from https://towardsdatascience.com/know-your-enemy-7f7c5038bdf3

Objective: for $x$ find $\delta$

- ✓ With $f(x) \neq f(x+\delta)$
- ✓ With $L_p(x, x+\delta) < \epsilon$

# Projected Gradient Descent (PGD)

*Iteratively compute:*

Step size

Normalize the gradient

model parameters

current example

$$x^{t+1} = \mathrm{Clip}_{x,\epsilon}(x^t + \alpha\, norm(\nabla_{x^t}(L(\theta_f, x^t, y))))$$

original label

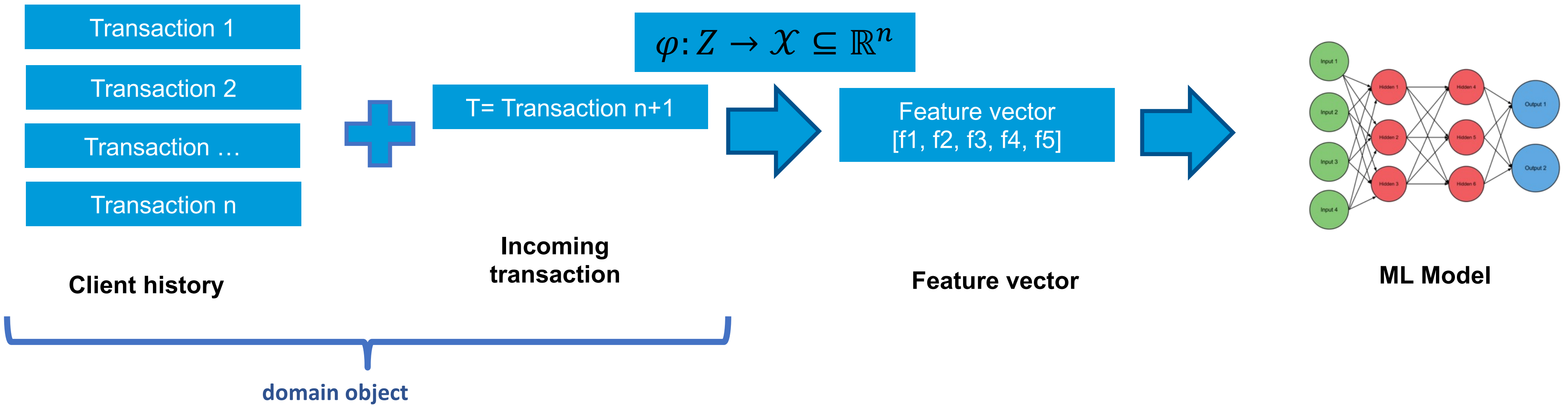
Project back on the hypervolume
Such that $D(x, x') \leq \epsilon$

Gradient of the loss around
$x^t$ the “slope”

Loss function

# Evasion attack in real-world software

ML model is integrated in a larger software system that takes as **input domain objects.**



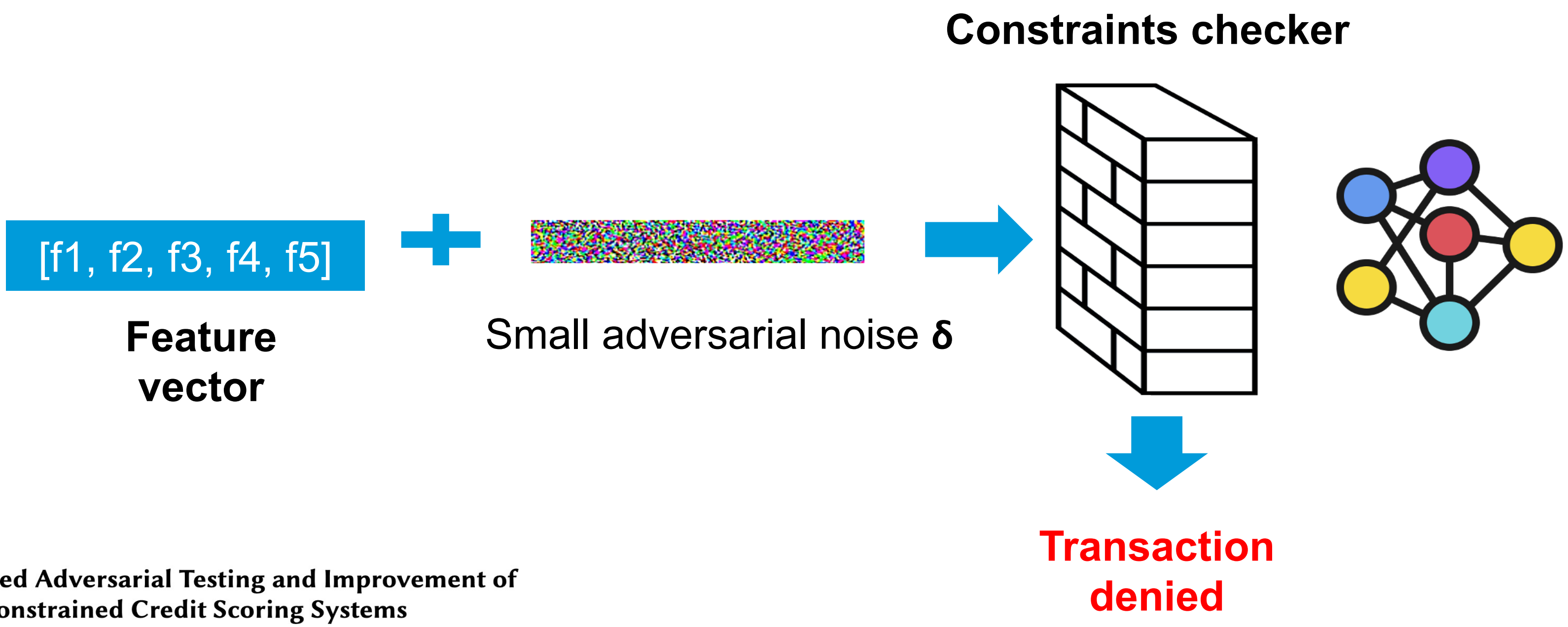

Domain object Space $Z$
respects some **natural conditions**

Feature space $\mathcal{X}_\Omega$
respects a **set of constraints** $\Omega$

$$open\_acc \leq total\_acc$$

$$installment = loam\_amount \times \frac{int\_rate \times (1 + int\_rate)^{term}}{(1 + int\_rate)^{term} - 1}$$

# Input validation as a first line of defense




**0% success rate from traditional evasion attacks**

**Search-Based Adversarial Testing and Improvement of Constrained Credit Scoring Systems**

Salah Ghamizi
University of Luxembourg
salah.ghamizi@uni.lu

Maxime Cordy
University of Luxembourg
maxime.cordy@uni.lu

Martin Gubri
University of Luxembourg
martin.gubri@uni.lu

Andrey Boystov
University of Luxembourg
andrey.boystov@uni.lu

Mike Papadakis
University of Luxembourg
michail.papadakis@uni.lu

Yves Le Traon
University of Luxembourg
yves.letraon@uni.lu

Anne Goujon
BGL BNP Parisbas

ESEC/FSE 2020

# Existing attacks generate infeasible examples!

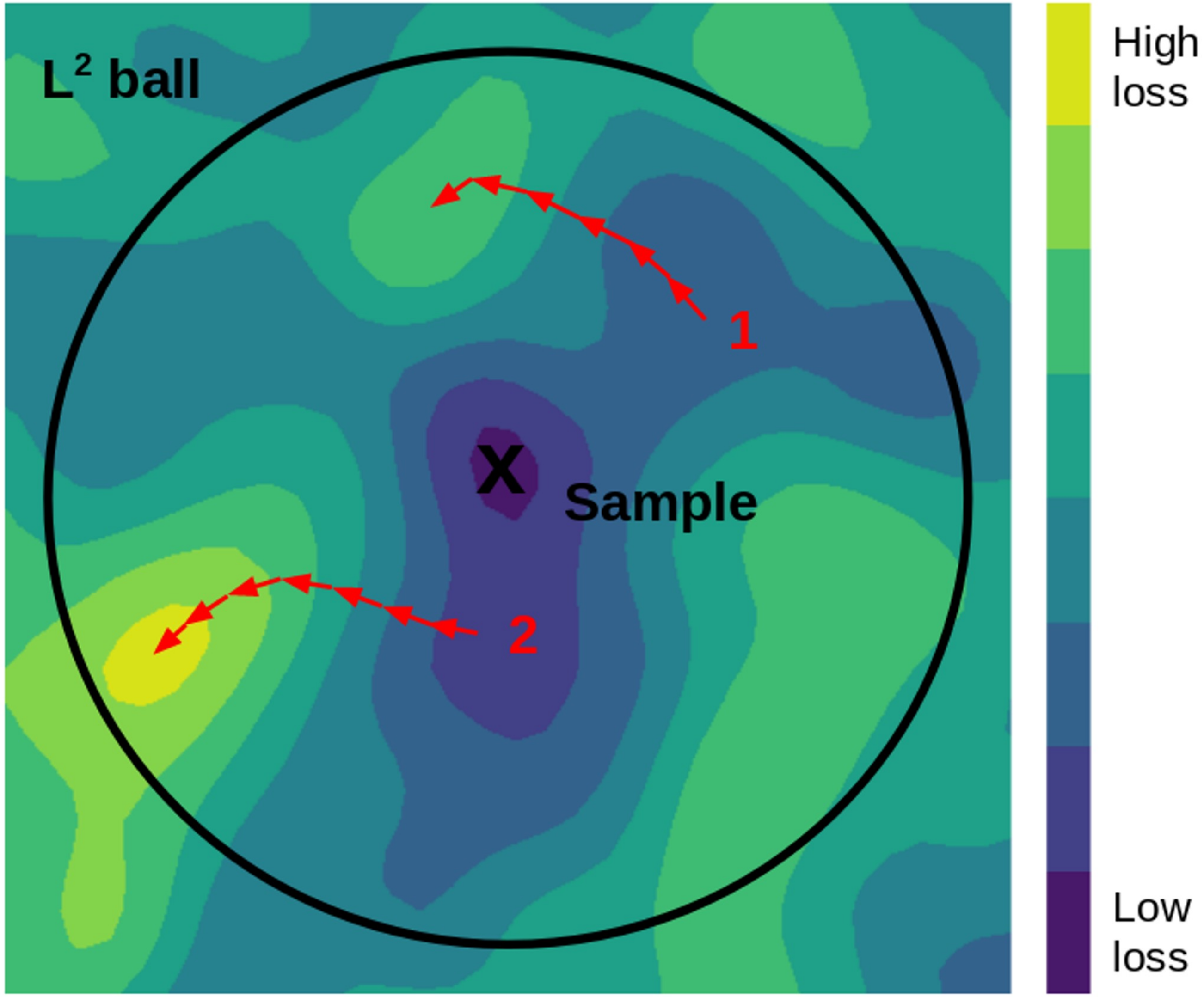


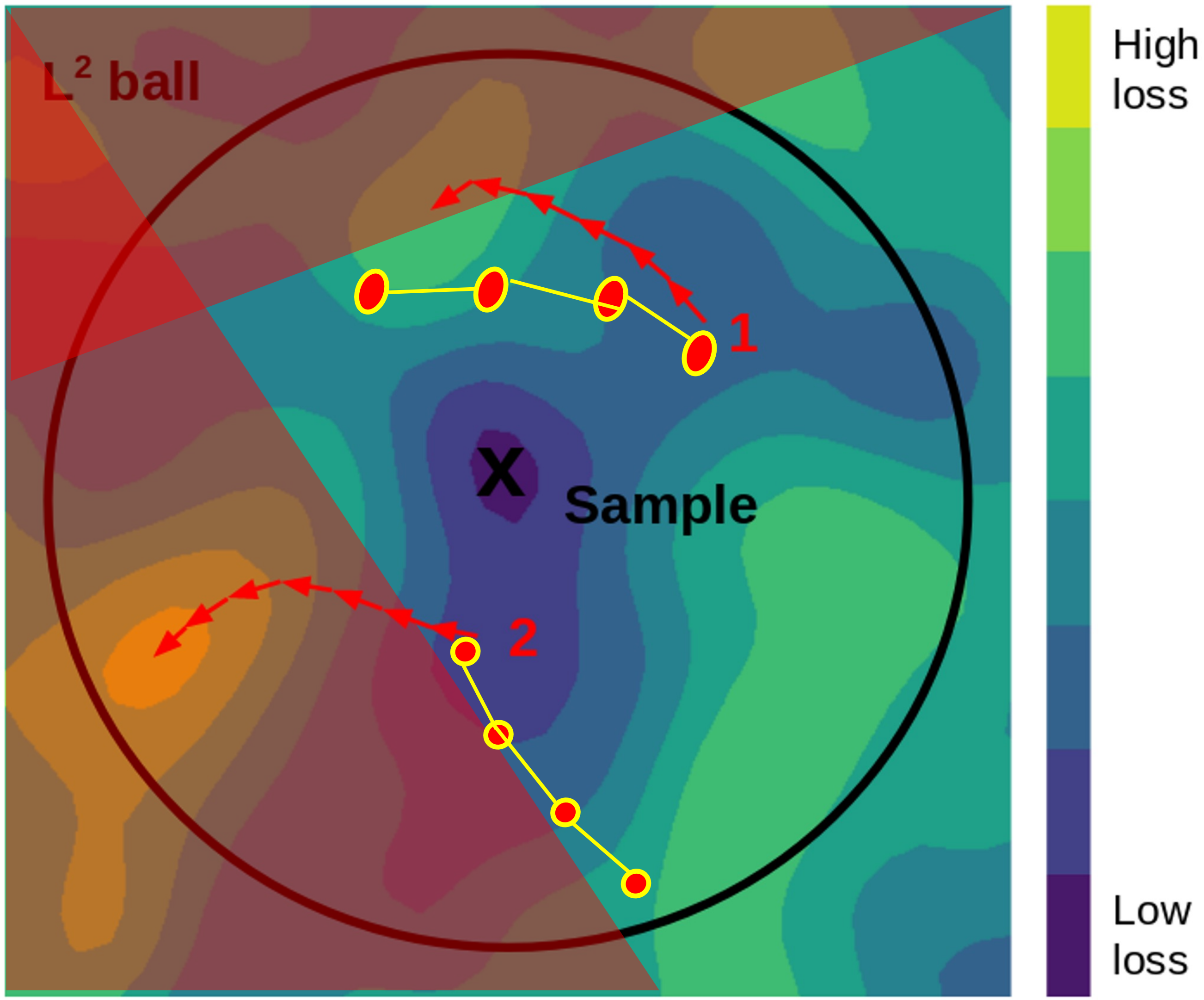

Image modified from https://towardsdatascience.com/know-your-enemy-7f7c5038bdf3

Objective: for $x$ find $\delta$

✓ With $f(x) \neq f(x+\delta)$

✓ With $L_p(x, x+\delta) < \epsilon$

# Existing attacks generate infeasible examples!




Objective: for $x$ find $\delta$

- ✓ With $f(x) \neq f(x+\delta)$
- ✓ With $L_p(x, x+\delta) < \epsilon$
- ✓ $x+\delta \in \mathcal{X}_\Omega$

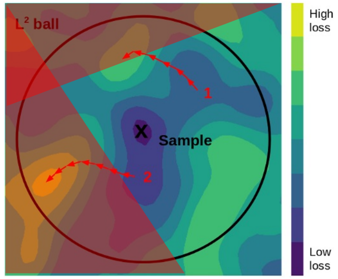
# Our Contributions

## A unified framework for adversarial attack and defense in constrained feature space

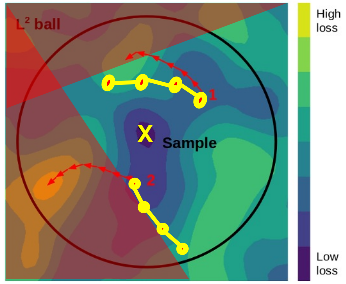
$$\omega := \omega_1 \wedge \omega_2 \mid \omega_1 \vee \omega_2 \mid \psi_1 \succeq \psi_2 \mid f \in \{\psi_1 \dots \psi_k\}$$
$$\psi := c \mid f \mid \psi_1 \oplus \psi_2 \mid x_i$$

| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \dots \psi_k\}$ | $\min(\{\psi_i \in \Psi : \mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

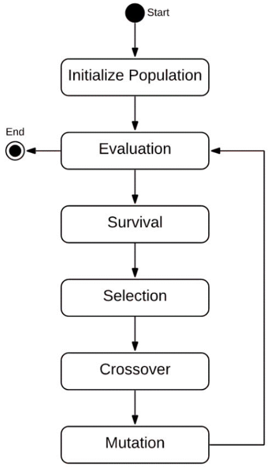
**Generic constraints language**




PGD + “Repair”




Constrained PGD




Multi-Objective Evolutionary Adversarial Attack (MoEvA2)

**Three constrained evasion attacks**

# Encoding constraints as a penalty function

Constraint grammar

$$\omega := \omega_1 \wedge \omega_2 \mid \omega_1 \vee \omega_2 \mid \psi_1 \succeq \psi_2 \mid f \in \{\psi_1 \ldots \psi_k\}$$
$$\psi := c \mid f \mid \psi_1 \oplus \psi_2 \mid x_i$$

$\boldsymbol{f} \in \boldsymbol{F}$ is the value of feature $f$ for a given input $x'$,
$\boldsymbol{c}$ is a constant real value,
$\boldsymbol{\omega}, \boldsymbol{\omega_1}, \boldsymbol{\omega_2}$ are constraint formulae,
$\succcurlyeq \in \{<, \leq, =, \neq, \geq, >\}$,
$\boldsymbol{\psi}, \boldsymbol{\psi_1}, \ldots, \boldsymbol{\psi_k}$ are numeric expressions,
$\oplus \in \{+, -, *, \backslash\}$, and
$\boldsymbol{x_i}$ is the value of the $i^{\text{th}}$ feature of the clean input $x$

Mapping to penalty functions

| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \ldots \psi_k\}$ | $\min(\{\psi_i \in \Psi :\mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

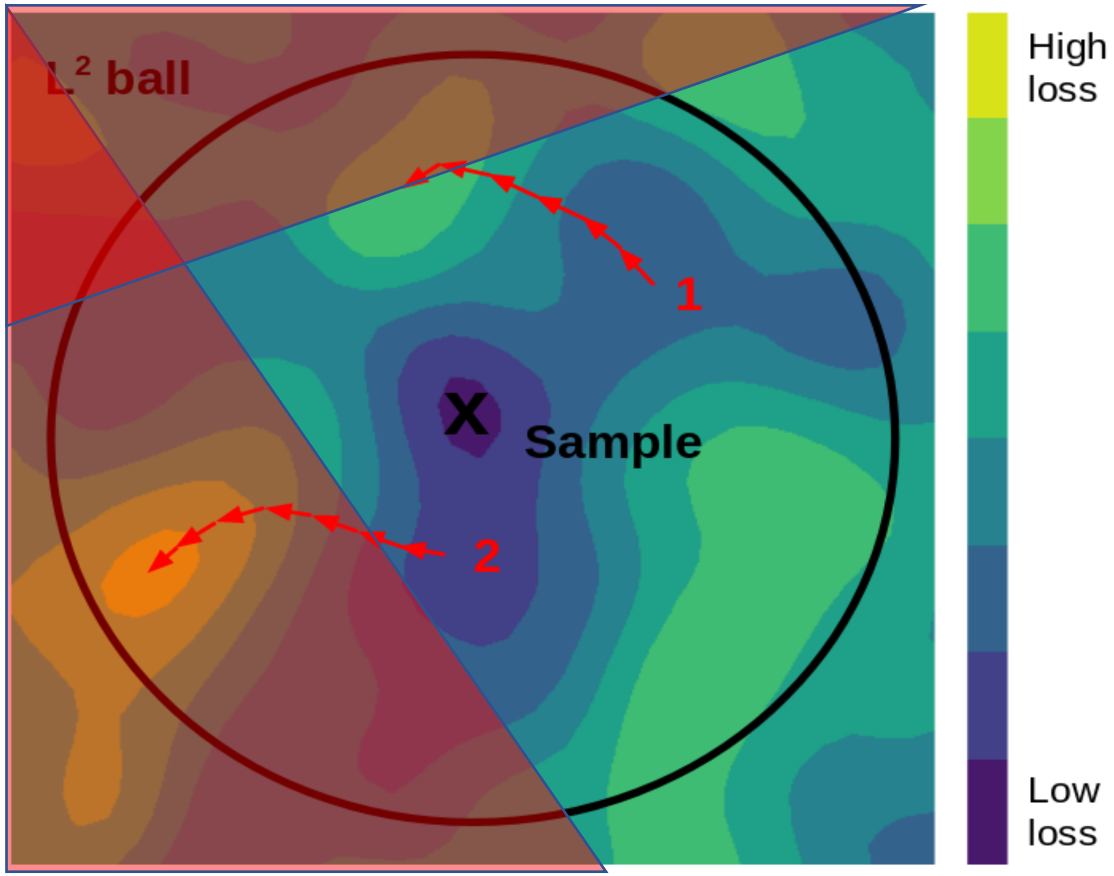
Table 1: From constraint formulae to penalty functions. $\tau$ is an infinitesimal value.

**Constraint is satisfied if and only if $\boldsymbol{g(\omega, x) = 0}$**

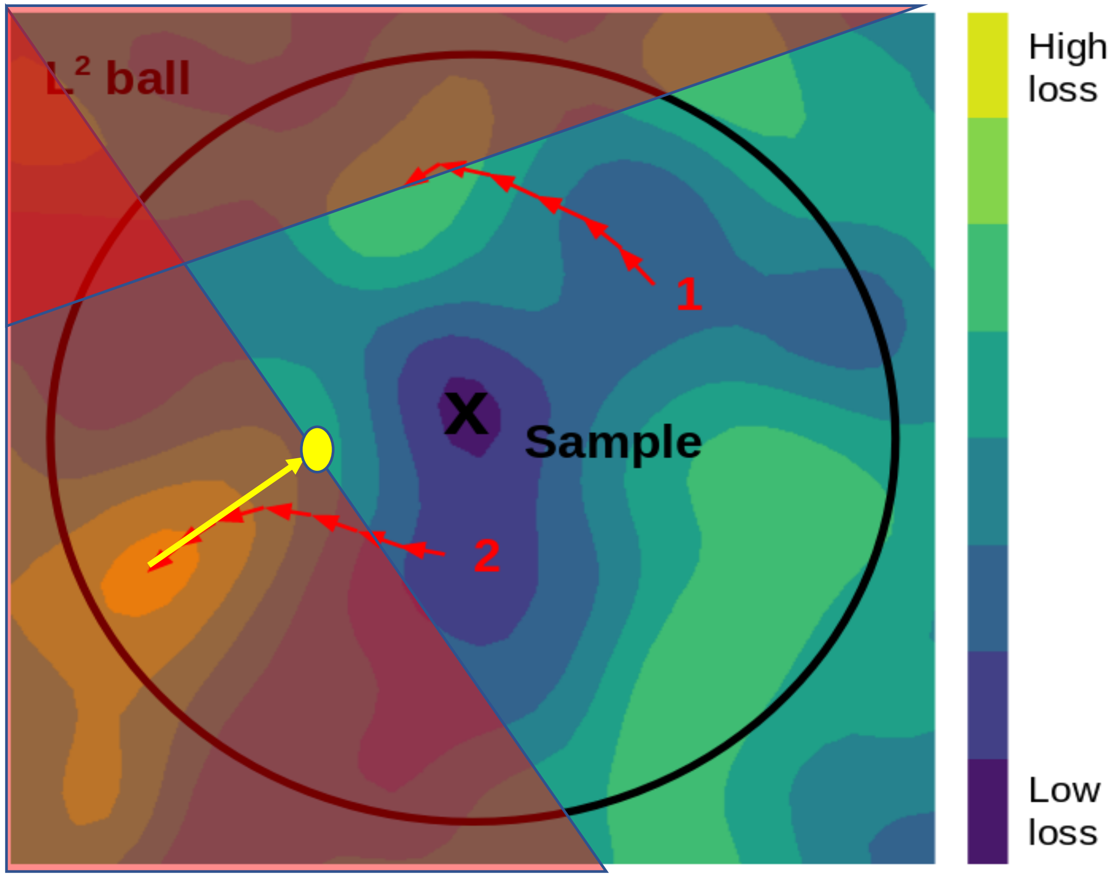
**Sufficient expressiveness to instantiate constraints in different domains**

# Approach 1: Vanilla PGD + "Repair"

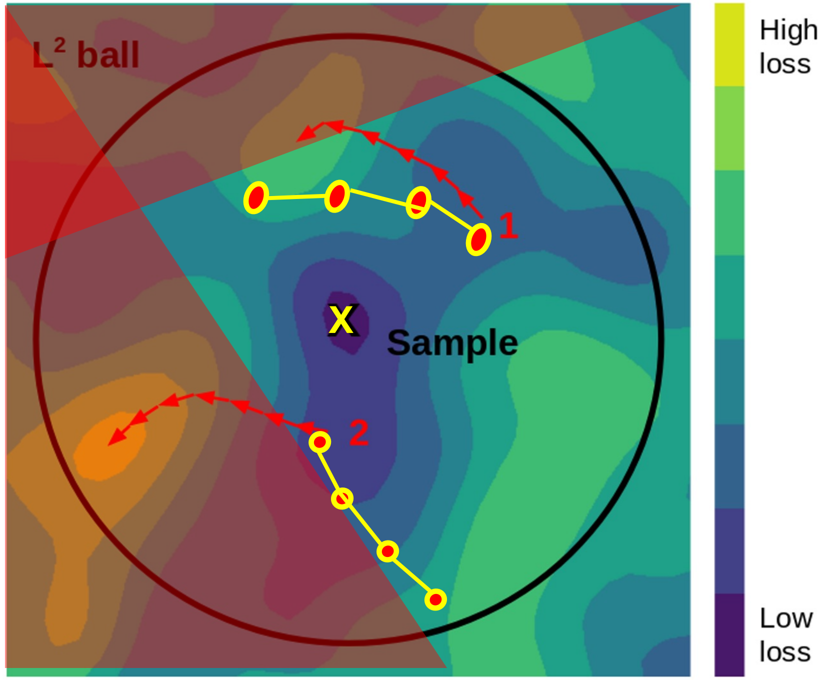
1) Apply PGD




2) Project the solution back to the feasible space (using mathematical programming solver)



# Approach 2: Constrained PGD: gradient-based constraint satisfaction




**Projected Gradient Descent (PGD)**

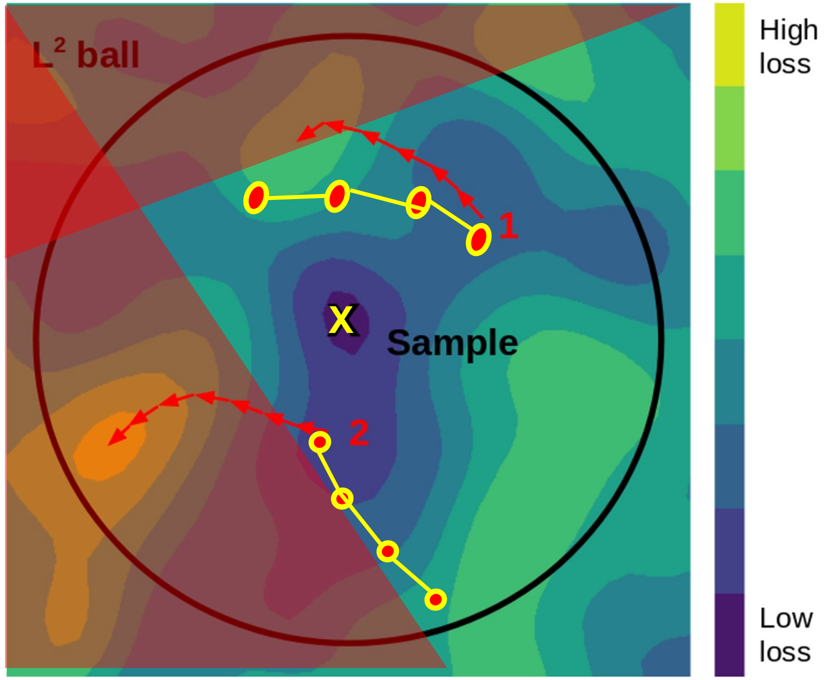
$$x^{t+1} = \text{Clip}_{x,\epsilon}(x^t + \alpha\, norm(\nabla_{x^t}(L(\theta_f, x^t, y))))$$

**Constraints regularization**

$$\nabla_{x^t} L(\theta_f, x^t, y) - \nabla_{x^t} penalty(x^t)$$

**Should be differentiable and ideally convex to increase convergence likelihood!**

# Approach 2: Constrained PGD: gradient-based constraint satisfaction




**Projected Gradient Descent (PGD)**

$$x^{t+1} = \mathrm{Clip}_{x,\epsilon}(x^t + \alpha\, norm(\nabla_{x^t}(L(\theta_f, x^t, y)))$$

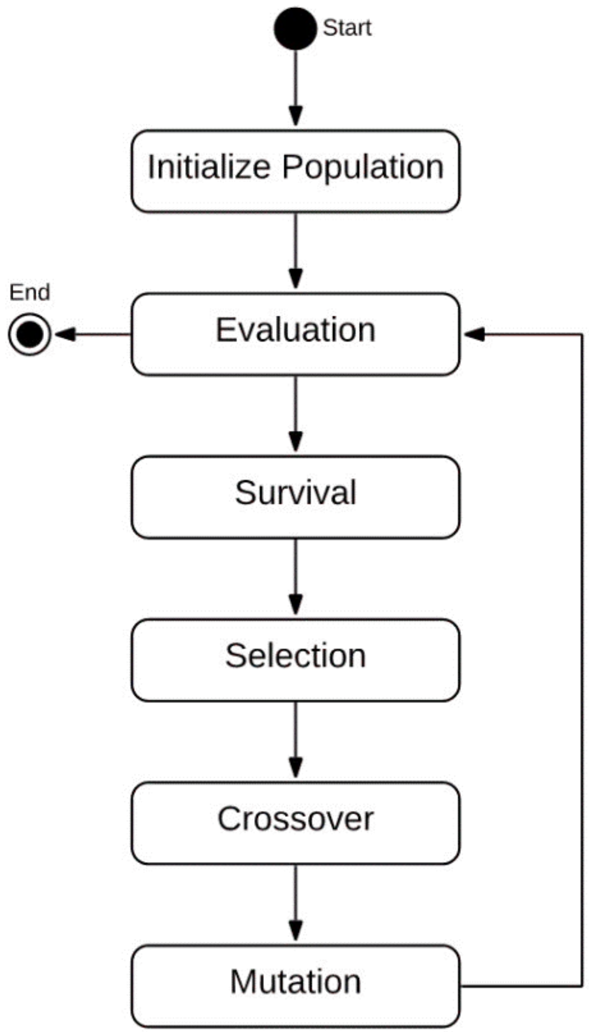
**Constraints regularization**

$$\nabla_{x^t} L(\theta_f, x^t, y) - \nabla_{x^t} penalty(x^t)$$

**Constrained Projected Gradient Descent (C-PGD)**

$$x^{t+1} = \mathrm{Clip}_{x,\epsilon}(x^t + \alpha\, norm\left(\nabla_{x^t} L(\theta_f, x^t, y) - \nabla_{x^t} penalty(x^t)\right)$$

# Approach 3: Multi-Objective Evolutionary Adversarial Attack (MoEvA2)



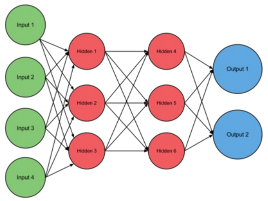

**Multi-objective** genetic algorithm (NSGA-III)

$$maximise\ g_1(x) \equiv L(\theta_f, x, y)$$

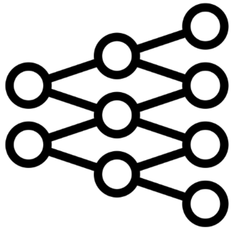
$$minimise\ g_2(x) \equiv\ L_p(x - x_0)$$

$$minimise\ g_3(x) \equiv \sum_{\omega_i \in \Omega} penalty(x, \omega_i)$$

# How effective are our approaches at generating adversarial examples?



Neural Network



Random Forest

| Model | Dataset | Attack | Success rate |
|---|---|---|---|
| Neural Network | LCLD | PGD | 0.00 |
| | | PGD + REP. | 0.00 |
| | | C-PGD | 9.85 |
| | | MoEvA2 | 97.48 |
| | CTU-13 | PGD | 0.00 |
| | | PGD + REP. | 0.00 |
| | | C-PGD | 0.00 |
| | | MoEvA2 | 100.00 |
| Random Forest | LCLD | Papernot * | 0.00 |
| | | MoEvA2 | 41.51 |
| | CTU-13 | Papernot * | 0.0 |
| | | MoEvA2 | 5.41 |
| | Malware | Papernot * | 0.00 |
| | | MoEvA2 | 39.30 |
| | URL | Papernot * | 8.50 |
| | | MoEvA2 | 31.89 |

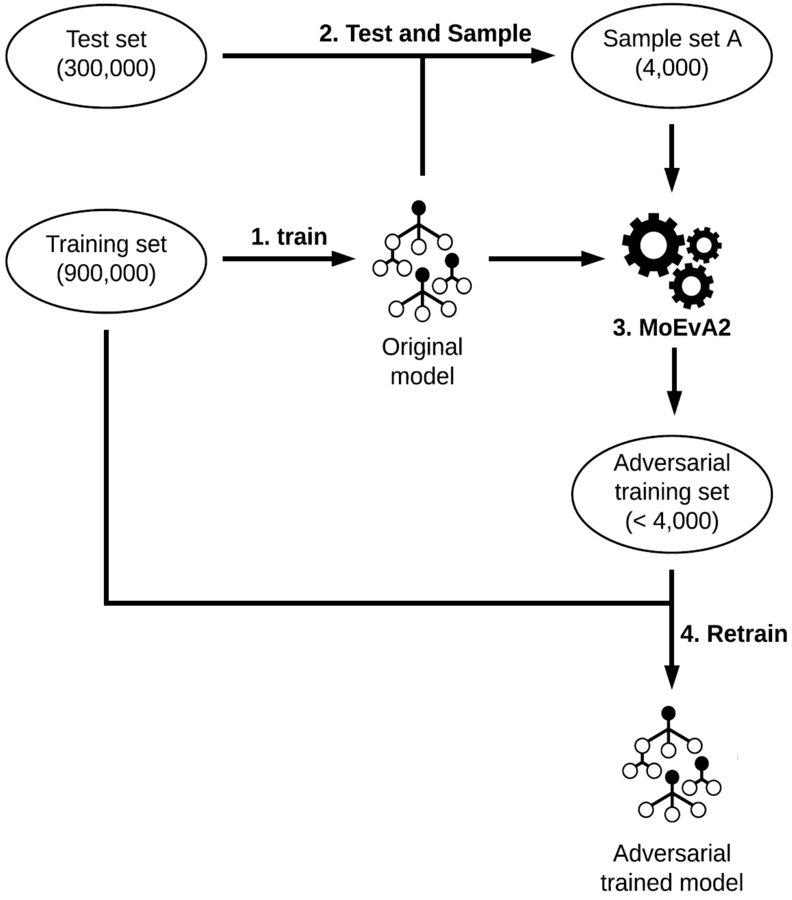
**Attacks unaware of domain constraints most often fail.**

**C-PGD worked on a single dataset (out of two).**

**MoEvA2 has successfully attacked all models.**

* Extended to random forest

# How to increase robustness?




**Adversarial retraining**

## How to increase robustness?

We hypothesize that **augmenting $\Omega$** with a set of engineered constraints can **robustify a model**.

We engineer a **new feature**

$$f_e = f_1 \oplus f_2$$

We have the **new constraint**

$$\omega_e \vDash (f_e = f_1 \oplus f_2)$$

# How effective are defense techniques ?


| Defense | Attack | LCLD | CTU-13 |
|---|---|---|---|
| None | C-PGD | 9.85 | 0.00 |
| None | MoEvA2 | 97.48 | 100.00 |
| C-PGD Adv. retraining * | C-PGD | 8.78 | NA |
| C-PGD Adv. retraining * | MoEvA2 | 94.90 | NA |
| MoEvA2 Adv. retraining * | C-PGD | 2.70 | NA |
| MoEvA2 Adv. retraining * | MoEvA2 | 85.20 | 0.8 |
| Constraints augment. | C-PGD | 0.00 | NA |
| Constraints augment. | MoEvA2 | 80.43 | 0.00 |
| MoEvA2 Adv. retrain. † | MoEvA2 | 82.00 | NA |
| Combined defenses † | MoEvA2 | 77.43 | NA |

Success rate of C-PGD and MoEvA2 after adversarial retraining and constraint augmentation (on neural networks). For a fair comparison, the model denoted by the same symbols (* or †) are trained with the same number of adversarial examples, generated from the same original samples.

# How effective are defense techniques ?

| Defense | Attack | LCLD | CTU-13 |
|---|---|---|---|
| None | C-PGD | 9.85 | 0.00 |
| None | MoEvA2 | 97.48 | 100.00 |
| C-PGD Adv. retraining * | C-PGD | 8.78 | NA |
| C-PGD Adv. retraining * | MoEvA2 | 94.90 | NA |
| MoEvA2 Adv. retraining * | C-PGD | 2.70 | NA |
| MoEvA2 Adv. retraining * | MoEvA2 | 85.20 | 0.8 |
| Constraints augment. | C-PGD | 0.00 | NA |
| Constraints augment. | MoEvA2 | 80.43 | 0.00 |
| MoEvA2 Adv. retrain. † | MoEvA2 | 82.00 | NA |
| Combined defenses † | MoEvA2 | 77.43 | NA |

Success rate of C-PGD and MoEvA2 after adversarial retraining and constraint augmentation (on neural networks). For a fair comparison, the model denoted by the same symbols (* or †) are trained with the same number of adversarial examples, generated from the same original samples.

**Adversarial training remains effective in constrained feature space**

# How effective are defense techniques ?


| Defense | Attack | LCLD | CTU-13 |
|---|---|---|---|
| None | C-PGD | 9.85 | 0.00 |
| None | MoEvA2 | 97.48 | 100.00 |
| C-PGD Adv. retraining * | C-PGD | 8.78 | NA |
| C-PGD Adv. retraining * | MoEvA2 | 94.90 | NA |
| MoEvA2 Adv. retraining * | C-PGD | 2.70 | NA |
| MoEvA2 Adv. retraining * | MoEvA2 | 85.20 | 0.8 |
| Constraints augment. | C-PGD | 0.00 | NA |
| Constraints augment. | MoEvA2 | 80.43 | 0.00 |
| MoEvA2 Adv. retrain. † | MoEvA2 | 82.00 | NA |
| Combined defenses † | MoEvA2 | 77.43 | NA |

Success rate of C-PGD and MoEvA2 after adversarial retraining and constraint augmentation (on neural networks). For a fair comparison, the model denoted by the same symbols (* or †) are trained with the same number of adversarial examples, generated from the same original samples.

**Adversarial training remains effective in constrained feature space**

**Constraint augmentation** is an **effective alternative defense** to adversarial retraining.

# How effective are defense techniques ?

| Defense | Attack | LCLD | CTU-13 |
|---|---|---|---|
| None | C-PGD | 9.85 | 0.00 |
| None | MoEvA2 | 97.48 | 100.00 |
| C-PGD Adv. retraining * | C-PGD | 8.78 | NA |
| C-PGD Adv. retraining * | MoEvA2 | 94.90 | NA |
| MoEvA2 Adv. retraining * | C-PGD | 2.70 | NA |
| MoEvA2 Adv. retraining * | MoEvA2 | 85.20 | 0.8 |
| Constraints augment. | C-PGD | 0.00 | NA |
| Constraints augment. | MoEvA2 | 80.43 | 0.00 |
| MoEvA2 Adv. retrain. † | MoEvA2 | 82.00 | NA |
| Combined defenses † | MoEvA2 | 77.43 | NA |

Success rate of C-PGD and MoEvA2 after adversarial retraining and constraint augmentation (on neural networks). For a fair comparison, the model denoted by the same symbols (* or †) are trained with the same number of adversarial examples, generated from the same original samples.

**Adversarial training remains effective in constrained feature space**

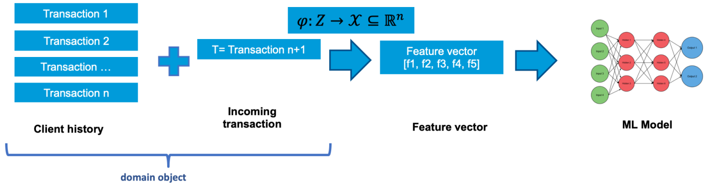
**Constraint augmentation is an effective alternative defense** to adversarial retraining.

**Constraint augmentation and adversarial retraining have complementary effects.**

# Conclusion

### Evasion attack in real-world software

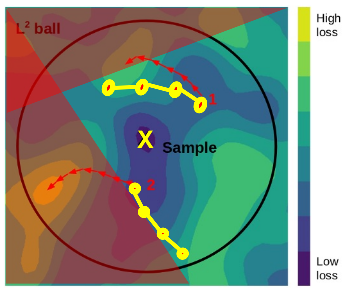
ML model is integrated in a larger software system that takes as **input domain objects.**



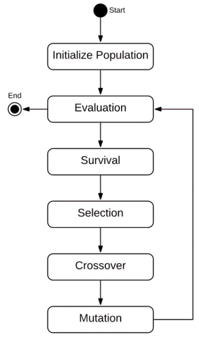

Domain object Space $Z$
respects some **natural conditions**

Feature space $\mathcal{X}_\Omega$
respects a **set of constraints** $\Omega$

$$open\_acc \leq total\_acc$$
$$installment = loam\_amount \times \frac{int\_rate \times (1 + int\_rate)^{term}}{(1 + int\_rate)^{term} - 1}$$




C-PGD

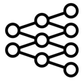



MoEvA2

Constrained Attacks

**How effective are our approaches at generating adversarial examples?**

NN



Random Forest

| Model | Dataset | Attack | C&M |
|---|---|---|---|
| NN | LCLD | PGD | 0.00 |
| | | PGD + SAT | 0.00 |
| | | C-PGD | 9.85 |
| | | MoEvA2 | 97.48 |
| | CTU-13 | PGD | 0.00 |
| | | PGD + SAT | 0.00 |
| | | C-PGD | 0.00 |
| | | MoEvA2 | 100.00 |
| Random Forest | LCLD | Papernot * | 0.00 |
| | | MoEvA2 | 41.51 |
| | CTU-13 | Papernot * | 0.0 |
| | | MoEvA2 | 5.41 |
| | Malware | Papernot * | 0.00 |
| | | MoEvA2 | 39.30 |
| | URL | Papernot * | 8.50 |
| | | MoEvA2 | 31.89 |

**Attacks unaware of domain constraints most often fail.**

**C-PGD worked on a single dataset (out of two).**

**MoEvA2 has successfully attacked all models.**

* Extended to random forest

$$\omega_e \vDash (f_e = f_1 \oplus f_2)$$

Constrained Augmentation

New defense method as effective as adversarial retraining and complementary

# Our related work…

## From white-box to black-box threat models: transferability of adversarial examples

### LGV: Boosting Adversarial Example Transferability from Large Geometric Vicinity

Martin Gubri[1], Maxime Cordy[1], Mike Papadakis[1], Yves Le Traon[1], and Koushik Sen[2]

[1] University of Luxembourg, Luxembourg, LU firstname.lastname@uni.lu
[2] University of California, Berkeley, CA, USA

**Abstract.** We propose transferability from Large Geometric Vicinity (LGV), a new technique to increase the transferability of black-box adversarial attacks. LGV starts from a pretrained surrogate model and collects multiple weight sets from a few additional training epochs with a constant and high learning rate. LGV exploits two geometric properties


### Efficient and Transferable Adversarial Examples from Bayesian Neural Networks

Martin Gubri[1] Maxime Cordy[1] Mike Papadakis[1] Yves Le Traon[1] Koushik Sen[2]

[1]University of Luxembourg, Luxembourg, LU
[2]University of California, Berkeley, CA, USA

**Abstract**

An established way to improve the transferabil-


our approach can reach 94% of success rate while

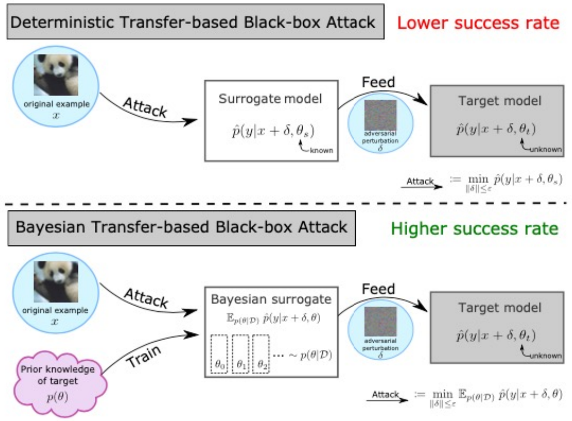



Figure 1: Illustration of the proposed approach.

gan et al., 2019]. However, a common pitfall of these models is that they are vulnerable to adversarial examples, i.e.,

# Our related work…

## Defense at low cost: using infeasible examples to protect against real-world attacks

**On The Empirical Effectiveness of Unrealistic Adversarial Hardening Against Realistic Adversarial Attacks**


***Abstract*—While the literature on security attacks and defenses of Machine Learning (ML) systems mostly focuses on unrealistic adversarial examples, recent research has raised concern about the under-explored field of realistic adversarial attacks and their implications on the robustness of real-world systems. Our paper paves the way for a better understanding of adversarial robustness against realistic attacks and makes two major contributions. First, we conduct a study on three real-world use cases (text classification, botnet detection, malware detection) and five datasets in order to evaluate whether unrealistic adversarial examples can be used to protect models against realistic examples. Our results reveal discrepancies across the use cases, where unrealistic examples can either be as effective as the realistic ones or may offer only limited improvement. Second, to explain these results, we analyze the latent representation of the adversarial examples generated with realistic and unrealistic attacks. We shed light on the patterns that discriminate which unrealistic examples can be used for effective hardening. We release our code, datasets and models to support future research in exploring how to reduce the gap between unrealistic and realistic adversarial attacks.**



***Index Terms*—adversarial attacks, constrained feature space, problem space, hardening**


However, recent studies [3], [10] have shown that in many domains, traditional adversarial attacks (e.g. PGD [11]) cannot be used for proper robustness assessment because these attacks produce examples that are not feasible (i.e. do not map to real-world objects). Indeed, while in computer vision the perturbations are simply independent pixel alterations that produce a similar image, in other domains the produced adversarial examples should satisfy specific domain constraints in order to represent real-world objects.

As a result, research has developed domain-specific adversarial attacks that either manipulate real objects through a series of problem-space transformations (i.e. *problem-space attacks*) or generate feature perturbations that satisfy domain constraints (i.e. *constrained feature space attacks*). These attacks produce examples that are realistic by design, however, at the cost of an increased computational cost compared to traditional attacks. This additional cost can be so high that it prevents the number of examples that ML engineers can use to assess and improve robustness.

In face of this dilemma between realism and computational cost, we pose the question whether we could improve model robustness against realistic examples through adversarial hardening on non-realistic examples. A positive

# THANK YOU

**Time for Q&A!**

maxime.cordy@uni.lu