“Real Attackers Don’t Compute Gradients”: Bridging the Gap between Adversarial ML Research and Practice

Giovanni Apruzzese*, Hyrum S. Anderson§, Savino Dambra¶, David Freeman†, Fabio Pierazzi‖, Kevin Roundy¶

*University of Liechtenstein, §Robust Intelligence, ¶Norton Research Group, †Meta, ‖King’s College London

{name.surname}@{uni.li*, nortonlifelock.com¶, kcl.ac.uk‖}, dfreeman@meta.com†, hyrum@robustintelligence.com§

Positions from Researchers and Practitioners

UNIVERSITÄT LIECHTENSTEIN · ROBUST INTELLIGENCE · Meta · NortonLifeLock · KING’S College LONDON

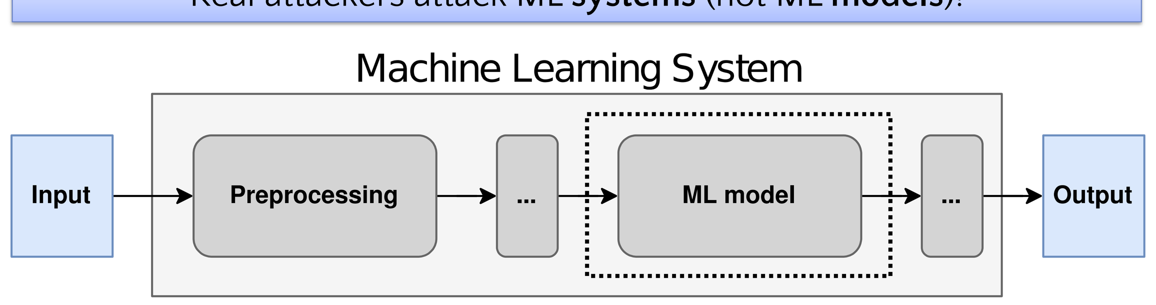
Abstract

Recent years have seen a proliferation of research on *adversarial machine learning*. Numerous papers demonstrate powerful algorithmic attacks against a wide variety of machine learning (ML) models, and numerous other papers propose defenses that can withstand most attacks. However, abundant real-world evidence suggests that actual attackers use simple tactics to subvert ML-driven systems, and as a result security practitioners have not prioritized adversarial ML defenses. Motivated by the apparent gap between researchers and practitioners, this position paper aims to *bridge* the two domains.

We first present **three real-world case studies** from which we can glean practical insights *unknown or neglected in research*. Next, we analyze **all adversarial ML papers recently published in top security conferences**, highlighting *positive trends and blind spots*. Finally, **we state positions** on precise and cost-driven threat modeling, collaboration between industry and academia, and reproducible research. We believe that our positions, if adopted, will increase the real-world impact of future endeavours in adversarial ML, bringing both researchers and practitioners closer to their shared goal of improving the security of ML systems.

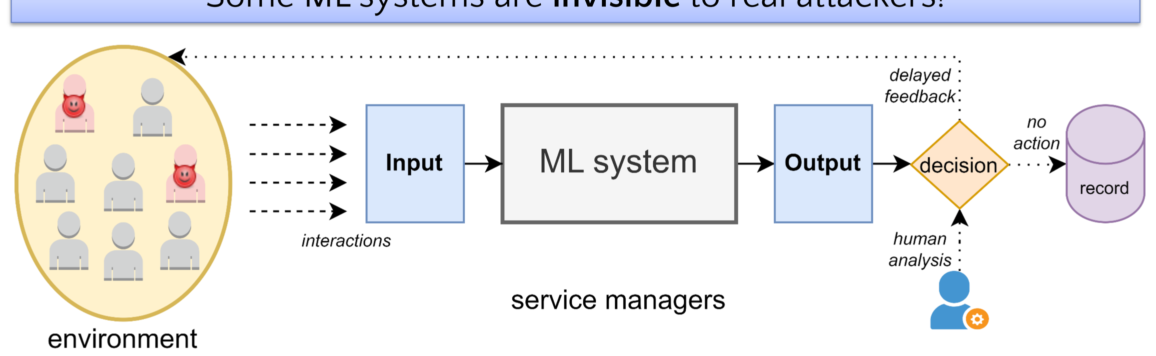
Cybersecurity and Machine Learning

Real attackers attack ML **systems** (not ML **models**)!



A ML model is just a **single component** within a much complex system. Real attackers interact with the ML system, and many things can happen *before* any input reaches the ML model, but also *after* the output is received by the attacker.

Some ML systems are **invisible** to real attackers!



Real ML systems are closed source, and some are “invisible” to real attackers, who may not receive any feedback usable for their attacks, and may not even know if such feedback is the result of their actions being analyzed by ML.

Cybersecurity is rooted in economics!

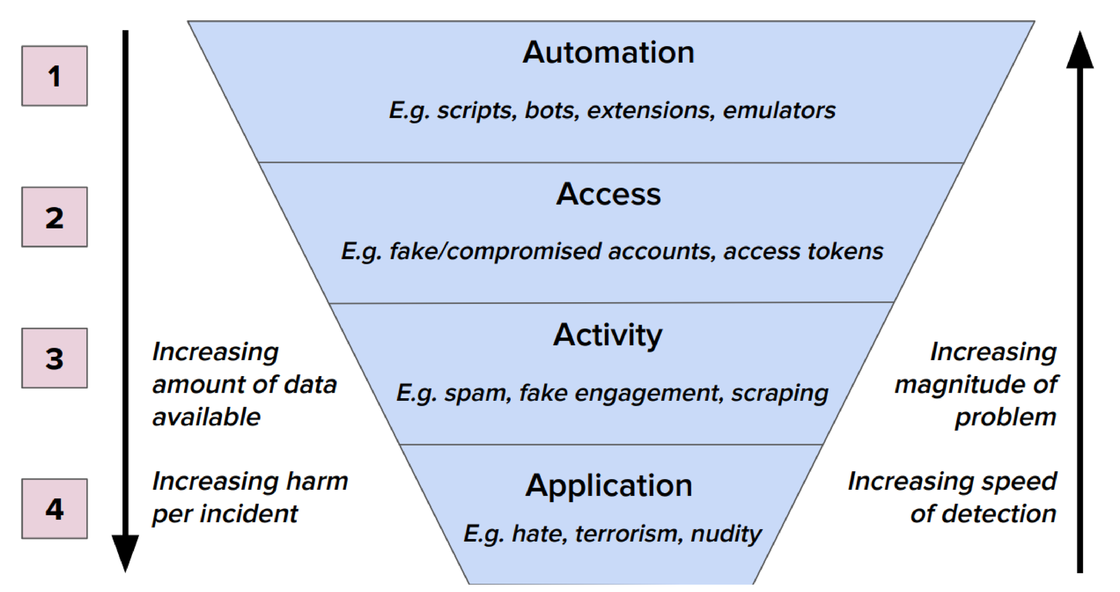
> “If you look at cybercrime in economical terms (as you should because it is a business) the optimization for an adversarial ex. is not the expensive part, it is the engineering part of building a tool that can create a diverse set of attacks with no obvious watermarks.”

A tweet by Konstantin Berlin**, head of Sophos AI**, in response to a Twitter thread entailing the participation of practitioners and researchers [63]. Operational cybersecurity is an optimization process, and developers have priorities (which apparently do not include the security of their ML components). Ultimately, “No system is foolproof”.

When asked if they secure their ML systems, practitioners reply “Why do so?” [5]

Case Studies (from industry practitioners)

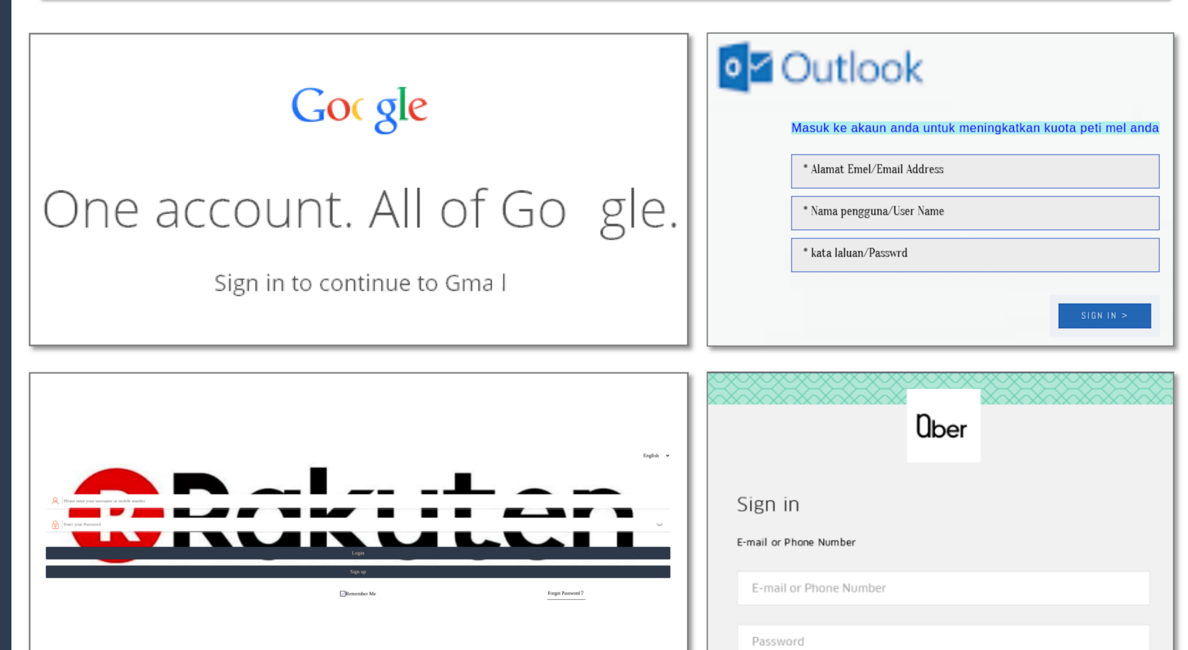
Real ML systems have **many defensive layers** (not all using ML)



The four-layered architecture of the ML-based spam detector used by **Facebook**. Most attacks can be blocked at the top layers, which not necessarily use ML (deep learning is mostly beneficial at the last layer).

→ The ML system uses both “deep” and “shallow” ML methods.
→ Only a small portion of malicious actions bypass the detector, which require huge effort

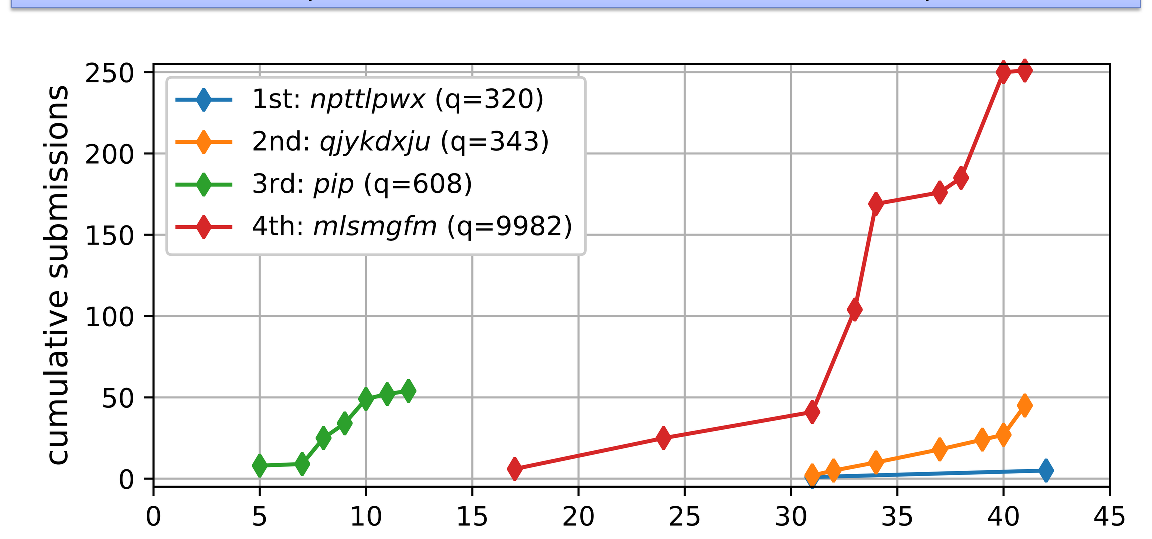
Operational ML detectors (still) **do not require gradients** to be fooled.



Some phishing webpages that are poorly recognized by a **commercial ML-based detector**. Attackers can circumvent Deep Learning methods via cheap tactics, which have been known for decades but which are still effective today.

→ Most failures of this ML detector are due to “natural” changes, unpredictable by developers.
→ We found no evidence of “adversarial examples” leveraging gradients (“probatio diabolica”)

Time is an important cost factor (more relevant than *queries*!)

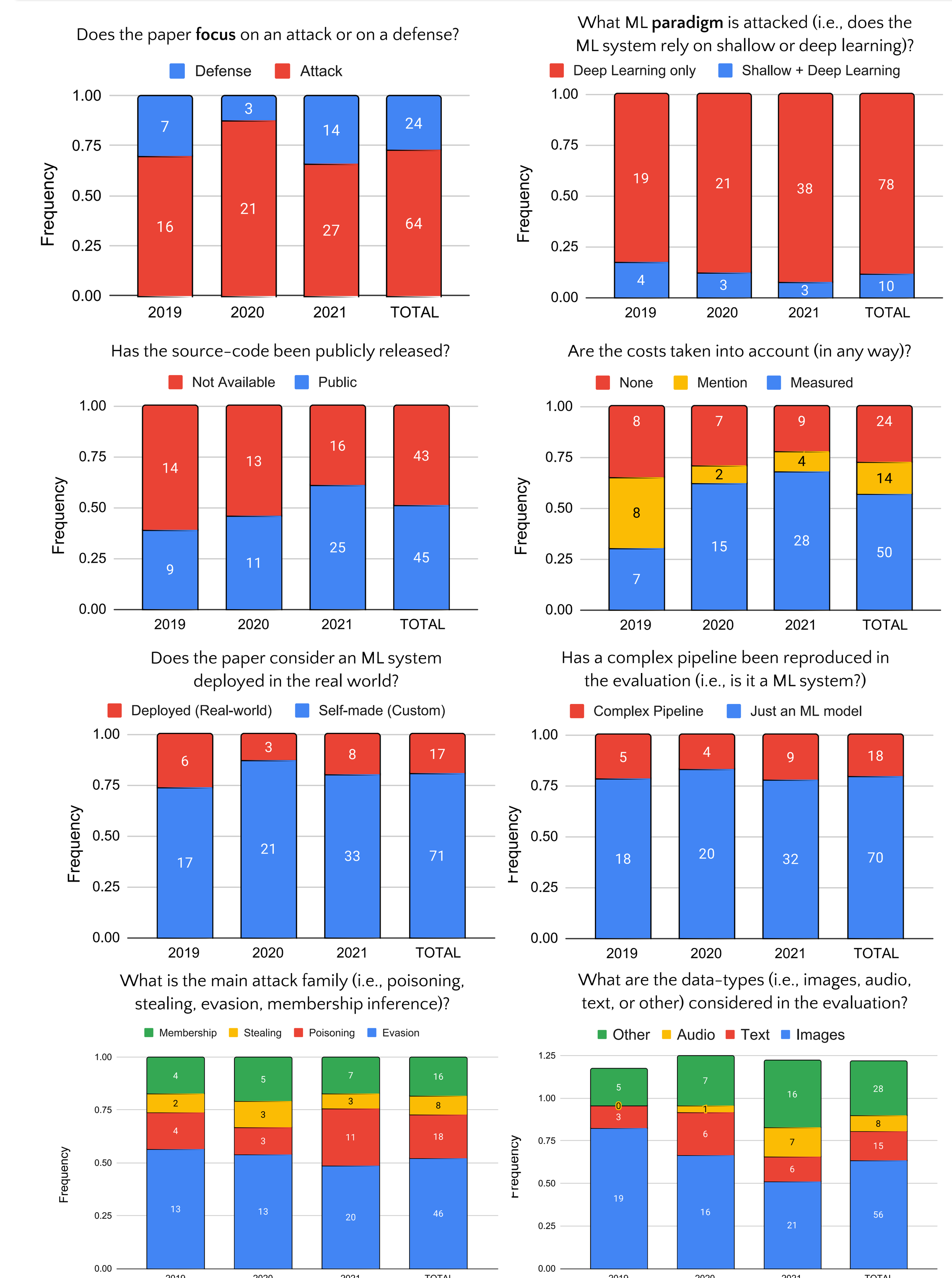


In-depth look at the **MLSEC anti-phishing evasion challenge** (2021). The plot reports the historical submissions by the top-4 ranked teams over the course of the challenge. The 1st-team (320 queries) was the last to submit their solutions.

→ Domain expertise is widely exploited by (real) attackers, who not necessarily will resort on adversarial ML techniques to reach their goals
→ Measuring “cost” via queries alone is an oversimplification: query-efficient attacks may require a lot of time!

State-of-Research (from “top-4” conferences)

Analysis of **all related papers** [2019–2021] from S&P, NDSS, Sec, CCS.



Some **inconsistencies**...

What does the attacker know? The terms “white-box” and “black-box” are widespread, but often denote different degrees of attacker’s *knowledge*.

> Co et al. [101]: “In **white-box** settings, the adversary has complete knowledge of the model architecture, parameters, and training data.[...] In a **black-box** setting, the adversary has no knowledge of the target model and no access to surrogate datasets.”

Aligns with Srndic and Laskov [43]

> Shan et al. [102]: “We assume a basic **white box** threat model, where adversaries have direct access to the the ML model, its architecture, and its internal parameter values [...] but *do not have access to the training data*.”

...this is different from [101] (“white-box”)!

> Xiao et al. [22]: “In this paper, we focus on the **white-box** adversarial attack, which means we need to access the target model (including its structure and parameters).”

...what about the training data?

> Suya et al. [103] assume a “**black-box**” attacker that “does not have direct access to the target model or knowledge of its parameters,” but that “has access to pre-trained local models for the same task as the target model” which could be “directly available or produced from access to similar training data.”

...this is different from [101] (“black-box”)!

> Hui et al. [104] envision a “**gray-box**” setting which “gives full knowledge to the adversary in terms of the model details. Specifically, except for the training data, the adversary knows almost everything about the model, such as the architecture and the hyper-parameters used for training.”

This is the exact same as [102]... which describes a “white-box” setting!

Disclaimer: taken *individually*, all past work are correct. The problems arise when analyzing the situation *as a whole*!

Our Four Positions (P)

P1: Adapt threat models to ML systems

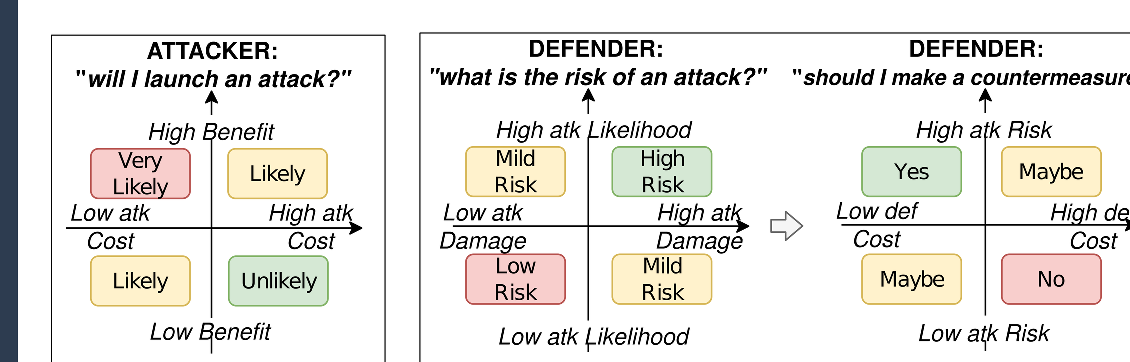
Attacker’s **Goal, Knowledge, Capabilities** and **Strategy** should reflect the ML system (and not just the ML model!)

→ Real attackers have **broader objectives** and do not want just to “evade the ML model.”

Each of those elements should be **precisely defined**.

→ Existing **terminology** is often used inconsistently.

P2: *Cost-based* threat modeling



Both attacks and defenses have a **cost**. Real attackers do not launch an attack if it is *too expensive*; and real developers will not develop a countermeasure if the attack is *unlikely to occur in reality*.

→ Measuring the cost should account for the **human factor** (queries / computation are not enough)

→ There is value also in defenses that work “only” against attackers with **limited knowledge** (since they are more common in reality).

P3: Collaborations between *industry* and *academia*

(!) Real ML systems are not open for research, and considerations on “custom-made” ML systems hardly portray realistic use cases.

(!) Even evaluations on real ML systems are hard to analyze for researchers if they cannot see what happens “inside the box.”

(!) Getting in touch with ML practitioners is daunting for researchers.

Practitioners should be **more willing** to cooperate with researchers: both have the same goal!

- Bug Bounties
- Releasing Schematics
- Streamline research collaboration process

P4: *Source-code* disclosure with “just culture”

Just Culture: assumes that mistakes are bound to occur and derive from organizational issues. Mistakes are avoided by understanding their root causes and using them as constructive learning experiences.

Embracing a just culture naturally promotes the **gradual improvement** at the base of research efforts.

→ The fast pace of research in ML can lead to errors in experiments (not always spotted during the peer-review)

→ By releasing the source code, future works can correct such mistakes, potentially systematizing them, and hence *turning “negative results” into positive outcomes* for our community.

Looking ahead, we also endorse research efforts on *forensics of adversarial examples*. Maybe real attackers **do** compute gradients... but we cannot prove it (yet)!

Acknowledgements and Remarks

The authors thank all participants of the **Dagstuhl Seminar** “Security of Machine Learning”, as most of the positions described in our paper derive from discussions originated during this event.

All our resources are available at: https://real-gradients.github.io