Funded by
the European Union

KINAITICS

From Attacks to Answers: Counterfactuals at the Intersection of Robustness and Explainability in AI

Davy Preuveneers
Industrial Research Manager, KU Leuven, Belgium

Adversarial Threats on Real Life Learning Systems
September 17th, 2025 – Esclangon building, Campus Pierre et Marie Curie, Paris

Funded by
the European Union

Overview

Counterfactuals at the Intersection of Robustness and Explainability in AI

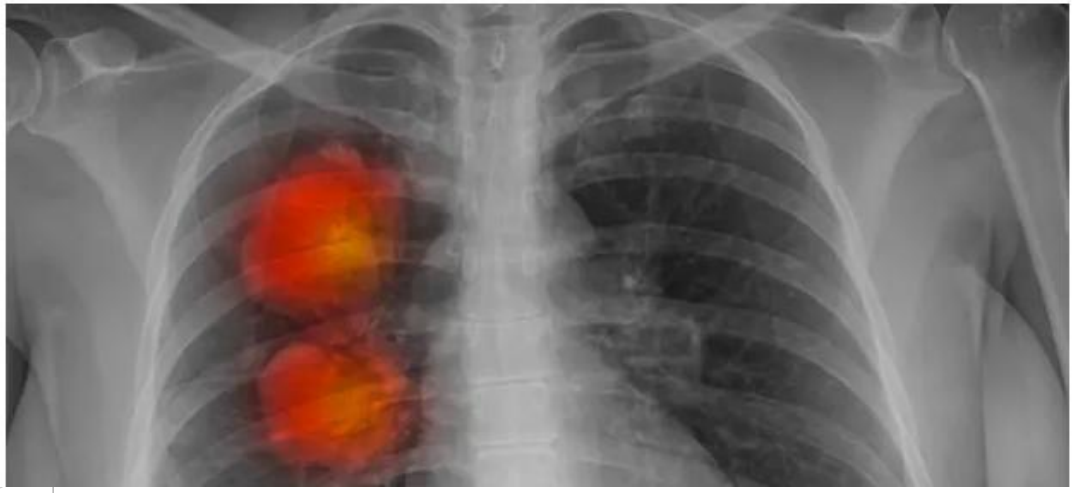
- Motivation and context
- Problem landscape
- Adversarial examples and counterfactual explanations
- Commonalities and key differences
- Future directions
- Summary and takeaways

Funded by
the European Union

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

4/27/2021
11:00 AM

Expect an Increase in Attacks on AI Systems

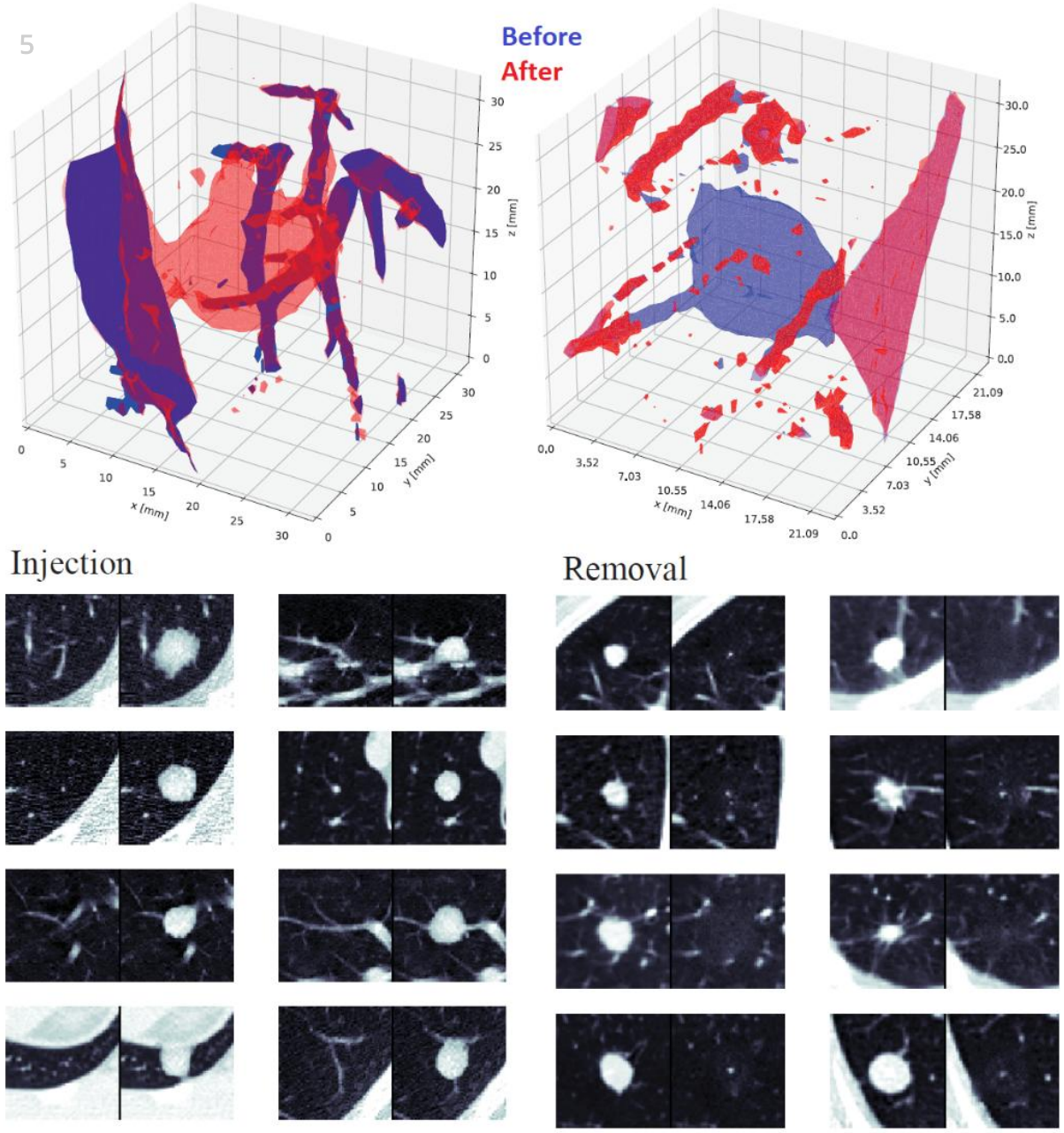
Robert Lemos
News

Companies are quickly adopting machine learning but not focusing on how to verify systems and produce trustworthy results, new report shows.

Research into methods of attacking machine-learning and artificial-intelligence systems has surged—with nearly 2,000 papers published on the topic in one repository over the last decade—but organizations have not adopted commensurate strategies to ensure that the decisions made by AI systems are trustworthy

https://www.darkreading.com/vulnerabilities---threats/advanced-threats/expect-an-increase-in-attacks-on-ai-systems/d/d-id/1340833

Funded by
the European Union



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

BAD BOTS ON THE RISE: INTERNET TRAFFIC HITS RECORD LEVELS

09 JUN 2025

CYBERTHREATS | CYBERATTACKS | CYBERSECURITY | DATA PRIVACY | DATA BREACHES | INNOVATION | DIGITAL TRANSFORMATION | REVIEW | DIGITAL IDENTITY AND SECURITY | AI

"More than half of all online traffic is generated by automated software programs known as bots."

https://www.thalesgroup.com/en/worldwide/digital-identity-and-security/magazine/bad-bots-rise-internet-traffic-hits-record-levels

2025 Bad Bot Report by Imperva

Funded by
the European Union

The web bot arms race: detection and evasion

Motivation and key challenges

- Modern web bots mimic human behavior to evade detection
- They exploit residential IPs and proxies
- Traditional detection methods (e.g., rule-based) ineffective
- AI models misclassify traffic: false positives and negatives
- AI models vulnerable to unseen behaviors and evasion attacks

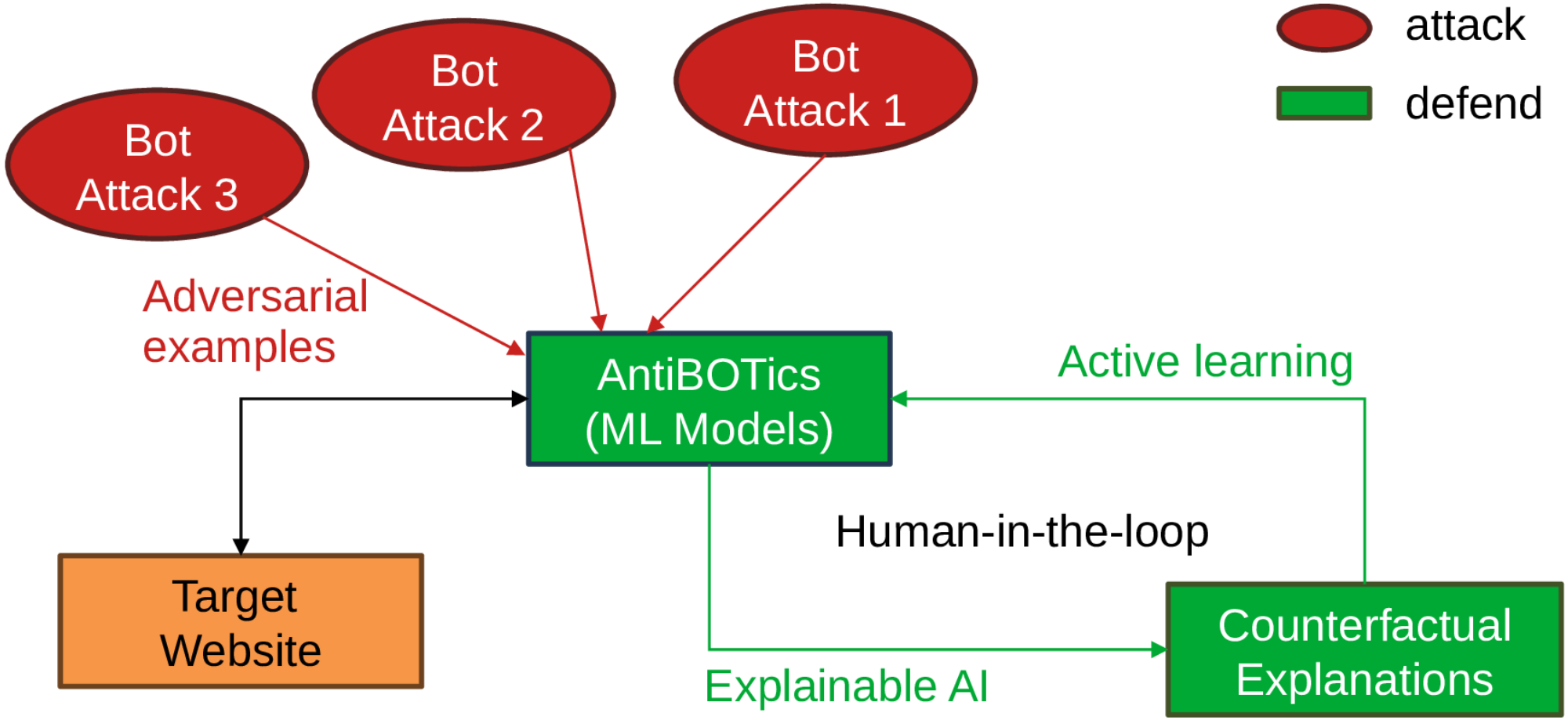
→ Detection must evolve continuously!

→ But manual verification and sample labeling is burdensome!

Funded by
the European Union

The web bot arms race: detection and evasion

AI-based attacks and defenses



Funded by
the European Union

AntiBOTics: AI-based web bot attack detection

Robust ML models for a hybrid AI defense system

- Feature extraction from HTTP logs over 10, 100 and 1000-second intervals

Feature Group	Description
Total Number of Requests *	The cumulative count of HTTP requests made.
HTTP Request Methods *	Distribution of HTTP methods used (e.g., GET, POST, PUT, HEAD).
HTTP Response Codes *	Distribution of HTTP response status codes received (e.g., 200, 301, 404).
Session Duration	Total time elapsed from the first to the last request in a session.
Time Between Requests *	Average interval between consecutive HTTP requests.
Unique URLs Accessed *	Count of distinct URLs visited.
User-Agent String	Analysis of the User-Agent header to identify the browser or bot type.
Referrer Analysis	Examination of the HTTP Referrer header to understand navigation paths.
Request Patterns *	Detection of repetitive or predictable request sequences.
Resource Types Requested *	Types of resources requested (e.g., HTML, images, scripts) and their frequencies.

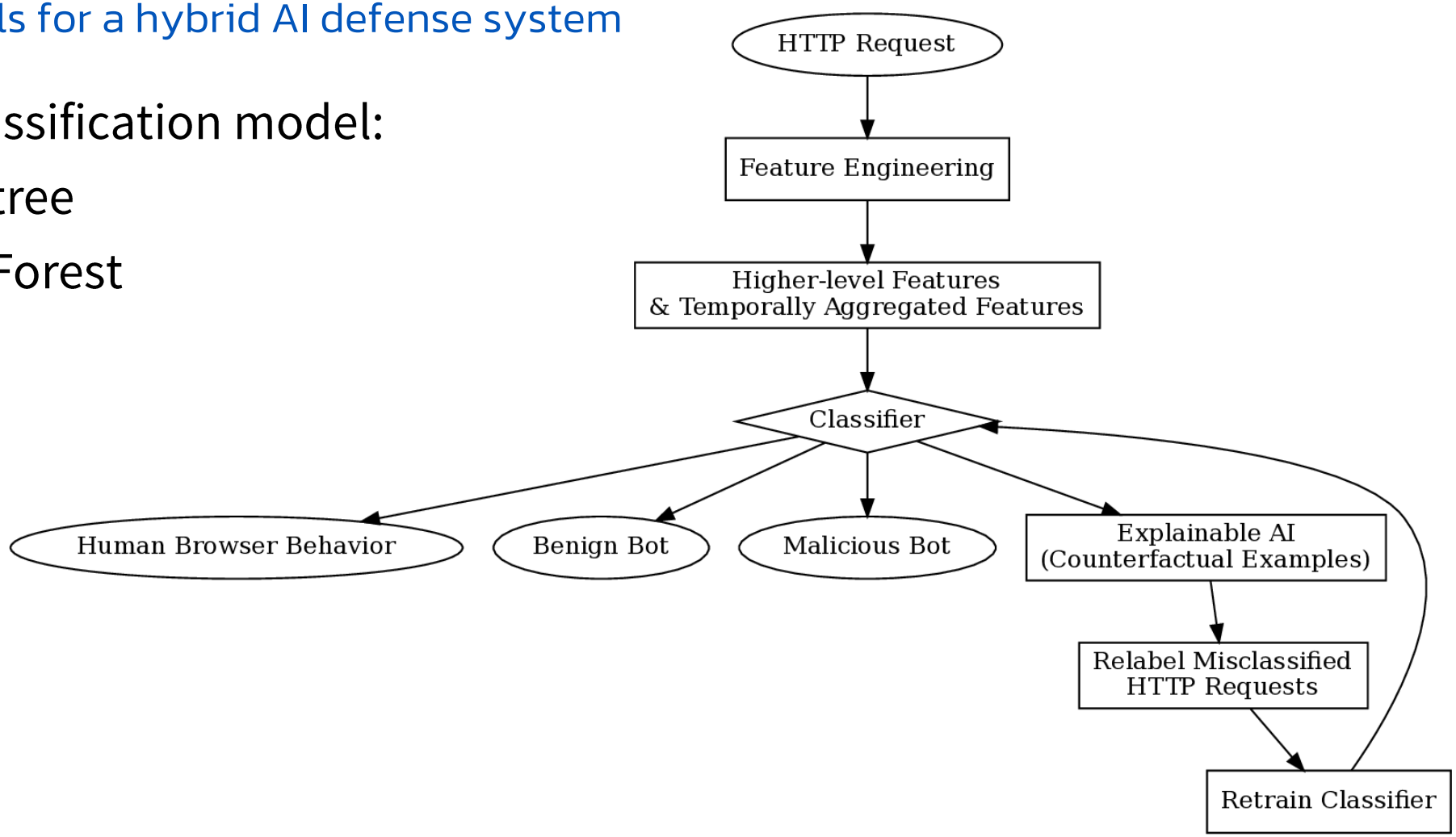
C. Iliou et al., *Detection of advanced web bots by combining web logs with mouse behavioural biometrics*, Digital threats: research and practice 2 (2021), 1–26.

Funded by
the European Union

AntiBOTics: AI-based web bot attack detection

Robust ML models for a hybrid AI defense system

- Training a classification model:
  - Decision tree
  - Random Forest
  - XGBoost
  - CatBoost
  - …

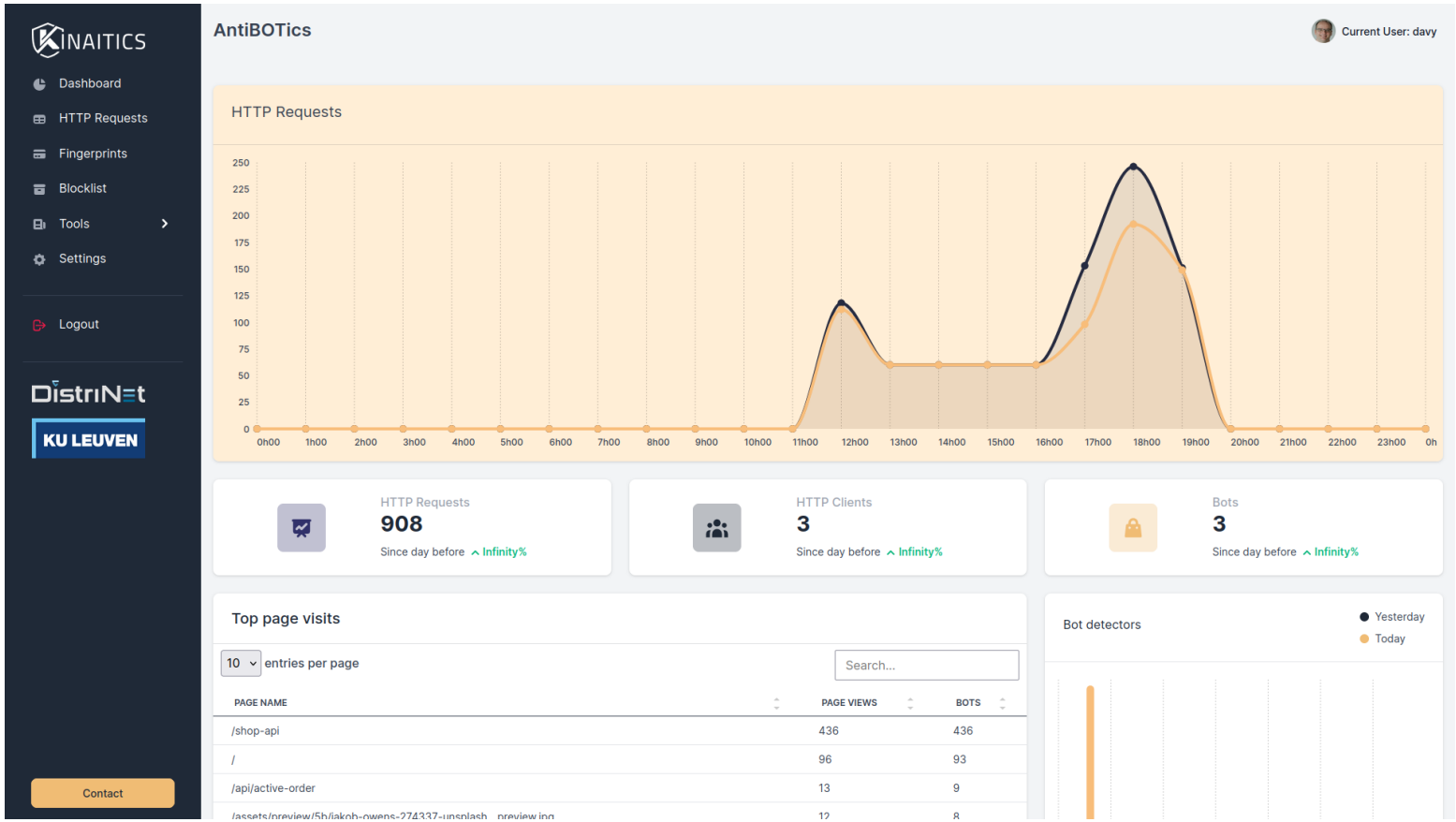


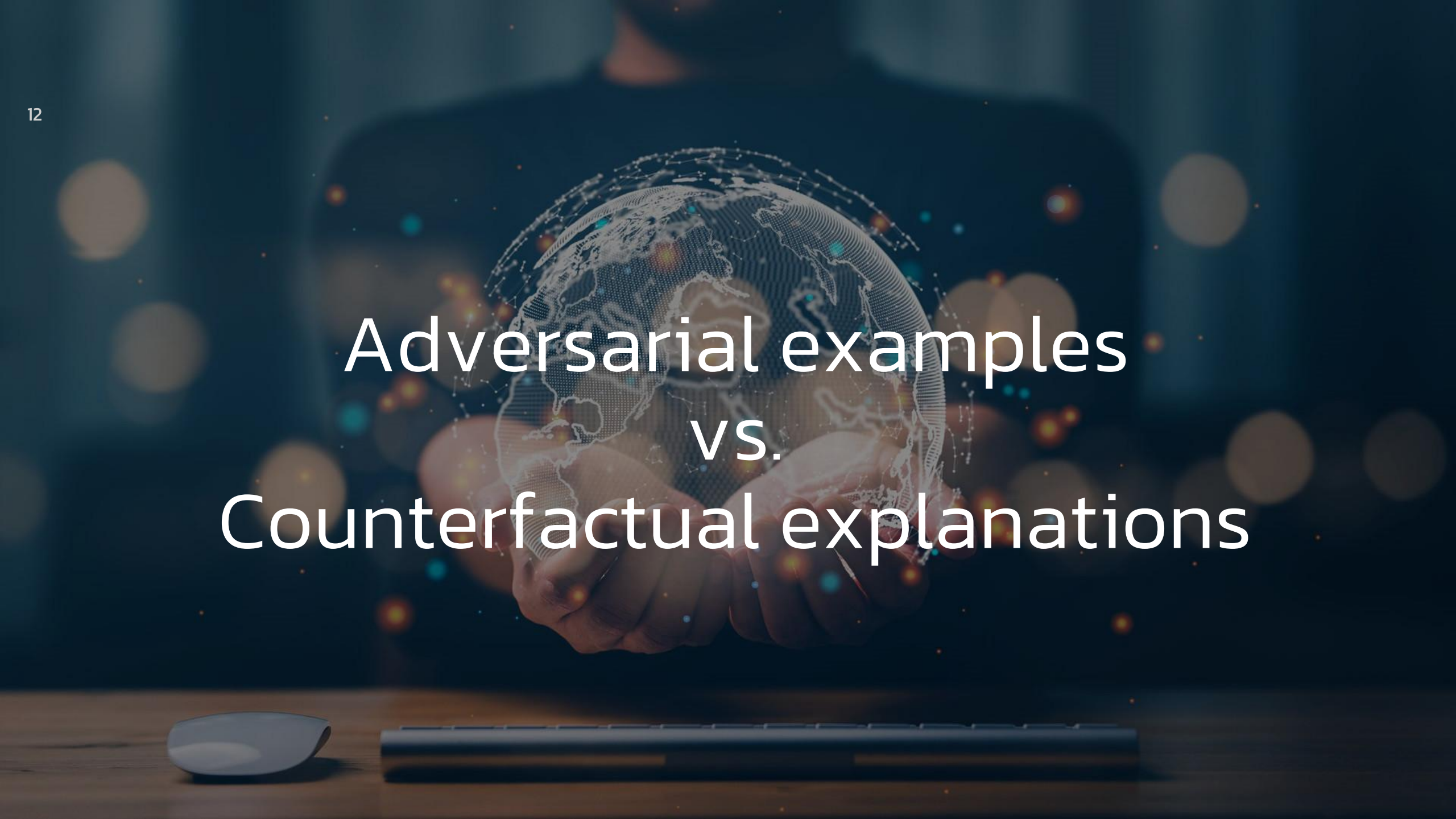
Funded by
the European Union

AntiBOTics: AI-based web bot attack detection

Robust ML models for a hybrid AI defense system

- **Key features**
  - Checklists (e.g. IP address)
  - Browser Fingerprinting
  - Heuristics / Statistics
  - Mix of ML Classifiers
- **Human-in-the-loop**
  - Explainable AI with Counter-factual Explanations
  - ML Model Updates Through Active Learning





Adversarial examples vs. Counterfactual explanations

Funded by
the European Union

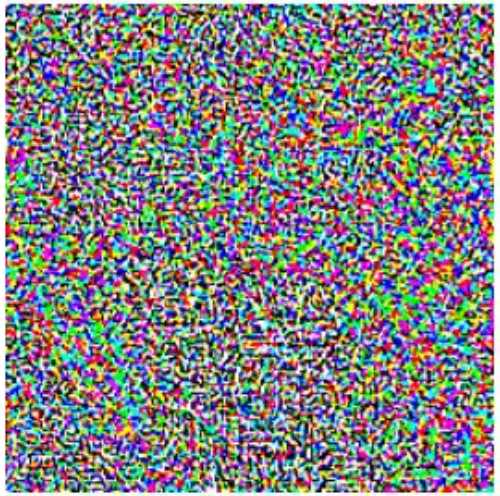
Adversarial examples

Modify inputs slightly to change model predictions

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

- Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014). https://arxiv.org/abs/1412.6572

Funded by
the European Union

Adversarial examples

Modify inputs slightly to change model predictions

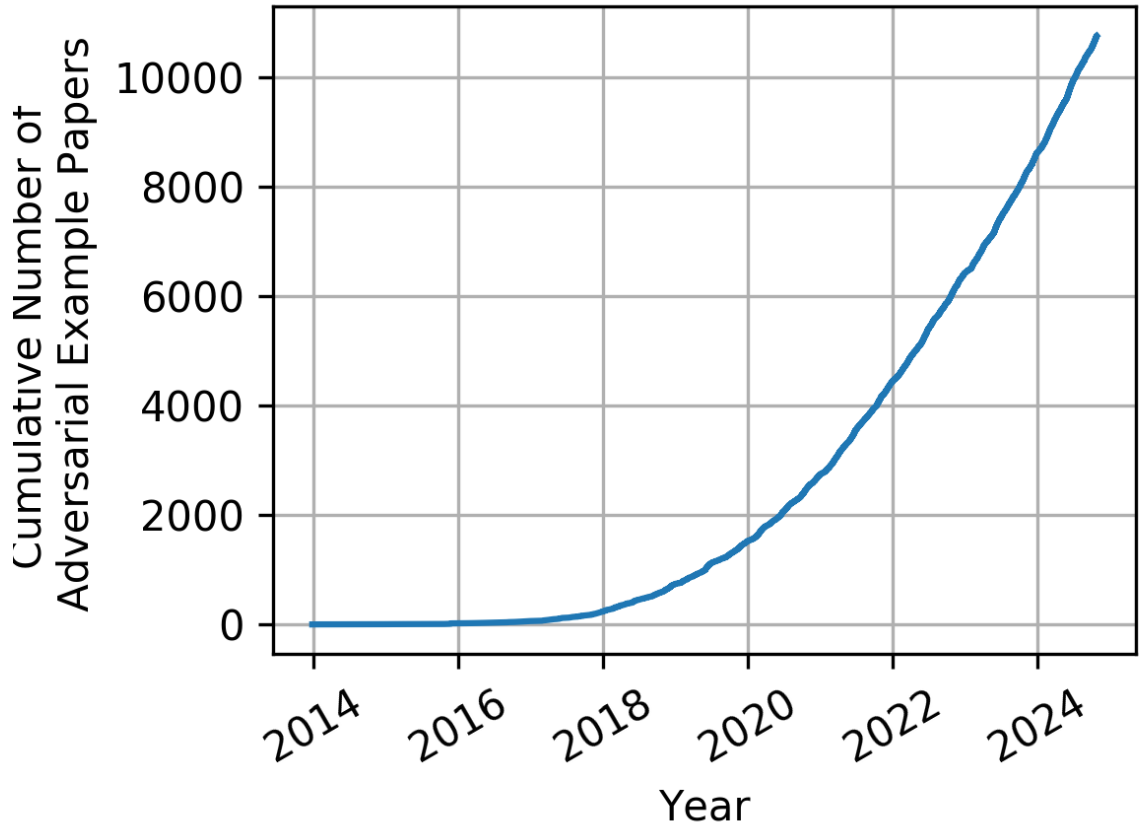
Attackers provide unusual inputs in the physical world

- Eykholt et al., *Robust Physical-World Attacks on Deep Learning Models*, CVPR, 2017.

Funded by
the European Union

Adversarial examples

A Complete List of All (arXiv) Adversarial Example Papers



- Nicolas Carlini (Google DeepMind)
- https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html

Funded by
the European Union

Adversarial examples with constrained domains

Modify inputs slightly to change model predictions

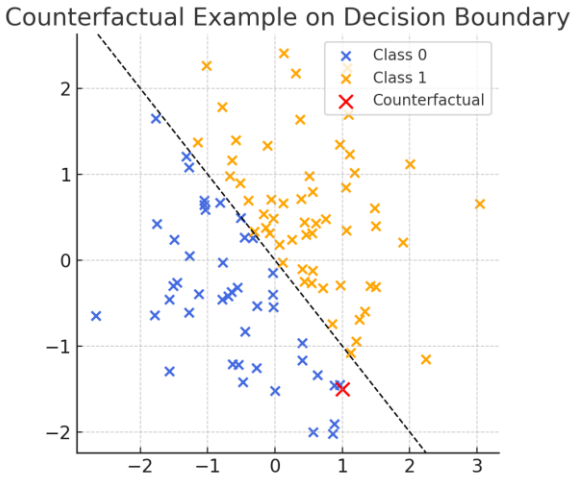
- Restrictions on Adversarial Examples:
  - **Malware**: *“feature changes can cause the application in question to lose its malware functionality in parts or completely”*
    - Grosse, Kathrin, et al. "Adversarial examples for malware detection." *European symposium on research in computer security*. Cham: Springer International Publishing, 2017.
  - **Network packets:** *“network packets that do not obey the TCP/IP protocol would not be permissible”*
    - Sheatsley, Ryan, et al. "Adversarial examples in constrained domains." *arXiv preprint arXiv:2011.01183* (2020).

Funded by
the European Union

Counterfactual explanations

Modify inputs slightly to change model predictions

- **Counterfactual example**: type of explanation that shows how a small change to the input would lead to a different prediction by the model.



Answer the 'what-if' question for stakeholders
Example: 'If income ↑ $5k → loan approved'

- Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." *Harv. JL & Tech.* 31 (2017): 841.
- Guidotti, Riccardo. "Counterfactual explanations and how to find them: literature review and benchmarking." *Data Mining and Knowledge Discovery* 38.5 (2024): 2770-2824.

Funded by
the European Union

Counterfactual examples as an “explainable AI” technique

When misclassifications happen

- DiCE [4,5]: a model-agnostic approach to implementing counterfactual examples
- Find them as an optimization problem, like constructing adversarial examples:
  - Feature perturbations alter the outcome of the classification model
  - Feature perturbations must be diverse and realistically feasible

- From high-level features to HTTP requests?

  ➡ Temporal constraints and dependencies across features

[4] https://interpret.ml/DiCE/

[5] R. K. Mothilal, A. Sharma, C. Tan, *Explaining machine learning classifiers through diverse counterfactual explanations*, in: Proceedings of the 2020 conference on fairness, accountability, and transparency, 2020, pp. 607–617.

Funded by
the European Union

Causally and temporally constrained counterfactual explanations

Extending DiCE with a new sampling technique

- Web logs (input space) to about 100 high-level features (feature space) (web logs)
- Examples of dependencies and constraints across features:
  1. **Causal constraint:** Increasing the "number of GET requests" feature:
     - Increases "total number of requests" feature
     - Alters "GET/POST/PUT ratio" feature
     - Potentially affect "HTML-to-image ratio" feature
     - …
  2. **Temporal constraint:** Aggregating smaller intervals (e.g. 10 s) into larger ones (e.g. 100 s) means that changes in the former feature propagate to the larger

Funded by
the European Union

Causally and temporally constrained counterfactual explanations

Extending DiCE with a new sampling technique

- **Solution**:
  1. Extending DiCE with a constrained random sampling method by modeling dependencies across features as a graph
  2. Validity checking and projection

- **Validity checking:** Is updated feature vector still a counterfactual example?
- **Projection:** Improve the quality of the counterfactual example by minimizing deviations from the original instance

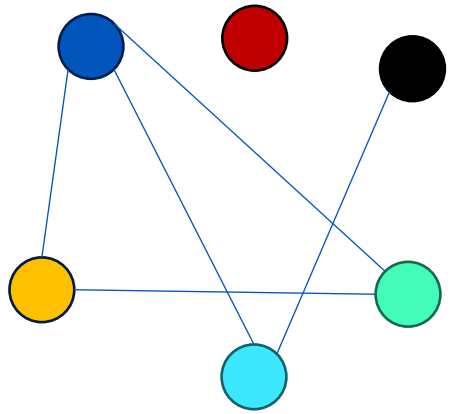
- Details: see paper later on

Funded by
the European Union

Causally and temporally constrained counterfactual explanations

Extending DiCE with a new sampling technique

- Extending DiCE with a constrained random sampling method by modeling dependencies across features as a graph



Algorithm 1 Iterative Adjustment of Causally Dependent and Time-Aggregated Features

Require: Feature list F, dependency graph G, original feature vector X_{orig}, perturbed feature vector X_{pert}
Ensure: Updated feature vector X_{final} satisfying dependencies and constraints

$X_{final} \leftarrow X_{pert}$
$updateQueue \leftarrow$ all perturbed features in X_{pert}
$updateCount \leftarrow$ dictionary mapping each feature to 0
$maxUpdates \leftarrow$ predefined upper limit for feature updates
while $updateQueue$ is not empty **do**
    $f \leftarrow$ dequeue from $updateQueue$
    if $updateCount[f] \geq maxUpdates$ **then**
        continue ▷ Avoid infinite loops due to cyclic dependencies
    end if
    $updateCount[f] \leftarrow updateCount[f] + 1$
    for each dependent feature g in G connected to f **do**
        Adjust $X_{final}[g]$ based on $X_{final}[f]$ and dependencies
        if g not in $updateQueue$ **then**
            Enqueue g into $updateQueue$
        end if
    end for
    for each time-aggregated feature h linked to f **do**
        Adjust $X_{final}[h]$ to maintain consistency over time scales
        if h not in $updateQueue$ **then**
            Enqueue h into $updateQueue$
        end if
    end for
end while
return X_{final}

Funded by
the European Union

Counterfactual explanations

Modify inputs slightly to change model predictions

```
X_test = pd.read_pickle("x_test.pkl.gz")
Y_test = pd.read_pickle("y_test.pkl.gz")
test_dataset = pd.concat([X_test, Y_test], axis=1)
column_names = list(X_test.columns)

num_features = [f for f in X_test.columns if "_10" in f]
num_features.extend((["response_size", "request_size"]))

# initiate DiCE
d = dice_ml.Data(dataframe=train_dataset, continuous_features=num_features, outcome_name="label")
model = pickle.load(open("catboost.pkl.gz", "rb"))
m = dice_ml.Model(model=model, backend="sklearn")
exp = dice_ml.Dice(d, m, method="constrained_random")  # custom sampling method

query_instance = test_dataset[test_dataset["label"] == 1].sample(n=1)
query_instance.drop(["label"], axis=1, inplace=True)

# generate counterfactuals
dice_exp = exp.generate_counterfactuals(query_instance, total_CFs=1, desired_class="opposite", proximity_weight=1.5,
                                        diversity_weight=1.0, features_to_vary=column_names, posthoc_sparsity_param=0)

print(dice_exp.visualize_as_list(show_only_changes=True))
```

Are adversarial and counterfactual examples the same?

Funded by
the European Union

Commonalities

Modify inputs slightly to change model predictions

Aspect	Adversarial Examples	Counterfactual Explanations
Minimal Perturbation	Both focus on small changes to the input	Both find the smallest change possible
Boundary Crossing	Crosses the model's decision boundary	Also crosses the decision boundary, but for explanation
Optimization Problem	Often solved by optimizing a loss function	Similarly uses optimization to find minimal modifications
Input Similarity	Changes are typically imperceptible or minimal	Changes are designed to be close to the original input

Funded by
the European Union

Key differences

Modify inputs slightly to change model predictions

Aspect	Adversarial Examples	Counterfactual Explanations
Primary Goal	Fool the model	Explain the model's decision
Perspective	Attacker-centric	User-centric
Optimization Objective	Minimize input perturbation while **maximizing misclassification**	Minimize input perturbation while **achieving a specific target outcome**
Plausibility of Change	Perturbations may be **unrealistic** (e.g., random pixel noise)	Changes must be **plausible and actionable** (e.g., higher income, lower debt)
Output Usefulness	Rarely interpretable for humans	Designed for human interpretability
Applications	Model robustness, security testing, adversarial training	Explainable AI, fairness auditing, regulatory compliance

Funded by
the European Union

Bridging explainability and adversarial-generation methods

Modify inputs slightly to change model predictions

- Towards a unified view on perturbations:
  - **Adversarial training** improves model resilience by exposing it to worst-case perturbations. However, adversarial examples are often unnatural and hard for humans to interpret (e.g., tiny imperceptible pixel changes).
  - **Counterfactual explanations**, on the other hand, generate perturbations aligned with human reasoning (e.g., “If this feature changed slightly, the prediction would flip”).
  - A hybrid approach ensures the model is robust to malicious, imperceptible changes **while also being transparent in how decisions could change under meaningful, human-aligned modifications.**

Funded by
the European Union

Bridging explainability and adversarial-generation methods

Modify inputs slightly to change model predictions

- Unified objective (minimize):

$$\mathcal{L}(x') = \underbrace{\lambda_1 d(x, x')}_{\text{proximity}} + \underbrace{\lambda_2 \ell_{task}\left(f(x'), y_{target}\right)}_{\text{target attainment}} + \underbrace{\lambda_3 \phi\left(C(x')\right)}_{\text{plausibility penalty}} + \underbrace{\lambda_4 \psi\left(A(x')\right)}_{\text{actionability penalty}} + \underbrace{\lambda_5 H(x')}_{\text{human preference / safety}}$$

- x = factual input, $y = f(x)$ predicted label or score.
- x′ = candidate modification (adversarial or counterfactual).
- d(·,·) = distance on input or latent manifold (e.g., L2 in latent space or edit distance in feature space).
- C(·) = plausibility / causal constraints (learned density model, generative prior, causal graph).
- $A(\cdot)$ = actionability constraints (which features a user can change).
- H(·) = human feedback / preference penalty (from HITL)

- Goldwasser, Jeremy, and Giles Hooker. "Unifying Image Counterfactuals and Feature Attributions with Latent-Space Adversarial Attacks." *arXiv preprint arXiv:2504.15479* (2025).

Funded by
the European Union

Bridging explainability and adversarial-generation methods

Modify inputs slightly to change model predictions

- Unified objective (minimize):

$$\mathcal{L}(x') = \underbrace{\lambda_1 d(x, x')}_{\text{proximity}} + \underbrace{\lambda_2 \ell_{task}\left(f(x'), y_{target}\right)}_{\text{target attainment}} + \underbrace{\lambda_3 \phi\left(C(x')\right)}_{\text{plausibility penalty}} + \underbrace{\lambda_4 \psi\left(A(x')\right)}_{\text{actionability penalty}} + \underbrace{\lambda_5 H(x')}_{\text{human preference / safety}}$$

- For **adversarial examples**, you set λ_2 to encourage any misclassification and usually omit C, ψ, H (or set them weak).
- For **counterfactuals**, you set λ_2 to target a specific outcome and **strongly** enforce C, ψ, H so outputs are plausible, causal, and actionable.
- Training and evaluation can share the same solver (gradient attack in latent space, mixed-integer solver for discrete features, or search over generative-model latent codes).

Funded by
the European Union

Experimental setup and evaluation

A human-in-the-loop analysis

- **Generation and retraining with counterfactual examples:** A HTTP request initially classified as malicious (label = 1) needs to be modified to benign (label = 0)

Feature	Original Value	Target Value
html_pages_10	21.0	1.5
label	1	0

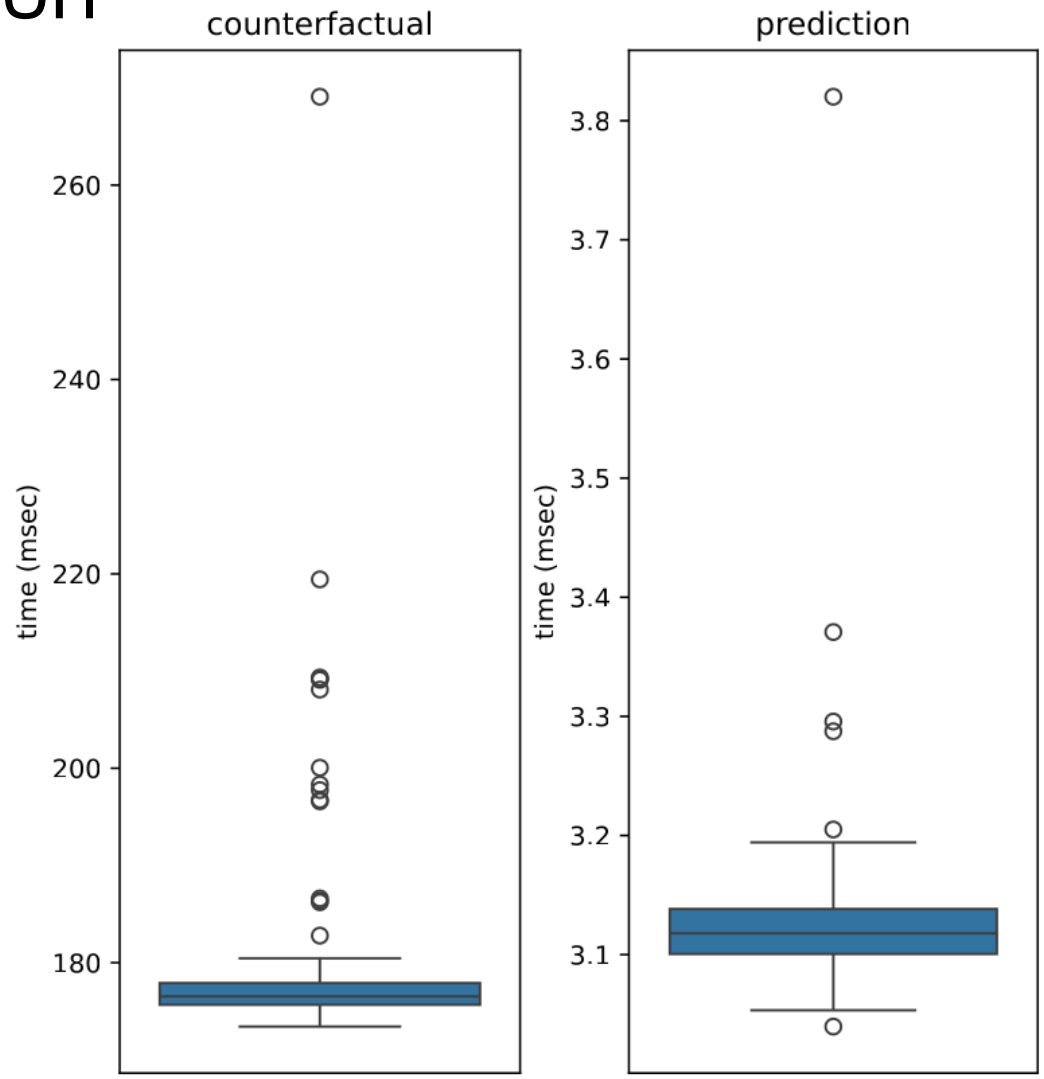
  - A malicious bot sent 21 HTTP requests for HTML pages within a 10-second window, significantly exceeding the browsing speed of a typical user
  - Such a perturbation would not be realistic, but a perturbation to 20.0 might
- Multiple attempts required to generate counterfactual examples!

Funded by
the European Union

Experimental setup and evaluation

A human-in-the-loop analysis

- **Benchmarking time:** counterfactual example vs. CatBoost prediction on a system with:
  - 12th Gen Intel Core i9-12900 CPU
  - 32 GB of memory
- Counterfactual example generation is almost **two orders of magnitude** slower
- Why? Costly iterative approach to optimizing counterfactual examples
- Only effective under selective generation



(a) Counterfactual example (b) CatBoost prediction

Funded by
the European Union

Ongoing work

Adversarial and counterfactual examples

- **Practical HITL loop (iterative):** simultaneously improves robustness and the quality of counterfactual explanations
  - Generate candidate x’ (adversarial or counterfactual) using latent-space perturbation or constrained search.
  - Present top-k candidates via UI with explanations of changed features.
  - Collect human feedback: accept/reject, plausibility score, improvement suggestions, or alternative target choices.
  - Update models: (i) retrain plausibility/actionability scorer, (ii) adversarially-train the predictor using accepted adversarial seeds, (iii) refine generator to prefer human-validated CFs.

Funded by
the European Union

Future directions

Unifying Adversarial Examples, Counterfactual Explanations & HITL

- **Causality-Aware Explanations**
  - Integrate **causal models** into counterfactual and adversarial generation.
  - Ensure perturbations respect **real-world causal dependencies.**
  - Study **counterfactual fairness** — guaranteeing equitable outcomes across groups.

- **Robustness + Explainability Co-Training**
  - Combine **adversarial training** with **counterfactual recourse** to improve both robustness and interpretability.
  - Develop multi-objective training pipelines balancing:
    - Prediction accuracy
    - Robustness against attacks
    - Quality of explanations

Funded by
the European Union

Summary & Takeaways

Adversarial examples and counterfactual explanations

- Adversarial robustness and explainability are interlinked.
  - Both AEs and CFs involve minimal input perturbations
  - Both operate near decision boundaries
  - Both can be formulated as optimization problems
- Why unification matters
  - For Research: Shared frameworks improve understanding of decision boundaries.
  - For Practice: Combining adversarial training + counterfactual recourse → robust, interpretable, and user-friendly models.
  - With HITL: Humans guide plausibility, fairness, and actionability, bridging security and interpretability goals.
- Unifying them yields more robust and interpretable AI systems.

Funded by
the European Union

References

Next steps

- Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." *Harv. JL & Tech.* 31 (2017): 841.
- Pawelczyk, Martin, et al. "Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis." *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.
- Guidotti, Riccardo. "Counterfactual explanations and how to find them: literature review and benchmarking." *Data Mining and Knowledge Discovery* 38.5 (2024): 2770-2824.
- Goldwasser, Jeremy, and Giles Hooker. "Unifying Image Counterfactuals and Feature Attributions with Latent-Space Adversarial Attacks." *arXiv preprint arXiv:2504.15479* (2025).

Funded by
the European Union

Thanks!

Any questions?

Davy Preuveneers

Davy.Preuveneers@kuleuven.be