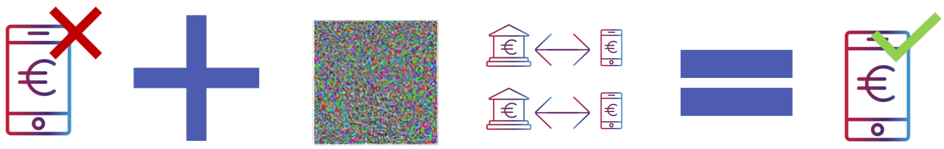
Towards Adaptive Attacks on Constrained Tabular ML

T. Simonetto[1], S. Ghamizi[2], M. Cordy[1]
[1]University of Luxembourg
[2]RIKEN AI, Luxembourg Institute of Science and Technology

Adversarial examples are subject to domain constraints



Refused transaction + Noise / Additional transactions = Accepted transaction

Objective: for x find δ

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

Adversarial examples are inputs carefully designed to cause **erroneous predictions** in machine learning systems. To fool **real-world** systems, they must **respect domain constraints**.

Constraints as a penalty function

Example of constraint:

$$\omega_1 \equiv installment = loan\_amount \times \frac{int\_rate \times (1 + int\_rate)^{term}}{(1 + int\_rate)^{term} - 1}$$

Penalty function:

$$penalty(x, \omega_1) = |installment - (loan\_amount \times \frac{int\_rate \times (1 + int\_rate)^{term}}{(1 + int\_rate)^{term} - 1})|$$

Loss function:

$$\mathcal{L}'(x) = \mathcal{L}(x, y, h, \Omega) = l(h(x), y) - \sum_{\omega_i \in \Omega} penalty(x, \omega_i)$$

Constrained Adaptive PGD

Gradient step momentum:

$$z^{(k+1)} = P_{\mathcal{S}}(x^{(k)} + \eta^{(k)}(\nabla \mathcal{L}'(x^{(k)}))$$

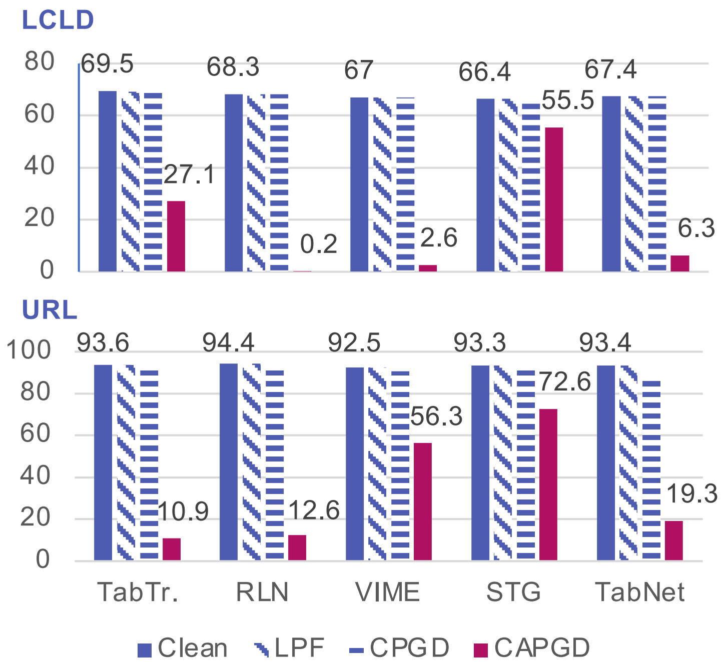
$$x^{(k+1)} = R_\Omega(P_{\mathcal{S}}(x^{(k)} + \alpha \cdot (z^{(k+1)} - x^{(k)}) + (1-\alpha) \cdot (x^{(k)} - x^{(k+1)})))$$

Step size adaptation

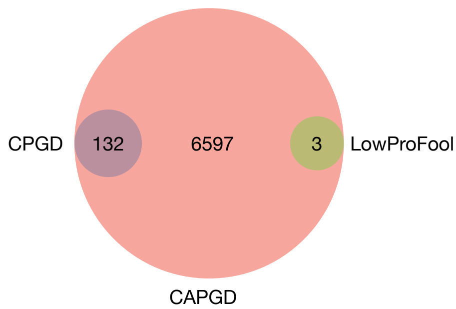
$$\sum_{i=w_{j-1}}^{w_j - 1} \mathbf{1}_{\mathcal{L}'(x^{(i+1)}) > \mathcal{L}'(x^{(i)})} < \rho \cdot (w_j - w_{j-1})$$

Repair operator: project values of feature in equality constraints.

Robust Accuracy (%)



CAPGD subsumes other gradient attacks



Number of successful adversarial examples

Next steps

CAA: An adaptive attack combining strong complementary attacks.

TabularBench: a comprehensive benchmark of robustness of tabular deep learning classification models.

More information