Northeastern University, Boston, MA, USA

Pluribus One
seeing one in many

Pattern Recognition and Applications Lab

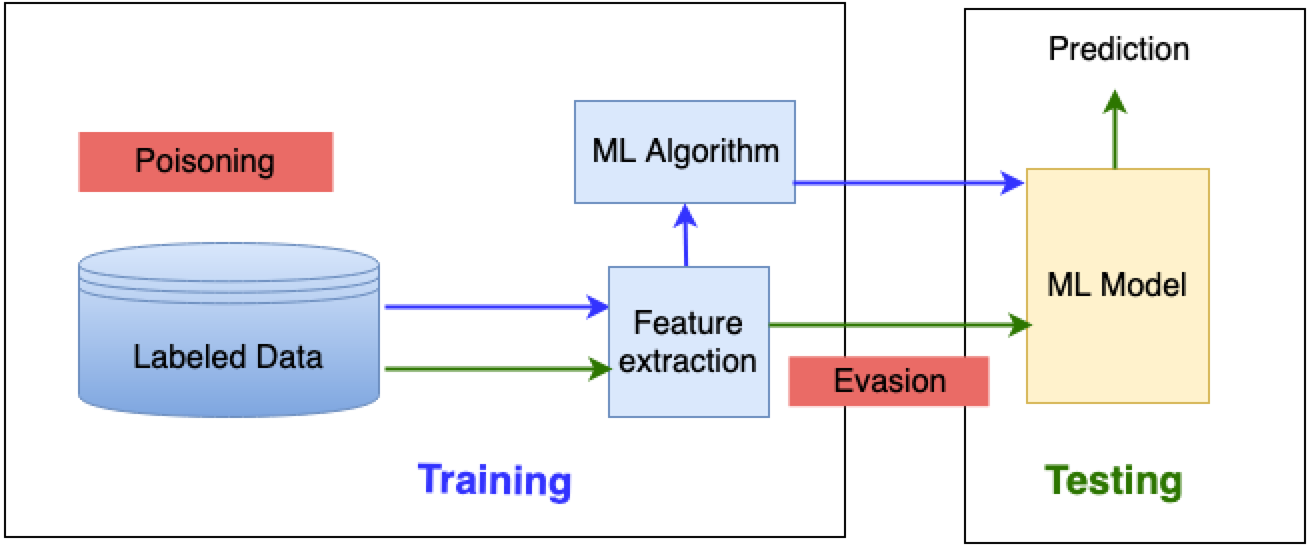
University of Cagliari, Italy

Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks

Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio,
Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli

Usenix Security Symposium 2019, Aug. 14-16, Santa Clara, California, USA

Attacks against machine learning

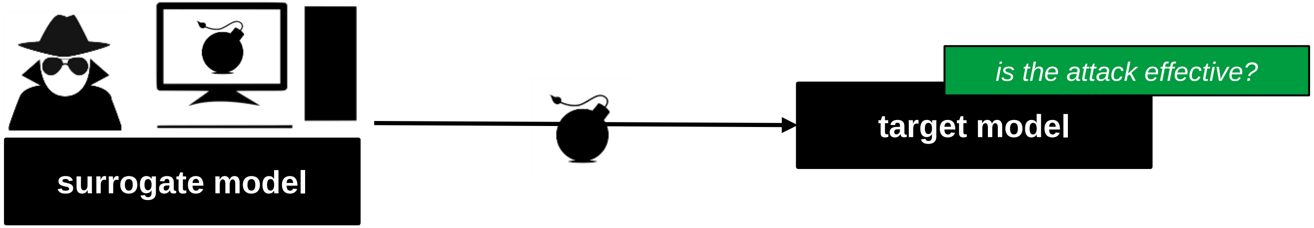


Threat model

- Evasion: add minimum amount of perturbation to a test point to change prediction
- Poisoning: add a fraction of poisoning points in training to degrade model accuracy (availability attack)
- Attacker Knowledge
  - White box: full knowledge of the ML system
  - Black-box: query access to the model

Why study transferability?

- **Transferability:** the ability of an attack, crafted against a **surrogate** model, to be effective against a different, *unknown* **target** model [1,2]



- **Open problems:**
  - What are the factors behind the transferability of evasion and poisoning attacks?
  - When and why do adversarial attacks transfer?

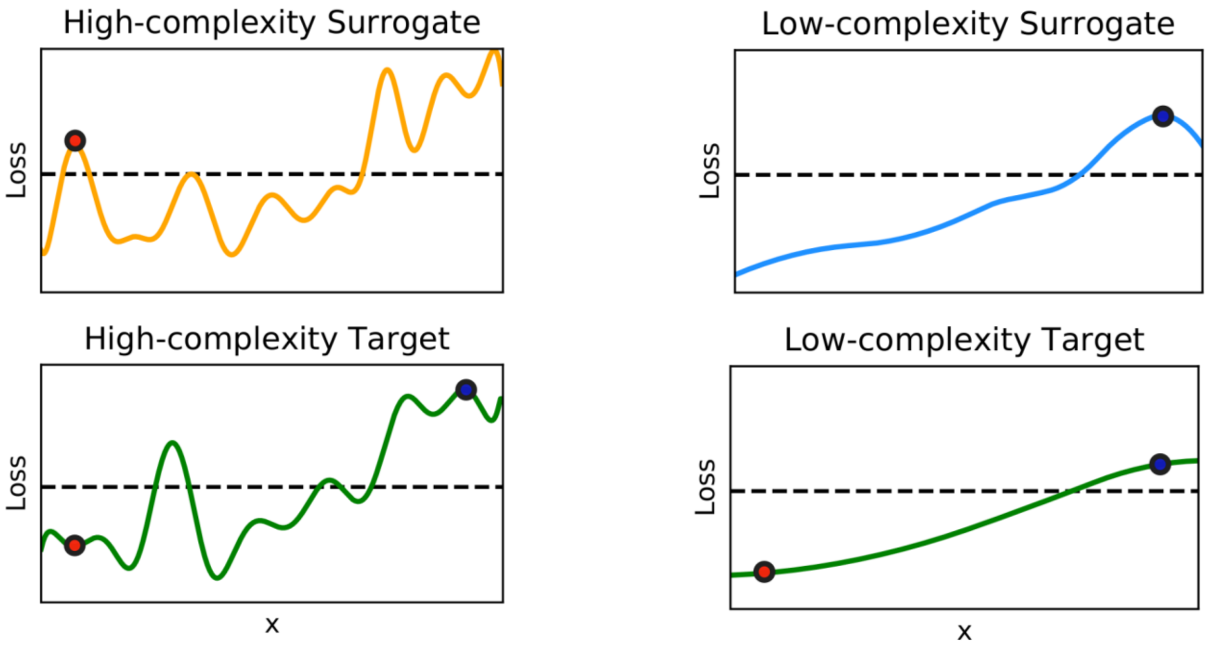
[1] Biggio et al., *Evasion attacks against ML at test time*, ECML PKDD 2013
[2] Papernot et al., *Practical Black-Box Attacks against Machine Learning*, ASIACCS 2017

Contributions

- Optimization framework for evasion and poisoning attacks
- Transferability definition and theoretical bound
  - Metric 1: Size of the input gradient
  - Metric 2: Gradient alignment
  - Metric 3: Variability of the loss landscape
- Comprehensive experimental evaluation of transferability
- Study the relationship between transferability and model complexity

Why complexity may influence transferability?

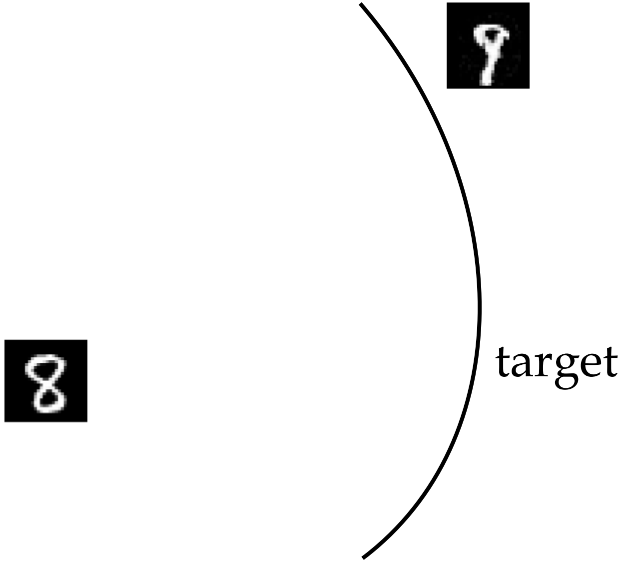
Model complexity: The capacity of the classifier to fit the training data
(can be controlled through regularization)



Our definition for transferability

Loss attained by the target on an adversarial point $\mathbf{x}^\star = \mathbf{x} + \hat{\delta}$ crafted against the surrogate

$$T = \ell(y, \boldsymbol{x} + \hat{\boldsymbol{\delta}}, \boldsymbol{w}) \cong \ell(y, \boldsymbol{x}, \boldsymbol{w}) + \underbrace{\hat{\boldsymbol{\delta}}^T \nabla_{\boldsymbol{x}} \ell(y, \boldsymbol{x}, \boldsymbol{w})}_{\Delta \ell_{bb}}$$

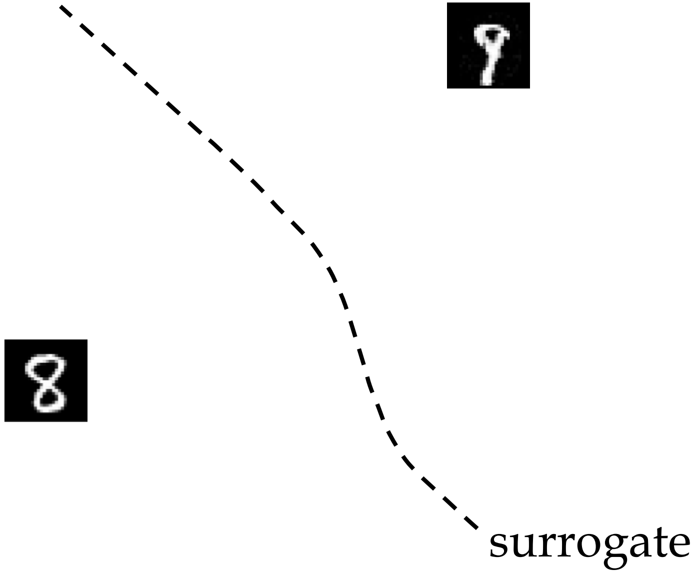


$\mathbf{w}$ target model
$\hat{\mathbf{w}}$ surrogate model

Our definition for transferability

Loss attained by the target on an adversarial point $\mathbf{x}^\star = \mathbf{x} + \hat{\delta}$ crafted against the surrogate

$$T = \ell(y, \boldsymbol{x} + \hat{\boldsymbol{\delta}}, \boldsymbol{w}) \cong \ell(y, \boldsymbol{x}, \boldsymbol{w}) + \underbrace{\hat{\boldsymbol{\delta}}^T \nabla_{\boldsymbol{x}} \ell(y, \boldsymbol{x}, \boldsymbol{w})}_{\Delta\ell_{bb}}$$



Gradient-based optimization:

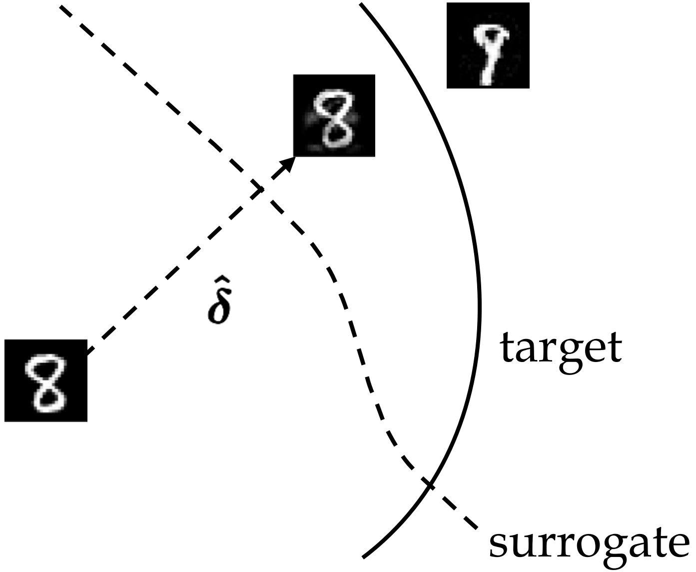
- Evasion:
  [Biggio et al. 13],
  [Szegedy et al. 14], [Goodfellow et al. 14],
  [Carlini and Wagner 17], [Madry et al. 18]
- Poisoning:
  [Biggio et al. 12, Suciu et al. 18]

Our definition for transferability

Loss attained by the target on an adversarial point $\mathbf{x}^\star = \mathbf{x} + \hat{\delta}$ crafted against the surrogate

$$T = \ell(y, \boldsymbol{x} + \hat{\boldsymbol{\delta}}, \boldsymbol{w}) \cong \ell(y, \boldsymbol{x}, \boldsymbol{w}) + \underbrace{\hat{\boldsymbol{\delta}}^T \nabla_{\boldsymbol{x}} \ell(y, \boldsymbol{x}, \boldsymbol{w})}_{\Delta \ell_{bb}}$$

$$\hat{\delta} = \epsilon \frac{\nabla_{\mathbf{x}} \ell(y, \mathbf{x}, \hat{\mathbf{w}})}{||\nabla_{\mathbf{x}} \ell(y, \mathbf{x}, \hat{\mathbf{w}})||_2}$$

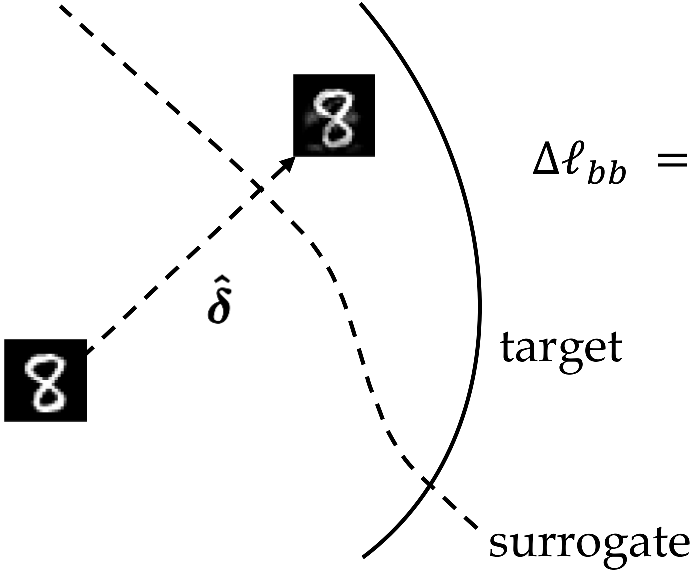


$\mathbf{w}$ target model
$\hat{\mathbf{w}}$ surrogate model

Our definition for transferability

Loss attained by the target on an adversarial point $\mathbf{x}^{\star} = \mathbf{x} + \hat{\delta}$ crafted against the surrogate

$$T = \ell(y, \boldsymbol{x} + \hat{\boldsymbol{\delta}}, \boldsymbol{w}) \cong \ell(y, \boldsymbol{x}, \boldsymbol{w}) + \underbrace{\hat{\boldsymbol{\delta}}^{T} \nabla_{\boldsymbol{x}} \ell(y, \boldsymbol{x}, \boldsymbol{w})}_{\Delta \ell_{bb}}$$



$$\Delta \ell_{bb} = \frac{\Delta \ell_{bb}}{\Delta \ell_{wb}} \Delta \ell_{wb} = \underbrace{\frac{\nabla_{x} \hat{\ell}^{T} \nabla_{x} \ell}{\|\nabla_{x} \hat{\ell}\|_2 \|\nabla_{x} \ell\|_2}}_{\mathbf{R}} \underbrace{\|\nabla_{x} \ell\|_2}_{\mathbf{S}}$$

R: *gradient alignment* measures *black-box to white-box loss increment ratio*

S: *size of input gradients* measures *white-box loss increment*

Poisoning attacks follow a similar derivation

Metric 1: Size of input gradients

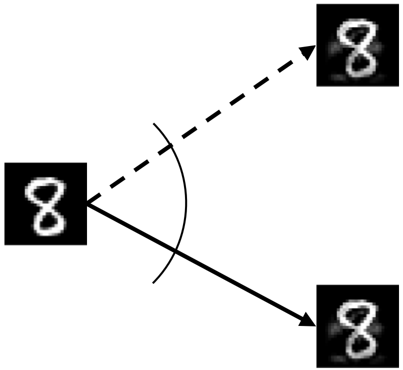
- Evaluates the loss increment $\Delta\ell_{wb}$ incurred by the target classifier under attack
  - **Intuition**: to capture sensitivity of the loss function to input perturbations, as also highlighted in previous work (at least for evasion attacks [1,2,3])

$$S(\mathbf{x}, y) = ||\nabla_x \ell||_2$$

1. C. Lyu et al., *A unified gradient regularization family for adversarial examples*, ICDM 2015
2. A. S. Ross and F. Doshi-Velez, *Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients*, AAAI 2018
3. C. J. Simon-Gabriel et al., *Adversarial vulnerability of neural networks increases with input dimension*, arXiv 2018

Metric 2: Gradient alignment

- Evaluates the ratio $\frac{\Delta \ell_{bb}}{\Delta \ell_{wb}}$ between the loss increment incurred in the black-box case and that incurred in the white-box case



Black-box attack against the surrogate model

White-box attack against the target model

Gradient alignment

$$R(\mathbf{x}, y) = \frac{\nabla_x \hat{\ell}^T \nabla_x \ell}{||\nabla_x \hat{\ell}||_2 ||\nabla_x \ell||_2}$$

Metric 3: Variability of the surrogate loss landscape

- This metric evaluates the variability of the surrogate classifier under training data resampling

$$V(\mathbf{x}, y) = \mathbb{E}_{\mathcal{D}}\{\ell(y, \mathbf{x}, \hat{\mathbf{w}})^2\} - \mathbb{E}_{\mathcal{D}}\{\ell(y, \mathbf{x}, \hat{\mathbf{w}})\}^2$$

Experimental setup

Datasets:

- Evasion: Drebin (Android Malware Detection)
- Poisoning: LFW (Face Verification task 1 vs 5)
- Evasion & Poisoning: MNIST89

Classifiers (8 surrogates, 12 target models):

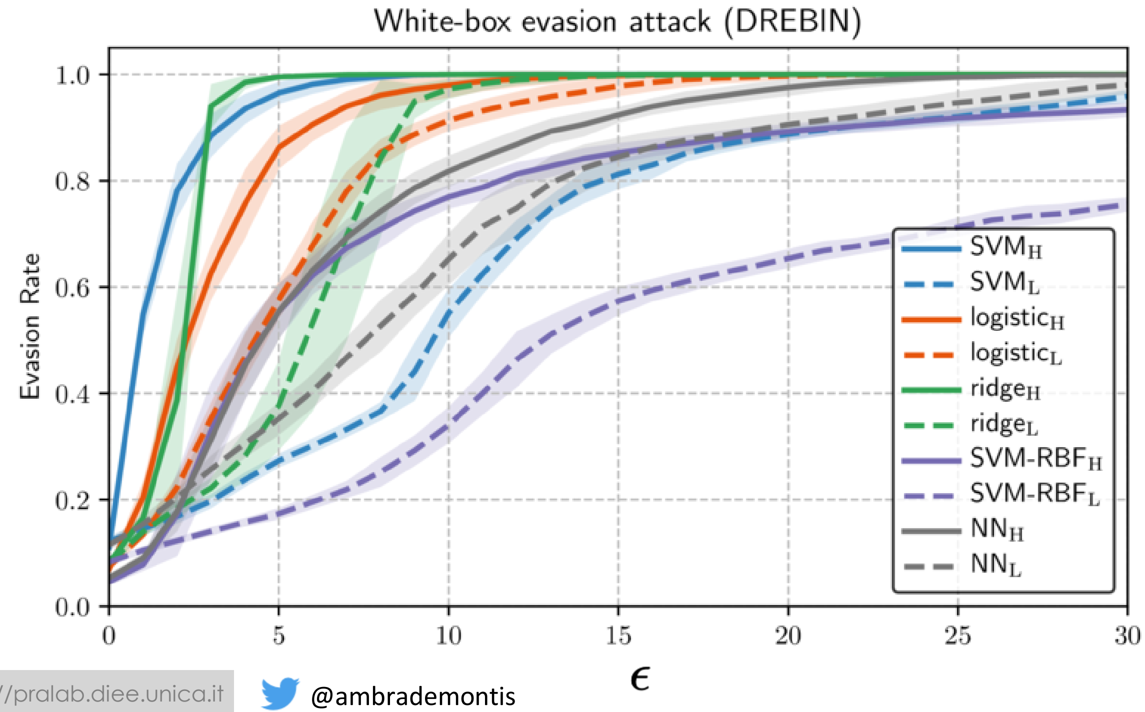
ridge, logistic regression, linear/RBF SVM, neural networks, random forests

Experiments:

- White-box security evaluation
- Black-box security evaluation (all combinations of targets and surrogates)
- Correlation between the proposed metrics, transferability and model complexity
- Statistical tests

Transferability of evasion attacks

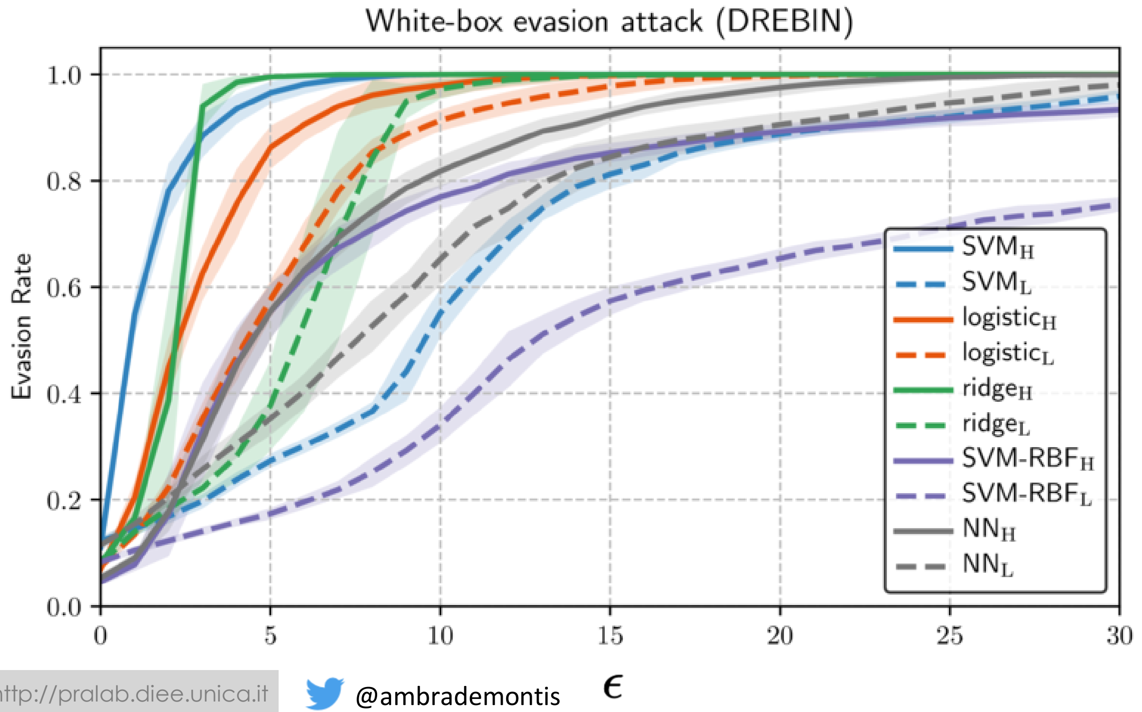
- **RQ1:** Are target classifiers with larger input gradients more vulnerable?
  - How does **model complexity** affect the size of input gradients?



Complexity	Gradient Size
SVM-RBF $_H$	0.16
SVM-RBF $_L$	0.08

Transferability of evasion attacks

- **RQ1:** Are target classifiers with larger input gradients more vulnerable?
  - How does **model complexity** affect the size of input gradients?



- Higher complexity models have larger gradients
- Target with larger gradients are more vulnerable

Transferability of evasion attacks

- **RQ2**: Is the **gradient alignment** correlated with the difference of the perturbations computed considering the target and the surrogate models?

gradient alignment

	SVM_H	SVM_L	$logistic_H$	$logistic_L$	$ridge_H$	$ridge_L$	$SVM\text{-}RBF_H$	$SVM\text{-}RBF_L$	NN_H	NN_L
SVM_H	0.22	0.17	0.32	0.24	0.03	0.10	0.23	0.17	0.32	0.17
SVM_L	0.13	0.64	0.19	0.36	0.03	0.12	0.32	0.55	0.47	0.51
$logistic_H$	0.43	0.24	0.53	0.44	0.16	0.25	0.31	0.24	0.42	0.24
$logistic_L$	0.36	0.59	0.54	0.77	0.18	0.35	0.42	0.47	0.50	0.56
$ridge_H$	0.04	0.07	0.09	0.12	0.53	0.33	0.07	0.04	0.11	0.05
$ridge_L$	0.44	0.15	0.46	0.43	0.51	0.61	0.25	0.13	0.31	0.20
$SVM\text{-}RBF_H$	0.27	0.44	0.40	0.56	0.08	0.20	0.50	0.44	0.44	0.44
$SVM\text{-}RBF_L$	0.29	0.78	0.37	0.68	0.04	0.16	0.41	0.82	0.50	0.78
NN_H	0.27	0.57	0.38	0.47	0.05	0.14	0.39	0.56	0.52	0.48
NN_L	0.26	0.76	0.34	0.63	0.03	0.11	0.33	0.73	0.46	0.80

perturbation correlation

	SVM_H	SVM_L	$logistic_H$	$logistic_L$	$ridge_H$	$ridge_L$	$SVM\text{-}RBF_H$	$SVM\text{-}RBF_L$	NN_H	NN_L
	.14	.11	.17	.19	.07	.12	.15	.10	.17	.11
	.12	.53	.17	.29	.01	.05	.29	.49	.29	.42
	.16	.17	.23	.27	.09	.14	.20	.14	.24	.17
	.17	.31	.28	.39	.05	.12	.27	.29	.34	.34
	.06	.01	.07	.05	.25	.19	.03	.00	.02	.00
	.13	.04	.15	.11	.18	.36	.09	.01	.06	.02
	.14	.27	.18	.25	.03	.07	.29	.29	.27	.25
	.09	.50	.15	.28	.00	.03	.30	.58	.29	.46
	.16	.28	.22	.30	.03	.06	.26	.29	.35	.34
	.11	.44	.18	.31	.00	.02	.28	.46	.34	.53

The gradient alignment metric is heavily correlated with the correlation between the perturbations

Does model complexity impact poisoning?

SVM_L SVM_H SVM-RBF_L SVM-RBF_H

- The findings are similar to evasion for input gradient and variability of loss landscape

- Differences from evasion:
  - For poisoning the best surrogates are the ones with similar level of model complexity

Summary

- Transferability definition and metrics to investigate connections between *attack transferability* and *complexity* of target and surrogate models

- Extensive experiments on 3 datasets and 12 classifiers have shown that:
  - High-complexity models are more vulnerable to both evasion and poisoning attacks
  - Low-complexity models are better surrogates to perform evasion attacks
  - The complexity of the best surrogate is the same as the one of the target for availability poisoning

- **Open-source code available within the Python library SecML:**
  - Code: https://gitlab.com/secml/secml
  - Docs: https://secml.gitlab.io