Towards Quality Assurance of SPLs with Adversarial Configurations

Paul TEMPLE [1] Mathieu ACHER [2] Gilles PERROUIN [1]
Battista BIGGIO [3] Jean-Marc JEZEQUEL [2] Fabio ROLI [3]

[1]PReCISE/Nadi/Université de Namur

[2]IRISA/Université de Rennes 1

[3]PRALab/University of Cagliari

April 2020

this work is funded by the EOS VeriLearn project

Modern software

Software is eating the world

Andreessen. Why software is eating the world? The Wall Street Journal 2011

Capability of being customized

Software Variability by Svahnberg *et al.*

The ability of a software system or artefact to be **efficiently extended, changed, customized or configured** for use in a **particular context**.

Svahnberg *et al.*, A taxonomy of variability realization techniques: Research Articles, Softw. Pract. Exper. 2005

Capability of being customized

Software Variability by Svahnberg *et al.*

The ability of a software system or artefact to be **efficiently extended, changed, customized or configured** for use in a **particular context**.

JHipster: **50** options

Linux Kernel: **15,000** options

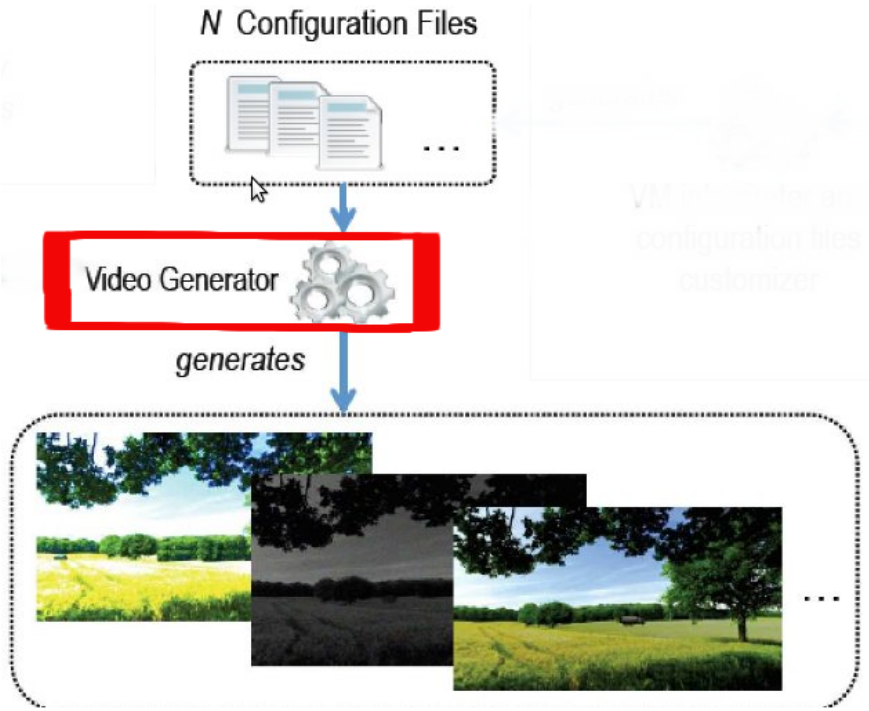
$$2^{15,000} \approx 10^{3,250} >> 10^{1,000} >> \text{estimated } \# \text{ of particules}$$

Svahnberg *et al.*, A taxonomy of variability realization techniques: Research Articles, Softw. Pract. Exper. 2005

Generate video sequences

Video generator

- config files
- Lua scripts
- video sequences



Generate video sequences

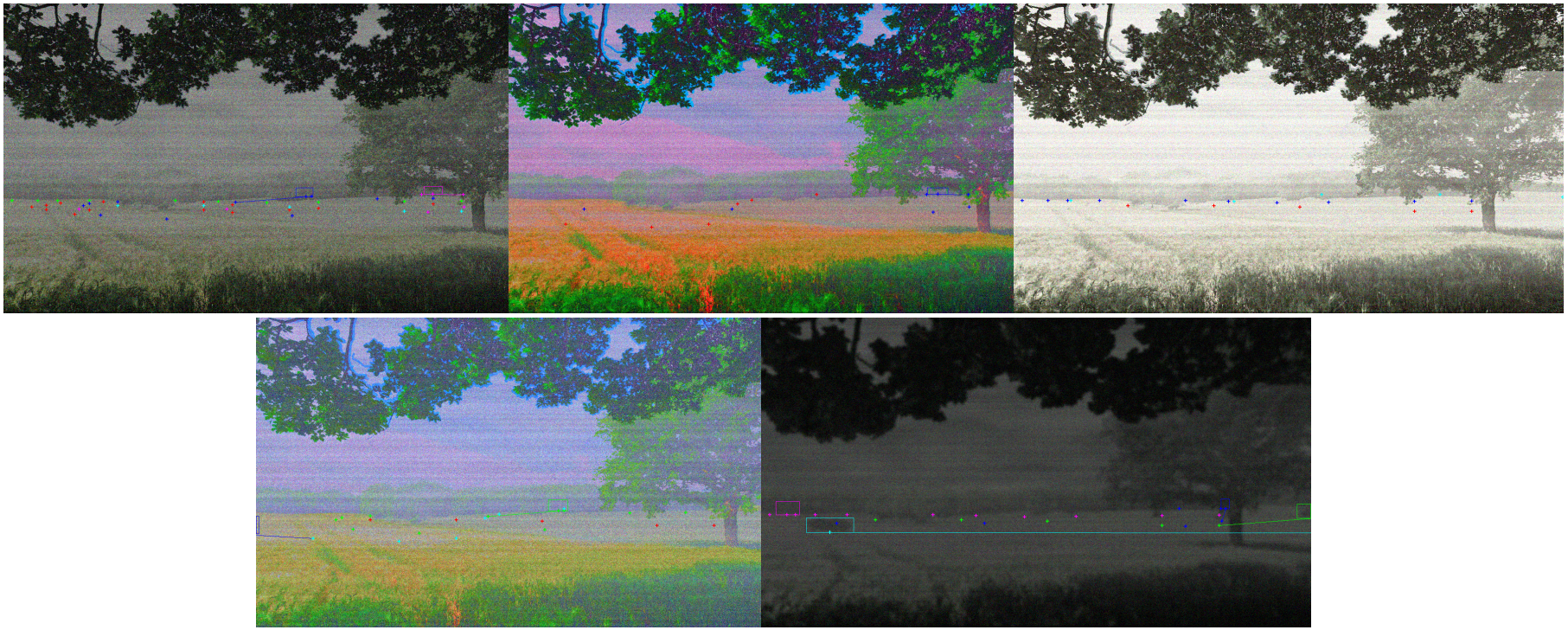
MOTIV video generator

- 100+ configuration options $\rightarrow$ Boolean, Integers, Float-values $\Rightarrow$ 10^{314} possible configurations
- $\approx$ 30 min per file

Generate video sequences

MOTIV video generator

- 100+ configuration options $\rightarrow$ Boolean, Integers, Float-values $\Rightarrow$ 10^{314} possible configurations
- $\approx$ 30 min per file



Combinatorial explosion

Validity VS acceptable

- Not all configurations can generate a video: what if I put an index 13 in a list composed of 10 items? $\Rightarrow$ not valid
- Some valid configurations do not meet some expectations: at night/during the day? $\Rightarrow$ not acceptable

how to check that **all valid** configurations/sequences are **acceptable**?

Alves Pereira *et al.*, *Learning Software Configuration Spaces: A Systematic Literature Review*

Combinatorial explosion

Validity VS acceptable

- Not all configurations can generate a video: what if I put an index 13 in a list composed of 10 items? $\Rightarrow$ not valid
- Some valid configurations do not meet some expectations: at night/during the day? $\Rightarrow$ not acceptable

how to check that **all valid** configurations/sequences are **acceptable**?

You need ML

- Reason on the configuration and not on final products
- Cover the feature space
- Get an idea of product performances
- Approximate performance distribution

Alves Pereira *et al.*, *Learning Software Configuration Spaces: A Systematic Literature Review*

Sample, Measure, Learn

Sampling configurations

- Detected faults depends on the sampling strategy (Medeiros *et al.* ; Sarkar *et al.*)
- Choosing the right sampling strategy is an open problem (Medeiros *et al.*)
- Some configurations may not be valid (Cohen *et al.*, Henard *et al.*, Lamancha *et al.*)

Medeiros *et al.*, A comparison of 10 sampling algorithms for configurable systems, ICSE'16

Sarkar *et al.*, Cost-efficient sampling for performance prediction of configurable systems, ASE'15

Cohen *et al.*, Constructing Interaction Test Suites for Highly Configurable Systems in the Presence of Constraints: A Greedy approach, IEEE TSE'08

Henard *et al.*, Bypassing the combinatorial explosion: Using similarity to generate and prioritize t-wise test configurations for SPL, IEEE TSE'14

Lamancha *et al.*, Testing product generation in SPLs using pairwise for features coverage, ICTSS'10

Sample, Measure, Learn

Predicting performances

- Previously executed configurations are kept into a database (Sincero *et al.*)
- Create a performance-influence model using Machine Learning (Guo *et al.*, Siegmund *et al.*)

Sincero *et al.*, Approaching non-functional properties of SPLs: Learning from products, APSEC,2010

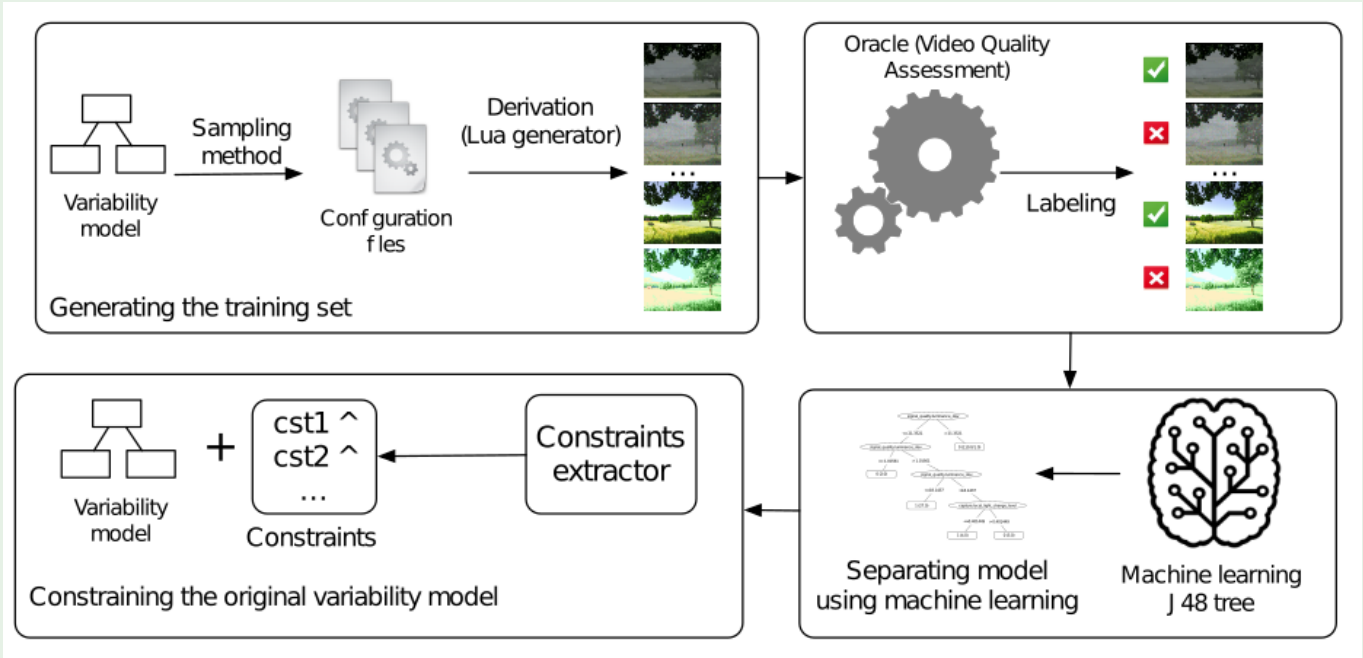
Siegmund *et al.*,Performance-influence models for highly configurable systems, FSE, 2015

Guo *et al.*, Variability-aware performance prediction: A statistical learning approach, ASE, 2013

Siemgund *et al.*, Scalable prediction of non-functional properties in SPLs: Footprint and memory consumption, Info. and Softw. Technol., 2013

Sample, Measure, Learn

Using ML to Improve SPLs



Temple *et al.*, *Using Machine Learning to Infer Constraints for Product Lines*, SPLC'16

What about testing the classifier?

Performance assessment

- Common practice
- 10-fold cross-validation $\rightarrow$ validate
- test set $\rightarrow$ evaluate performances on unknown set

What about testing the classifier?

Performance assessment

- Common practice
- 10-fold cross-validation $\rightarrow$ validate
- test set $\rightarrow$ evaluate performances on unknown set

But...

- Data gathered at the same time
- From the same data distribution
- Assumption: stationary data distribution

Adversarial Machine Learning

Quick history of spams

- "brute" spams → clear text, words that you do not want to see
- "obfuscated" spams → change some letters; try to find synonyms
- "highly obfuscated" spams → even more changes! Message is not clear, mix different stuff together, hard to understand
- "change media" spams → from text to image

⇒ **every time**, anti-spam filters adapted
(same with viruses, system intrusions, *etc.*)

Adversarial Machine Learning

Stationarity assumption

- Does not hold
- game between attacker and defender

Biggio and Roli, *Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning*, Pattern Recognition Vol. 84 2018

Adversarial Machine Learning

Stationarity assumption

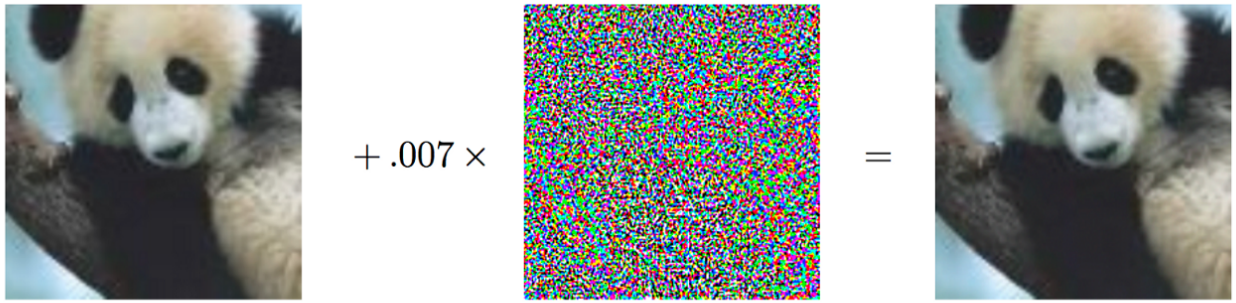
- Does not hold
- game between attacker and defender

Adversarial Machine Learning

- Appeared in mid 2000's
- Warned about the excessive use of ML
- Better understand ML algorithms and their assumptions

Biggio and Roli, *Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning*, Pattern Recognition Vol. 84 2018

Adversarial Machine Learning



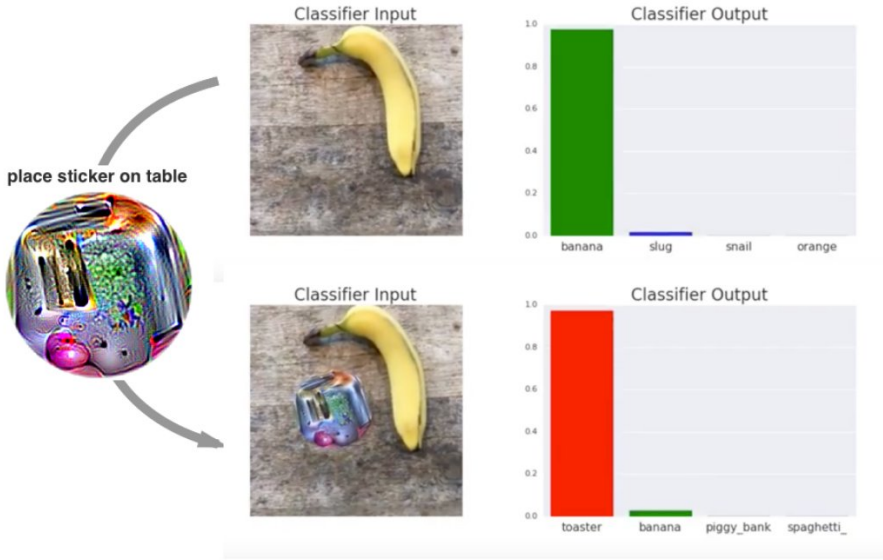
"panda"

noise

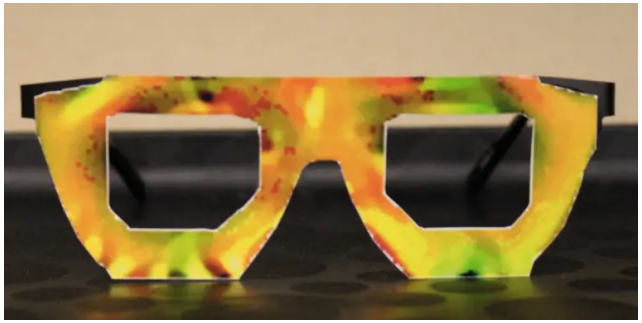
"gibbon"

57.7% confidence

99.3% confidence



Adversarial Machine Learning



CMU adversarial glasses

1.1 original

1.2 with added rain

result from the DeepTest paper

Adversarial Machine Learning

Different techniques

- GAN
- Poisoning attacks
- Evasion attacks

Goodfellow *et al.*, *Generative Adversarial Nets*, NIPS 2014

Biggio *et al.*, *Poisoning attacks against support vector machines*, ICML 2012

Biggio *et al.*, *Evasion attacks against machine learning at test time*, ECML/PKDD 2013

Biggio and Roli, *Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning*, Pattern Recognition Vol. 84 2018

Adversarial Machine Learning

Different techniques

- GAN
- Poisoning attacks
- Evasion attacks

Used everywhere

- Spam filtering, intrusion detection systems
- Deep Fake, Fake news
- Testing ML-based systems

Goodfellow *et al.*, *Generative Adversarial Nets*, NIPS 2014

Biggio *et al.*, *Poisoning attacks against support vector machines*, ICML 2012

Biggio *et al.*, *Evasion attacks against machine learning at test time*, ECML/PKDD 2013

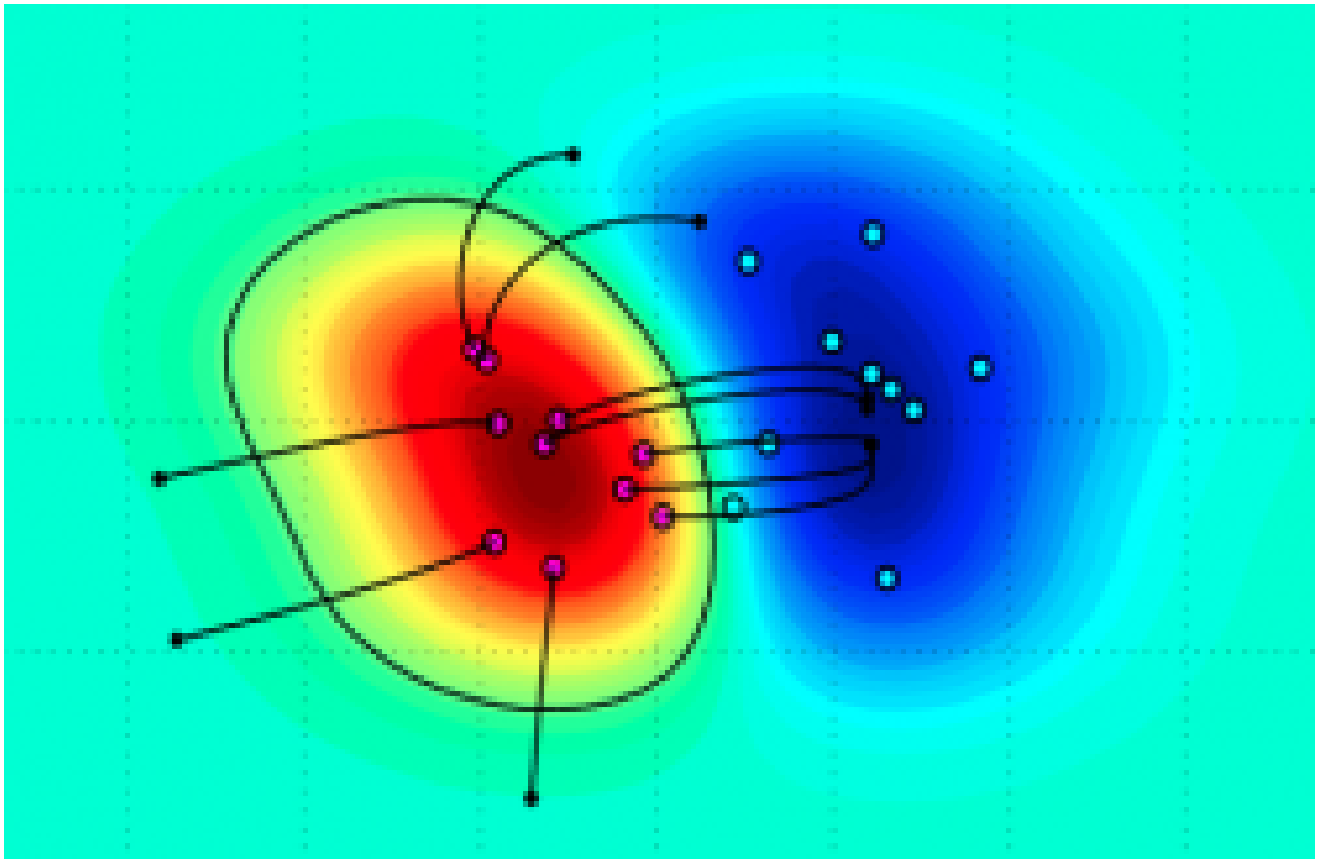
Biggio and Roli, *Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning*, Pattern Recognition Vol. 84 2018

Follow the same process

- Probe the target classifier
- Train your own surrogate classifier
- Select a point x for which the class y_c is known
- Compute $\nabla \mathrm{F}(x^i)$ as a unit vector: $\nabla \mathrm{g}(x^i)$ - λ $\nabla \mathrm{p}(xi \mid y_c^i = -y_c)$
- $x^{i+1} = x^i$ - $t \times \nabla \mathrm{F}(x^i)$
- Repeat computations until stopping criteria is met

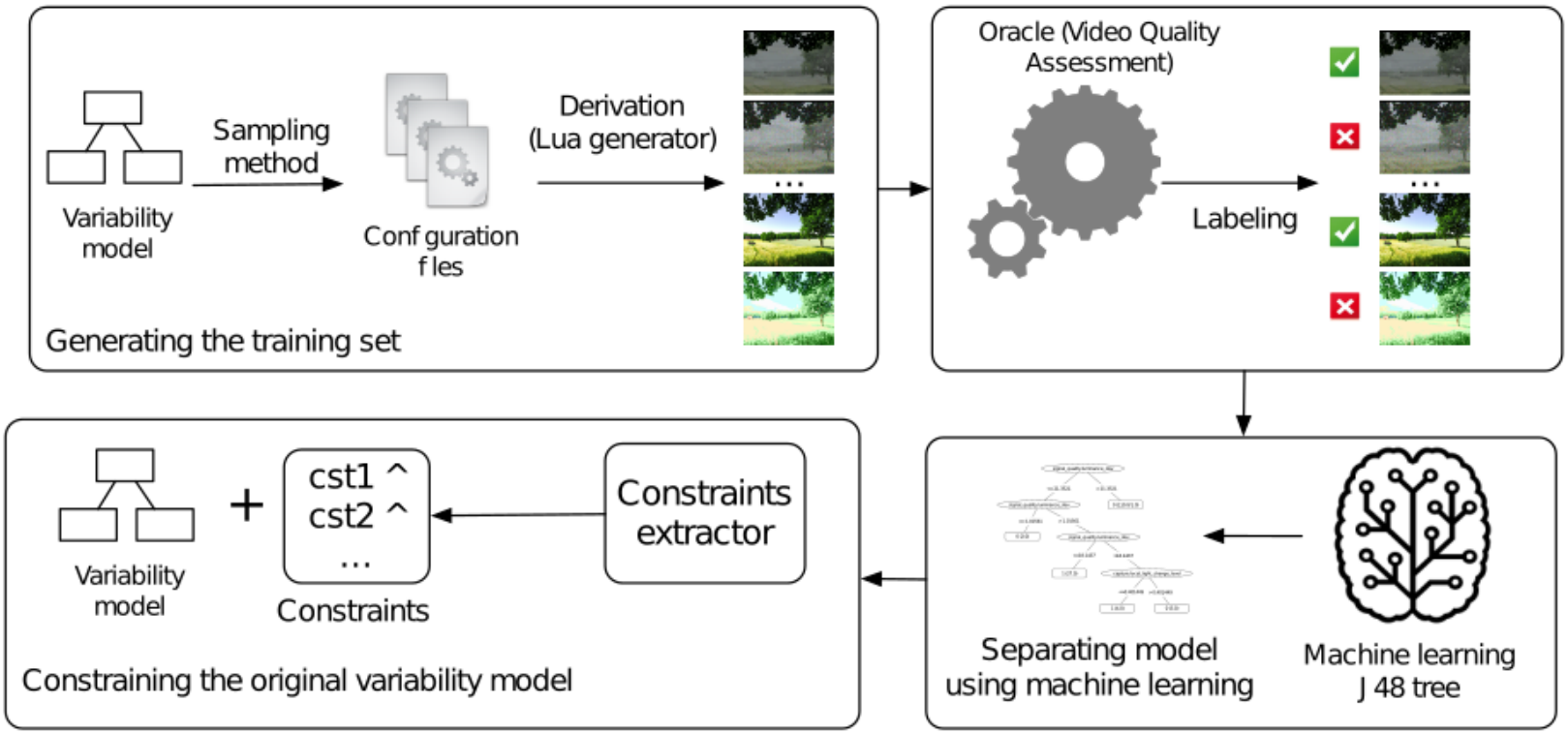
Biggio *et al.*, *Evasion attacks against machine learning at test time*, ECML/PKDD 2013

Follow the same process



Biggio *et al.*, *Evasion attacks against machine learning at test time*, ECML/PKDD 2013

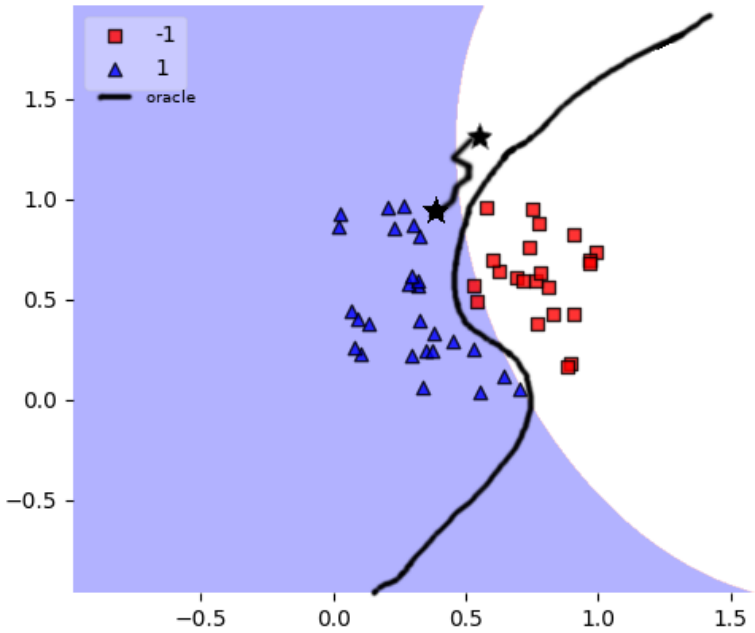
Can use advML to improve our classifier?



Can use advML to improve our classifier?

Can we **automatically** find new configurations that:

- are "far" from other known configurations
- fool the classifier



Evaluation

Our contribution: AdvML & SPL

- Adapation from Biggio et al.
- Configurable attacks $\rightarrow$ # displacement; step size
- Can take some constraints into account

Biggio *et al.*, *Evasion attacks against machine learning at test time*, ECML PKDD 2013

Evaluation

Our contribution: AdvML & SPL

- Adapation from Biggio et al.
- Configurable attacks $\rightarrow$ # displacement; step size
- Can take some constraints into account

Data set

- 4500 config. available; $\approx$ 10% are not acceptable
- 500 $\rightarrow$ training set
- 4000 $\rightarrow$ test set

Biggio *et al.*, *Evasion attacks against machine learning at test time*, ECML PKDD 2013

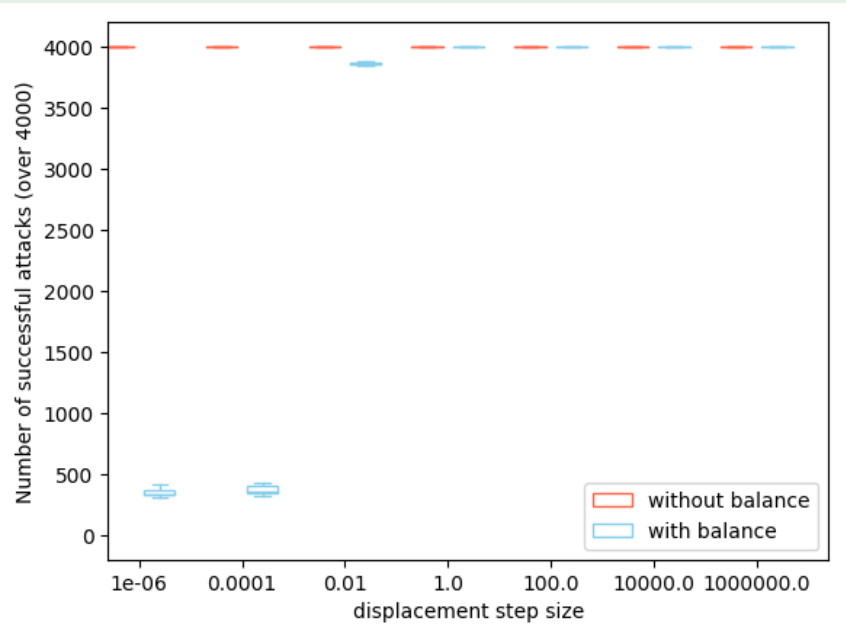
Research Questions:

- **RQ1**: Can we generate wrongly classified adversarial config. but valid w.r.t MOTIV's VM?
- **RQ2**: Comparison with random modif. on existing config?
- **RQ3**: What happens if we retrain the classifier?

Research Questions:

- **RQ1**: Can we generate wrongly classified adversarial config. but valid w.r.t MOTIV's VM?
- **RQ2**: Comparison with random modif. on existing config?
- **RQ3**: What happens if we retrain the classifier?

RQ1

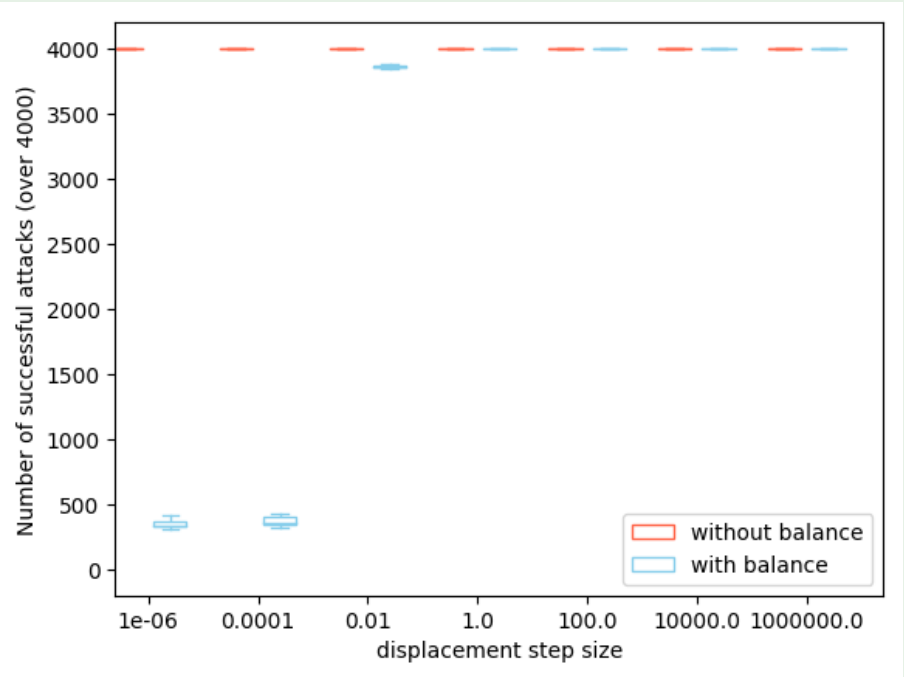


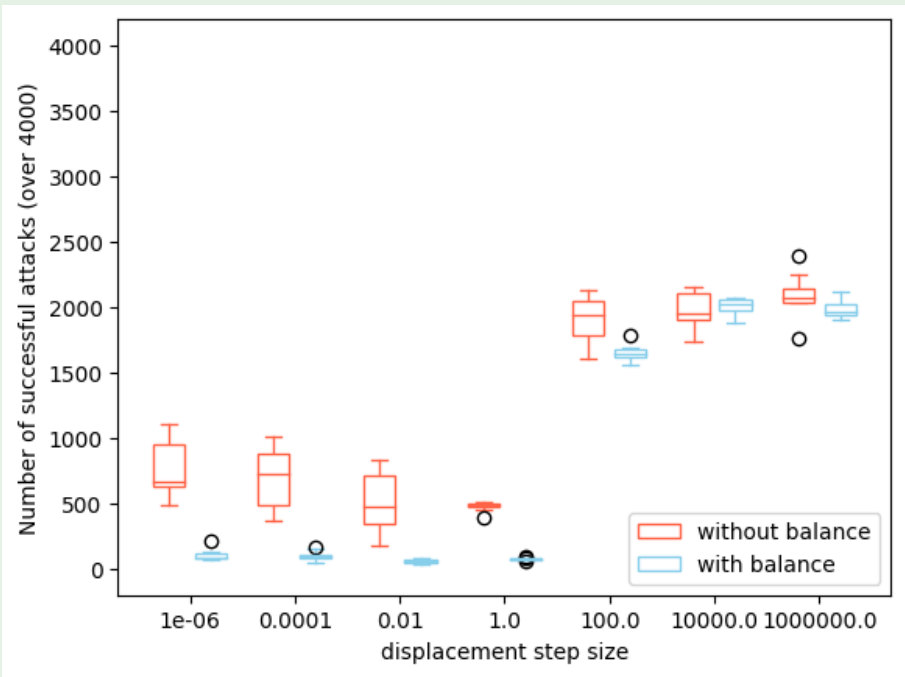
- Preprocess made all features homogeneous (between 0 and 1)
- Checks for Boolean features
- **All adv. config. are valid**

Research Questions:

- **RQ1**: Can we generate wrongly classified adversarial config. but valid w.r.t MOTIV's VM?
- **RQ2**: Comparison with random modif. on existing config?
- **RQ3**: What happens if we retrain the classifier?

RQ2





Research Questions:

- **RQ1**: Can we generate wrongly classified adversarial config. but valid w.r.t MOTIV's VM?
- **RQ2**: Comparison with random modif. on existing config?
- **RQ3**: What happens if we retrain the classifier?

RQ3

- Added 25 adv. config. in the training set and classif. perf. assessment on test set
- Accuracy can increase up to 3% (from 92% to 95%)

On an other software?

JHipster

- Generate web app with complete technological stack
- 58 config. options $\rightarrow \approx$ 90k config.
- Build in about 10 min.

On an other software?

JHipster

- Generate web app with complete technological stack
- 58 config. options $\rightarrow \approx$ 90k config.
- Build in about 10 min.

Results

Results for all 3 RQs are similar

Conclusion

Synthesis

- Adapt an advML attack to the context of SPL
- Find new configurations of a video generator
- Adv config can be valid and misclassified
- Using advML is more efficient than random modif
- With a right configuration of the attack, classifer performances can increase

Conclusion

Synthesis

- Adapt an advML attack to the context of SPL
- Find new configurations of a video generator
- Adv config can be valid and misclassified
- Using advML is more efficient than random modif
- With a right configuration of the attack, classifer performances can increase

Future Work

- In depth understanding on how to use these techniques
- How to integrate cross-tree constraints?
- Adversarial ML as a new sampling technique?