Audition CRCN CNRS 2022
Efficient Exploration of Colossal Configurable Spaces

Paul TEMPLE

March 2022

Equipes : Spirals (Lille) ; ProGresS (Bordeaux) ; NaoMod (Nantes)

Software variability & system complexity

Linux Kernel: **15,000** options

Software variability & system complexity

Linux Kernel: **15,000** options

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

Sébastien Mosser @petitroll · 25 févr.

"the number of atoms in the visible universe is 10^80. There are 2^15000 different versions of the Linux kernel. So astrophysicists works with things way simpler than software engineers". @jmjezequel

Software variability & performances

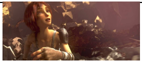
no-mbtree = F
qblur = 0.90
nr = 200

processing time = 5 min

processing time = 2 h

processing time = 10 h

Evaluating performance is complex

	Program Variants				
Inputs		264	264	...	264
		12	1	...	5
		1	348	...	10
	...				
		50	101	...	260

Link to PhD manuscript and slides for defense

Do we need to measure?

Assumptions

- Exploring all configurations is **impossible**
- Measuring performances is **costly**

Siegmund *et al.*, Perf. Prediction with feature interaction, ICSE'12
Siegmund *et al.*, Perf.-Influence models for config. systems, FSE'15
Guo *et al.*, Var.-aware perf. prediction, ASE'13

Do we need to measure?

Assumptions

- Exploring all configurations is **impossible**
- Measuring performances is **costly**

Assign a measure without measuring

- Similar configurations produce similar performances
- Performance prediction

		Program Variants			
		264	264	...	264
Inputs		12	1	...	5
		1	348	...	10
			...		
		50	101	...	260

Siegmund *et al.*, Perf. Prediction with feature interaction, ICSE'12
Siegmund *et al.*, Perf.-Influence models for config. systems, FSE'15
Guo *et al.*, Var.-aware perf. prediction, ASE'13

Do we need to measure?

Assumptions

- Exploring all configurations is **impossible**
- Measuring performances is **costly**

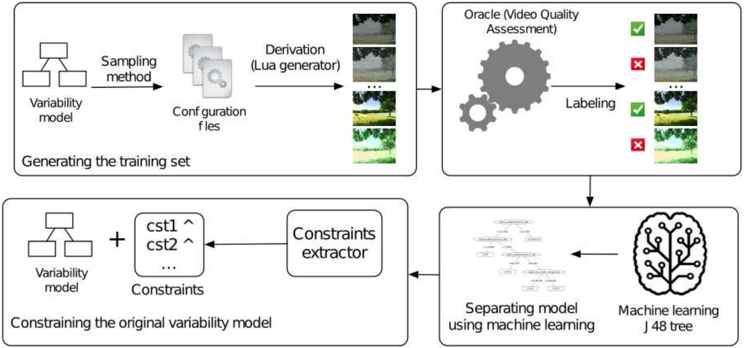
Scope the configuration space

- **Explicit** requirements
- **Few** configurations are **acceptable**

		Program Variants			
		264	264	...	264
Inputs		12	1	...	5
		1	348	...	10
	...				
		50	101	...	260

Temple *et al.*, Using machine learning to infer constraints for product lines, SPLC'16

Reducing configuration space with ML



Temple *et al.*, Using machine learning to infer constraints for product lines, SPLC'16

Adversarial configurations

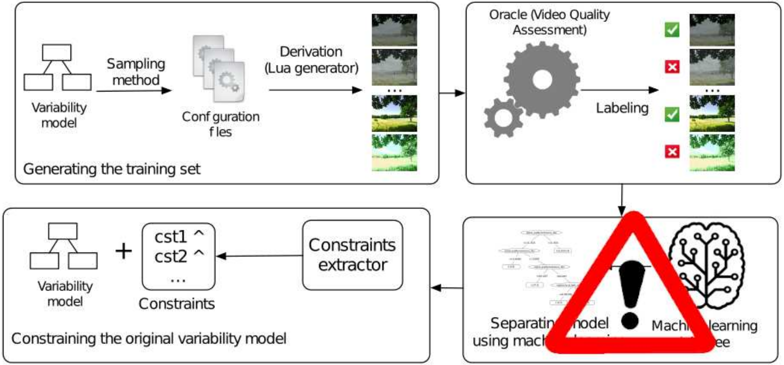
Temple *et al.*, Adv. Configs for config. systems, EMSE'21
Link to PRALab website

Adversarial configurations

- 1^{st} **use** of adversarial ML techniques for configurable systems
- Collaboration between:
  - Battista Biggio and Fabio Roli, PRALab, Sardinia
  - Mathieu Acher and Jean-Marc Jézéquel, IRISA, Rennes
  - Gilles Perrouin and myself, Namur
- Published at SPLC'19 → extended in EMSE'21

Temple *et al.*, Adv. Configs for config. systems, EMSE'21
Link to PRALab website

Reducing configuration space with ML



Machine Learning is based on statistics → errors

- Over-constraining: lack of flexibility
- Under-constraining: waste of resources

Improving the pipeline

Robustifying the model

- Show new configurations
- Configurations with **high risk** of misclassification

Goodfellow *et al.*, Adversarial examples, ICLR'15

Elsayed *et al.*, Fool both humans and computers, NeurIPS'18

Improving the pipeline

Robustifying the model

- Show new configurations
- Configurations with **high risk** of misclassification

Goodfellow *et al.*, Adversarial examples, ICLR'15
Elsayed *et al.*, Fool both humans and computers, NeurIPS'18

Robustifying the model

Evasion attacks

- Iterative modifications of features
- Move towards the separation

no-mbtree = F
qblur = 0.90
nr = 200

predict.: acceptable

Biggio *et al.*, Evasion attacks against SVMs, ECML'13

Evasion attacks

- Iterative modifications of features
- Move towards the separation

no-mbtree = F
qblur = 0.90
nr = 200

predict.: acceptable

no-mbtree = F
qblur = 0.74
nr = 220

predict.: acceptable

Biggio *et al.*, Evasion attacks against SVMs, ECML'13

Robustifying the model

Evasion attacks

- Iterative modifications of features
- Move towards the separation

no-mbtree = F
qblur = 0.90
nr = 200

predict.: acceptable

no-mbtree = F
qblur = 0.74
nr = 220

predict.: acceptable

no-mbtree = F
qblur = 0.65
nr = 260

predict.: non-acceptable

Biggio *et al.*, Evasion attacks against SVMs, ECML'13

Robustifying the model

Evasion attacks

- Iterative modifications of features
- Move towards the separation

no-mbtree = F
qblur = 0.90
nr = 200

predict.: acceptable

no-mbtree = F
qblur = 0.74
nr = 220

predict.: acceptable

no-mbtree = F
qblur = 0.65
nr = 260

predict.: non-acceptable

Adversarial Configurations

- Adversarial retraining
- Analyze them to understand what is wrong

Biggio *et al.*, Evasion attacks against SVMs, ECML'13

Research project

- ML for configurable systems
  - Adversarial configurations for improvement
  - Find a more efficient representation for configurations
- ML models exploration with variability management

Adversarial configurations for improvement

Adversarial configurations

- Used to improve ML model fairness

Delobelle *et al.*, Ethical Adversaries, SIGKDD Exploration NewsLetters 2021

Adversarial configurations for improvement

Adversarial configurations

- Used to improve ML model fairness

Support for constraints

- Constraints on values of options and their combinations
- Constraints may be complex $\rightarrow$ involve several options
- Constraint checking strategy that scales

Delobelle *et al.*, Ethical Adversaries, SIGKDD Exploration NewsLetters 2021

Find an efficient representation for configurations

What is wrong?

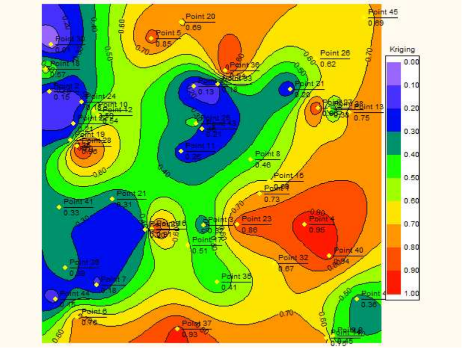
- Similar configurations → similar performances
- Options as a feature vector → interactions?

Iscen *et al.*, Metric Learning, CVPR'19

Find an efficient representation for configurations

What is wrong?

- Similar configurations → similar performances
- Options as a feature vector → interactions?



Iscen *et al.*, Metric Learning, CVPR'19

ML models design with variability management tools

Modern ML models

- 100 epochs ImageNet to train AlexaNet in 24*minutes* for **only 1.2M dollars**

⇒ Impossible if you are not GAFAM

You *et al.*, ImageNet trained in 24 Minutes, ICPP'18

ML models design with variability management tools

Modern ML models

- 100 epochs ImageNet to train AlexaNet in 24*minutes* for **only 1.2M dollars**

$\Rightarrow$ Impossible if you are not GAFAM

Explicitly managing variability to reduce costs

- Many models exist
- Neural Architecture Search $\rightarrow$ subparts can be combined

$\rightarrow$ PhD Antoine Gratia: manage the exploration of DL design space

---

You *et al.*, ImageNet trained in 24 Minutes, ICPP'18

ML models design with variability management tools

Modern ML models

- 100 epochs ImageNet to train AlexaNet in 24*minutes* for **only 1.2M dollars**

$\Rightarrow$ Impossible if you are not GAFAM

Explicitly managing variability to reduce costs

- Many models exist
- Neural Architecture Search $\rightarrow$ subparts can be combined

$\rightarrow$ PhD Antoine Gratia: manage the exploration of DL design space

Goal of variability management

- Green computing
- Reduce complexity of models $\rightarrow$ explainability

You *et al.*, ImageNet trained in 24 Minutes, ICPP'18

Integration

- Spirals, Lille
- ProGresS, Bordeaux
- NaoMod, Nantes

Teams

- Software Engineering teams with lack of ML background
- Centered around variability

Teams

- Software Engineering teams with lack of ML background
- Centered around variability

Spirals

- Run-time adaptation // Performance
- Different aspects of SE

ProGresS

- Evolution
- Towards collaborations with the BKB team

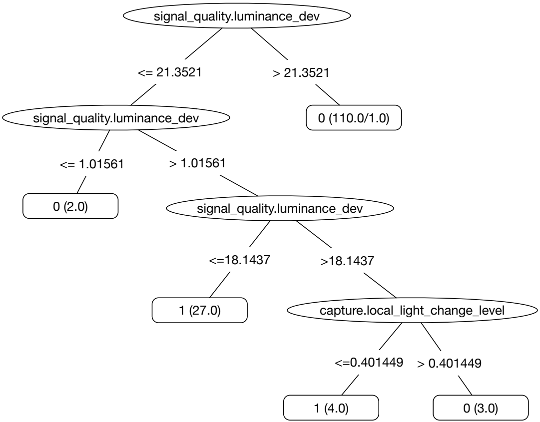
NaoMod

- Architecture
- Projects in collaboration with companies

Efficient Exploration of Colossal Configurable Spaces

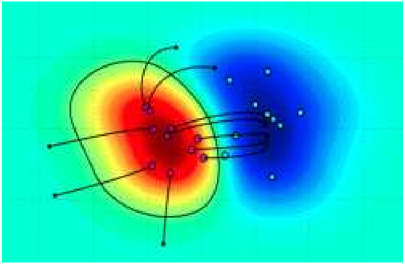
- Software variability; Machine learning; Performance
- Testing performances of configurable systems is difficult
- Adversarial configurations
- Research Project:
  - representation problem
  - adversarial for improvement $\rightarrow$ fairness; adversarial sampling
  - var. management for models $\rightarrow$ green computing, reducing complexity
- Possible Teams:
  - Spirals, Lille
  - ProGresS, Bordeaux
  - NaoMod, Nantes

Infering constraints with ML

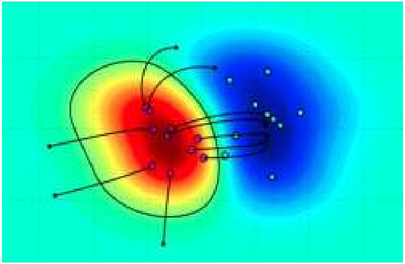



```
!(signal_quality.blur_level > 0 &&
 signal_quality.static_noise_level <=0.135519)
```

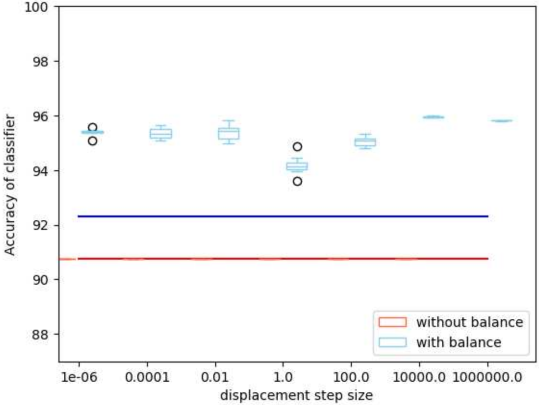
Adversarial configurations



Adversarial configurations

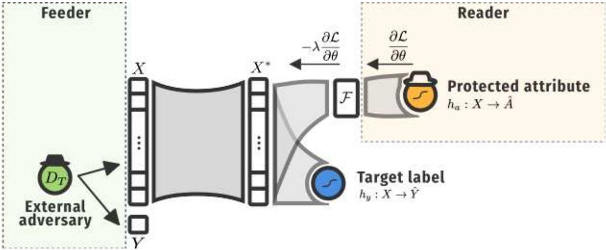


Adversarial retraining

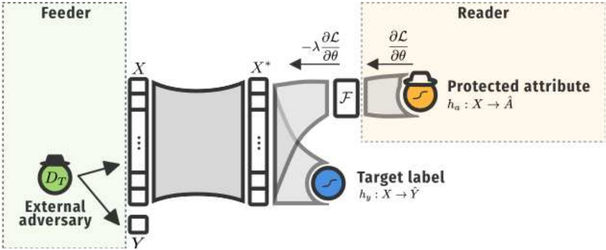


25 adversarial configurations added to the training set (video variants)
Initial accuracy: 90.766% (red); 92.315% (blue)

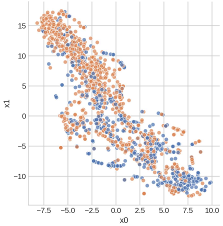
Ethical adversaries

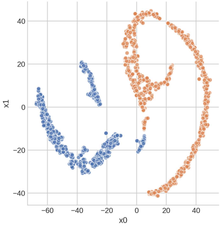


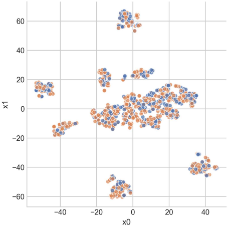
Ethical adversaries: architecture



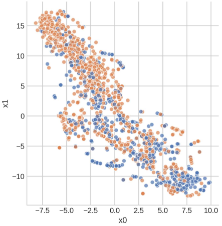
Ethical adversaries: results

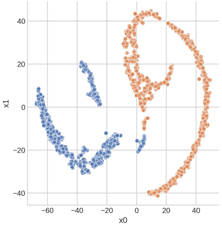


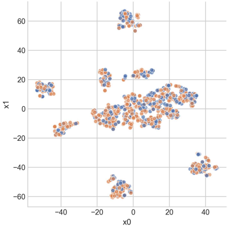


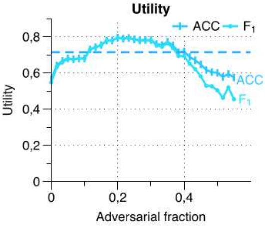


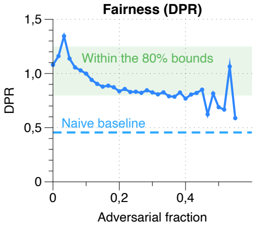
Ethical adversaries: results

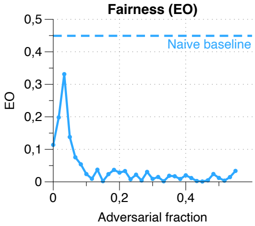




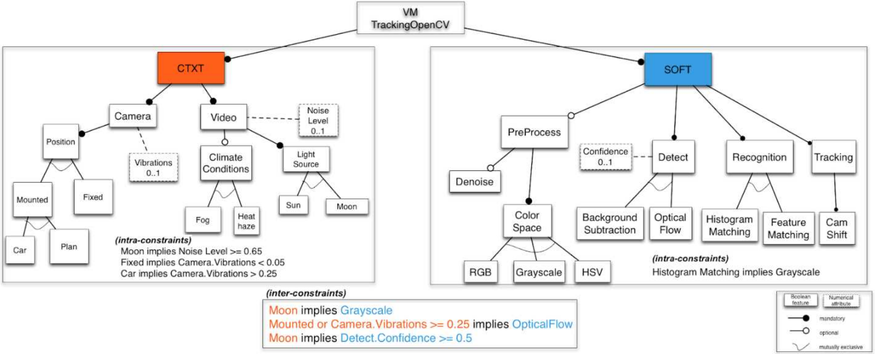




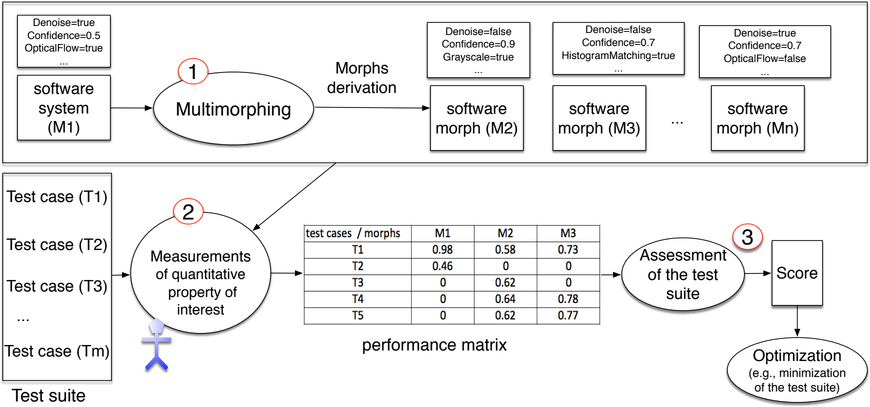




Learning Contextual Variability



Multimorphic Testing: process



test cases / morphs	M1	M2	M3
T1	0.98	0.58	0.73
T2	0.46	0	0
T3	0	0.62	0
T4	0	0.64	0.78
T5	0	0.62	0.77

performance matrix

Multimorphic Testing: definition of score

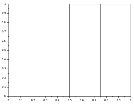
Properties

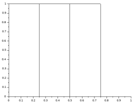
- P1: Be positive
- P2: Given 2 test suites A and B, A $\subseteq$ B, score(A) $\leq$ score(B)
- P3: $\forall$ test suites A and B, score($A \cup B$) $\geq$ max(score(A),score(B))

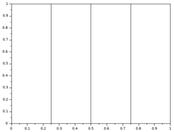
Multimorphic Testing: definition of score

Properties

- P1: Be positive
- P2: Given 2 test suites A and B, A $\subseteq$ B, score(A) $\leq$ score(B)
- P3: $\forall$ test suites A and B, score($A \cup B$) $\geq$ max(score(A),score(B))

T1: 
$= \frac{2}{4}$

T2: 
$= \frac{3}{4}$

$T1 \cup T2$: 
$= \frac{4}{4}$

Multimorphic Testing: evaluation

Case	App. Domain	# morphs	# test suites
OpenCV	Tracking in videos	252	49
COCO	Obj. rec. in images	52	12
Haxe	Code generation	21	84

Multimorphic Testing: evaluation

Case	App. Domain	# morphs	# test suites
OpenCV	Tracking in videos	252	49
COCO	Obj. rec. in images	52	12
Haxe	Code generation	21	84

COCO

- 12 categories → 40k images
- Can we keep a similar ranking with a smaller test suite
- 5 categories → few permutations (Spearman correl: 0.998)

Haxe

- 84 test suites
- 1 bug (wrong data structure)
- With 5 test suites, the bug is found