

# Machine Learning and Configurable Systems: A Gentle Introduction

## (tutorial at SPLC'19)

Hugo Martin, Juliana Alves Pereira,  
Paul Temple, Mathieu Acher

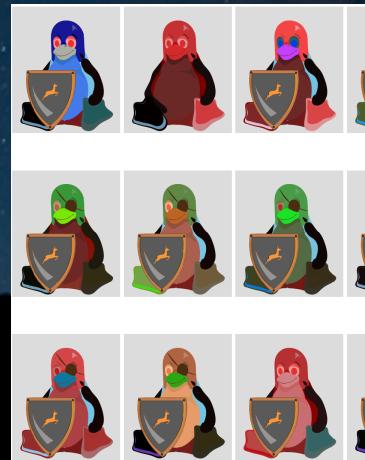
<https://github.com/VaryVary/>







# How to master configuration space? (with machines and humans)



# Process



## ① Variability annotations and modeling

```

{{#if ACK}}
{{#if BOLD_ACK}}\textbf{Acknowledgements.}{/{if}}
{{#if PARAGRAPH_ACK}}\paragraph{Acknowledgements}{/{if}} We thank anonymous re
{{#if LONG_ACK}} We thank Pierre Laperdrix for the newspaper example. {/{if}}
% project fundings also
{{/{if}}}
%
\scriptsize
% \vspace{-2mm}
\vspace{-({vspace\_bib})mm} \caption{\label{fig:generator}Video generator: modularity and variants}
\bibliographystyle{abbrv}
\bibliography{Modularity15}

```

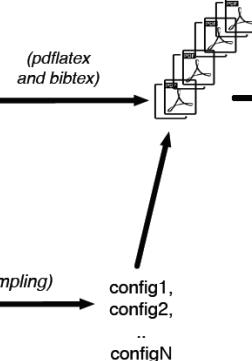
**LaTeX source files**

```

// Boolean options (features)
fmLaTeX = FM (VARY_LATEX : BREF BIB [PL_FOOTNOTE] [ACK] JS_STYLE
[LONG_AFFILIATION];
JS_STYLE : (JS_SCRIPTSIZE I JS_TINY I JS_FOOTNOTESIZE); // mutually exclusive
ACK : [LONG_ACK] (BOLD_ACK I PARAGRAPH_ACK); // LONG_ACK is optional
LONG_AFFILIATION : [EMAIL];
// numerical options (attributes)
real BIB.vspace_bib: [1.0..5.0] precision 1 // 1 decimal digit precision
real BREF.bref_size: [0.7..1.0] precision 1 // either 0.7 0.8 0.9 or 1.0
real cserver_size: [0.6..0.9] precision 1 // either 0.6 0.7 0.8 or 0.9
// specific constraints can be added a priori if needs be
...

```

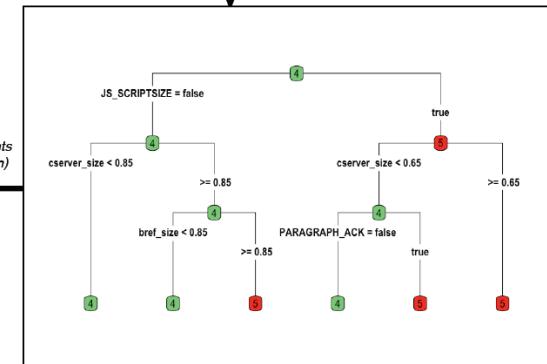
**variability model**



## ② Paper variants building and measurements

JS_SCRIPTSIZE	JS_STYLE	JS_TINY	LONG_ACK	LONG_AFFILIATION	PARAGRAPH_ACK	PL_FOOTNOTE	VARY_LATEX	bref_size	cserver_size	vspace_bib	nbPages
false	true	true	true	false	false	false	true	0.7	0.9	4.0	4 ✓
false	true	true	false	false	false	false	true	0.8	0.6	2.2	4 ✓
false	true	true	false	false	false	false	true	0.9	0.6	2.3	4 ✓
false	true	true	true	true	false	true	true	0.7	0.8	1.1	4 ✓
true	true	false	false	true	true	false	true	0.8	0.9	1.8	5 ✗
true	true	false	false	true	false	false	true	0.7	0.8	2.8	5 ✗
true	true	false	false	false	false	false	true	0.7	0.8	2.9	5 ✗
false	true	true	false	true	false	false	true	0.9	0.7	4.9	4 ✓
true	true	false	true	true	false	true	true	1.0	0.7	1.7	5 ✗
true	true	false	false	false	false	true	true	1.0	0.6	1.8	5 ✗
true	true	false	false	true	false	true	true	0.7	0.6	2.8	4 ✓

## ③ Machine Learning (Classification problem)



```

// same original variability model
fmLaTeX = FM (VARY_LATEX ... )
// ...
real cserver_size: [0.6..0.9] precision 1
// constraints (^ is AND, ! is NOT, => is IMPLIES)
// we negate the paths leading to class "5" (non-acceptable)
// !(JS_SCRIPTSIZE ^ cserver_size >= 0.65) or more readable:
(JS_SCRIPTSIZE => cserver_size < 0.65) ^
// !(JS_SCRIPTSIZE ^ cserver_size < 0.65 ^ PARAGRAPH_ACK)
// equivalent to
(JS_SCRIPTSIZE => (cserver_size < 0.65 => !PARAGRAPH_ACK)) ^
!(JS_SCRIPTSIZE ^ cserver_size >= 0.9 ^ bref_size >= 0.9)

```

**variability model + constraints**



## ④ Ready-to-configure paper

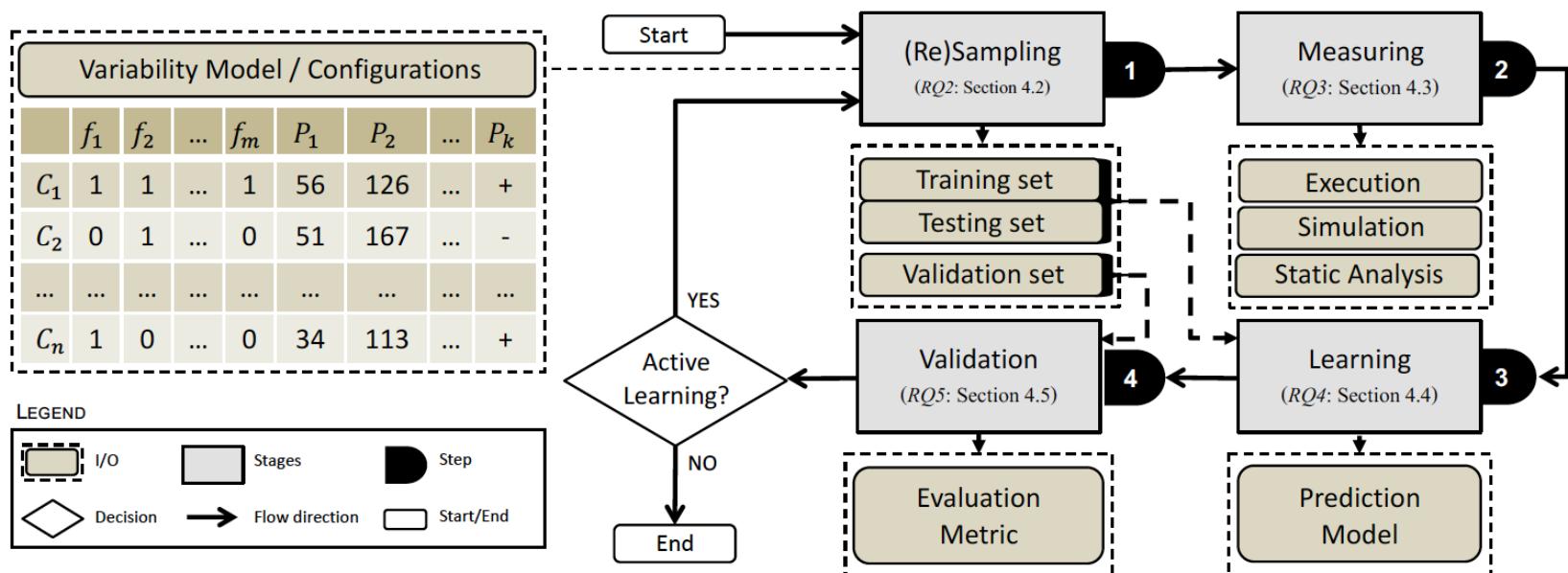
<input checked="" type="checkbox"/> VARY_LATEX
<input checked="" type="checkbox"/> ACK
<input type="checkbox"/> PARAGRAPH_ACK
<input checked="" type="checkbox"/> BOLD_ACK
<input type="checkbox"/> LONG_ACK
<input checked="" type="checkbox"/> JS_STYLE
<input type="checkbox"/> JS_FOOTNOTESIZE
<input type="checkbox"/> JS_TINY
<input checked="" type="checkbox"/> JS_SCRIPTSIZE
<input type="checkbox"/> PL_FOOTNOTE
<input type="checkbox"/> LONG_AFFILIATION
<input type="checkbox"/> EMAIL
<input checked="" type="checkbox"/> BIB
<input checked="" type="checkbox"/> BREF

```

// same original variability model
fmLaTeX = FM (VARY_LATEX ... )
// ...
real cserver_size: [0.6..0.9] precision 1
// constraints (^ is AND, ! is NOT, => is IMPLIES)
// we negate the paths leading to class "5" (non-acceptable)
// !(JS_SCRIPTSIZE ^ cserver_size >= 0.65) or more readable:
(JS_SCRIPTSIZE => cserver_size < 0.65) ^
// !(JS_SCRIPTSIZE ^ cserver_size < 0.65 ^ PARAGRAPH_ACK)
// equivalent to
(JS_SCRIPTSIZE => (cserver_size < 0.65 => !PARAGRAPH_ACK)) ^
!(JS_SCRIPTSIZE ^ cserver_size >= 0.9 ^ bref_size >= 0.9)

```

# Sampling, Measuring, Learning



Learning Software Configuration Spaces: A Systematic Literature Review

Juliana Alves Pereira, Hugo Martin, Mathieu Acher, Jean-Marc Jézéquel, Goetz Botterweck, Anthony Ventresque <https://arxiv.org/abs/1906.03018>

# Huge applicability!

Name	Domain	Non-Functional Properties	Name	Domain	Non-Functional Properties
Thingiverse's 3D printer	3D printer	defects	Wget	Data transfer	memory footprint, code complexity
IBM WebSphere Application server	Application server	throughput	Actian Vector	Database system	runtime
Clasp ASP solver	ASP solver	response time	Apache Cassandra	Database system	latency
SNW Asset management	Asset management	area and throughput	Berkeley DB	Database system	I/O time, memory footprint, performance, response time, code complexity, maintainability, binary size
Binpacking Binpacking algorithm	Binpacking algorithm	execution time and accuracy	FAME-DBMS	Database system	maintainability, binary size, performance
XGBoost Boosting algorithms	Boosting algorithms	training time	MySQL	Database system	defects, throughput, latency
SaaS system Cloud computing	Cloud computing	response time	Postgres	Database system	throughput, latency
Clustering Clustering algorithm	Clustering algorithm	execution time and accuracy	Prevayler	Database system	memory footprint, performance
AJStats Code analyzer	Code analyzer	response time	SQLite	Database system	memory footprint, performance, response time, code complexity, runtime
SaC Code analyzer	Code analyzer	I/O time, response time	StockOnline	Database system	response time
POLLY Code optimizer	Code optimizer	runtime	Kafka	Distributed systems	throughput
Libssh Combinatorial model	Combinatorial model	defects	DNN	DNNs algorithms	accuracy of predictions
Telecom Communication system	Communication system	defects	Curriculum vitae	Document	number of pages
LLVM Compiler	Compiler	memory footprint, performance, response time, code complexity, compilation time	Paper	Document	number of pages
Compressor SPL Compression library	Compression library	compression time, memory usage and compression ratio	RUBiS	E-commerce application	response time
7Z Compression library	Compression library	compression time	EMAIL	E-mail client	time
LRZIP Compression library	Compression library	compressed size, compression time, compilation time	MBED TLS	Encryption library	response time
RAR Compression library	Compression library	code complexity	SAP ERP	Enterprise Application	response time
XZ Compression library	Compression library	compression time	noc-CM-log	FPGA	CPU power consumption, runtime
ZipMe Compression library	Compression library	memory footprint, performance, code complexity, time	sort-256	FPGA	area, throughput
WordPress Content management	Content management	CPU power consumption	E-Health System	Health	response time
LinkedList Data structures	Data structures	memory footprint, performance, maintainability, binary size	HIPA <sup>cc</sup>	Image processing	response time
Curl Data transfer	Data transfer	code complexity	Disparity SPL	Image processing	energy consumption
			PKJab	Instant messenger	memory footprint, performance
			IBM ILOG CPLEX	Integer solver	runtime
			SPECjbb2005	Java Server	response time, throughput
			WEKA	Learning algorithm	accuracy of predictions
			SVD	Linear algebra	execution time and accuracy
			Trimesh	Mesh solver	iterations, response time
			MBENCH	Micro benchmark	time
			ACE+TAO system	Middleware software	defects
			SensorNetwork	Network simulator	memory footprint, performance
			Simonstrator	Network simulator	latency
			NoC	Network-based system	energy and runtime
			Helmholtz 3D	Numerical analysis	execution time and accuracy
			Poisson 2D	Numerical analysis	execution time and accuracy
			Linux kernel	Operating system	memory footprint, performance
			DNN	Optimization algorithm	response time

Learning Software Configuration Spaces: A Systematic Literature Review

Juliana Alves Pereira, Hugo Martin, Mathieu Acher, Jean-Marc Jézéquel, Goetz Botterweck,  
Anthony Ventresque <https://arxiv.org/abs/1906.03018>

# RQ6: Limitations and Open Challenges

Introduction

Methodology

Framework  
Classification

Results

Conclusion

- RQ1: Applicability
  - Is there a one size fits all solution for any application scenario?
- RQ2: Sampling
  - Explore sampling techniques used in other domains, such as SPL testing
- RQ3: Measuring
  - How many times do you need to repeat measurements? Can we trust?
- RQ4: Learning
  - Parameter tuning and resampling
- RQ5: Validation
  - User empirical studies



Aren't we there  
yet? To what extent  
learning approaches  
are effective for real-  
world software  
systems?



Synergy between communities?

# We are not there yet!

- End-users fear variability and stick to default configurations that may be sub-optimal.
  - There are not concrete evidence about the use of learning techniques in practice by users of the evaluated systems. **How**

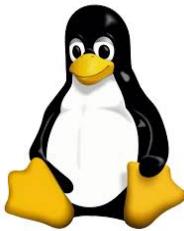
Name	Domain	NFP	References	Name
ATGAT	Video compression	P14	[1-9]	ATGAT
ATGAT	Code analyzer	P14	[17-49]	ATGAT
ATGAT	Protocol	P14	[50-52]	ATGAT

References	Name	Domain	NFP	References	Name	Domain	NFP
[12, 45, 51]	VideoStream	Video processing	P14	[1, 46–49]	PKI	Instant messenger	P10, P11
[12, 22, 45]	AJStats	Code analyzer	P14		Prevayler	Database	P10, P11
[10, 12, 22, 33, 36, 41, 45, 46, 51]	Apache	Web server	P13, P14, P18		RAR	Compression	P19
[33]	Apache Storm	Stream processing	P9, P16		RollingSort	Stream processing	P9, P16
[10, 12, 22, 22, 33, 36, 41, 45–49, 51]	Berkeley DB	Database system	P8, P10, P11		SaaS system	Cloud computing	P14
[12, 22, 45]	Clasp	ASP solver	P14		SaC	Code analyzer	P8, P14
[19]	CoBot System	Robotic system	P4		SensorNetwork	Simulation	P10, P11
[48]	Curl	Data transfer	P19		Sol	Stream processing	P9
[22, 33, 45, 51]	DUNE	Stencil code	P14		sort-256	FPGA	P1, P16
[22, 33, 45, 51]	HSMGP	Stencil code	P14, P16		SPEAR	SAT solver	P14
[45, 51]	JavaGC	Runtime environment	P14	[1, 46–49, 51]	SQLite	Database system	P10, P11, P14, P19
[47–49]	LinkedList	Data structures	P10, P11		StockOnline	Database system	P14
[48?–50]	Linux kernel	Operating system	P10, P11		Trimesh	Mesh solver	P7, P14
[10, 12, 22, 34, 36, 41, 45, 46, 48, 51]	LLVM	Compiler	P10, P11, P14		Violet	UML editor	P10, P11
[12, 22, 33, 45]	LRZIP	Compression library	P2, P14		Wget	Data transfer	P10, P19
[51, 52]	MOTIV	Video encoder	P17	[1, 46–49, 54]	WordCount	Stream processing	P9, P16
[34]	noc-CM-log	FPGA	P3, P15	[32]	WordPress	Content management	P3
[19]	NoSQL	Database system	P9	[10, 12, 18, 22, 33, 34, 36, 41, 45, 46, 48, 51]	x264	Video encoder	P3, P5, P14, P19
[51]	OpenCV	Video tracking	P11	[47–49]	ZipMe	Compression	P10, P11, P19



# Advanced topics

# Learning the Configuration Space of Linux



Enormous configurations space eg Linux has 15K+ options, tri-state values {y, n, m}. A build takes 15 minutes on average on a recent machine

```
[...] KConfig file
config PRINTK
    default y
    bool "Enable support for printk" if EXPERT
    select IRQ_WORK
    help
        This option enables normal printk support. Removing it
        eliminates most of the message strings from the kernel image
        and makes the kernel more or less silent. As this makes it
        very difficult to diagnose system problems, saying N here is
        strongly discouraged.

config PRINTK_NMI
    def_bool y
    depends on PRINTK
    depends on HAVE_NMI

config BUG
    bool "BUG() support" if EXPERT
    default y
    help
        Disabling this option eliminates support for BUG and WARN, reducing
        the size of your kernel image and potentially quietly ignoring
        numerous fatal conditions. You should only consider disabling this
        option for embedded systems with no facilities for reporting errors.
        Just say Y.

config ELF_CORE
    depends on COREDUMP
    default y
    bool "Enable ELF core dumps" if EXPERT
    help
        Enable support for generating core dumps. Disabling saves about 4k.

[...]
config AIO
    bool "Enable AIO support" if EXPERT
    default y
    help
        This option enables POSIX asynchronous I/O which may be used
        by some high performance threaded applications. Disabling
        this option saves about 7k.
```



Configurator



# Input Sensitivity and Transferability of Performance Prediction Models

(ongoing work)

**What if I change the input video?  
Can I reuse my performance prediction model?**

```
x264 --no-mbtree  
--rc-lookahead 40  
--ref 9  
-o trailer_480p24.x264  
trailer_2k_480p24.y4m
```

```
x264 --no-mbtree  
--rc-lookahead 40  
--ref 9  
-o football.x264  
football.y4m
```



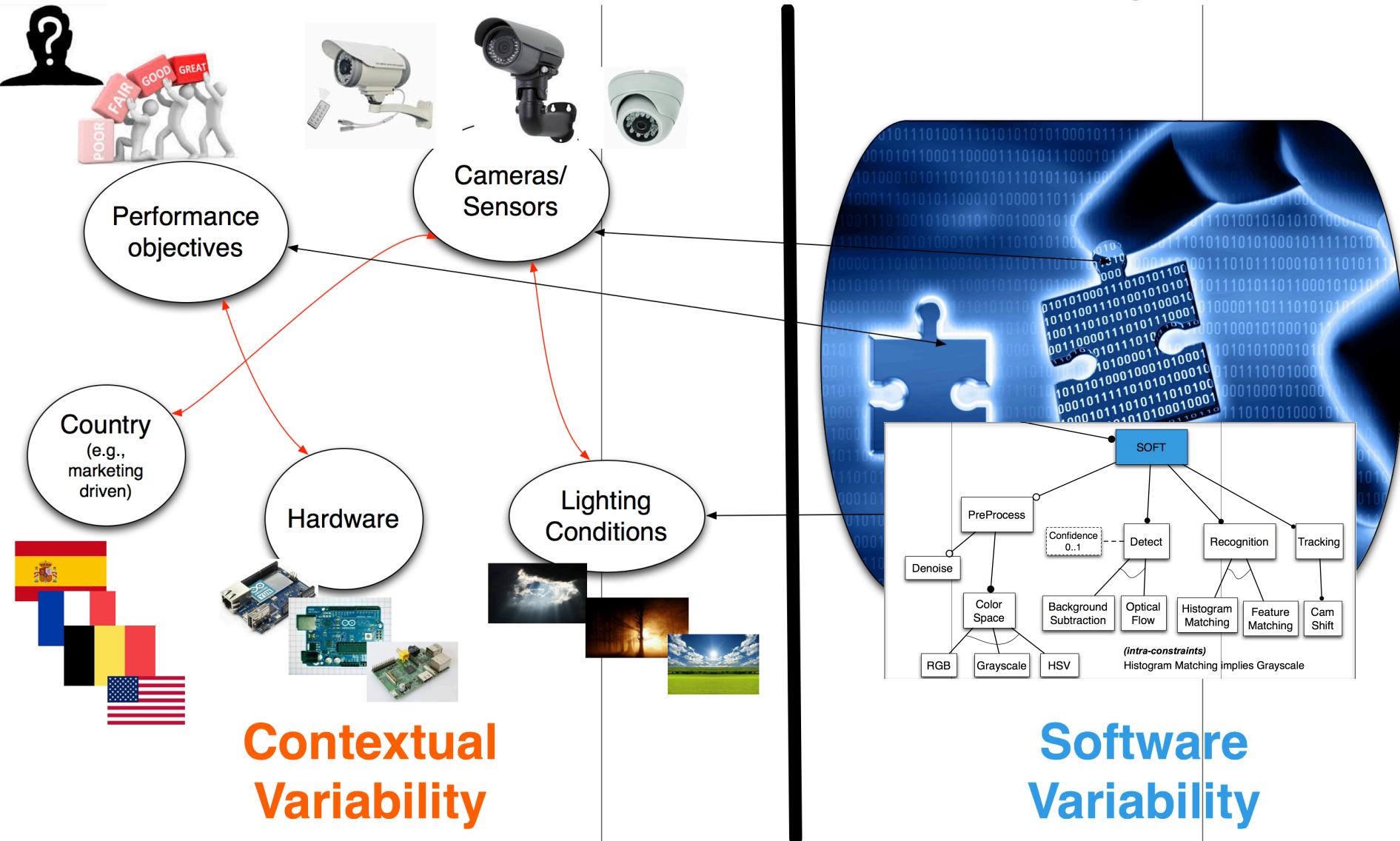
55 seconds

?

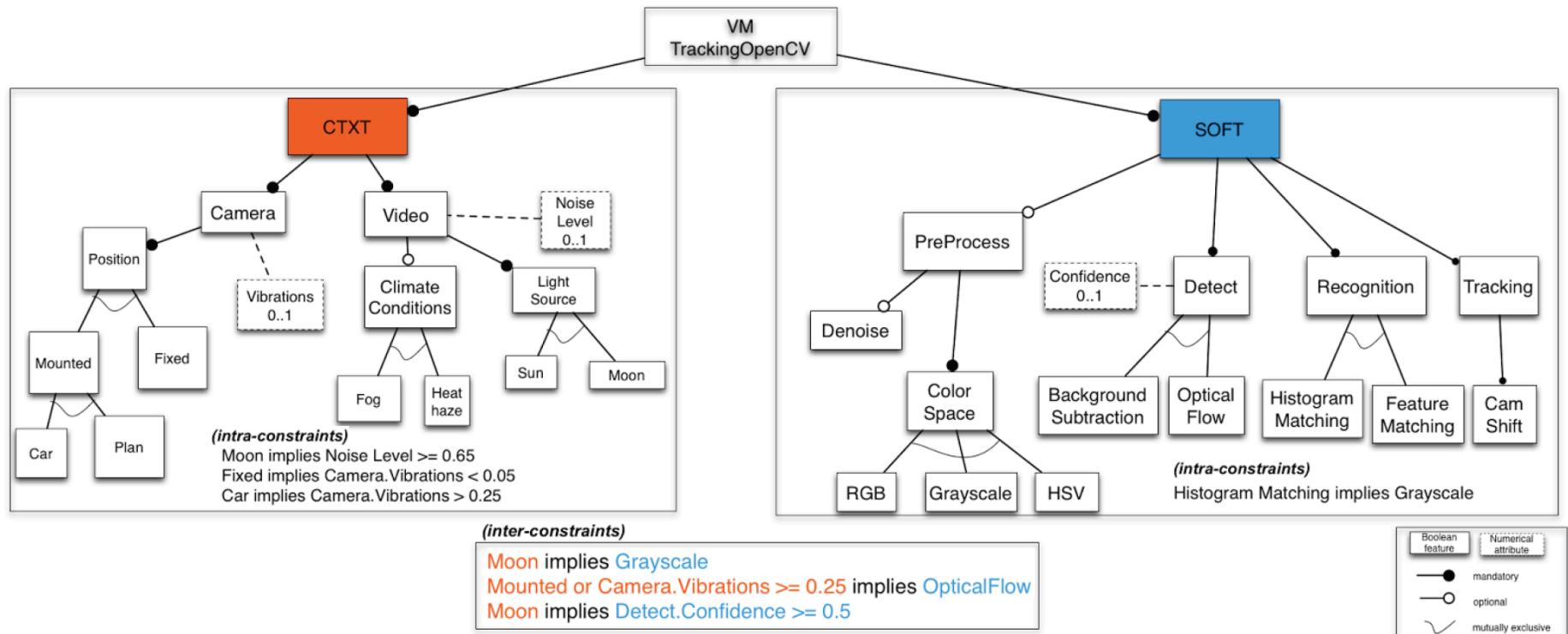


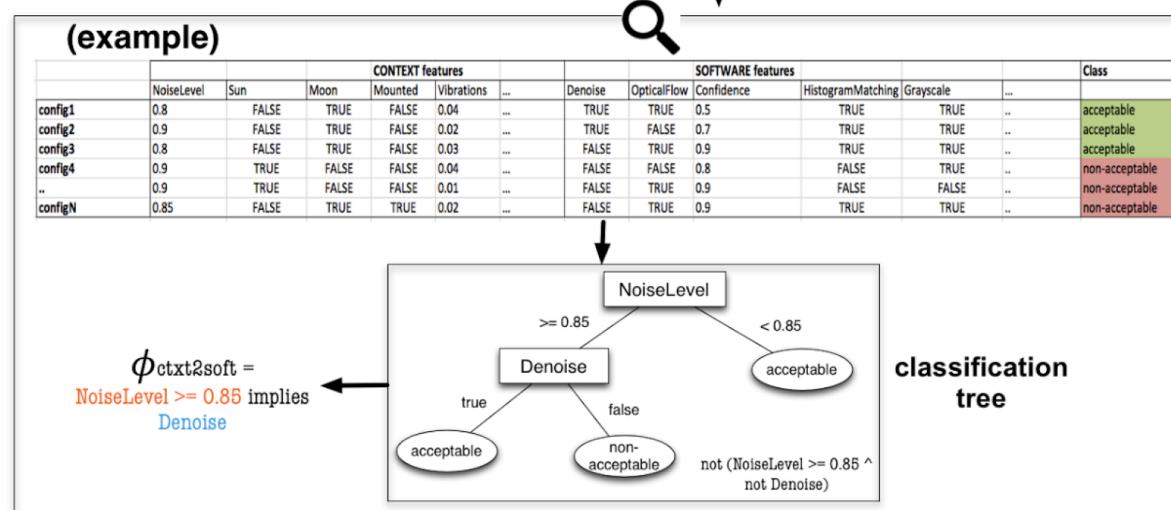
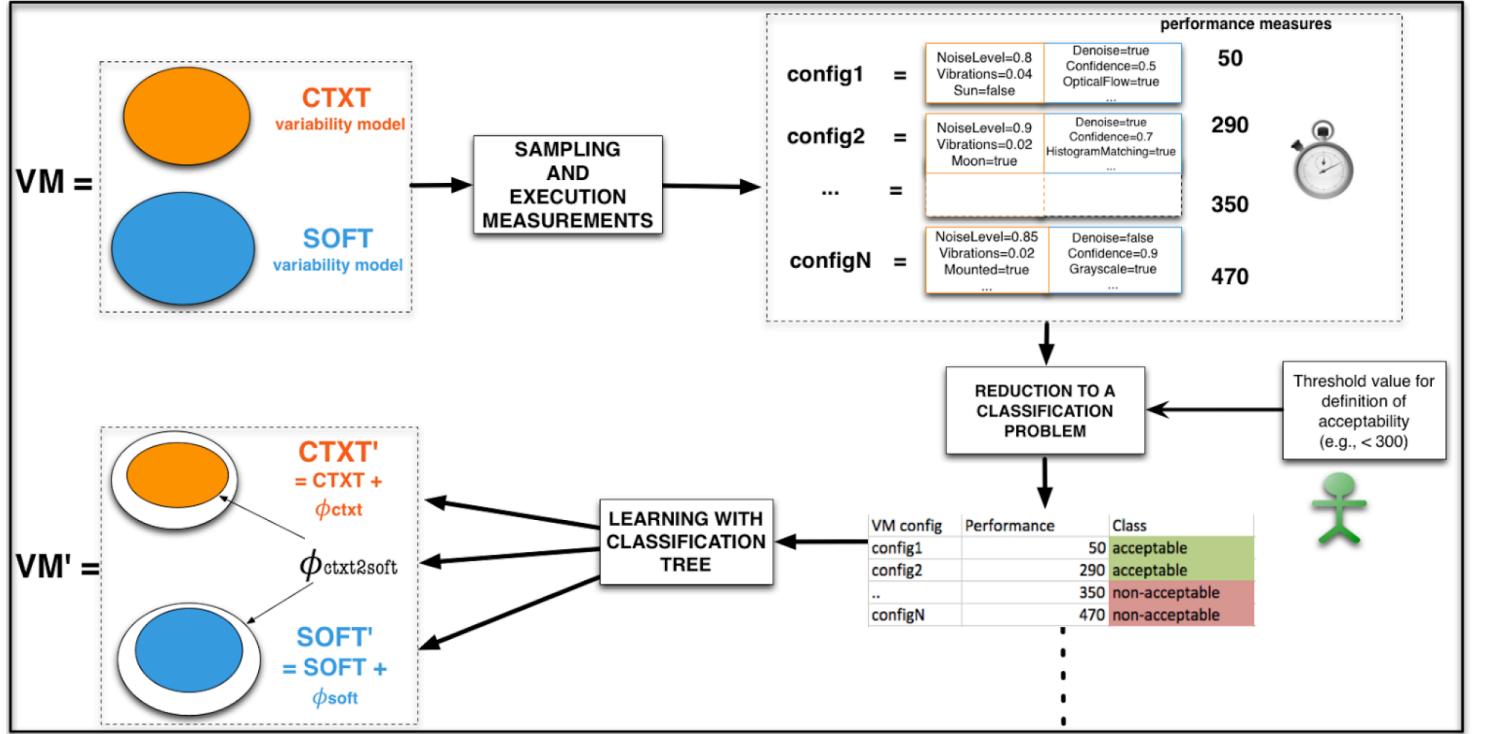
?? seconds

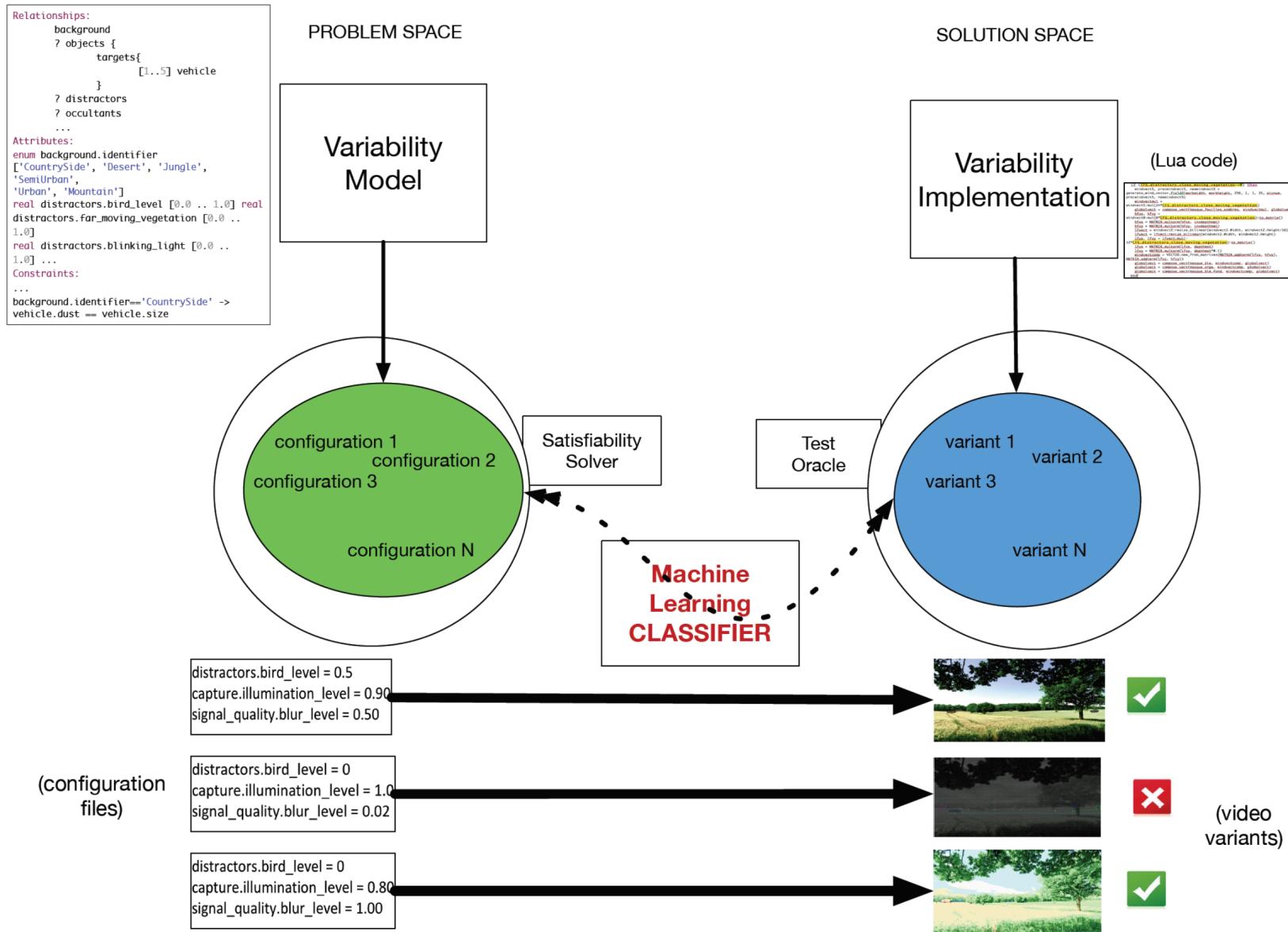
# Context and Variability



# Learning Contextual Variability Models





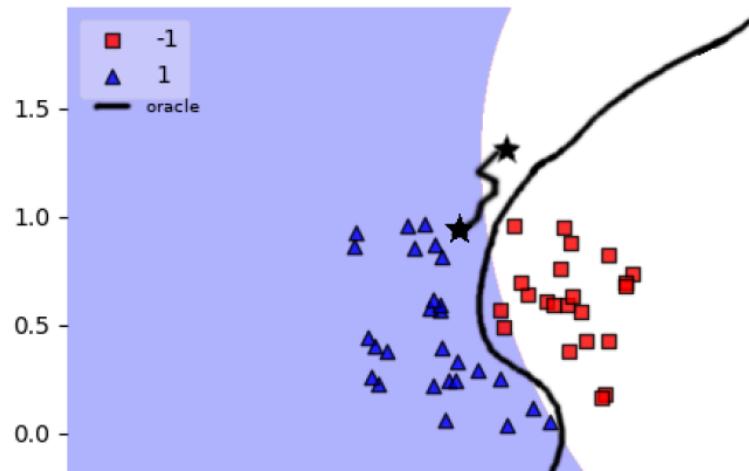


## Adversarial ML

- Appeared in 2004
- Popular around 2014 with GANs<sup>a</sup>
- Continue to be popular today<sup>b</sup>

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

b. Zhang et al., *DeepRoad : GAN-based Metamorphic Autonomous Driving System Testing*, ASE'18



- Can we use Adv ML with SPLs to create new configs that are misclassified ?
- Can Adv. Config. be valid ?
- Is it worth it using Adv ML technique ?

*answers on Friday 10 :30am, Session 11 : Configuration and Sampling*