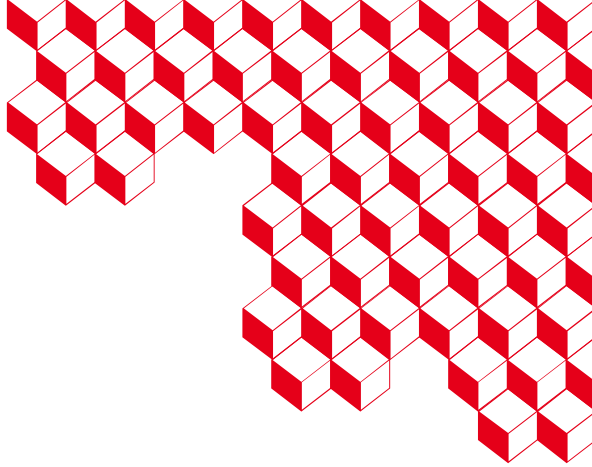


AIMOS: Metamorphic Testing of AI – An Industrial Application

Augustin Lemesle (CEA List)

Aymeric Varasse (CEA List)

Zakaria Chihani (CEA List)

Dominique Tachet (Renault)

Table of Contents

Introduction

Artificial Intelligence is becoming more and more prevalent, but is far from being flawless.

However, reliability is a critical *sine qua non* for the adoption of AI-based components.

Thus, we need to have tools and methodologies to ensure the reliability of AI models in various scenarios.

Metamorphic testing

Metamorphic property:

Considering relationships (R_1, R_2) and some inputs (a, b, c), then a sound software S should induce other relationships (Q_1, Q_2) on the outputs:

$$\forall a, b, c, R_1(a, b) \wedge R_2(b, c) \rightarrow Q_1(S(a), S(b)) \vee Q_2(S(b), S(c))$$

Example:

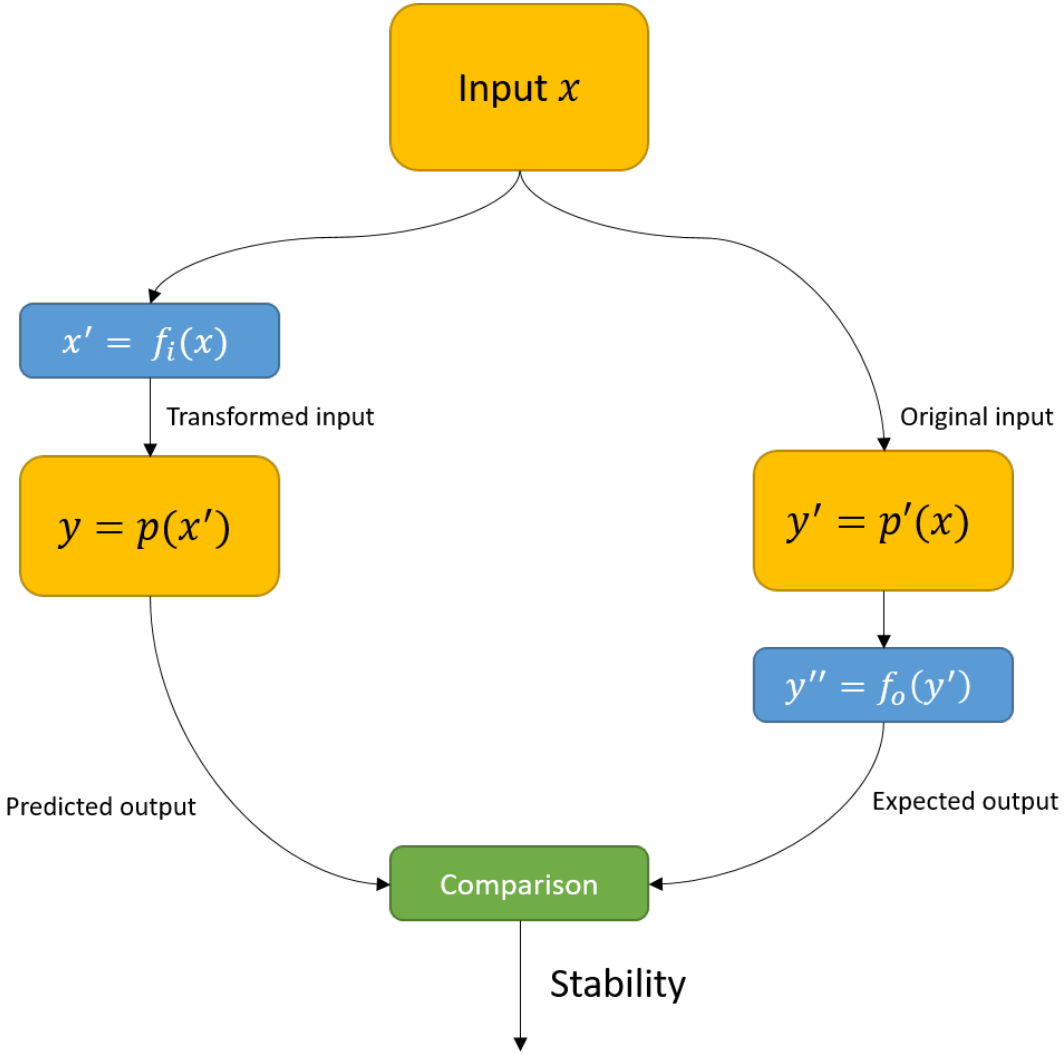
A software that computes the minimal cost of travel between two points in an undirected graph should be impervious to symmetry.

AIMOS

AIMOS (Artificial Intelligence Metamorphic Observing Software) is a tool to assess the stability of AI systems using metamorphic testing.

- No need to label data for testing.
- Automates the entire process of applying metamorphic properties on the inputs and outputs of models, comparing them and compiling the results into a stability score.
- Model agnostic (Neural Networks, Support Vector Machines, *etc.*).

AIMOS – Basic Principle



AIMOS

AIMOS was built with a few key principles in mind:

- Ease of use
- Extensibility
- Modularity

AIMOS – Ease of use

- Written in Python
- Model agnostic: only the inference functions are needed.
- Built-in support for various frameworks, input formats and model types.

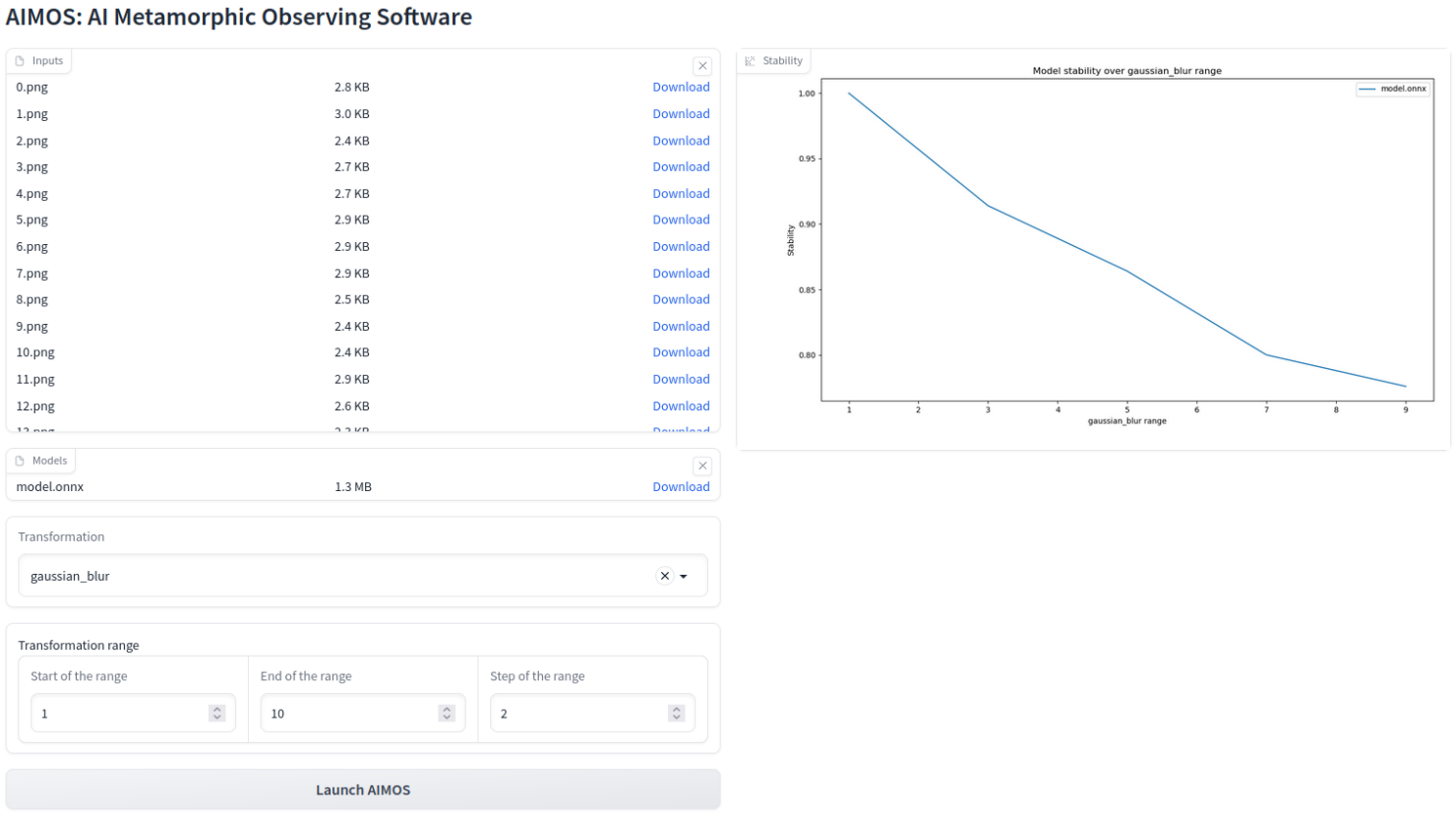



- Built-in classical transformations (rotation, noise, symmetry, *etc.*).



AIMOS – Ease of use

- With a configuration file
- As a Python library
- With a Graphical User Interface



AIMOS – Modularity and Extensibility

Any operation can be replaced with a custom made Python function (loading the model, the inputs, new metrics, *etc.*).

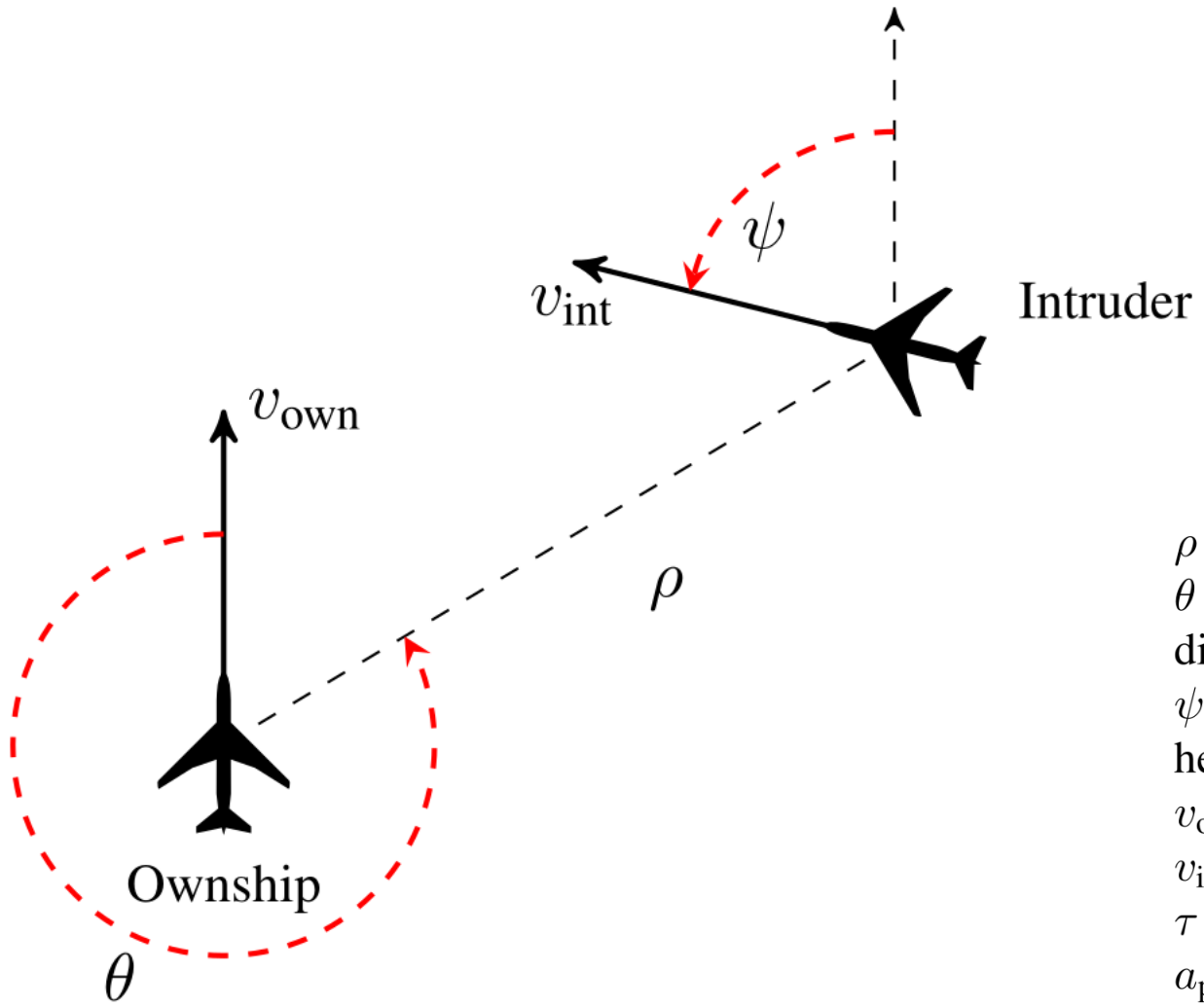
```python
def dead_columns(input, columns=np.uint8([50, 100, 150])):
    """ Adds dead pixel columns to an image. """
    input[:, columns, :] = 0
    return input
```

Use Cases

AIMOS has already been used on two industrial use cases:

- ACAS Xu
- Renault Welding

Use Cases – ACAS Xu



ρ (m): Distance from ownship to intruder.
θ (rad): Angle to intruder relative to ownship heading direction.
ψ (rad): Heading angle of intruder relative to ownship heading direction.
v_{own} (m/s): Speed of ownship.
v_{int} (m/s): Speed of intruder.
τ (sec): Time until loss of vertical separation.
a_{prev} (°/s): Previous advisory.

Use Cases – ACAS Xu

- Airborne Collision Avoidance System for unmanned vehicles.
- 5 geometrical parameters + time until loss of vertical separation + previous advisory.
- 45 Deep Neural Networks to replace 2GB of lookup tables.

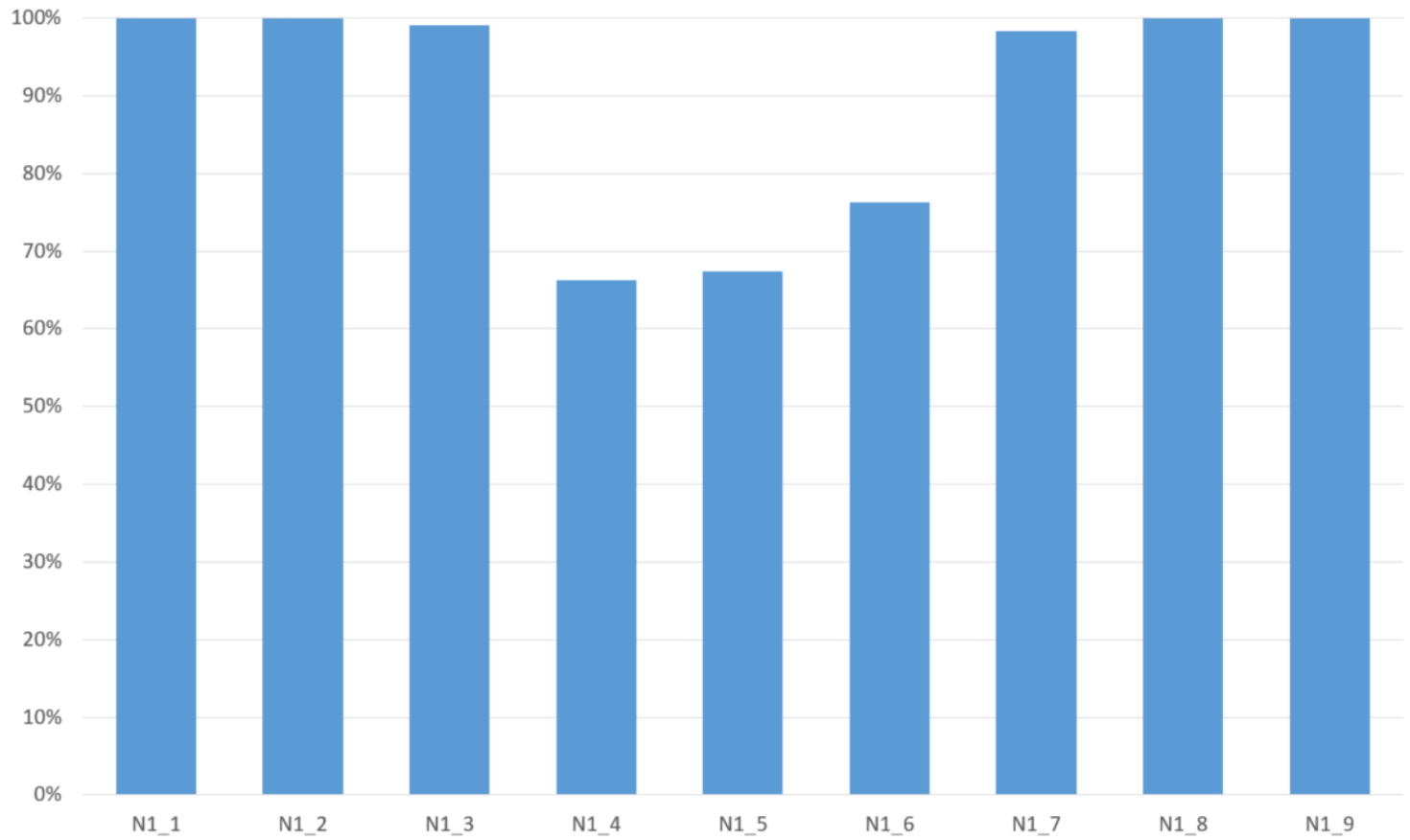
Use Cases – ACAS Xu

- For each model, we tested a symmetry alongside the ownership heading axis.
- 2 test scenarios on different ranges of input.
  - 100,000 tests points, uniformly distributed over the **full range** of the five inputs.
  - 100,000 tests points, uniformly distributed over a more **restricted part** of the input space.
- As the 45 networks were discretized through the previous advisory, when considering a symmetry on the system, the previous advisory should also be symmetrized.

Use Cases – ACAS Xu

Stability of networks in function of τ when the previous advisory is “Clear-of-conflict”.

Drop of stability in N1_4, N1_5 and N1_6 shows potential need for further training.



Use Cases – Renault Welding

- Control of the conformity of welds of rear axles on a vehicle production line of a Renault factory in Le Mans.
- Control realized by the analysis of an image of the weld by an algorithm which has been trained on labelled weld images.

Use Cases – Renault Welding

Operational Domain Design: Operating conditions under which a given automation system is specifically designed to function.

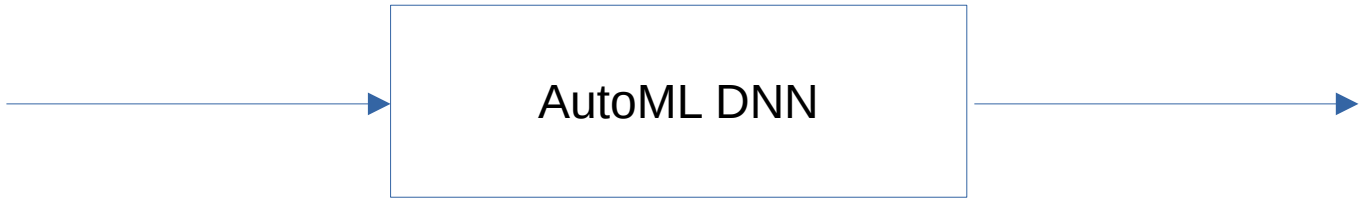
- The degradations applied were selected by Renault by analyzing their Operational Design Domain (ODD) for this system.
- Renault's context analysis on the problem with both the fixed input parameters (AD) and the ODD parameters to take into account for testing.

Image		
1	Dimension	AD
2	Size	AD
3	Blur	ODD
4	Colorimetry	ODD
5	Rotation	ODD
6	Translation	ODD
7	Transport noise	ODD
8	Number of colors (RGB)	AD
9	Transparency (A)	AD
10	Image format	AD
11	Histogram	ODD
12	Zone of interest	ODD

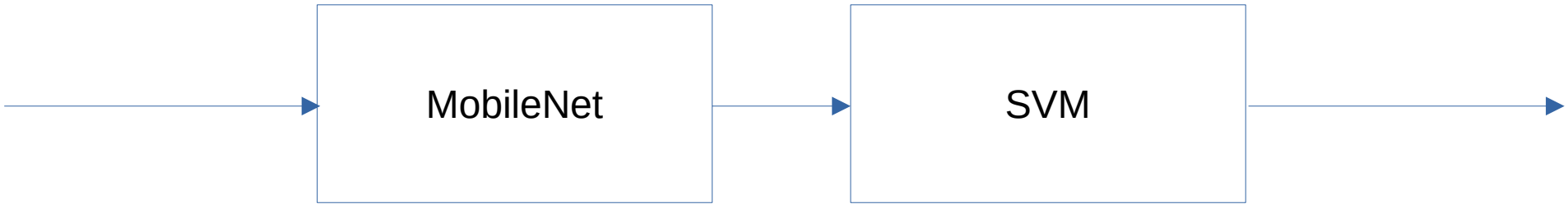
Use Cases – Renault Welding

Two types of models are currently being used by Renault depending empirically on where they achieve the best results:

- Models generated automatically through Google's AutoML framework (Neural Network).

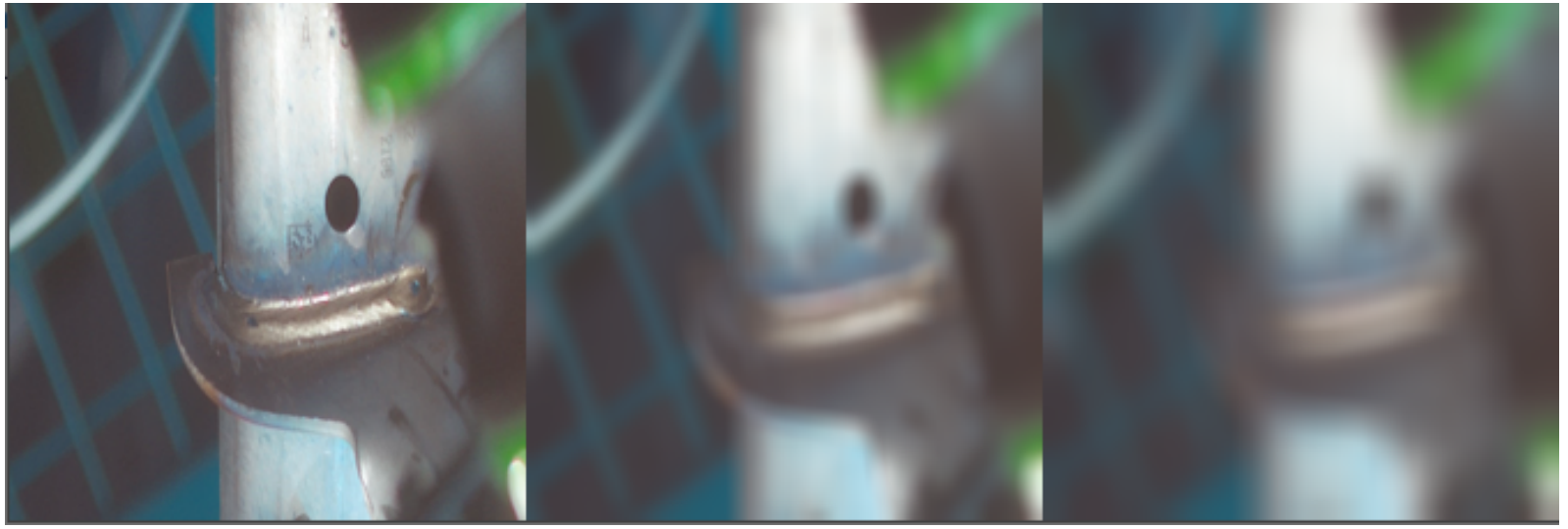


- Specific models created by data scientists at Renault internal R&D laboratory (Neural Networks + Support Vector Machine).



Use Cases – Renault Welding

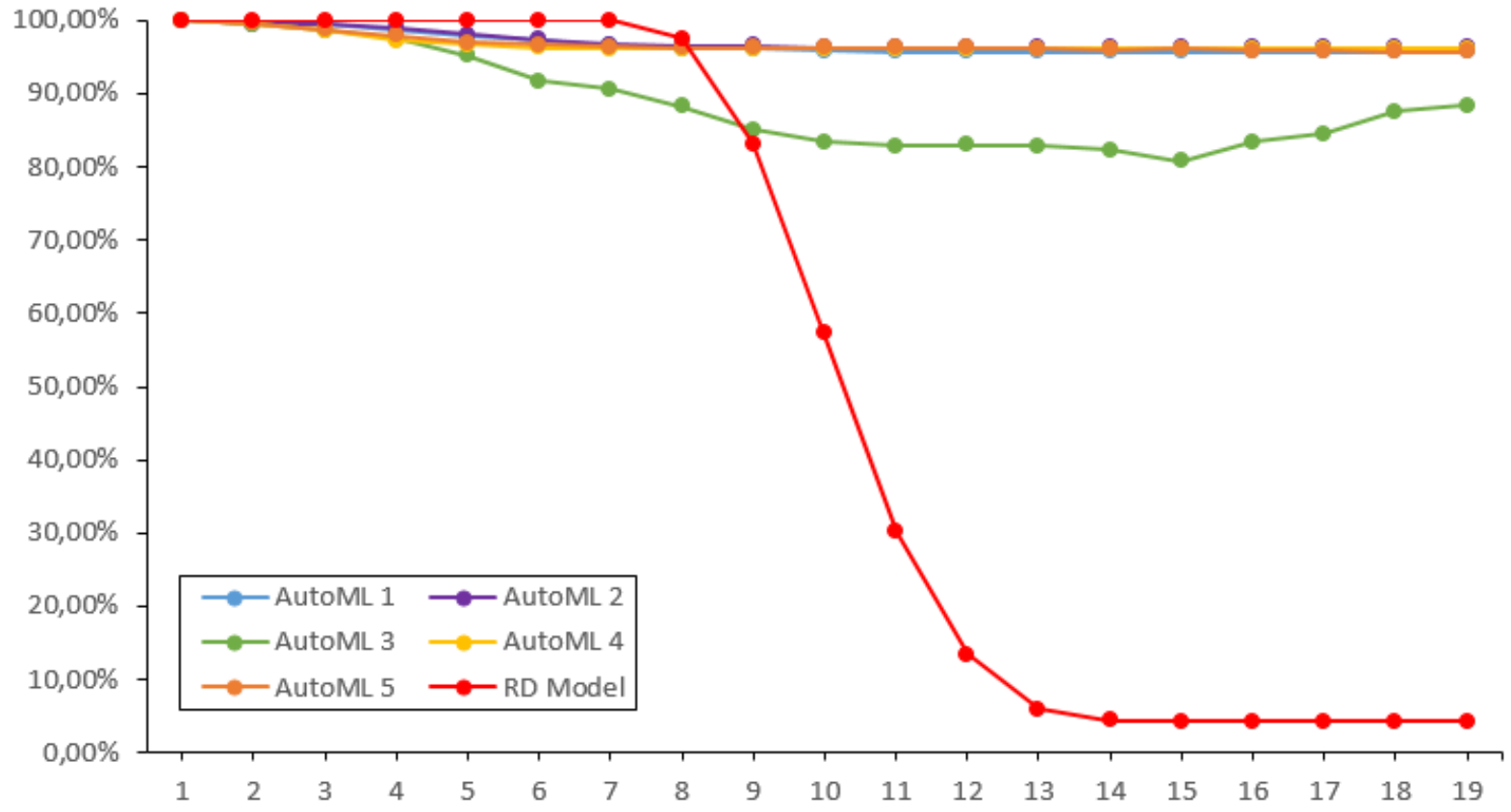
- 3 different production lines called C10, C20 and C34 and their corresponding weld.
- 5 AutoML models and 1 RD model per production line.
- For each model, we focused on a blur perturbation.



Use Cases – Renault Welding

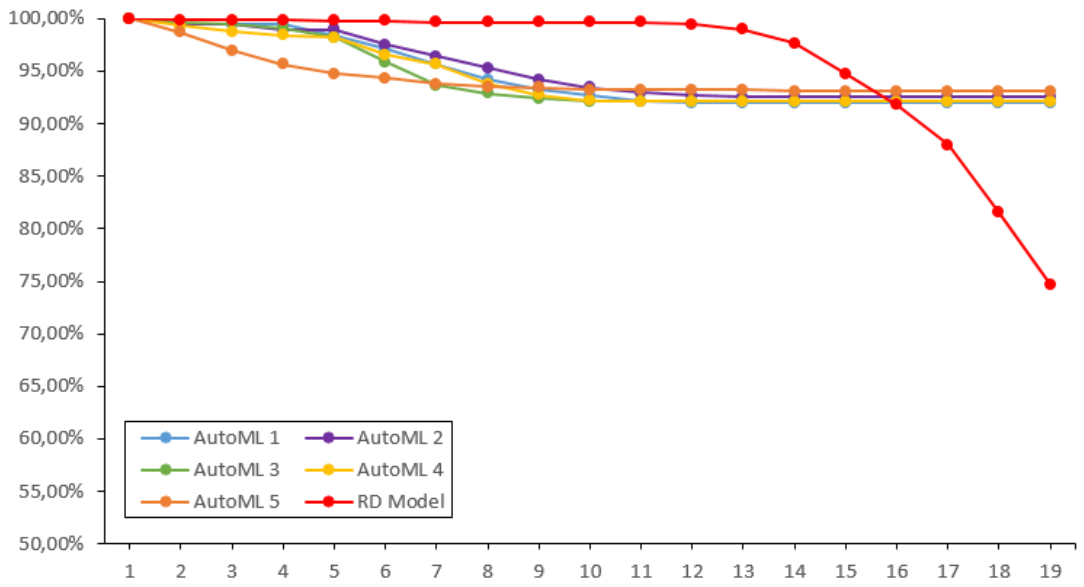
C34

- **AutoML**:
  Stability drops quickly for low perturbations then plateaus at a lower level.
- **RD**:
  Greater stability up until a certain point, then drops drastically in stability.

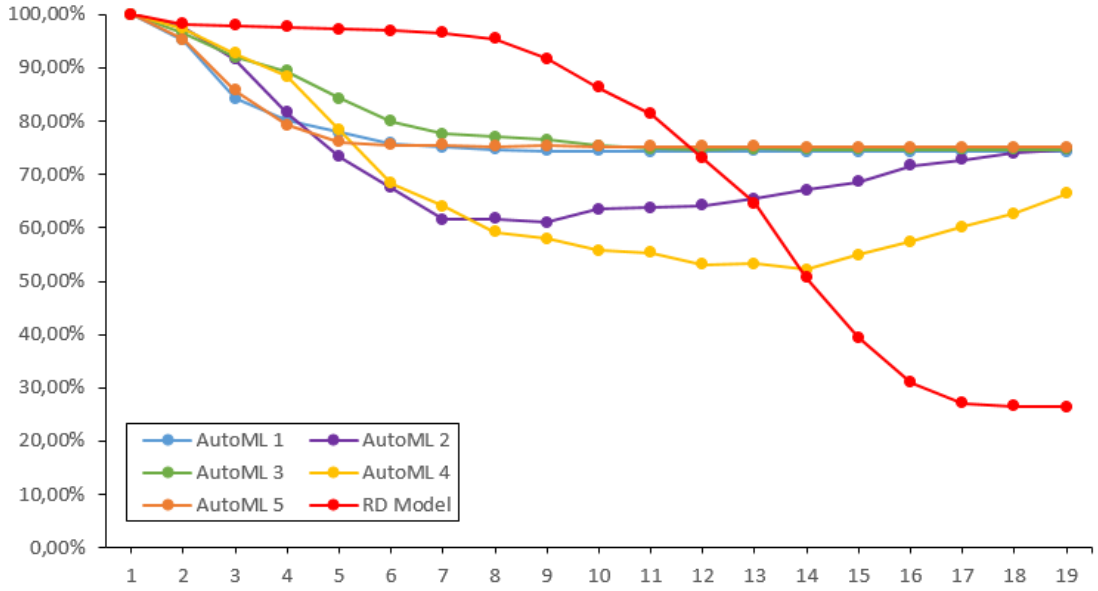


Use Cases – Renault Welding

C10



C20



Conclusion

AIMOS is a tool that can be integrated in the verification and validation process of AI-based components.

- Freely available for teaching and research purposes.
- Integrated in CAISAR, an open-source platform for characterizing safety in AI systems.

CAISAR

- Next step: built-in support for Time Series.

AIMOS: AI Metamorphic Observing Software

Inputs

0.png	2.8 KB	Download
1.png	3.0 KB	Download
2.png	2.4 KB	Download
3.png	2.7 KB	Download
4.png	2.7 KB	Download
5.png	2.9 KB	Download
6.png	2.9 KB	Download
7.png	2.9 KB	Download
8.png	2.5 KB	Download
9.png	2.4 KB	Download
10.png	2.4 KB	Download
11.png	2.9 KB	Download
12.png	2.6 KB	Download

Models

model.onnx	1.3 MB	Download

Transformation

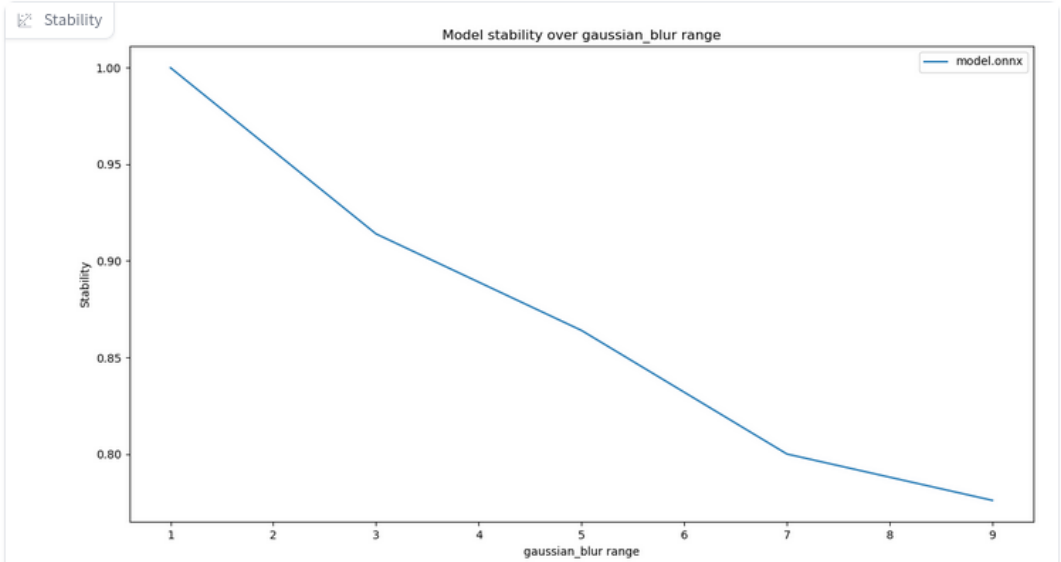
gaussian_blur

Transformation range

Start of the range	End of the range	Step of the range
1	10	2

Launch AIMOS

Stability



AIMOS – Configuration file

```yaml
options:
  plot: True
  inputs_path: "inputs"
  transformations:
    - name: "gaussian_blur"
      fn_range: range(1, 10, 2)

models:
  - defaults:
      models_path: "models/model.onnx"
```

AIMOS – Python library

```python
from aimos import core

core.main(
    "./inputs",
    "./models/model.onnx",
    "average_blur",
    fn_range=range(1, 10, 2),
    plot=True,
)
```