Members of the chair

ONERA
Claire Pagetti (COVNI)
Kevin Delmas (RIME)
Charles Lesire (SEAS)

LAAS
Jérémie Guiochet

IRIT
Thomas Carle

Airbus
Mélanie Ducoffe (plan de relance)
Adrien Gauffriau (MAD)

CS
Mohammed Belcaid (MAD)

Thésards ANITI
Iryna De Albuquerque
Iban Guinebert
Noémie Cohen
Anthony Faure-Gignoux

Post doc ANITI
Joris Guérin (2020 – 2022) now CR @IRD

Stagiaires M1/M2
9

Lightening talks
Certification
- evaluation of an argumentation, to convince that a system (i.e., its architecture, its settings, including mitigation means...) is compliant with the regulatory requirements
- accepted mean of compliance with the requirements is to rely on mature standards

Applicative scope:
- ML (Machine Learning) based systems

Active contribution to
- DEEL mission certif
- EUROCAE/FAA ED 324 / ARP 6983 (more particularly on the implementation section)

System requirements
- System architecture
- Item requirements
- System integration / verification

ODD definition
- Data Management
- Model design
- Model verification and mitigation means
- Item allocation and requirements
- Verification and validation of implemented model

Item specification and requirements
- Item implementation and verification

DO 178 C SW / DO 254 HW

General objective: End-to-end development process to achieve the expected level of performances and provide some of the evidences required by certification.
Outline & contributions

- Trained ML model
  - Semantic
    - ML model description in formal format
  - 2. Formal verification
    - Compliance with the requirements
  - 4. Implementation
    - Efficient and predictable code generation

System and Safety Requirements

1. ODD definition

Use cases
- 1. Academic ACAS Xu
- 2. Academic UAV emergency landing
- 3. VBL (vision-based landing)
  - with LARD open source
- 4. DDT (drone detection tracking)
  - proprietary

Semantic ML model description in formal format

Runtime verification

Implementation

Efficient and predictable code generation

3 targets: ARM v7 + NEON

CPU

Vector extension

L1I

L1D

L2

DDR

NVIDIA Xavier AGX

HW accelerator: LeNet 5 streaming architecture

3 targets: ARM v7 + NEON

ODD definition

Conv1

Pool1

Conv2

Pool2

FC1

Sel1

FC2

Sel2

FC

MAC

X

+
Avoidance Collision System for vertical and horizontal cooperative and non-cooperative avoidance (Multi-Intruders)

**Intended function** "the intruder should not enter in the ownship envelope"

**ODD:** pre-defined ranges of inputs
- Initially based on look-up tables (LUT)
- Replacing the LUT by neural networks proposed by Standford
- Interest: Gain in memory footprint (from 4Go to 3Mo)

**ANITI focus:**
- Formal verification
- Hybrid architecture definition (safety net as a backup when the NNs do not replicate well the LUT)
- Implementation

**ACAS Xu**

- Decision algorithm
  - Clear of Conflict (CoC)
  - Weak Left (WL)
  - Weak Right (WR)
  - Strong Left (SL)
  - Strong Right (SR)

**LUT-based controller**

- Choose 5 LUTs from pa
- argmin on the selected 5 LUTs

45 NN (fully connected NN with 6 layers of 50 neurons each)
**UAV emergency landing**

**Intended function:** identify a landing zone in urban environment ensuring a safe emergency landing. If no suitable landing is found, the flight is aborted.

**ANITI focus:**
- Runtime verification
**Intended function:** computing the position of the aircraft from the position of the runway within an image taken during the approach and landing phases of an aircraft.

**ANITI focus:**
- Dataset design
- ODD definition
- Formal verification
- Implementation
LARD – Landing Approach Runway Detection – Dataset

- **Training dataset**
  - Google Earth Studio and Microsoft Flight Simulator synthetic images of 33 runways

- **Test dataset**
  - synthetic images of 79 runways
  - real footage of 38 runways

Collaboration with **DEEL** mission certif

**Tarbes:** Comparison of a real landing footage (left) with synthetic replicas (Google Earth Studio center, Microsoft Flight Simulator right)

https://github.com/deel-ai/LARD
Intended function: camera-based detection of intruder drone:

- Focus and tracking of the drone by the camera

ANITI focus:

- Extension of existing academic and CS Group proprietary dataset
- ODD definition
- Model design
- Implementation on the NVIDIA Xavier

Main difficulties:

- Object size ≤ 25 pixels
- Birds and drones
- Position within the image

- Collaboration with IRT Saint Exupéry – CS Group within ARCHEOCS project
ODD definition via several operational scenarios

- Category of objects (bird, drone), size of the objects, range of velocities
- Possible trajectories
- Diversity of backgrounds (empty grass background, buildings …)
- Possible out of ODD (helicopter, plane)

Existing dataset

- Internal company collection of data
  - limitations: not all the ODD is covered and many biases in the dataset (position of the drone in the image, type of background…)

Example of extension

- unbiased position of the object

<table>
<thead>
<tr>
<th>Parameters Test Set</th>
<th>Combinatorial - Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes (Objects)</td>
<td></td>
</tr>
<tr>
<td>Object Size</td>
<td>Min 20 pixels – Max 400 pixels</td>
</tr>
<tr>
<td>Background</td>
<td>Sky, buildings, Landscape</td>
</tr>
<tr>
<td>Position</td>
<td>Uniform Spatial distribution</td>
</tr>
<tr>
<td>Texture</td>
<td></td>
</tr>
<tr>
<td>Brightness</td>
<td>Monotonic Function</td>
</tr>
<tr>
<td>Geometric Transformation</td>
<td>Rotation – Symmetric</td>
</tr>
</tbody>
</table>

Distribution of object position in the dataset

Transformation to unbiase the Position

Modification with uniform spatial distribution
Outline & contributions

1. ODD definition

2. Formal verification
Compliance with the requirements

3. Runtime verification
Purpose verify that the ML-based system fulfills the intended function in the ODD.

4. Implementation
Efficient and predictable code generation

Semantic
ML model description in formal format

Trained ML model

System and Safety Requirements

3 targets: ARM v7 + NEON

NVIDIA Xavier AGX

HW accelerator: LeNet 5 streaming architecture
Reachability problems

- **Property:** Given an input set $X$ and a NN model realising the function $F$, what is the reachable set $F(X)$?

- **Practical property:**
  - $X$ is approximated with an abstract domain
  - solver computes (an over-approximation of) $F(X)$

- **Existing solvers:**
  - Exact solvers: Reluplex/Marabou, Planet, ...
  - Approximating solvers: Auto-Lirpa (IBP, Crown, ...), DecoMon, ...

**What you know:**
- Last year ANITI Days presentation with Mélanie
- Lightning talk of Noémie
DecoMon – verification tool developed by Mélanie

Build a toolbox to ease the use of **formal methods** among Airbus Data Scientists.

[GitHub link]

**Automatic** Keras Conversion (Formal Training)

**Automatic** Common properties built with **divisions**

---

**Business**

- Open-source Airbus library
- Used by BUs
- Airbus ML-flow
- Compatibility with ANITI’s libraries

**Research**

- CVPR 2023: FM for XAI
- ICAF 2023: FM for predictive maintenance
- Ongoing submission: NASA FM ...

[CertifAI chair]

16/11/2023
**Definition [p-box]:** $p \in \mathbb{N}$, a $p$-dimensional box $[b]^p$ is a set of $\mathbb{R}^p$ defined as the cartesian product of $p$ intervals:

$$[b]^p = \times_{i<p+1} [l_i, u_i]$$

**Definition (Similar behaviour).**


We consider that a $\text{NN}_{pa,\text{range}}$ behaves similarly to the $\text{LUT}_{pa}$ on $l$ if

$$\text{decisions } \text{NN}_{pa,\text{range}}(l) \subseteq \text{decisions } \text{LUT}_{pa}(l)$$

**Case 1 3D (fixing $v_{\text{own}}, v_{\text{int}}$):** 304 000 p-boxes

<table>
<thead>
<tr>
<th></th>
<th>Verif time</th>
<th>Number of true</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reluplex NN</td>
<td>5 days</td>
<td>254 670</td>
<td>84%</td>
</tr>
<tr>
<td>adversarial</td>
<td>17 hours</td>
<td>286 023</td>
<td>94%</td>
</tr>
<tr>
<td>Corner</td>
<td>32 hours</td>
<td>272 212</td>
<td>90%</td>
</tr>
<tr>
<td>deellip</td>
<td>26 min</td>
<td>280 000</td>
<td>92%</td>
</tr>
</tbody>
</table>

**Case 2 5D:** $36.10^6$ p-boxes

<table>
<thead>
<tr>
<th></th>
<th>Verif time</th>
<th>Number of true</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reluplex NN</td>
<td>&gt; year</td>
<td></td>
<td></td>
</tr>
<tr>
<td>adversarial</td>
<td>29 days</td>
<td>34 352 549</td>
<td>93.4%</td>
</tr>
<tr>
<td>Corner</td>
<td>&gt; 1 month</td>
<td></td>
<td></td>
</tr>
<tr>
<td>deellip</td>
<td>3 days</td>
<td>34 173 698</td>
<td>92.9%</td>
</tr>
</tbody>
</table>
Outline & contributions

1. ODD definition

Trained ML model

System and Safety Requirements

Semantic
ML model description in formal format

3. Formal verification
Compliance with the requirements

3. Runtime verification

4. Implementation
Efficient and predictable code generation

Purpose

➔ In addition to formal verification
➔ Detect hazardous behaviors at runtime

Research focus: Evaluation of safety monitors

➔ What are the relevant metrics to assess safety monitors?
➔ Is it possible to unify the evaluation method to enable a fair comparison of safety monitors

3 targets: ARM v7 + NEON

NVIDIA Xavier AGX

HW accelerator: LeNet 5 streaming architecture

1. ODD definition

CPU          Vector extension
L1I          L1D
L2           DDR

bus

conv1
pool1
conv2
pool2
FC1
seq1
FC2
seq2
FC

MAC

X

+
What is Safety Monitoring of Deep Neural Networks?

Sensory input (e.g., Image)  
Deep Neural Network  
Prediction used by critical system

Safety Monitor (may observe many parts of the DNN)

Warning
unsafe to use the DNN predictions

Images: Cordts et al. "The cityscapes dataset for semantic urban scene understanding."
Out-of-Model-Scope: detecting situations where DNN may fail
⇒ dependent on the DNN

Out-of-Distribution: detecting situations that are not sampled according to the training distribution
⇒ independent from the DNN (dependent on the dataset)

- The definition of OMS is **objective**
- OMS is not assuming which situations are difficult for the DNN

- The definition of OOD is **ambiguous** e.g., how much perturbation is required to define OOD?
- A perfect OOD detector can **discard valid predictions** the DNN
- The best monitor for OOD is not always the best to **detect errors**
Evaluation framework inspired from RL training process

→ **Assess the safety and capacity** to complete the mission (i.e., mission reward)
→ Applicable to any monitor for perceptive functions
→ Request a clear formalisation of the assumptions that are made regarding the impact of a ML error on the system's safety

Applied on automotive and UAV use cases and demonstrates that:
→ Monitors developed for OOD may not be the best to ensure the safety

Selecting a rejection threshold is pivotal during the design of a safety monitor but:

→ Many works use threshold-agnostic evaluation metrics (e.g., AUROC)
→ Very-few works are addressing the problem of selecting an optimal threshold
→ The actual evaluation the safety monitor should be done for the selected rejection threshold

Develop a threshold optimisation method for safety monitors:

→ Assess the impact of prior knowledge on problematic situations (i.e., runtime threats)
→ Select threshold-aware metrics to evaluate a monitor
→ Provide an automated optimization of the rejection threshold

Red Pill or Blue Pill? Thresholding Strategies for Neural Network Monitoring, Tran Khoi, Joris Guérin, Kévin Delmas & Jérémie Guiochet, ICLR 2024 (under review)
Outline & contributions

1. ODD definition

2. Formal verification
   Compliance with the requirements

3. Runtime verification

4. Implementation
   Efficient and predictable code generation

Purpose
Implement the ML models on embedded hardware while preserving the semantic and being able to compute WCET

Semantic
ML model description in formal format

Trained ML model

System and Safety Requirements

3 targets: ARM v7 + NEON
NVIDIA Xavier AGX
HW accelerator: LeNet 5 streaming architecture

CPU
Vector extension
L1I
L1D
L2
DDR
bus

Purpose
Implement the ML models on embedded hardware while preserving the semantic and being able to compute WCET
Certification context (DO 178-C)

- Traceability between the requirements and the (source) code
- Capacity to compute tight WCET
- Intense testing

ACETONE Automatic sequential C code generation from inference model

https://github.com/idealbug/NNCodeGenerator

- Convolution, 3 implementation: direct conv, naïve gemm, gemm (block matrices)

Criteria:

- Semantic preservation: similar results in the order of $10^{-6}$
- WCET
- MET

<table>
<thead>
<tr>
<th>Framework</th>
<th>Execution time [cycles]</th>
<th>WCET [cycles]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACAS-Xu</td>
<td>LeNet-5</td>
</tr>
<tr>
<td>ACETONE</td>
<td>decr128</td>
<td></td>
</tr>
<tr>
<td>Keras2C</td>
<td>1 104 134</td>
<td>25 786 401</td>
</tr>
<tr>
<td>uTVM static</td>
<td>681 708</td>
<td>10 201 249</td>
</tr>
</tbody>
</table>

Extending a predictable machine learning framework with efficient gemm-based convolution routines. De Albuquerque et al. RTS. 2023

Purpose: Impact of hardware faults on the execution of DNN

- **Type of accelerator**: streaming architecture

- HW fault injection
- Formal methods to assess the quality of fault injection strategy
- Reduction of fault injection points while preserving the coverage

Collaboration with NXP

Fault injection strategies: identifying HW failures with functional impact. Guinebert et al. ETS industrial paper. 2023
End-to-end development process to achieve the expected level of performances and provide some of the evidences required by certification.

**ANITI2**: industrial chair on “Embeddability and safety assurance of ML-based systems under certification (CertifEmbAI)”

**Conclusion & future work**

- **Aware Training**
  - Embeddability and verifiability aware training
  - Embeddability and verifiability aware quantization

- **Semantic**

- **Formal verification**
  - Image based systems

- **Implementation**
  - More parallelism, more programming languages ...

- **Hybrid architecture** – more accelerators – WCET estimation

  - **CPU**
  - **CPU**
  - **DDR**
  - **GPU**
  - **FPGA**
  - **DLA**

- **System and Safety Requirements**

- **Runtime verification**

- **Use cases**
  - Continue contributing to use case definition and analysis (e.g. LARD, DDT, drone emergency landing)

- **ODD definition**

CertifAI chair
16/11/2023