Advanced Testing of Deep Learning Models:
Towards Robust AI

Winter Semester – 2024-25

Vivek V. Vekariya
Simon Speth
Prof. Dr. Alexander Pretschner

Lehrstuhl für Software and Systems Engineering
Technische Universität München
04.07.2024
The world of AI testing

- **DNN Testing**: How to identify corner cases?
- **Dataset Quality Evaluation**: How to evaluate dataset quality for safety assured performance?

### Hidden defects of DNN (Corner Cases)

1. Hand Picked Test set
2. Inadequate test dataset
3. Statistical measure about the quality of test dataset?
Exploring Latent Space Coverage

- **Dataset Quality Aspects:**
  - Robust test dataset: e.g. Accuracy - 0%
  - Diverse test dataset: Test more underlying faults

- **Latent Space Coverage:**
  - Coverage, Density & Sparsity Estimation
    - Verify training policies
    - Estimate potential data collection gap
Exploring Latent Space Coverage

- **Directly using Latent space vectors:**
  - GANs & VAEs

- **Corner Case Identification:**
  - Coverage-guided Fuzz Testing
  - Latent Space based Testing
  - Metamorphic Relation Testing

Is this a true maximization of latent space coverage?

Dense and Sparse test data points in Latent Space

Ideal test data points in latent space
Coverage-Guided Fuzzing

1) Failed test
2) Coverage Information
Metamorphic Testing

- Metamorphic Testing (MT) is one method to solve the oracle problem for Deep Learning Models
  - There are usually no oracles for DL models
  - Metamorphic testing can be seen as a pseudo-oracle/model
  - Reverse engineering of a part of the specification

- Metamorphic Relations (MR) need to be defined in order to compute test cases
  - Source test inputs are used to compute follow-up inputs
  - Both inputs (source and follow-up) are fed into the System Under Test (SUT)
  - Both outputs and both inputs are compared to check whether the MR holds true
Metamorphic Testing

- **Example:** Testing the implementation of the $\sin(x)$ function
- **Assumption:** We implement a test case $\sin(2)$ but don't know what the correct output is.
- **Metamorphic Testing:** Creation of a follow-up test case $\sin(2 + 2\pi)$ which is expected to have the same output as the source test case $\sin(2)$.
- **Test Case Evaluation:** We check if the relation $\sin(2) = \sin(2 + 2\pi)$ holds. If yes, the test case passed.

```
<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2$</td>
<td>$\sin(2)$</td>
</tr>
<tr>
<td>$2 + 2\pi$</td>
<td>$\sin(2 + 2\pi)$</td>
</tr>
</tbody>
</table>
```

**sin() Example:**

```
2 \downarrow
\Rightarrow
2 + 2\pi \Rightarrow \sin(2 + 2\pi)
```
Metamorphic Testing

Example: Deep Learning LiDAR object detection model:

\[ M = (R, \varphi) \]

\[ x \rightarrow f(x) \]

\[ \varphi(x) \rightarrow f(\varphi(x)) \]
Metamorphic Testing

Example: Deep Learning LiDAR object detection model:

• Testing of a LiDAR object detection model:
• $\varphi(x)$: Rotation of the follow-up point cloud by 180°
• $R$: Inverse 180° rotation of all output 3D bounding boxes. Then we check if all follow-up bounding boxes have a corresponding bounding box in the source output.
Learning Outcomes

• **Implementation, testing & evaluation** of state-of-the-art Classification & 2D Object Detectors DNNs
• Corner Case data generation using fuzzing, metamorphic relations and latent space properties
• GANs & VAEs for latent space coverage maximization
• Adversarial Attacks for state-of-the-art Classifiers and 2D Object Detectors
Prerequisites

**Required**

- Python (of course 😊)
- Deep Learning Frameworks (PyTorch, Keras, TensorFlow)
- Linux / Windows

**Good to have**

- Insights of 2D Object Detector Networks (SSD, Yolo, RCNN)
- Understanding of latent space and vector space modelling
- Passion for Safe AI

....But every smart work requires sincere dedication & commitment!
Agenda

- **Pre-course Meeting**: 04.07.2024
- **Apply with additional documents**: till 20.07.2024
- **Acceptance Notification**: 25.07.2024
- **Kick-off Meeting - 1**: XX.10.2024 (Di.)
- **Project Discussions & Allocation**: XX.10.2024 (Di.)
- **Weekly Follow-ups**
- **Mid-term Presentations**: TBD (Preliminary-Do.)
- **Final Presentations**: Feb.2025 (Preliminary-Do.)
1. Give your 1\textsuperscript{st} priority to this course in the matching system
2. Tell us more about you (motivation, CV, transcripts & Gitlab link) by filling out: 
   \url{TUM_I4_student_wiki}
Thank you for your attention 😊

Vivek V. Vekariya
Simon Speth
Garching bei München