Exploitable Results by Third Parties

18022 IVVES

**Project details**

<table>
<thead>
<tr>
<th>Project leader</th>
<th>Mark van Helvoort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email:</td>
<td><a href="mailto:mark.van.helvoort@philips.com">mark.van.helvoort@philips.com</a></td>
</tr>
<tr>
<td>Website:</td>
<td><a href="https://ivves.eu/">https://ivves.eu/</a> &amp; <a href="https://learn.ivves.eu/">https://learn.ivves.eu/</a></td>
</tr>
</tbody>
</table>
### Name: Synthetic 3D brain data

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
</table>
| ▪ Real 3D brain scans  
▪ MRI, CT, … | ▪ Data preprocessing, registration, augmentation  
▪ Multi-GPU GAN training | ▪ Synthetic 3D brain data |

| Unique Selling Proposition(s): | ▪ Full pipeline from data preprocessing to synthetic data generation  
▪ Designed to work with moderately low amount (≈1000 samples) of input data |
| Integration constraint(s): | ▪ Implemented as Docker images, which can be converted to Singularity images  
▪ Multiple GPUs and lots of RAM (>100 GB) recommended |
| Intended user(s): | ▪ Researchers, data scientists |
| Provider: | ▪ VTT |
| Contact point: | ▪ Harri Pölönen, harri.polonen@vtt.fi  
▪ Mark.Pijnenburg, Mark.Pijnenburg@philips.com |
| Condition(s) for reuse: | ▪ Mostly built on open-source technology  
▪ Component licenses: Apache 2.0 and MIT |

**Latest update: 29 Nov 2022**
## Name: RELOAD

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ The URL of Web-based SUT + the set of jmx files recorded by JMeter representing initial load test plans for each functional use case of SUT</td>
<td>▪ Automated ML-assisted Load Testing</td>
<td>▪ Load test report</td>
</tr>
</tbody>
</table>

### Unique Selling Proposition(s):
- Automated performance and load testing of a system using reinforcement learning

### Integration constraint(s):
- Can be provided as SaaS

### Intended user(s):
- Software testers
- Product managers
- Software engineers

### Provider:
- RISE Research Institutes of Sweden

### Contact point:
- Mahshid Helali Moghadam [Mahshid.helali.moghadam@ri.se](mailto:Mahshid.helali.moghadam@ri.se)
- Mehrdad Saadatmand [mehrdad.saadatmand@ri.se](mailto:mehrdad.saadatmand@ri.se)

### Condition(s) for reuse:
- Licensing and permission required

*Latest update: Nov 2022*
### Name: DeepAD

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
</table>
| • Sensors’ data or data from other sources in timeseries format | • Deep Learning-based Anomaly Detection tool  
• Suitable for *unsupervised* datasets  
• Autoencoders (AE) and Long-Short Term Memory (LSTM) Neural Networks | • Discovers patterns in data that do not conform to the expected normal behaviour |

<table>
<thead>
<tr>
<th>Unique Selling Proposition(s):</th>
<th>Integration constraint(s):</th>
<th>Intended user(s):</th>
</tr>
</thead>
</table>
| • Condition monitoring  
• Predictive maintenance  
• Abnormal behavior detection | • Input data in a sequence format with time sample | • Software developers and designers of industrial application need to implement Anomaly Detection techniques. It has applications in cyber-security intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, detecting ecosystem disturbances, defect detection, and medical diagnosis. |

<table>
<thead>
<tr>
<th>Provider:</th>
<th>Contact point:</th>
<th>Condition(s) for reuse:</th>
</tr>
</thead>
</table>
| • RISE Research Institutes of Sweden | • Sima Sinaei [sima.sinaei@ri.se](mailto:sima.sinaei@ri.se)  
• Mehrdad Saadatmand [mehrdad.saadatmand@ri.se](mailto:mehrdad.saadatmand@ri.se) | • Licensing and permission required |

*Latest update: Nov 2022*
Name: Generating Adversarial Examples

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image datasets</td>
<td>Enhancing security and robustness of Neural Networks especially in the face of an adversary who wishes to fool the model</td>
<td>A slightly perturbed image, still easily recognizable by human observers with the goal of producing a wrong output from the correct target class</td>
</tr>
</tbody>
</table>

Unique Selling Proposition(s):

- Adversarial attacks and defense in Machine Learning applications

Integration constraint(s):

- A labeled dataset and a primary machine learning model for classification are needed. The security and Robustness of this Neural Network can be improved by generating an adversarial dataset

Intended user(s):

- AI-based system's developer

Provider:

- RISE Research Institutes of Sweden

Contact point:

- Sima Sinaei sima.sinaei@ri.se
- Mehrdad Saadatmand mehrdad.saadatmand@ri.se

Condition(s) for reuse:

- Licensing and permission required

*Latest update: Nov 2022*
Name: Data Quality Wrapper

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Tabular data (incl. synthetic)</td>
<td>• Automated EDA of inputs</td>
<td>• EDA report</td>
</tr>
<tr>
<td>• Text data</td>
<td>• Automated preprocessing of tabular data into training data</td>
<td>• Training data</td>
</tr>
<tr>
<td>• Images</td>
<td>• Augmentation of audio and image files</td>
<td></td>
</tr>
<tr>
<td>• Audio files</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Unique Selling Proposition(s):    | Automated EDA and preprocessing                                               |                              |
| Integration constraint(s):        | Easy report generation and sharing                                             |                              |
|                                  | Educational component of the solution                                          |                              |

| Intended user(s):                 | Students and ML practitioners                                                  |                              |
| Provider:                         | Sogeti NL                                                                     |                              |
| Contact point:                    | Tijana Nikolic – tia.nikolic@sogeti.com                                       | Almira Pillay – almira.pillay@sogeti.com |
| Condition(s) for reuse:           | MIT license                                                                   |                              |

Latest update: Nov 2022
Name: CodeAssist

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code repository</td>
<td>Code Complexity prediction</td>
<td>Code complexity</td>
</tr>
<tr>
<td></td>
<td>Explainable AI layer</td>
<td>Explainable AI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>artefacts</td>
</tr>
</tbody>
</table>

Unique Selling Proposition(s):
- Improve peer review process and code maintainability
- Shift-left

Integration constraint(s):
- Code repository needs to be in a standardized format, tagged with bugs and the pull requests need to follow a naming convention

Intended user(s):
- Developers

Provider:
- Sogeti NL

Contact point:
- Tijana Nikolic – tia.nikolic@sogeti.com
- Almira Pillay – almira.pillay@sogeti.com

Condition(s) for reuse:
- Commercial product

*Latest update: Nov 2022*
<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML files</td>
<td>ETL pipeline</td>
<td>Visualization</td>
</tr>
<tr>
<td></td>
<td>Dashboard</td>
<td></td>
</tr>
</tbody>
</table>

**Unique Selling Proposition(s):**
- A standardized and reusable pipeline and dashboard
- Improve move towards shift-left
- Improve observability in testing

**Integration constraint(s):**
- Azure
- Grafana

**Intended user(s):**
- Product owners and project managers
- Testers

**Provider:**
- Sogeti NL

**Contact point:**
- Tijana Nikolic – tia.nikolic@sogeti.com
- Almira Pillay – almira.pillay@sogeti.com

**Condition(s) for reuse:**
- Commercial product

*Latest update: Nov 2022*
<table>
<thead>
<tr>
<th>Name: Quality AI Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input(s):</strong></td>
</tr>
<tr>
<td>- Al Use Cases</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Unique Selling Proposition(s):</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Ethical AI considerations</td>
</tr>
<tr>
<td>• Quality AI considerations</td>
</tr>
<tr>
<td>• Education of practitioners</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Integration constraint(s):</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• None from a technical perspective, this is a framework</td>
</tr>
<tr>
<td>• Agile way of work</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Intended user(s):</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Product owners and project managers</td>
</tr>
<tr>
<td>• ML Practitioners</td>
</tr>
<tr>
<td>• Students</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Provider:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Sogeti NL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Contact point:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Tijana Nikolic – <a href="mailto:tia.nikolic@sogeti.com">tia.nikolic@sogeti.com</a></td>
</tr>
<tr>
<td>• Almira Pillay – <a href="mailto:almira.pillay@sogeti.com">almira.pillay@sogeti.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Condition(s) for reuse:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Free to reuse, it is a framework</td>
</tr>
</tbody>
</table>

*Latest update: Nov 2022*
Name: Failure analysis

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>xunit test result files</td>
<td>Tests failure analysis provides fast and reliable way to find and group similar failures in your CI pipeline</td>
<td>Find similar errors between tests</td>
</tr>
</tbody>
</table>

Unique Selling Proposition(s):
- When failure grouping and similarity scoring is done automatically by a machine, it will free resources from development team member to fix the most important failures in their CI/CD pipeline. It is tedious work for a human to download, open and read all the test failures and analyze which failures belong to the same group. The failure-analysis package solves this problem by processing xunit xml files and failures found within by calculating the similarity score of failures using cosine similarity.

Integration constraint(s):
- Python 3.8 or never
- numpy 1.23.4 or never
- pandas 1.5.1 or never
- sklearn 1.1 or never
- lxml 4.9.1 or never
- drain3 0.9.11 or never

Intended user(s):
- Software development team members, like developers and QA people. Teams with CI pipeline with decent amount of test automation result.

Provider:
- Open Source Code: [https://github.com/F-Secure/failures-analysis](https://github.com/F-Secure/failures-analysis)

Contact point:
- [https://github.com/F-Secure/failures-analysis/issues](https://github.com/F-Secure/failures-analysis/issues)

Condition(s) for reuse:
- Apache License 2.0

Latest update: Nov 2022
### Name: Flaky test detection

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
</table>
| - xunit test result files  
  - csv file | - Visualize tests which state changes most often. | - Calculation of test names which state changes most often  
  - Heatmap |

### Unique Selling Proposition(s):
- During software development, it is often common that some automated tests start to randomly fail, but finding those tests is a tedious and time-consuming problem. Flaky tests detection solves that problem by processing historical xunit test results and checks which tests changes state most often.

### Integration constraint(s):
- Python 3.7 or never  
  - pandas  
  - junitparser  
  - seaborn  
  - matplotlib

### Intended user(s):
- Software development team members, like developers and QA people. Teams with CI pipeline with decent amount of test automation result.

### Provider:
- Open Source Code: https://github.com/F-Secure/flaky-tests-detection

### Contact point:
- https://github.com/F-Secure/flaky-tests-detection/issues

### Condition(s) for reuse:
- Apache License 2.0

*Latest update: November 23, 2022*
### Name: Annotated software requirement corpus

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Software and system requirement descriptions annotated in universal dependencies</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Unique Selling Proposition(s):**

- English software requirement text corpus annotated with universal dependencies syntactic and part-of-speech information. Requirements are taken from a variety of domains contained in the open source PURE corpus (accessible at [http://nlreqdataset.isti.cnr.it/](http://nlreqdataset.isti.cnr.it/)).

**Integration constraint(s):**

- None

**Intended user(s):**

- Machine learning or natural language processing experts that require training or evaluation data for automatic analysis of software requirements in universal dependencies grammar as a standalone task or as part of a multiobjective task. This corpus can help improve natural language understanding tasks aiming to interpret, validate or analyse software requirements in a requirement validation or verification scenario.

**Provider:**

- Computer Research Institute of Montréal
- Data available: [https://github.com/UniversalDependencies/UD_English-CTeTex/](https://github.com/UniversalDependencies/UD_English-CTeTex/)

**Contact point:**

- Pierre André Ménard - pierre-andre.menard@crim.ca

**Condition(s) for reuse:**

- Open source licence [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

*Latest update: November 22, 2022*
<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>• An application</td>
<td>• Automated run-time test generation on GUI level</td>
<td>• Reports on test sequences leading to failures</td>
</tr>
<tr>
<td>with a graphical</td>
<td>• State model generation based on observed behavior of the GUI</td>
<td></td>
</tr>
<tr>
<td>user interface (GUI)</td>
<td>• ML-based GUI exploration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Unique Selling Proposition(s): | • Open source tool for automated robustness testing on GUI level |
| Integration constraint(s):    | • Run-time (scriptless) test generation               |
| Intended user(s):            | • The system under test has to be one of the following: |
| Provider:                    | • Windows GUI application                           |
| Contact point:               | • Web application                                   |
| Condition(s) for reuse:      | • Mobile (Android or iOS) application               |
|                            | • Software developers of applications with a GUI     |
|                            | • OUNL                                              |
|                            | • Tanja.Vos@ou.nl                                    |
|                            | • BSD-3 open source license                          |

Latest update: November 15, 2022
Name: Conformal Prediction Framework

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>Machine Learning framework for constructing predictive models that can estimate the confidence of their own predictions</td>
<td>Model predictions in the form of confidence intervals</td>
</tr>
<tr>
<td>datasets</td>
<td>Used on top of machine learning models, as built-in quality assurance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Good for safety-critical systems</td>
<td></td>
</tr>
</tbody>
</table>

Unique Selling Proposition(s):
- Small memory and energy footprint
- Works on top of any machine learning model of Ekkono’s SDK
- Aids towards validation of evolving systems
- Confidence bounds are guaranteed to contain the correct output value with some pre-defined probability
- Necessary for sensitive and high-risk applications

Integration constraint(s):
- The framework is part of the Ekkono SDK
  - Modeling: .NET 2.0+ or Python 3.7+ (Windows, macOS, or Linux)
  - Deployment: C++17 (or C++14 with included MPark.Variant), delivered as source to be compiled by the customer
  - Integration will be done through Ekkono’s C++ API of the compiled library

Intended user(s):
- Data Scientists, machine learning engineers, and software developers that want to run machine learning and conformal prediction on any type of device

Provider:
- Ekkono Solutions

Contact point:
- Rikard@ekkono.ai
- Eva@ekkono.ai

Condition(s) for reuse:
- Commercial license

Latest update: November 28, 2022
### Name: Change-Analyzer

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s):</th>
<th>Output(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Software Application with GUI (Windows / Web)</td>
<td>• Software Application exploration using Reinforcement Learning Automated Test Case generation done while exploring the Software Application Change Detection done by comparing collected data from different Software versions</td>
<td>• Generated Test Cases Change Detection report comparing two different Software versions</td>
</tr>
</tbody>
</table>

| Unique Selling Proposition(s): | • Open-source framework which explores Software Applications using Reinforcement Learning and uses the collected data to generate Test Cases and to detect changes across different Software versions |
| Integration constraint(s):     | • Python 3.7 or newer                                                             |
|                                 | • Pandas                                                                          |
|                                 | • Matplotlib                                                                      |
|                                 | • LudwigAI                                                                       |
|                                 | • OpenAI                                                                          |
|                                 | • PyTorch                                                                         |

| Intended user(s): | • Software development team members, QA & DEV                                    |
| Provider:         | • Open Source Code: [https://github.com/F-Secure/change-analyzer](https://github.com/F-Secure/change-analyzer) |
| Contact point:    | • Sorin Patrasoiu, sorin.p trasoiu@withsecure.com                              |
| Condition(s) for reuse: | • Apache License 2.0                 |

*Latest update: November 23, 2022*
Name: Anomaly Detector

<table>
<thead>
<tr>
<th>Input(s):</th>
<th>Main feature(s)</th>
<th>Output(s):</th>
</tr>
</thead>
</table>
| • Regression datasets | • Multi-variate algorithm that detects sudden or instantaneous deviations from a normal state  
• Monitors the health or performance of a device | • Anomaly score between 0 and 1, indicating how likely (1) or unlikely (0) the observed data point is an anomaly |

Unique Selling Proposition(s):
• Unsupervised algorithm, no need for labeled data  
• Algorithm is trained on normal state data, then used in unseen data to detect deviations  
• Applicable to streaming and offline datasets

Integration constraint(s):
• The anomaly detector is part of the Ekkono SDK  
  o Modeling: .NET 2.0+ or Python 3.7+ (Windows, macOS, or Linux)  
  o Deployment: C++17 (or C++14 with included MPark.Variant), delivered as source to be compiled by the customer  
  o Integration will be done through Ekkono’s C++ API of the compiled library

Intended user(s):
• Data Scientists, machine learning engineers, and software developers that want to run machine learning and anomaly detection on any type of device

Provider:
• Ekkono Solutions

Contact point:
• Rikard@ekkono.ai  
• Eva@ekkono.ai

Condition(s) for reuse:
• Commercial license

Latest update: November 28, 2022