ITEA 3

ITEA Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands

T + 31 88 003 6136
E info@itea3.org
W www.itea3.org

ITEA 3 is a EUREKA strategic ICT cluster programme

Exploitable Results by Third Parties

18022 IVVES

Project details

Project leader:	Mark van Helvoort
Email:	mark.van.helvoort@philips.com
Website:	https://ivves.eu/ & https://learn.ivves.eu/

Name: Synthetic 3D brain data

Input(s):	Main feature(s)	Output(s):
▪ 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: Apace 2.0 and MIT

Latest update: 29 Nov 2022

Name: RELOAD

Input(s):	Main feature(s)	Output(s):
▪ 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	▪ Automated ML-assisted Load Testing ▪	▪ Load test report ▪

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 ▪ Mehrdad Saadatmand mehrdad.saadatmand@ri.se
Condition(s) for reuse:	▪ Licensing and permission required

Latest update: Nov 2022

Name: DeepAD

Input(s):	Main feature(s)	Output(s):
▪ 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

Unique Selling Proposition(s):	▪ Condition monitoring ▪ Predictive maintenance ▪ Abnormal behavior detection
Integration constraint(s):	▪ Input data in a sequence format with time sample
Intended user(s):	▪ 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.
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: Generating Adversarial Examples

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

Unique Selling Proposition(s):	adversarial attacks and defence 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

Input(s):	Main feature(s)	Output(s):
▪ Tabular data (incl. synthetic) ▪ Text data ▪ Images ▪ Audio files	▪ Automated EDA of inputs ▪ Automated preprocessing of tabular data into training data ▪ Augmentation of audio and image files	▪ EDA report ▪ Training data

Unique Selling Proposition(s):	▪ Automated EDA and preprocessing ▪ Easy report generation and sharing ▪ Educational component of the solution
Integration constraint(s):	▪ App is deployed on Streamlit cloud and can freely be used ▪ Adding new features in app requires coding them in
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

Input(s):	Main feature(s)	Output(s):
▪ Code repository	▪ Code Complexity prediction ▪ Explainable AI layer	▪ Code complexity ▪ Explainable AI artefacts

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

Name: DevAssist

Input(s):	Main feature(s)	Output(s):
- XML files	- ETL pipeline - Dashboard	- Visualization

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

Name: Quality AI Framework

Input(s):	Main feature(s)	Output(s):
▪ AI Use Cases	▪ A framework to help organize AI projects ▪ Education of practitioners on quality and ethical AI aspects	▪ A set of tools and methods to ensure quality and ethical AI development

Unique Selling Proposition(s):	▪ Ethical AI considerations ▪ Quality AI considerations ▪ Education of practitioners
Integration constraint(s):	▪ None from a technical perspective, this is a framework ▪ Agile way of work
Intended user(s):	▪ Product owners and project managers ▪ ML Practitioners ▪ Students
Provider:	▪ Sogeti NL
Contact point:	▪ Tijana Nikolic – tia.nikolic@sogeti.com ▪ Almira Pillay – almira.pillay@sogeti.com
Condition(s) for reuse:	▪ Free to reuse, it is a framework

Latest update: Nov 2022

Name: Failure analysis

Input(s):	Main feature(s)	Output(s):
▪ xunit test result files	▪ Tests failure analysis provides fast and reliable way to find and group similar failures in your CI pipeline	▪ Find similar errors between tests

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
Contact point:	▪ https://github.com/F-Secure/failures-analysis/issues
Condition(s) for reuse:	▪ Apache License 2.0
	Latest update: Nov 2022

Name: Flaky test detection

Input(s):	Main feature(s)	Output(s):
▪ 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

Input(s):	Main feature(s)	Output(s):
N/A	Software and system requirement descriptions annotated in universal dependencies	N/A

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/).
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/
Contact point:	▪ Pierre André Ménard - pierre-andre.menard@crim.ca
Condition(s) for reuse:	▪ Open source licence CC BY-SA 4.0

Latest update: November 22, 2022

Name: TESTAR

Input(s):	Main feature(s)	Output(s):
▪ An application with a graphical user interface (GUI)	▪ Automated run-time test generation on GUI level ▪ State model generation based on observed behavior of the GUI ▪ ML-based GUI exploration	▪ Reports on test sequences leading to failures

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

Latest update: November 15, 2022

Name: Conformal Prediction Framework

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

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

Input(s):	Main feature(s)	Output(s):
▪ Software Application with GUI (Windows / Web)	▪ 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	▪ Generated Test Cases Change Detection report comparing two different Software versions

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
Contact point:	▪ Sorin Patrasoiu, sorin.patrasoiu@withsecure.com
Condition(s) for reuse:	▪ Apache License 2.0

Latest update: November 23, 2022

Name: Anomaly Detector

Input(s):	Main feature(s)	Output(s):
▪ 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 ○ 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 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