IML4E

Industrial Machine Learning for Enterprises

Deliverable D3.5

Second version of tools for advanced model engineering

Project title:	IML4E
Project number:	20219
Call identifier:	ITEA AI 2020
Challenge:	Safety & Security

Work package:	WP3
Deliverable number:	D3.4
Nature of deliverable:	Report
Dissemination level:	PU
Internal version number:	1.0
Contractual delivery date:	2024-05-31
Actual delivery date:	2024-07-17
Responsible partner:	University of Helsinki

Contributors

Editor(s)	Mikko Raatikainen (University of Helsinki)
Contributor(s)	Dorian Knoblauch, Abhishek Shresta, Marek Feldo (Fraunhofer Fokus), Janis Lapins (Spicetech), Mikko Raatikainen (University of Helsinki),
Quality assuror(s)	Harry Souris (Silo AI), Jürgen Grossman (Fraunhofer Fokus)

Version history

Version	Date	Description
1.0	24-07-17	Version for publication

Abstract

This document describes the second version of the tools for advanced model engineering. It revisits the earlier tools that require updating and introduces new tools. These tools complement the methods and techniques covered in deliverable D3.4.

Keywords

MLOps, Model engineering

Executive Summary

This document describes the second version of tools for advanced model engineering. It revisits the earlier tools that require updating and introduces new tools. The tools include the autonomously adaptive experimentation-driven pipeline tools, the data and model monitoring dashboard, the adversarial test toolbox, VALICY, and the model cards toolbox. This document serves as a complementary description of these tools, which are all software tools. We briefly introduce and summarize the tools using a common technology sheet format. The documentation for each individual tool provides a more detailed technical description.

Table of contents

1 INTRODUCTION ... 6

1.1 Role of this Document ... 6
1.2 Intended Audience ... 6
1.3 Definitions and Interpretations ... 6
1.4 Applicable Documents ... 6

2 AUTONOMOUSLY ADAPTIVE EXPERIMENTATION-DRIVEN PIPELINE TOOLS ... 7

Usage Instructions ... 8

3 DATA AND MODEL MONITORING DASHBOARD ... 9

Usage Instructions ... 9

4 ADVERSARIAL TEST TOOLBOX ... 10

Usage Instructions ... 11

5 VALICY ... 13

Usage Instructions ... 13

6 MODEL CARDS TOOLBOX ... 15

Usage Instructions ... 15

7 SUMMARY ... 16

1 Introduction

1.1 Role of this Document

The purpose of this document is to provide a complementary description of the second version of tools for advanced model engineering in the IML4E project. Detailed technical descriptions for each tool are provided in, e.g., their respective GitHub README files. These tools are software tools developed within the project. The methodology for applying the tools is covered in deliverable D3.4. This document focuses on ML model engineering and quality assurance, paralleling the data engineering-focused deliverables of work package 2 in the IML4E project.

1.2 Intended Audience

The intended audience of the present document is composed primarily of the IML4E consortium for the purpose of understanding the tools and advancing ML model engineering. However, this document is public and can provide an overview of the advances in the IML4E project to wider audience. This document describes tools for the technically oriented audience rather than the general public or layman.

1.3 Definitions and Interpretations

The terms used in this document have the same meaning as in the contractual documents referred in [FPP] with Annexes and [PCA] unless explicitly stated otherwise.

1.4 Applicable Documents

Reference	Referred document
[FPP]	IML4E – Full Project Proposal 20219
[PCA]	IML4E Project Consortium Agreement
{D3.4]	Second version of methods and techniques for advanced model engineering

Table 1: Contractual documents.

2 Autonomously Adaptive Experimentation-Driven Pipeline Tools

General Information	
Title	Autonomously Adaptive Experimentation-Driven Pipeline tools
Partners	University of Helsinki
Research area(s)	Life cycle
Description	A fully automated MLOps pipeline can be autonomously adaptive and experimentation-driven to maintain the model's performance in changing conditions. Autonomous includes continuous training (CT) by automatic model retraining and continuous deployment (CD) by automatically deploying retrained models to production. Retraining is triggered periodically or by model monitoring results or repository updates. In addition, the pipeline conducts experimentation by A/B testing before promoting a better model to serve all requests. The tools include necessary additional tools for the IML4E OSS platform.
Innovation	☐I1: High quality and interoperable data preparation infrastructures for trustworthy ML ☐I2: Scalable MLOps techniques and tools for critical application domains ☒ I3: An MLOps Methodology ☒ I4: An experimentation and training platform ☐I5: Pre-standardization work on cross-domain engineering for AI-systems
Related KPIs	☒ ML service and process automation ☒ Increased service delivery capability/new products ☐Human or/and computational resources ☐Effectiveness of data usage ☒ Finding defects
Business Impact	☐ New AI enabled services ☒ Fast and efficient deployment of ML products and services ☒ Increased trust in AI enabled products and services ☐ New MLOps consulting service
Impact	Open access and source releases.
Technology Environment	Built on IML4E OSS platform.
Synergies	IML4E OSS platform. Autonomously Adaptive Experimentation-Driven Pipeline approach (D3.5)
Access	☐Proprietary/Confidential ☒ Open source/access: MIT
Links	https://version.helsinki.fi/luoyumo/ctcd-e-mlops-pipeline

Usage Instructions

The prerequisite, installation, and usage instructions are detailed in the GitLab repository defined above. The tools are designed and tested in a Linux environment where the specified additional open-source tools are required.

Detailed additional configuration instructions are given for Prometheus and Grafana, as well as self-implemented Joiner, Monitor, Retraining-Triggering Webhook, and Model Comparison Runner.

The repository includes a sine experiment source code and a video demonstrating running the sine experiment.

3 Data and model monitoring dashboard

General Information	
Title	Data and model monitoring dashboard
Partners	Granlund, Software AG
Research area(s)	ML application monitoring and maintenance
Description	The data and model monitoring dashboard is a service that supports machine learning systems working on a large number of models. It is built on Grafana and displays crucial information about model performance, drifts, and other metrics. Data monitoring helps to understand the data and minimize the negative impact on the service. The dashboard also includes infrastructure monitoring, providing information about workflows and resources in production. It is a valuable tool for ensuring the proper function of machine learning systems. The work was aided by Software AG by study of model drift method
Innovation	☒I1: High quality and interoperable data preparation infrastructures for trustworthy ML ☐I2: Scalable MLOps techniques and tools for critical application domains ☐I3: An MLOps Methodology ☐I4: An experimentation and training platform ☐I5: Pre-standardization work on cross-domain engineering for AI-systems
Related KPIs	☒ ML service and process automation ☐Increased service delivery capability/new products ☐Human or/and computational resources ☐Effectiveness of data usage ☒ Finding defects
Business Impact	☐ New AI enabled services ☐ Fast and efficient deployment of ML products and services ☒ Increased trust in AI enabled products and services ☐ New MLOps consulting service
Impact	It helps with monitoring and fault detection of ML models, allowing for timely intervention and resolution of issues. This reduces downtime and improves customer satisfaction. Impact isn't quantifiable
Technology Environment	Grafana, EvidentlyAI, Prometheus, MLflow
Synergies	WP2
Access	☒ Proprietary/Confidential ☐ Open source/access
Links	

Usage Instructions

Data and model monitoring dashboard is Granlund’s internal tool with proprietary information.

4 Adversarial Test Toolbox

General Information	
Title	Adversarial Test Toolbox
Partners	Fraunhofer (DEU)
Research area(s)	Model Adversarial Robustness Assessment
Description	The Adversarial Test Toolbox provides in-depth assessment of adversarial robustness of object detection models. The tool enables users to use a variety of algorithms to generate powerful attacks and apply them to the target models in both white-box and black-box scenarios. Given the usability threats posed by adversarial vulnerability of deep learning models, we use our recent research results on adversarial transferability to develop the automated tool to test models against transfer-based attacks. The tool supports multiple object detection models and attack algorithms.
Innovation	☐I1: High quality and interoperable data preparation infrastructures for trustworthy ML ☒I2: Scalable MLOps techniques and tools for critical application domains ☐ I3: An MLOps Methodology ☐I4: An experimentation and training platform ☐I5: Pre-standardization work on cross-domain engineering for AI-systems
Related KPIs	☒ ML service and process automation ☐Increased service delivery capability/new products ☐Human or/and computational resources ☐Effectiveness of data usage ☒ Finding defects
Business Impact	☐ New AI enabled services ☐ Fast and efficient deployment of ML products and services ☒ Increased trust in AI enabled products and services ☐ New MLOps consulting service
Impact	By identifying vulnerabilities in deep learning models, the toolbox helps improve the security and robustness of AI systems, reducing the risk of adversarial attacks in real-world applications.
Technology Environment	Windows/UNIX-based os with Python (>3.10.8) and PyTorch 2.2.1
Synergies	PipelineProbe
Access	☒Proprietary/Confidential ☐ Open source/access
Link	https://iml4e.org/en/iml4e/toolbox https://gitlab.fokus.fraunhofer.de/ml-cse/adversarial_test_toolkit (restricted access only)

Usage Instructions

Prerequisites

It is recommended to use newer version of Python (>=3.10.8) and Pytorch (>=2.2.1).

Installation Steps

1. **Download the Tool**:
   - Obtain the latest version of the Adversarial Test Toolbox from the official repository at: https://gitlab.fokus.fraunhofer.de/ml-cse/adversarial_test_toolkit
2. **Install Dependencies**:
   - Install Python if not already installed: Download from python.org and follow the installation instructions for your OS.
3. **Set up the Environment**:
   - Create a virtual environment to avoid conflicts with other packages:

     ```
     python -m venv attacktoolbox
     ```

   - Activate the environment:

     ```
     source attacktoolbox/bin/activate # On Windows use
     `dqevaltool\Scripts\activate`
     ```

   - Install required Python libraries:

     ```
     pip install -r requirements.txt
     ```

Using the tool:

1. **Configuration:**
   - Use the config.yml file in the repository to provide necessary configuration before running the tool.

     This configuration file can be used to set global configurations which include:
     - the location of the trained model (source) to be used to create adversarial examples,
     - adversarial attack algorithm, number of samples to create (selected at random from the source dataset), model where the attack is to be applied (target), and
     - dataset to use for creating the examples.

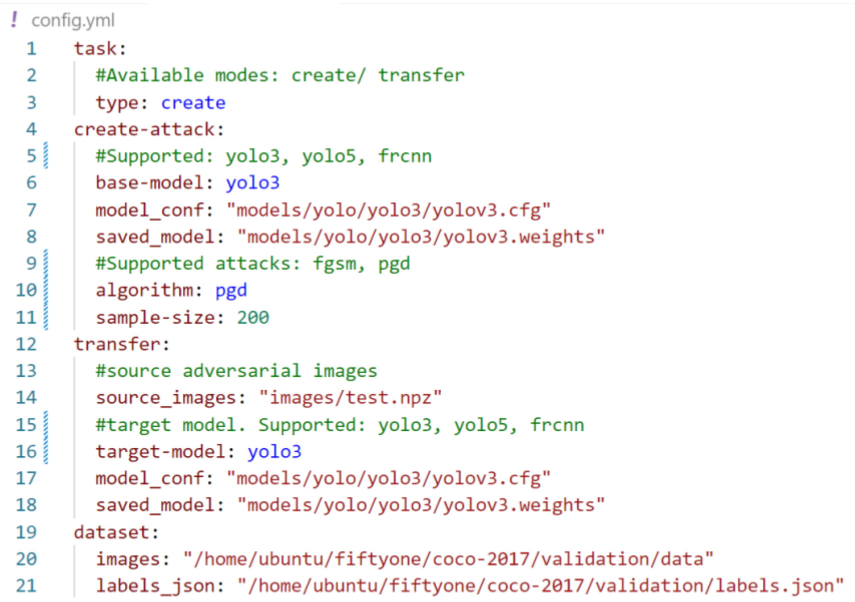
     Further, the tool can be run on either “create” or “transfer” mode.
2. **Execution:**
   a. First create adversarial samples by running the tool in “create” mode:

      ```
      python begin_exp.py
      ```

      This creates and saves adversarial examples in .npz format inside the *logs* folder.

   b. To apply the created samples on a defined target model, set the mode as “transfer”. Give the location of the saved images and execute the tool.

      Depending on the attack algorithm and the number of samples, it may take some time to complete the assessment.
3. **Practical example:**
   - An example configuration of the tool where an attack algorithm called the Projected Gradient Descent (PGD) (https://arxiv.org/abs/1706.06083) is used is shown below. Both source and target models are Yolo3. 200 random samples from COCO datasets were used to create adversarial samples. We compute mAP (mean average precision) on both clean and resulting adversarial samples.



config.yml

```yaml
task:
  #Available modes: create/ transfer
  type: create
create-attack:
  #Supported: yolo3, yolo5, frcnn
  base-model: yolo3
  model_conf: "models/yolo/yolo3/yolov3.cfg"
  saved_model: "models/yolo/yolo3/yolov3.weights"
  #Supported attacks: fgsm, pgd
  algorithm: pgd
  sample-size: 200
transfer:
  #source adversarial images
  source_images: "images/test.npz"
  #target model. Supported: yolo3, yolo5, frcnn
  target-model: yolo3
  model_conf: "models/yolo/yolo3/yolov3.cfg"
  saved_model: "models/yolo/yolo3/yolov3.weights"
dataset:
  images: "/home/ubuntu/fiftyone/coco-2017/validation/data"
  labels_json: "/home/ubuntu/fiftyone/coco-2017/validation/labels.json"
```

The results are then stored in a json file. Contents as below (for this run):

{"source_images": "logs/images.npz", "target_base_model": "yolo3", "target_model_location": "models/yolo/yolo3/yolov3.weights", "mAp_on_clean_images": 0.4095084983498349, "mAp_on_adversarial_images": 0.23793140814081398}

4. Best Practices:

- **Regular Updates**: Regularly update the Adversarial Test Toolbox to benefit from the latest features and bug fixes.
- **Data Backup**: Always back up your data from logs (saved images and results) before new runs.
- **Documentation**: Maintain thorough documentation of all assessments to ensure traceability and repeatability.

5. Troubleshooting:

- The application should run without any problems if all the dependencies are installed as per the requirements.txt file in the repository. In the case of errors, please make sure that the versions of the Python packages and Python are as recommended in this guide.

5 VALICY

General Information	
Name	VALICY – a tool for virtual validation of AI & complex software applications
Provider(s)	Spicetech GmbH
Topic(s) Covered	Virtual validation of AI & complex software application, training of state dependent field data to train an AI model for prediction of states
Description	An AI core that runs different competing AI instances to train from application data and drive the test proposals of input parameters towards critical parameter conditions close to the decision boundarie(s) of the application under test, thereby identifying characteristics. With an increasing number of evaluated results trained by AI models, the AIs within VALICY always improve their own prediction capabilities. The estimated remaining uncertainty of the sampled multi-dimensional space is provided as a stop criterion for VALICY jobs, along with the number of evaluated runs. Data to and from the AI application is stored in a database and transferred via a REST-API. For ease of data transfer, an additional API class writes results using pandas.DataFrame via the API. The frontend allows inspecting the results.
Innovation	☐I1: High quality and interoperable data preparation infrastructures for trustworthy ML ☒I2: Scalable MLOps techniques and tools for critical application domains ☐I3: An MLOps Methodology ☐I4: An experimentation and training platform ☒I5: Pre-standardization work on cross-domain engineering for AI-systems
Related KPIs	☒ML service and process automation ☐Increased service delivery capability/new products ☒Human or/and computational resources ☐Effectiveness of data usage ☒Finding defects
Business Impact	☒ New AI enabled services ☒ Fast and efficient deployment of ML products and services ☒ Increased trust in AI enabled products and services ☐ New MLOps consulting service
Examples (Use Cases)	The VITAREX Pose Estimation Use Case was successfully integrated to VALICY within the course of the IML4E Plenary meeting in Budapest in November 2022 and was further refined for the pose estimation use case and published 2024.
Technology Environment	Python machine learning, MySQL, Docker, REST-API, Swagger
License	☐Open Source ☒ Proprietary
Link	https://Valicy.de , API: https://api.valicy.de/docs, https://github.com/SpicetechGmbH/Valicy-Interface-Example

Usage Instructions

The "Usage Instructions" section will include the following information for each tool:

- Prerequisites: Any necessary prerequisites or dependencies required to use the tool effectively.
- Installation: Step-by-step instructions on how to install and set up the tool.

- Configuration: Details on how to configure the tool for optimal performance and customization.
- Examples: Practical examples demonstrating how to use the tool in real-world scenarios.

□ Best Practices: Tips and best practices for utilizing the tool efficiently and avoiding common pitfalls.

Prerequisites:

- AI application to test
- Python > 3.7
- Contact team@valicy.de to get a user account with an API key

Installation:

- Go to https://github.com/SpicetechGmbH/Valicy-Interface-Example to download the example for the virtual validation with VALICY
- Integrate the results coming from your AI applications to above VALICY-Interface-Example

Configuration:

- Start with a moderate certainty (e.g. 0.8) as VALICY stop criterion
- Start with a moderate number of test points (e.g. 10 – 20 k)
- To see how VALICY samples the test space and make sense of the results
- On your side: VALICY is a tool that samples your test space, the more input parameter dimensions, the longer the response time: the response to the request may take a while (in the orders of seconds)

Examples:

- An explanation in combination with a toy sample is provided on this GitHub page: https://github.com/SpicetechGmbH/Valicy-Interface-Example
- API key can be obtained through mail request

Pitfalls:

- Be patient: Do not expect an immediate answer from VALICY from the API. Rather, re-try after a waiting time. The more complex the problem under validation the longer the waiting time (in range of seconds) for a response, although it is mostly in the beginning when there is little information about the test space.

6 Model Cards Toolbox

General Information	
Title	Model cards toolbox
Partners	University of H
Research area(s)	Model engineering
Description	Model cards toolbox is a set of tools integrated to GitHub actions to create, validate and visualize model cards semi-automatically. Model cards are ledgers for model-related information, such as performance tests or measures and their results for ethical concerns, in a machine-readable form that can be rendered to suitable presentation for different stakeholders, including the non-technical audience.
Innovation	☐I1: High quality and interoperable data preparation infrastructures for trustworthy ML ☒ I2: Scalable MLOps techniques and tools for critical application domains ☒ I3: An MLOps Methodology ☐I4: An experimentation and training platform ☐I5: Pre-standardization work on cross-domain engineering for AI-systems
Related KPIs	☒ ML service and process automation ☐Increased service delivery capability/new products ☐Human or/and computational resources ☐Effectiveness of data usage ☐ Finding defects
Business Impact	☐ New AI enabled services ☐ Fast and efficient deployment of ML products and services ☒ Increased trust in AI enabled products and services ☐ New MLOps consulting service
Impact	Improved quality
Technology Environment	Model card representation in YAML. GitHub Actions
Synergies	CABC
Access	☐Proprietary/Confidential ☒ Open source/access: MIT
Link	https://github.com/CompliancePal/modelcard-action https://helda.helsinki.fi/items/df04410c-2b48-4f32-88c5-bfcaae4f6cae

Usage Instructions

The model card toolbox works relying on GitHub actions. The detailed instructions are given in the GitHub repository. A detailed description is in the master thesis in the above link.

7 Summary

The primary objective of WP3 was to develop methods, techniques, and tools for various industrial machine learning use cases in ML model engineering. This document outlines advancements in the tools, including the autonomously adaptive experimentation-driven pipeline tools, the data and model monitoring dashboard, the adversarial test toolbox, VALICY, and the model cards toolbox. This document serves as a complementary description of these tools, which are all software tools. The methods and techniques for the tools are described in D3.5 while access to each tool is provided, e.g., by GitHub link if openly available. These tools complement the overall MLOps methodology and framework defined in the IML4E project.