Industrial Machine Learning for Enterprises

Deliverable D2.3

First Version of Tools for Data Collection, Processing, and Valorisation

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

Work package:	WP2
Deliverable number:	D2.3
Nature of deliverable:	Report/Prototype
Dissemination level:	PU
Internal version number:	1.0
Contractual delivery date:	2022-11-30
Actual delivery date:	2022-12-05
Responsible partner:	Software AG

Contributors

Editor(s)	Mohamed Abdelaal (Software AG)
Contributor(s)	Mohamed Abdelaal (Software AG), Timo Sinisalmi (Basware), Luca Szegletes (BUTE), Dorian Knoblauch (Fraunhofer), Heikki Ihasalo (Granlund), Anna Korolyuk (Granlund), Harry Souris (Silo AI), Kimmo Sääskilahti (Silo AI)
Quality assuror(s)	Lalli S Myllyaho (University of Helsinki), Gabor Gulyas (Vitarex)

Version history

Version	Date	Description
1.0	22-12-05	Version for publication

Abstract

The deliverable provides an overview of the tools and technologies developed within IML4E WP2. The deliverable covers the tools which serve the objectives of the main three tasks in WP2, including data preparation automation (Task 2.1), data management and version control (Task 2.2), and continuous data quality assurance (Task 2.3).

Keywords

Data preparation, data valorisation, anomaly detection, data version control, data quality assurance, ML certification, quality metrics, data cleaning, data privacy

Executive Summary

The overall objective of IML4E WP2 is to develop tools and techniques to realize a data layer according to the requirements imposed by the different use cases. In this deliverable, we present the WP2 tools and techniques that have been developed during the first year of the project.

The set of tools and techniques are as follows:

- The automated detection tool from Software AG, supporting the automatic detection of errors in tabular data
- The data quality dashboard from Granlund Oy, which is a tool designed to continuously monitor the quality of large data volumes.
- The Mosquito data cleaner from Basware Oyj, which automatically performs error detection on data extracted from digital invoices.
- The data version control tool from Silo AI, which is a version control solution that can flexibly serve different use cases and different data modalities.
- The continuous audit-based certification (CABC) tool from Fraunhofer FOKUS, supporting the continuous data quality assurance.

Table of contents

1 INTRODUCTION 6

2 CHARACTERISTICS OF THE IDENTIFIED WP2 TOOLS 7

2.1 META LEARNING-BASED TABULAR DATA ERROR DETECTOR 7
2.2 DATA QUALITY DASHBOARD 8
2.3 MOSQUITO DATA CLEANER 9
2.4 PRIVACY-FRIENDLY HUMAN POSE ESTIMATOR 10
2.5 CABC (DATA QUALITY ATTRIBUTES) 11

3 DESCRIPTION OF THE IDENTIFIED WP2 TOOLS 12

3.1 AUTOMATED ERROR DETECTION 12
3.1.1 Planned Features 12
3.1.2 Current Status 12
3.2 DATA QUALITY DASHBOARD 12
3.2.1 Planned Features 13
3.2.2 Current Status 13
3.3 MOSQUITO DATA CLEANER 13
3.3.1 Planned Features 13
3.3.2 Current Status 14
3.4 PRIVACY-FRIENDLY HUMAN POSE ESTIMATOR 15
3.4.1 Planned Features 15
3.4.2 Current Status 15
3.5 DATA VERSION CONTROL 16
3.5.1 Planned Features 16
3.5.2 Current Status 16
3.6 CABC (DATA QUALITY ATTRIBUTES) 16
3.6.1 Planned Features 16
3.6.2 Current Status 17
3.6.3 Syntactic data accuracy 17
3.6.4 Semantic data accuracy 18
3.6.5 Attribute completeness 18
3.6.6 Risk of Data inconsistency 19

4 SUMMARY 20

REFERENCES 21

1 Introduction

This report represents a documentation of our tool development efforts and an introduction to the prototypes developed within the IML4E WP2. The prototypes have been implemented to realize the methods and techniques that are developed in this work package. The reader is referred to deliverable D2.2 for an overview of the WP2 research results after the first year of the IML4E project.

The prototypes introduced, in this report, serve as a part of the IML4E toolbox for data preparation, data version control, and continuous data quality monitoring, which will be utilized by the advanced model engineering methods developed in the IML4E WP3. In WP3, we develop and implement techniques and tools for easier, faster, and more automated ML operations in development, deployment, and operational stages. It is important to mention that the tools developed in WP2 and WP3 will be integrated in the IML4E MLOps platform which enables ML engineers and practitioners to readily develop and deploy their ML-powered applications.

The tools that are presented in the next two chapters comprise: meta learning-based error detection for tabular data, data quality dashboard, Mosquito data cleaner for structured data extracted from invoices, privacy-friendly image processing for pose estimation, data version control, and continuous audit-based certification (CABC). In Chapter 2, we provide a brief overview of the main characteristics of the tools, and in Chapter 3, we present the tools in some more details. Specifically, we explain the planned features and the current status for the development of the reported tools. Finally, we present a summary in Chapter 3.

2 Characteristics of the Identified WP2 Tools

In this section, we provide an overview of various tools developed within WP2 of the IML4E project.

2.1 Meta Learning-Based Tabular Data Error Detector

General Information	
Name	Meta learning-Based Error Detection
Provider(s)	Software AG
Topic(s) Covered	Task 2.1: Data Preparation Automation
Description	This technology formulates the task of error detection in tabular data as a classification problem. Meta learning is adopted to transfer knowledge from a set of historical dirty datasets to new dirty datasets, i.e., the datasets to be cleaned. Specifically, the technology consists of two modules, a *knowledge gathering* module and a *detection* module. The former module trains a set of ML models to identify errors in the historical datasets. The latter module matches the new dirty dataset with a set of the historical datasets, before using the corresponding models to generate the feature vector for the meta classifier.
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)	Examined with the craft beers data set (shorturl.at/eyQV6), the individual income tax data set (shorturl.at/ejuMV), and the smart factory data set (shorturl.at/iFW09)
Technical Information	
OS	Linux
Technology Environment	Python3
Synergies (Other Tools)	Data Quality Dashboard
Additional Information	
License	☒Open Source ☐Proprietary
Link	https://github.com/mohamedyd/SAGED

2.2 Data Quality Dashboard

General Information	
Name	Data quality dashboard for continuous monitoring of large data volumes
Provider(s)	Granlund
Topic(s) Covered	Task 2.1: Data Preparation Automation
Description	The dashboard summarizes data quality issues in KPIs (e.g., percentage of abnormal values). It highlights the key data points and aids in managing large data volumes.
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)	Use case in built on energy consumption data
Technical Information	
OS	Linux, Windows
Technology Environment	The end result is a report that could be built on several technologies. Proof of concept is built using Azure tools and Power BI reporting.
Synergies (Other Tools)	Meta Learning-Based Tabular Data Error Detector
Additional Information	
License	☐Open Source ☒Proprietary
Link	NA

2.3 Mosquito Data Cleaner

General Information	
Name	Automatic removal of human errors from dataset with anomaly detection technique
Provider(s)	Basware Oy
Topic(s) Covered	Task 2.1: Data Preparation Automation
Description	This task is meant for removing human errors originating from wrong manual data extraction from invoices. The automatic error removal is done by a family of algorithms called Mosquito. The original generation of Mosquito algorithms, G1, considered invoice as a bag-of-words. In this task, two new generations of Mosquito algorithms were implemented, G2 and G3. G2 considers invoice as a bag or geometrical blocks and G3 uses Two-Dimensional Stochastic Context Free Grammar (2D-SCFG) to parse the invoice and deeply understand its structure including labels, values, tables, line items, and street addresses. This approach helps producing rich and descriptive signature of the invoice, which is then used for invoice clustering and field identification. Once the fields are identified, we use the training data to map the field names with majority voting technique. This way Mosquito learns to extract the data from invoice. Then Mosquito is re-applied to its own training set, and all mismatches are considered anomalies. The sample weight for anomalous sample is lowered during the main model training.
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)	New training iteration was run with 10M invoices. Significantly more efficient clustering was observed. Main model coverage increased by 5% and accuracy increased by 1%.
Technical Information	
OS	Linux
Technology Environment	Python3
Synergies (Other Tools)	Mosquito GT Refinery UI
Additional Information	
License	☐Open Source ☒Proprietary
Link	NA

2.4 Privacy-friendly Human Pose Estimator

General Information	
Name	Privacy-friendly Image Preparation for AI Pipelines
Provider(s)	Budapest University of Technology and Economics
Topic(s) Covered	Task 2.1: Data Preparation Automation
Description	In this task, we develop a processing pipeline for images. This processing pipeline involves facial anonymization, and it is adopted to pose estimation. In pose estimation the person 's movement are tracked by finding the location of a set of selected keypoints. Since the introduction of GDPR by the EU in early 2018, privacy protection became an indispensable task using personal data. The regulation leaves space for non-consensual use of images if the individual is unrecognizable. In the processing pipeline we remove facial information from images. However, for our use case, facial keypoints have a great information value. They are not negligible and needed to be recognized for accurate pose estimation. The preprocessing pipeline needs to eliminate personal information from images and still keep the keypoints for further analysis.
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)	Use case for pose estimation
Technical Information	
OS	Linux
Technology Environment	Python, Pytorch
Synergies (Other Tools)	
Additional Information	
License	☒Open Source ☐Proprietary
Link	TBA

2.5 CABC (Data Quality Attributes)

General Information	
Name	Data Quality Test Suite
Provider(s)	Fraunhofer Fokus
Topic(s) Covered	Data Quality, Automated Testing, CABC
Description	This test suite loads a dataset and performs a number of tests on the dataframe. In the current state it computes four data Quality Measures (QMs) defined in the ISO/IEC 25024 standard. Namely: Syntactic data accuracy, Semantic data accuracy, Attribute completeness and Risk of data inconsistency. Additionally, it is capable of providing a histogram of the labels. In the future, the data quality test suite will be extended with more measurements and will cover more aspects of the ISO/IEC 25024 measurements. It's intended to be used as a standalone tool as well as in the context of CABC, therefore it provides the data points for the calculation.
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)	
Technical Information	
OS	Python, Fiftyone
Technology Environment	Python3
Synergies (Other Tools)	CABC
Additional Information	
License	☒Open Source ☐Proprietary
Link	https://gitlab.fokus.fraunhofer.de/ml-cse/datatestsuite(Invite on request: dorian.knoblauch@fokus.fraunhofer.de)

3 Description of the Identified WP2 Tools

3.1 Automated Error Detection

In the context of IML4E WP2, the automated error detection tool is used to enhance the level of automation while preparing tabular data in ML pipelines. The tool and its core principles are introduced in the deliverable D2.2. Therefore, in this deliverable, we focus on the planned features and the current status of the automated error detection tool.

3.1.1 Planned Features

The following features are planned to be developed and implemented within the IML4E WP2:

- Design time knowledge: A database of historical tabular data which has been previously cleaned either using other data cleaning tools or cleaned by subject matter experts. This knowledge serves as a valuable asset which can be used to enhance the cleaning process of the newly collected data.
- Similarity measures: a tool to measure the similarity between the input dirty data sets and the historical data. Based on these measures, a set of meta features will be generated
- Labelling budget: To train a meta classifier, users have to label a set of data instances. However, this labelling process may be overwhelming if we are dealing with large volumes of data. Hence, several techniques can be used to reduce the labelling budget, including active learning and semi-supervision.
- Self-supervision: To further increase the level of automation, we seek to avoid any user intervention. To this end, we plan to exploit self-supervision models to detect the errors in tabular data.
- Data valuation: ML-based error detectors typically suffer from poor scalability. As a workaround to combat the scalability problems, we intent to valuate the input data instance to clean only the most important data instances.

3.1.2 Current Status

The automated error detection tool is currently able to detect errors with high accuracy, compared to the state-of-the-art data cleaning methods. However, the tool currently relies on clustering to find the matching historical data sets. We examined two clustering algorithms, namely K-Means clustering and Hierarchical clustering. The latter method is found to be more accurate. However, adopting clustering seems to be a workaround, which can still be improved. For the labelling budget, we examined active learning and self-supervision, and random sampling. Surprisingly, the random sampling methods achieve good and stable results together with minimum overhead. To achieve all the above mentioned features, we still need to refine and adopt the features. Hence, the main tasks in further developing the automated error detection tool in WP2 can be summarized as follows:

- Extending the design time knowledge by adding more historical datasets. In the one hand, this extension will further improve the detection accuracy. On the other hand, it will require more time to perform the similarity measure process. Therefore, clustering of the historical data sets is needed to combat the complexity of searching the entire historical database.
- Integrating a data valuation method to the data cleaning method. To this end, we carried out a microbenchmark of the available data valuation methods.
- Examining the cosine similarity as a correlation metric between the historical data sets and the input dirty data sets. Other similarity measures, such as the Jaccard similarity, can also be examined.

3.2 Data Quality Dashboard

This section describes the development work related to the data quality dashboard. It is designed to be used in continuous data quality monitoring of large data volumes.

3.2.1 Planned Features

The use case for the data quality dashboard is energy consumption data of hundreds or thousands of buildings. In energy data, the typical data quality issues are related to missing values and abnormal values. For missing values, existing commercial tools provide solutions, but for abnormal values more advanced methods are needed. Abnormal values can be meter jams to certain consumption level or changes in a typical energy consumption level. In addition to finding abnormal values, the plan is to evaluate the suitability of the data to energy prediction purposes. Predictability is improved if there are no changes in the consumption pattern and the correlation between outside temperature, day of the week and consumption is similar during the analysed time period. The goal is to develop quality and suitability detection methods, present the results of the methods in KPIs (e.g., percentage of abnormal values) and visualize KPIs in a dashboard.

3.2.2 Current Status

We have built the infrastructure and data pipelines to visualize data quality issues in a dashboard. For the dashboard we use Power BI. We can visualize statistical information of the data and have started gathering user requirements for the dashboard. During the next reporting period, we plan to develop data quality KPIs to the dashboard based on the user requirements.

To detect abnormal values in the data, we have tested several methods, for example neural network, ARIMA and Random Forest algorithms. Based on our experiments neural network algorithms required huge computational resources and long calculation times. More suitable methods were ARIMA and Random Forest with Random Forest giving better accuracy. We have tried several features and the latest algorithm uses outside temperature, time variables and previous consumption. Time variables are related to cyclic nature of the building usage. For example, weekday and weekend will differ in terms of consumption as well as hour of the day. Results on detecting abnormal consumption are promising and we continue in improving the algorithm to enhance the accuracy.

For the suitability of the data to energy prediction, we have tested method called energy signature. The method gives the correlation of building energy use with outdoor climatic variables. We have also tried dividing the results into office hours (7 am to 18 pm) and night-time hours to see the differences in energy usage. In our test dataset, we could find differences in energy signature results between buildings. We will research in more detail, what are the reasons behind these differences and how well energy signature works in evaluating the suitability of the data for ML purposes.

In the future, we will improve the methods developed so far, research techniques in finding pattern changes in the data and develop method results in to KPIs and a dashboard.

3.3 Mosquito Data Cleaner

3.3.1 Planned Features

Two more generations of Mosquito, the G4 and G5, are planned, both requiring intense scientific research. The weakness of the current version G3 is that it is based on programmable heuristics rather than machine learning. This prevents optimization of the model and tuning the weights and probabilities automatically with gradient descent. Manual tuning of heuristics parameters is time consuming and unreliable. This is why we consider G3 more like a proof of concept rather than a final solution.

The G4 is based on machine learning approach, which uses the output of G3 as a training data. The model used behind G4L1 is Microsoft LayoutLMv2 open-source multimodality model, which mixes graphical aspects of the invoice with the words extracted with OCR. The model uses BERT embeddings to produce high-quality representations of the invoice labels, which then results in invoice segmentation with five categories of regions: fields, labels, tables, street addresses, other text. The segmentation is then used to produce the invoice signature or embedding.

G4L2 is a clustering model that uses invoice signatures or embeddings provided by G4L1. This can be signature-based clustering similar to G3L2, or otherwise we can consider embeddings-based clustering. Once the clustering is done, the next task is the field identification. This is the most difficult part, as we need to reliably separate all fields within invoice, still keeping the generalization over similar invoices. This task reminds the contrastive learning setup, and we plan to use the contrastive "Barlow Twins" loss to train the unsupervised model for generating field embeddings.

G4L3 is a classical supervised model that uses ground truth data to learn the mappings between fields and their labels using the field embeddings provided by G4L2.

Our next challenge is data efficiency and fast learning. Some training data is coming to our training pipe from so-called "self-validation service", which is based on single-click learning. We promise our customers to learn from as little amount of data as possible. Preferably – single click. Also, the learning time must be reduced to a few minutes. Our customers don't want to wait for the main model to re-train in a couple of weeks, while the old model keeps making the same mistakes that have already been reported.

This is why single-shot learning and continual learning are considered as key research activities within the remaining capacity of WP2, and Mosquito G5 is the version that is planned to provide the final solution, while G3 is considered as a proof of concept. The important feature of G3 is the ability to identify the fields within invoice in an unsupervised manner. Learning this skill is considered as meta-learning or learning-to-learn, because reliable and generalizable field embeddings help memorizing the field-to-label mapping in one shot.

G5 is a ML-based improvement over G3, which provides memorization functionality using neural networks. Our interest is thus concentrated on the neural memory models developed by world leading companies such as Numenta and Cerenaut. Their models are biologically plausible imitations of memory-focused regions of the human brain, such as the hippocampus. Numenta provides the so-called Hierarchical Temporal Memory model (HTM) using the proprietary software. Cerenaut, in contrast, provides the highly biologically plausible model of the hippocampus using classical artificial networks.

We plan to use these approaches in order to build the biologically plausible network mimicking the structure and the functionality of G3. Thus, G5L1 will provide the invoice segmentation and embeddings, G5L2 will be responsible for invoice clustering and field embeddings, and G5L3 will be responsible for single-shot continual memorization of the sparse training data. We hope that Mosquito G5 will be self-sufficient, self-supervised, meta-learning-based, memory-based model featuring one-shot continual learning along with high generalization capability enabling the stability over invoice variations and high coverage of the unpredictable corner cases.

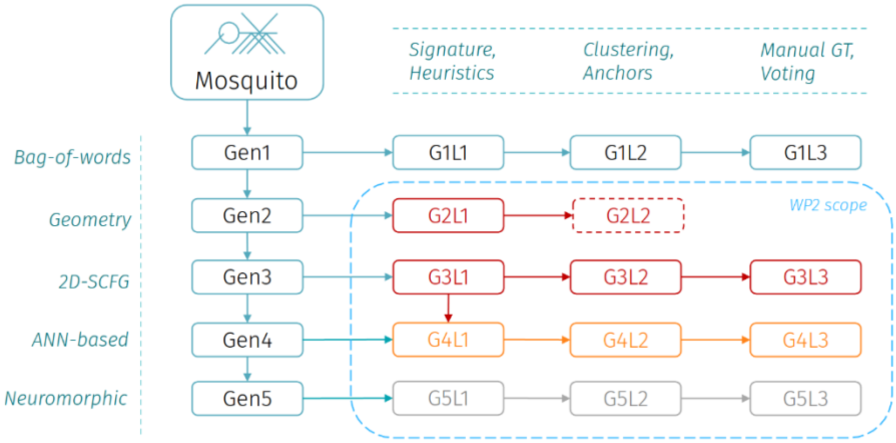

Figure 1. Five generations of Mosquito

3.3.2 Current Status

Mosquito G2 and G3 work is completed. The models are deployed to the training pipe. As a result, the main model coverage is increased by 4% (20% manual cases covered), and the accuracy is increased by 1% (10% of errors removed). The maximum size of the training set is increased from 3M to 20M.

The G4 work has started. Microsoft LayoutLMv2 model has been trained with G3L1 output as a training data. The desired segmentation image has been received.

The G5 studies are started, reading the papers published by Numenta and Cerenaut, listening to lectures about biologically plausible memory models and the continual learning capability of the human brain.

3.4 Privacy-friendly Human Pose Estimator

Pose estimation is an emerging use-case of machine learning which involves predicting peoples pose in images and videos, by predicting keypoints (usually joints in the human body) and their connections. Privacy is also more important than ever: with regulations like the GDPR, people's right to individual privacy is more and more regulated, and taken very seriously by companies and research groups. This section describes our attempt to create a privacy-preserving pose estimation workflow. In section 3.4.1 we elaborate on the planned features, then in section 3.4.2, we describe the current status of the project.

3.4.1 Planned Features

Our expectation for the work we're doing is very simple: anonymizing the workflow of pose estimation, while maintaining accuracy. Our privacy-preserving workflow attains privacy by anonymizing the data that the machine learning model is fed with. By removing facial features from the images, and then applying pose estimation on the already anonymized image, the result of the pose estimation can be considered private (or at least to a certain degree, more on this later…), and can be made public, or used in such ways that would be illegal if the images would contain recognizable individuals.

The workflow needs a pose estimation network to operate with. For this purpose, we used Daniil Osokin's lightweight human pose estimation network, which is open source and available on GitHub[1]. The input of the workflow is the image that needs to be searched for poses, and the output is the anonymized image, with the pose estimation (i.e., annotated with keypoints and their connections). We achieve the anonymization by applying the DeepPrivacy network[2] on the input image. This fully automatic anonymization technique replaces faces in images in a way, that the distribution of the generated face matches the originals. This intuitively means that the keypoints of the replacement faces reside very closely to the ones that they replace, which allow us to get accurate keypoints from the pose estimation model, despite applying it on the modified image.

In the beginning of our work, we experimented with many anonymization techniques, and DeepPrivacy yielded the best results in terms of how much keypoints from the modified image matched the original keypoints. The other methods we tried were:

- using blacked out faces in images,
- applying gaussian blur to the image, and
- pixelation of the face.

3.4.2 Current Status

We have implemented the workflow in Python, and it shows very promising performance. We have created a script that applies the DeepPrivacy network on the input images, a Python script that runs the pose estimation network on input images, and the performance evaluation script, that compares the model's performance on the two image variants (original and privacy-preserving modified).

One area where we have room to improvement, is the validation of our method. We evaluated the performance of the pose estimation to validate the performance of the model on the privacy modified images, but we did not validate the **privacy-preserving performance** of our method. Ideally, we want to guarantee some degree of privacy for people whose photos are processed by our workflow. We consider this our next task, and we are currently looking for ways to measure this privacy guarantee, and thus fully validate our workflow.

[1] https://github.com/Daniil-Osokin/lightweight-human-pose-estimation.pytorch

[2] https://github.com/hukkelas/DeepPrivacy

3.5 Data Version Control

3.5.1 Planned Features

We are in good position to apply the tools to a use-case in IML4E project. We look for collaborating with other partners in IML4E to apply the tools to their projects. We are interested in the use cases provided by Basware and Granlund.

3.5.2 Current Status

We have evaluated the tooling landscape in deliverable D2.2. We have built internal proofs-of-concept with DVC, Delta Lake, Apache Hudi and LakeFS.

Because of the departure of two key contributors, we have made slower progress than we expected.

3.6 CABC (Data Quality Attributes)

3.6.1 Planned Features

As described in D2.2, we are striving to match the data quality attributes laid out in ISO 25012 and the corresponding measurement from ISO 25024 with suitable measurement implementations. In this deliverable we are describing out current implementation of a data test suite. Each high-level quality attribute needs to be assessed via measurements. In the final version, we are planning to assess the quality attributes through the respective measurements listed in **Error! Reference source not found.**.

Table 1: Quality Attributes and corresponding measurements

Quality Attribute	**Measurement**
Accuracy	
	Syntactic Accuracy
	Semantic Accuracy
	Data Accuracy Assurance
	Risk of dataset Inaccuracy
	Metadata Accuracy
	Data Accuracy Range
Completeness	
	Data File Completeness
	Attribute Completeness
	Metadata Completeness
Consistency	
	Referential Consistency
	Data Format Consistency
	Risk of Data Inconsistency
	Data Values Consistency Coverage
	Semantic Consistency
Credibility	
	Values Credibility
	Source Credibility
	Data Dictionary Credibility
	Data Model Credibility
Currentness	
	Update Frequency

	Timeliness of update
	Update Item Requisition
Compliance	
	Regulatory Compliance of Value and/ or Format
Confidentiality	
	Encryption Usages
Efficiency	
	Efficient Data Item Format
Understandability	
	Symbols Understandability
	Data Values Understandability

3.6.2 Current Status

In its current state, the test suite computes four data Quality Measures (QMs) defined in the ISO/IEC 25024 standard. These are:

1. Syntactic data accuracy
2. Semantic data accuracy
3. Attribute completeness
4. Risk of data inconsistency

The Syntactic data accuracy and semantic data accuracy are measures of Accuracy, while the attribute completeness is one of the measures for Completeness. Similarly, risk of data inconsistency is a measure for consistency.

Apart from the four listed above, the app also provides the number of objects in each class. This is helpful in evaluating other data quality characteristics like data imbalance.

The quality checks are performed using the FiftyOne[3] tool. The tool can be used both for evaluation and visualization of data. Using FiftyOne, various Quality Measure Elements (QMEs) are computed, which are then used for measuring the QMs. It is also worth mentioning that the app currently only supports COCO formatted datasets.

3.6.3 Syntactic data accuracy

Duplicate image annotations are used as a QME to measure Syntactic Accuracy. For a given dataset, images with multiple bounding boxes representing the same object are counted. Labelling the same object multiple times (although with same label) is a Syntactic error. Thus, more images with duplicate object annotations means less syntactic accuracy. FiftyOne computes IoUs (intersection over union) for each object of the same class in an image, this can be used to track if same object is labelled twice. Objects with high IoU could mean duplicate labels.

$$\text{Syntactic Accuracy} = \left(1 - \frac{\text{Number of images with duplicate object labels}}{Total\ number\ of\ images\ in\ the\ dataset}\right)$$

[3] FiftyOne — FiftyOne 0.18.0 documentation (voxel51.com)

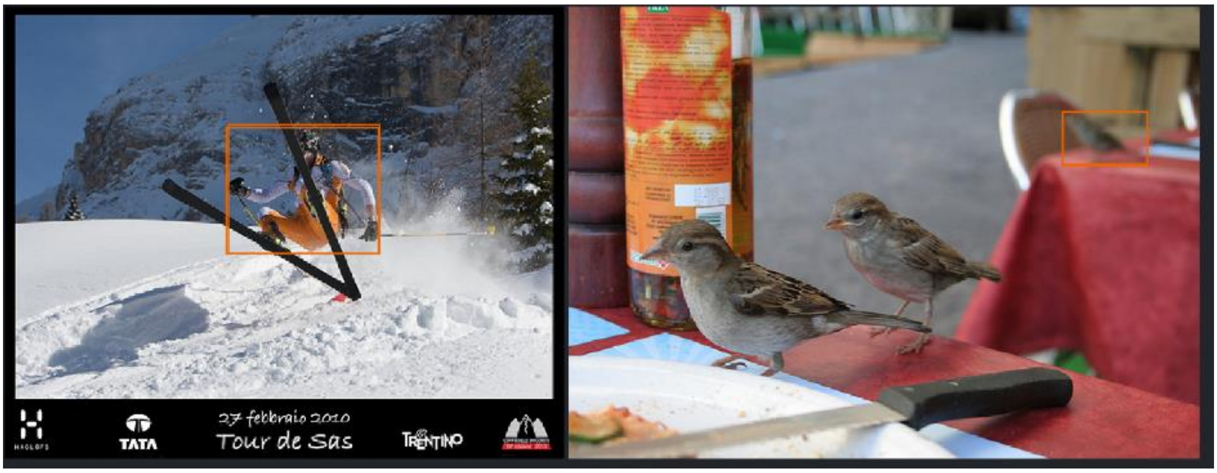

Figure 2. Syntactic Accuracy

3.6.4 Semantic data accuracy

Labelling errors in an image is used as a QME for calculating semantic data accuracy. FiftyOne uses a reference model (PyTorch Faster-RCNN[4]) to make predictions on the given dataset and then compares the predictions with the ground truth (input dataset) to find errors in ground truth labels.

$$Semantic\ Accuracy = \left(1 - \frac{\text{Images with incorrectly labeled objects}}{Total\ number\ of\ images\ in\ the\ dataset}\right)$$

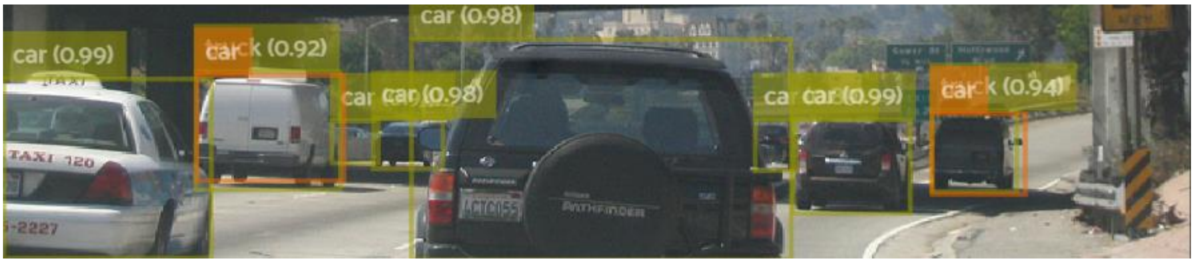

Figure 3: Semantic Accuracy, groundtruth car vs truck predictions, green is predicted and orange is groundtruth

3.6.5 Attribute completeness

The number of missing annotations in the ground truth image is used as a QME for measuring Attribute Completeness. To find missing annotations, FiftyOne uses the F-RCNN reference model as specified above to annotate images in the given dataset. The predicted annotation from the model is then compared with ground truth to find missing annotations in the ground truth.

$$\text{Attribute Completeness} = \left(1 - \frac{\text{Number of records with possbily missing annotations}}{Total\ number\ of\ images\ in\ the\ dataset}\right)$$

[4] vision/faster_rcnn.py at main · pytorch/vision · GitHub

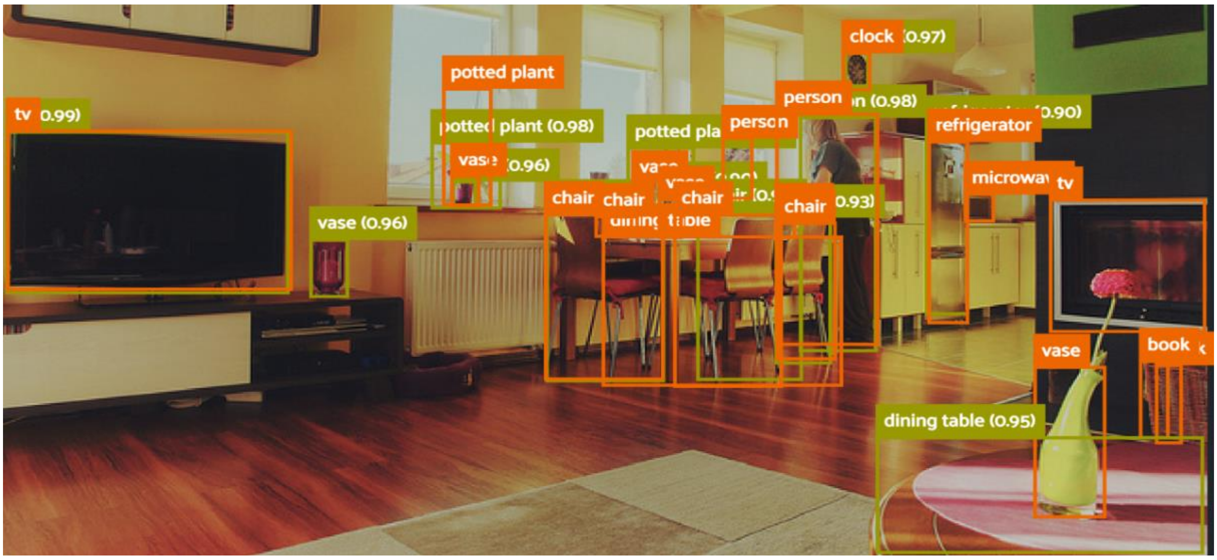

Figure 4: Attribute Completeness, orange bounding boxes are annotation in ground truth, while the green ones are predictions from the reference model

3.6.6 Risk of Data inconsistency

The number of repeated hash values are used to count the number of duplicate images in the dataset. This is done by simply computing image hash and then comparing hash of each image with each other. This gives the absolute duplicate of images (if any in the dataset). This QME is then used to compute Risk of Data inconsistency QM.

$$\text{Attribute Completeness} = \left(1 - \frac{\text{Number of duplicate images}}{Total\ number\ of\ images\ in\ the\ dataset}\right)$$

4 Summary

In this report, we have given an overview of the IML4E WP2 tools that are delivered as prototype deliverable D2.3. In the context of WP2, these tools have been developed to support the techniques and methods of this work package for data collection, processing and valorisation. Together with the methodologies developed and implemented in WP3, the WP2 tools are planned to support the IML4E MLOps platform.

The tools presented in this deliverable are meta learning-based error detection for tabular data (Software AG, Germany), data quality dashboard (Granlund Oy, Finland), Mosquito data cleaner for structured data extracted from invoices (Basware Oyj, Finland), privacy-friendly image processing for pose estimation (Budapest University of technology and economics, Hungary), data version control (Silo AI, Finland), and continuous audit-based certification (CABC) (Fraunhofer FOKUS, Germany).

References

Hukkelås, H., Mester, R. and Lindseth, F., 2022. *DeepPrivacy: A Generative Adversarial Network for Face Anonymization*.