Industrial Machine Learning for Enterprises

Deliverable D3.4

Second version of methods and techniques 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, Martin Große-Rhode (Fraunhofer Fokus), Janis Lapins (Spicetech), Mikko Raatikainen (University of Helsinki),
Quality assuror(s)	Johannes Gasthuber (Siemens) Johan Himberg (Reaktor)

Version history

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

Abstract

This document outlines the second version of methods and techniques for advanced model engineering. It revisits some of the earlier methods that have been further developed and introduces new methods and techniques. This document provides a concise, structured overview of these methods and techniques, with pointers to additional detailed resources.

Keywords

MLOps, model engineering, model management.

Executive Summary

This document describes the second version of methods and techniques for advanced model engineering. The included methods and techniques are the autonomously adaptive experimentation-driven pipeline, data and model monitoring dashboard, adversarial test toolbox, discrepancy scaling for unsupervised anomaly detection and localization, calibrated confidence estimator, inference scaling, monitoring rare coactivations, validation of pose estimation models, and ML lineage. We provide a brief summary of these methods and techniques using a common technology sheet format, followed by a more detailed description. Each method and technique include references to additional resources, if applicable.

Table of contents

1 Introduction

1.1 Role of this Document

The purpose of this document is to provide a description of the second version of methods and techniques for advanced model engineering in the IML4E project. These methods and techniques are technical solutions for ML model engineering within MLOps. This document focuses on ML model engineering and quality assurance, paralleling the data engineering-focused deliverables of work package 2

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 methods and technologies 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
[D3.5]	Second version of tools for advanced model engineering

Table 1: Contractual documents.

2 Autonomously Adaptive Experimentation-Driven Pipeline approach

General Information	
Title	Autonomously Adaptive Experimentation-Driven Pipeline
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.
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.
Access	☐Proprietary/Confidential ☒ Open source/access: CC-BY 4.0
Links	https://researchportal.helsinki.fi/en/publications/autonomously-adaptive-machine-learning-systems-experimentation-dr

2.1 Description

What is it?

An MLOps pipeline takes care of the ML model life cycle, including various tasks such as model training, deployment, and serving. Continuous training (CT) and continuous deployment (CD) are a means to maintain the model's performance. CT enables automatic model retraining, and CD automatically deploys retrained models to production. They enable ML systems to respond to changes in production by keeping models up to date. Retraining can be triggered periodically or by model monitoring results or repository updates. One additional commonly used strategy during CD in today's software engineering is A/B testing, meaning experimenting with a redeployed model on a small percentage of user traffic. A/B testing can validate the performance of a retrained model in production and mitigate the risk of deploying a poorly performing model, further elevating the effectiveness of model CT and CD.

Why is it necessary?

Especially when changes are frequent, uncertainty is high, or many models are being served, CT and CD are used to operate autonomously to adapt the ML model requiring advanced tools on top of the MLOps pipeline to handle automation. However, simply CT and CD are not enough, the resulting retrained ML model needs to be validated so that it, at least, outperforms the existing model requiring additional infrastructure to handle validations.

How does it work?

CTCD-e (continuous-training-and-continuous-deployment-enabling) pipeline works on top of the IML4E OSS pipeline so that it can autonomously adapt ML systems to changing data by providing flexible CT and CD support for models. It can automatically start to retrain a model when its performance degrades, and automatically A/B test the retrained model against its predecessor in production.

Further reading

https://helda.helsinki.fi/items/dbe17b14-b030-4d04-98fc-00aed4529db2

https://researchportal.helsinki.fi/en/publications/autonomously-adaptive-machine-learning-systems-experimentation-dr

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 SoftwareAG 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	

3.1 Description

Figure 1. Model monitoring dashboard: Building info detail

Figure 2. Model monitoring dashboard: Training status monitoring

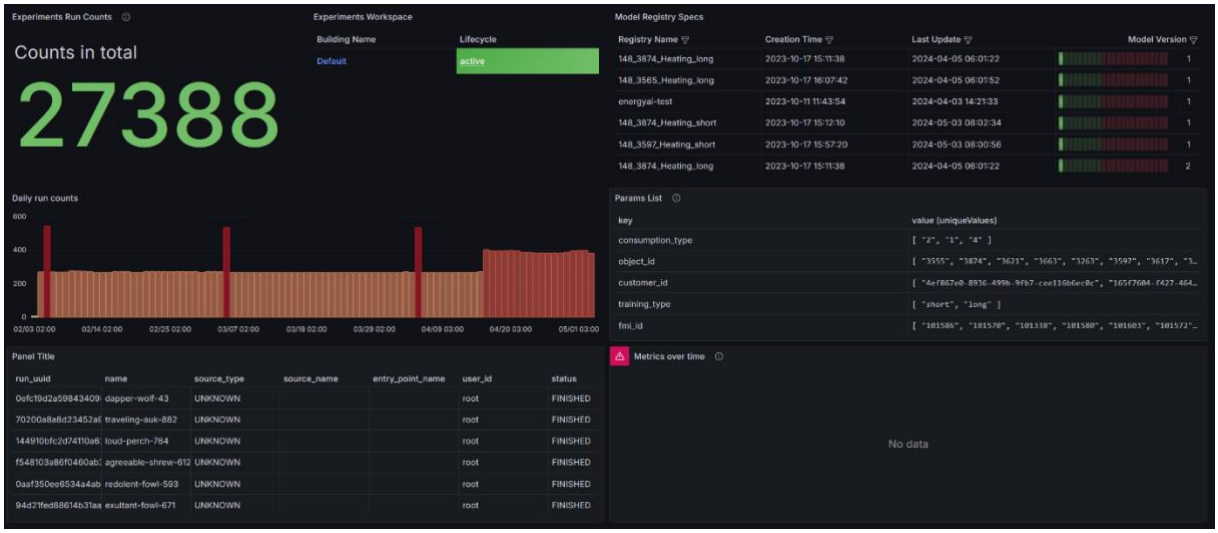


Figure 3. Model monitoring dashboard: MLflow info detail

What is it?

The Model Monitoring Dashboard is a specialized tool designed for the comprehensive oversight of multiple machine learning (ML) models deployed across various buildings. Each building uses four models to manage and predict energy consumption for two key consumption types: electricity and heating. The dashboard offers a centralized platform for monitoring these models, providing aggregated outputs such as energy consumption predictions and anomaly detection.

Why is it necessary?

The necessity of our Model Monitoring Dashboard arises from the unique structure and complexity of our ML deployment, where each building requires multiple models to accurately predict different types of energy consumption (electricity and heating). This setup, involving two models per consumption type, is not typically addressed by conventional ML monitoring tools, which often focus on single-model scenarios.

Our dashboard caters to this complexity by enabling detailed oversight of each model's performance across numerous buildings. It ensures the reliability and precision of our energy predictions and anomaly detections, which are critical for operational efficiency. Furthermore, comprehensive monitoring is crucial to identify and resolve issues promptly, maintaining system integrity and avoiding potential disruptions in energy management. The dashboard also supports compliance and governance needs by providing transparent and traceable monitoring metrics, crucial for regulatory and internal audit requirements.

This tailored monitoring approach not only enhances operational effectiveness but also ensures that our ML-driven insights remain robust and trustworthy across all applications.

How does it work?

The dashboard is equipped with several advanced features for detailed monitoring and analysis:

- Model Training Metrics: Tracks metrics such as training durations and inference times to gauge model efficiency.
- Model Registry Details: Provides information on model versions, parameter settings, and configuration details.
- API and System Monitoring: Includes monitoring of API gateway responses, training processing events, error logs, system warnings, and container application issues.
- Visualization and Alerts: Utilizes Grafana for data visualization, Prometheus for event monitoring, and EvidentlyAI for targeted ML model monitoring. It also integrates MLflow for model lifecycle management.

Deployed on Azure, the dashboard also taps into logs from container applications to provide a holistic view of system health and model performance. Users can navigate through different layers of data, from customer and building down to individual model outputs, ensuring thorough scrutiny and management of all deployed models.

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 an object detection model. 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: <INSTRUCTION: Select and if open source/access, add a license, such as MIT or CC-BY 4.0>
Links	https://gitlab.fokus.fraunhofer.de/ml-cse/adversarial_test_toolkit

4.1 Description

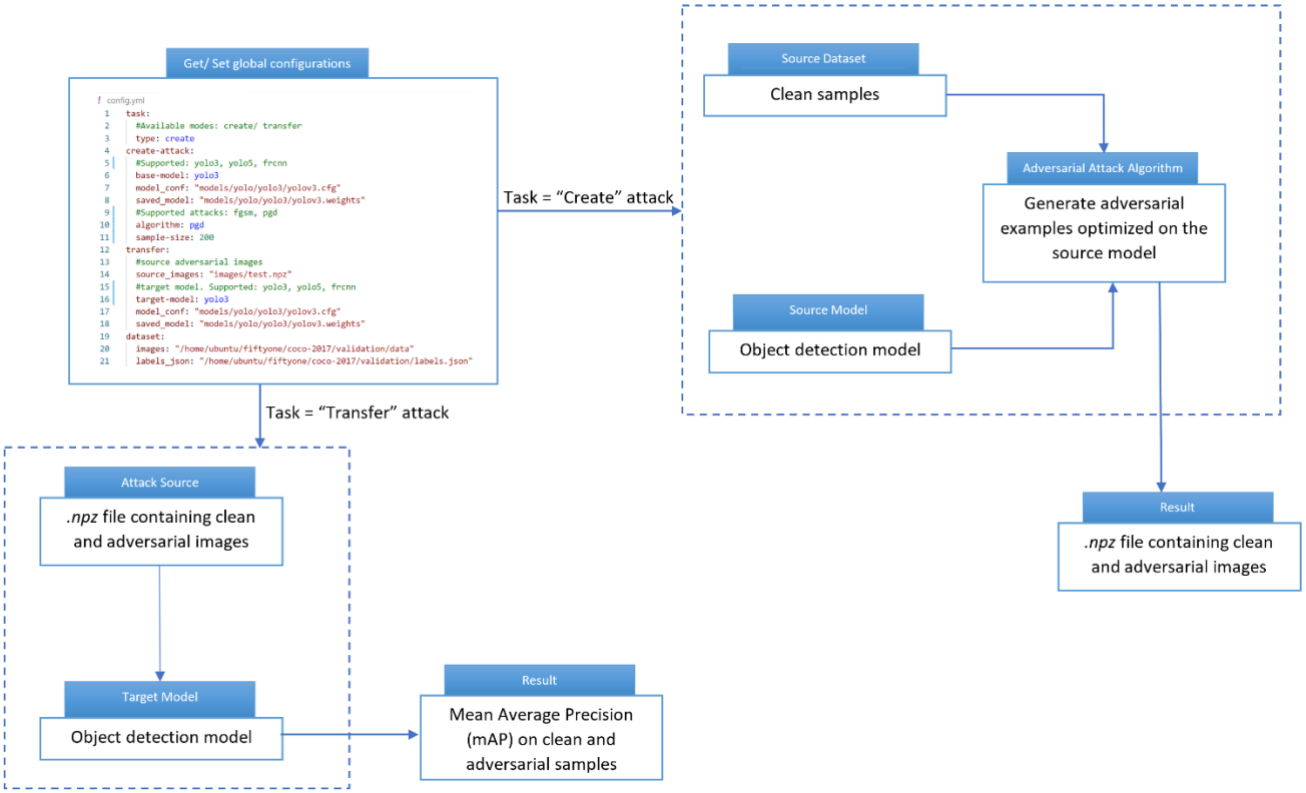


Figure 4: Adversarial Test Toolbox works in two modes. The figure depicts the workflow on both modes.

What is it?

The Adversarial Test Toolbox enables users to test the adversarial robustness of object detection models against adversarial examples. The tool supports a range of attack algorithms that users can use to create adversarial examples. These samples can then be applied on any selected target model to assess its adversarial robustness.

Why is it necessary?

DNNs are found to be vulnerable to data samples with deliberately added and often imperceptible perturbations [1]. Benign images that are otherwise classified correctly by a network, when subjected to these perturbation vectors, can cause classifiers to misclassify the perturbed images at a high rate. These perturbed images or adversarial examples are a severe threat to the usability of DNNs in safety-critical domains as they can effectively fool a network into making wrong decisions inspired by an adversary. Moreover, adversarial examples are observed to be transferable. Examples generated on one classifier are found to be effective on other classifiers trained to perform the same task [2]. This enables an adversary to mount a black-box attack on a target network with adversarial images crafted in another network. Thus, given the usability threats posed by adversarial vulnerability of deep learning models, it becomes relevant to assess the robustness of deployed models against both white-box and black-box (transferred) attacks.

Within the project, we conducted an in-depth analysis of properties that affect the transferability of adversarial examples under various scenarios with some notable findings specifically relating to the algorithms used to create adversarial examples and model-related properties like model capacity and architecture [5]. The goal is to use these findings within an MLOps settings to help users to assess model robustness against these types of attacks and build more resilient models. By allowing users to perform both white-box (attacks created and applied on the same network) and black-box (transfer-based) attacks on target models of different properties (than source network), the application is a step towards this goal.

As an example, we created 50 adversarial samples on Yolo3 model by sampling random images from the COCO dataset and found that the mAP on clean samples was 0.41 while the mAP on adversarial mAP was 0.24. However, when these samples were applied to Yolo5 mAP was 0.20 (mAP on clean samples were 0.43). This highlights the threat posed by black-box transfer-based attacks, emphasizing the need for deployed models to be rigorously tested against them.

How does it work?

A simple workflow of the Adversarial Test Toolbox is as shown in the diagram above. The application works on two modes. The "create" mode can be used to create adversarial examples on the selected base model. Users can also provide the adversarial attack algorithm and a dataset of clean images. The application then uses Adversarial Robustness Toolbox (ART) [4] library to generate adversarial examples and saves the generated samples. The "transfer" mode then allows users to apply the created adversarial samples on a target model. The application in this case computes mAP (Mean Average Precision) on both clean and adversarial samples. The input configurations are provided through a yaml file.

The figure below shows detections from yolo3 model on clean sample compared to detections on adversarial sample (on yolo3). As can be seen, both adversarial and clean samples look identical but when provided as input to the model, the results for the adversarial sample are incorrect (zebras detected as person).

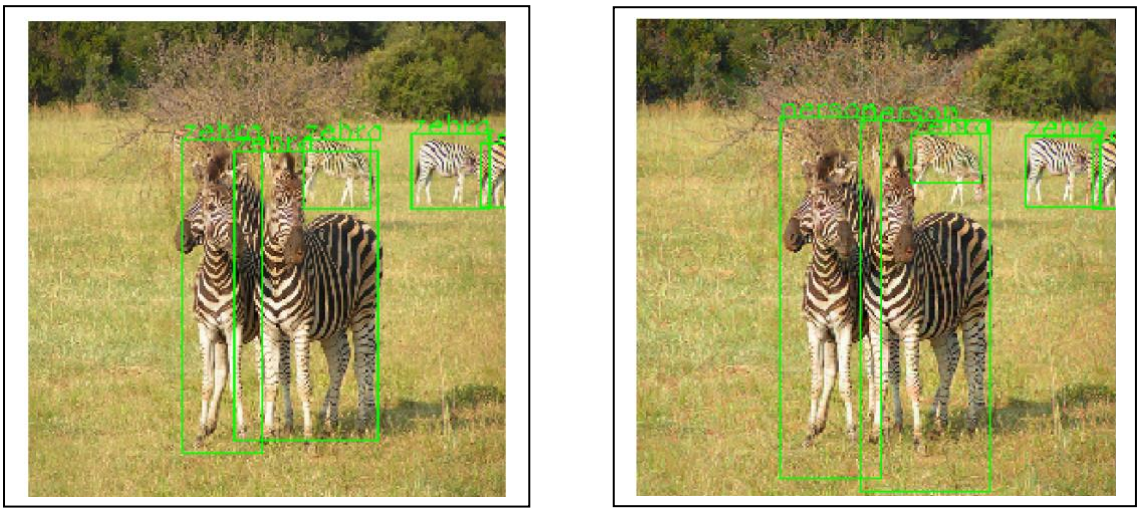


Figure 5: Predictions on a clean sample from COCO dataset on Yolo3 model (left). Predictions on adversarial sample on Yolo3 (right). Adversarial examples were created using an attack algorithm called Project Gradient Descent (PGD) [3].

References and further reading

[1] Christian Szegedy,Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. (2014). arXiv:1312.6199 http://arxiv.org/abs/1312.6199

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. (2015). arXiv:1412.6572 http://arxiv.org/abs/1412.6572

[3] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2019). Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083 [cs, stat]. https://arxiv.org/abs/1706.06083

[4] Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I. M., & Edwards, B. (2019). Adversarial robustness toolbox v1.0.0. arXiv:1807.01069 [cs, stat]. https://arxiv.org/abs/1807.01069

[5] Abhishek Shrestha and Jürgen Großmann. Properties that allow or prohibit transferability of adversarial attacks among quantized networks. (2024). arXiv:2405.09598 https://arxiv.org/abs/2405.09598

5 Discrepancy Scaling for Unsupervised Anomaly Detection and Localization

General Information	
Title	Discrepancy Scaling for Unsupervised Anomaly Detection and Localization
Partners	University of Helsinki
Research area(s)	Anomaly detection and quality assurance
Description	A fast and accurate deep learning based unsupervised anomaly detection (AD) and localization (AL) method for image data.
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	Discrepancy Scaling produces a significant improvement in AL accuracy and a slight improvement in AD accuracy over Student-Teacher Feature Pyramid Matching (STFPM), the AD/AL method on which it is built.
Technology Environment	Discrepancy Scaling is implemented in Python 3 using the PyTorch deep learning library. It can be applied to both natural images (photographs) and artificial images such as spectrograms.
Synergies	
Access	☐Proprietary/Confidential ☒ Open source/access: Code released under the GPL-3.0 license.
Links	Code: https://github.com/juhamyllari/discrepancy-scaling Publication: https://doi.org/10.1109/COMPSAC57700.2023.00042

5.1 Description

What is it?

Discrepancy Scaling [1] is a fast and accurate deep learning based unsupervised anomaly detection (AD) and localization (AL) method for image data.

Why is it necessary?

AD methods can automatically identify defective or atypical inputs or outputs in industrial or other processes; furthermore, AL methods can point out the location of the anomaly. Unsupervised methods are particularly valuable as they require only normal (non-anomalous) data as training examples. Discrepancy Scaling is a computationally light unsupervised AD and AL method that nevertheless provides good accuracy.

How does it work?

Discrepancy Scaling builds upon the Student-Teacher Feature Pyramid Matching (STFPM) [2] method for AD and AL. In STFPM, two convolutional neural networks (CNNs) of identical architecture are used. One CNN, the teacher, is pre-trained and frozen, while the other, known as the student, is trained to mimic the activations of the teacher on normal data. When the model is shown an anomalous image in inference, the student is unable to mimic the teacher in the activations that correspond to the anomalous region; this information is used to determine both the anomalousness of the image as a whole and the location and extent of the anomaly.

We have demonstrated that STFPM's way of calculating student-teacher discrepancies leaves information on the table. Namely, when calculated on normal data, each element of the array of discrepancies may have a non-zero mean and different elements may have different variances. In Discrepancy Scaling, we calculate the mean and standard deviation of each discrepancy array element on normal training data. In inference, we use these statistics to standardize the student-teacher discrepancy values, producing more accurate anomaly scores.

Anomaly localization in action. Left: an image of an object with anomalies. Center: the ground truth anomaly map produced by a human annotator. Right: an anomaly map produced by an unsupervised machine learning model. Data source: MVTec AD Dataset [2].

References and further reading

[1] Mylläri, Juha, and Nurminen, Jukka K. "Discrepancy scaling for fast unsupervised anomaly localization." 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2023.

[2] Guodong Wang et al. "Student-Teacher Feature Pyramid Matching for Anomaly Detection", arXiv:2103.04257, 2021.

[3] Paul Bergmann et al. "The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection". International Journal of Computer Vision 129.4, April 2021. Template for reporting

6 Calibrated Confidence Estimator

General Information	
Title	Calibrated Confidence Estimator
Partners	University of Helsinki, Basware
Research area(s)	Uncertainty estimation, model monitoring
Description	A confidence estimation pipeline for failure prediction in 2D document information extraction.
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	The processing of documents predicted with high confidence can be fully automated, whereas documents with low confidence predictions can be manually inspected. In the Basware SmartPDF AI case, this led to 6-7% more coverage of automatically processed invoices.
Technology Environment	The calibrated confidence estimator is implemented in Python 3 using the following libraries: TensorFlow, XGBoost, Betacal.
Synergies	
Access	☐Proprietary/Confidential ☒ Open source/access: cc_by_nc_nd
Links	• https://researchportal.helsinki.fi/en/publications/failure-prediction-in-2d-document-information-extraction-with-cal • https://helka.helsinki.fi/permalink/358UOH_INST/q5v72t/alma9934346122606253 • http://hdl.handle.net/10138/352138

6.1 Description

What is it?

The confidence estimator is used to estimate the uncertainty in the predictions of the Basware distillation pipeline, which in turn is used to extract information from commercial invoices. The uncertainty is expressed in the form of a calibrated confidence score for each processed invoice. The calibrated confidence score can be

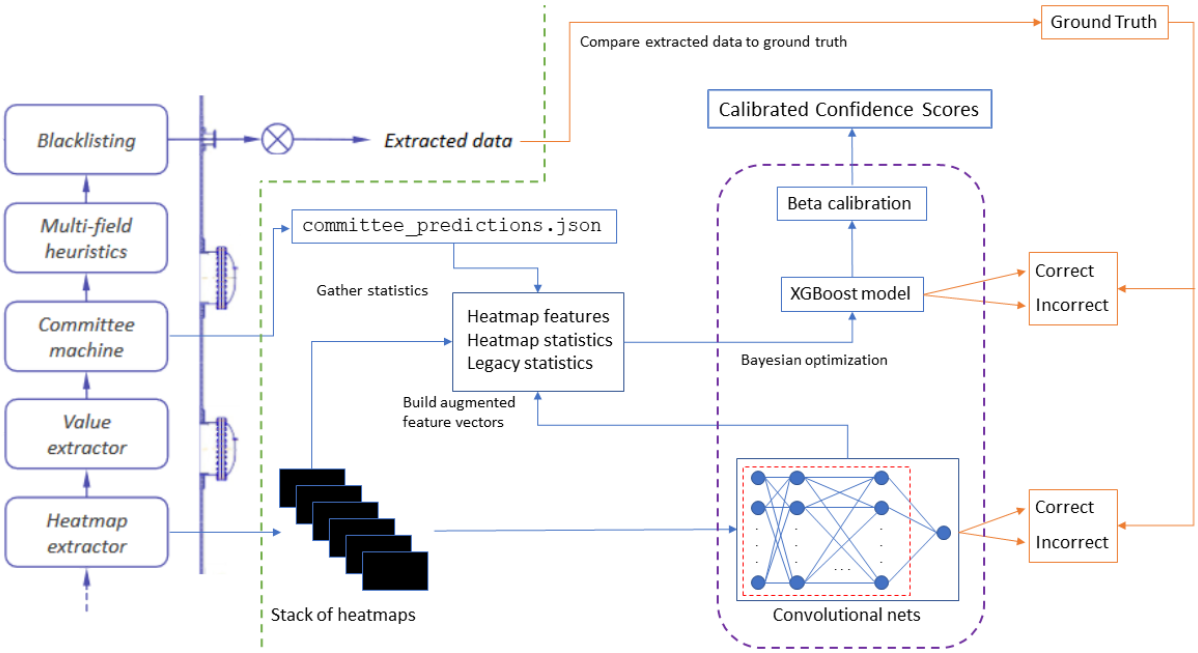
seen as a reliable probabilistic assessment that the information extracted from an invoice conforms to preset quality criteria.

Why is it necessary?

In theory, the calibrated confidence scores could be used as a failure prediction mechanism; only predictions with high enough confidence are to be trusted whilst predictions with insufficient confidence are sent to manual inspection. Furthermore, the calibrated confidence scores can be used in model monitoring in cases where there is no access to ground truth labels after deployment or the labels are obtainable only after an unacceptable lag. In these cases, one can use the calibrated confidence scores to estimate the predictive performance of a deployed machine learning model to detect and alert the user if the predictive performance deteriorates.

How does it work?

The confidence estimator is a complex hybrid machine learning pipeline consisting of convolutional networks, an XGBoost classifier, and a Beta calibration mapping. It gathers latent representations used by the base model during inference and uses convolutional neural networks to extract informative features from these representations. These features are optionally augmented with statistics from the inference process. An XGBoost model uses these augmented features to assign a confidence score for each prediction of the base model. Finally, a Beta calibration mapping is used to calibrate these confidence scores.



An illustration of the confidence estimator [2] trained to produce calibrated confidence scores for the predictions of the Basware DPL (on the left side of the green dashed line) on the document level. Parts with orange colour are only present during training. The trainable models in our method are marked with the purple dashed line.

7 Inference Scaling

General Information	
Title	Inference Scaling
Partners	University of Helsinki
Research area(s)	Inference Serving and Model deployment
Description	Testing scaling characteristics of ML deployments from an inference protocol perspective.
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	In some ML settings, trading off a model's accuracy to gain performance may not be convenient or an acceptable approach. It means that other performance optimization avenues should be explored. Optimizing inference protocols (REST or gRPC) provide an opportunity to improve performance of deployed models.
Technology Environment	ML Models can be deployed in a variety of ways. Using custom-built servers, open-source runtimes such as Tensorflow serving, Torch serve, TensorRT, OpenVino, serving platforms such as KServe, SeldonCore or managed services such as Sagemaker or Vertex AI. The deployment approach depends on the stage and maturity of MLOPs activities. Managed services can be easily adopted but provide less flexibility.
Synergies	
Access	☐Proprietary/Confidential ☒ Open source/access:
Links	

7.1 Description

What is it?

Optimising the performance of machine learning (ML) models is often based on factors intrinsic to the architecture of models, such as the number of neurons, number of layers and/or adjusting of numerical precision. Techniques such as quantisation and pruning are often adopted when performance is focused on the intrinsic

aspects of the model. Tuning performance by such methods requires changing the internal features of a model and often results in changes in accuracy. For example, assuming a lower precision from floating point representation to integer representation results in a loss of numerical accuracy. Implications of loss in accuracy may differ across different machine learning domains.

Why is it necessary?

In some ML settings, trading off a model's accuracy for improved performance may not be convenient. This means other channels to optimise performance should be explored. We evaluate inference performance from a protocol perspective. Model serving run-times support two protocols, REST and gRPC. These two protocols provide different performance profiles when serving models. Understanding how these protocols affect inference performance for a deployed model can be helpful in designing inference architectures and utilization of infrastructure resources. REST primarily uses JSON as a serialisation format to send data between client and server, while gRPC, uses protocol buffers as the serialisation format. Due to REST's popularity in web applications, it is also widely adopted in machine learning settings. Serving runtimes are required to support REST and gRPC endpoints.

The performance profile of these protocols across two different frameworks shows that REST's performance is sensitive to the type of payload. Figure 1 shows the performance profiles of two serving frameworks (Tensorflow serving and Torchserve) at different load intensities where the intensity of the load is controlled by the lambda parameter of the Poisson distribution. Two key observations emerge from the experiments, i) the difference between inference protocols, ii) effects of caching, iii) improving performance at higher workloads.

On both frameworks, the gRPC endpoint performs better than REST, this is indicated by lower latencies on the gRPC endpoints on the two frameworks. The difference between the protocols is more pronounced in TFServing compared to Torchserve. This difference in TFServing REST and gRPC endpoints can be accounted for by the payload type in the request indicating that REST's performance is sensitive to the type of payload. Serializing and de-serializing non-binary data into JSON leads to higher latencies hence the higher latency on TFServing's REST endpoint. By Default, gRPC's protobuf are optimized for binary payloads. The payload on Torchserve is binary data on both endpoints, as such REST and gRPC can achieve relatively close performance. A separate experiment is conducted on TFServing where a binary payload is sent over REST and gRPC endpoints.

The effect of caching is induced by sending the same payload content while changing the payload ensures cache-misses, the latter is considered a more realistic scenario for a production environment. TorchServe shows a clear caching effect, while the effect is not distinctly visible on TFServing across load profiles.

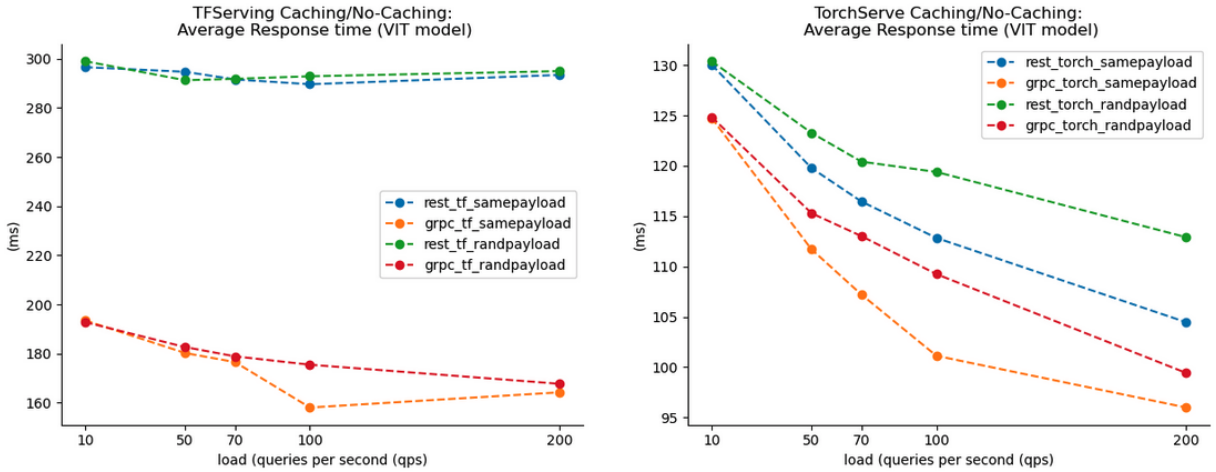


Figure 1. Performance profile of Tensorflow serving and TorchServe under different load profiles and payload characteristics. Results of each load profile (qps) are based on 16384 inference queries; this value is calculated to model the 90% quantile at a 95% confidence internal. The experiment involves sending the same payload (induce caching) and randomising the payload (induce cache misses).

Improving performance (lower latency) at higher workloads is a result of higher CPU resource utilization. gRPC endpoints can deliver more payload to the CPU compared to REST hence higher CPU utilization.

To control for the differences in payload types, the same experiment is repeated with TFServing. Binary payloads are sent over the REST and gRPC endpoints. Figure2 indicates that gRPC still performs better than REST, however, the significant difference between REST and gRPC on TFServing disappears. There is a distinct difference between server frameworks, but lesser difference between protocols as shown by the distribution plots. This experiment confirms REST's sensitivity to payload types due to the penalty of serialization.

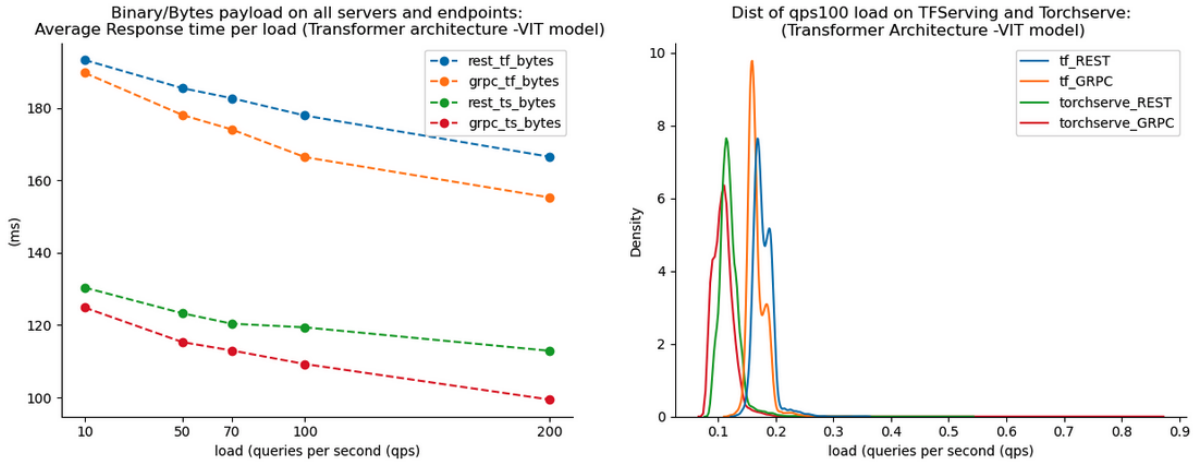


Figure 2. Performance profile of Tensorflow serving and TorchServe under different load profiles and same payload characteristics (binary payload). The two servers are subject to similar Load profiles (10, 50, 70, 100, 200). The distributions on the right indicate less difference on the protocols but distinct server difference.

How does it work?

To test a model's performance and scalability characteristics, a model is deployed using a standard ML serving runtime such as Tensorfow Serving [1]. Once a model is deployed, inferences requests can be generated towards the REST and gRPC endpoints using a load-testing framework such as locust [2] or a custom load testing tool as done in this project. A custom tool was used to increase flexibility of gathering different statistics transparently. Similar experiment and tools can be used to simulate the scaling characteristics of deployed models for a given infrastructure configuration. The experiments are designed to follow the server scenario [3] where requests to the server are generated following a Poisson distribution and one request contains one payload (image). The reported experiments were conducted on CPU, but similar can be extended on GPU settings.

References and further reading

[1] Olston, Christopher, et al. "Tensorflow-serving: Flexible, high-performance ml serving." arXiv preprint arXiv:1712.06139 (2017).

[2] Locust. 2023. Locust: An open source load testing tool. (Last visited: 08/05/2023).

[3] Reddi, Vijay Janapa, et al. "Mlperf inference benchmark." 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2020.

8 Monitoring Rare Coactivations

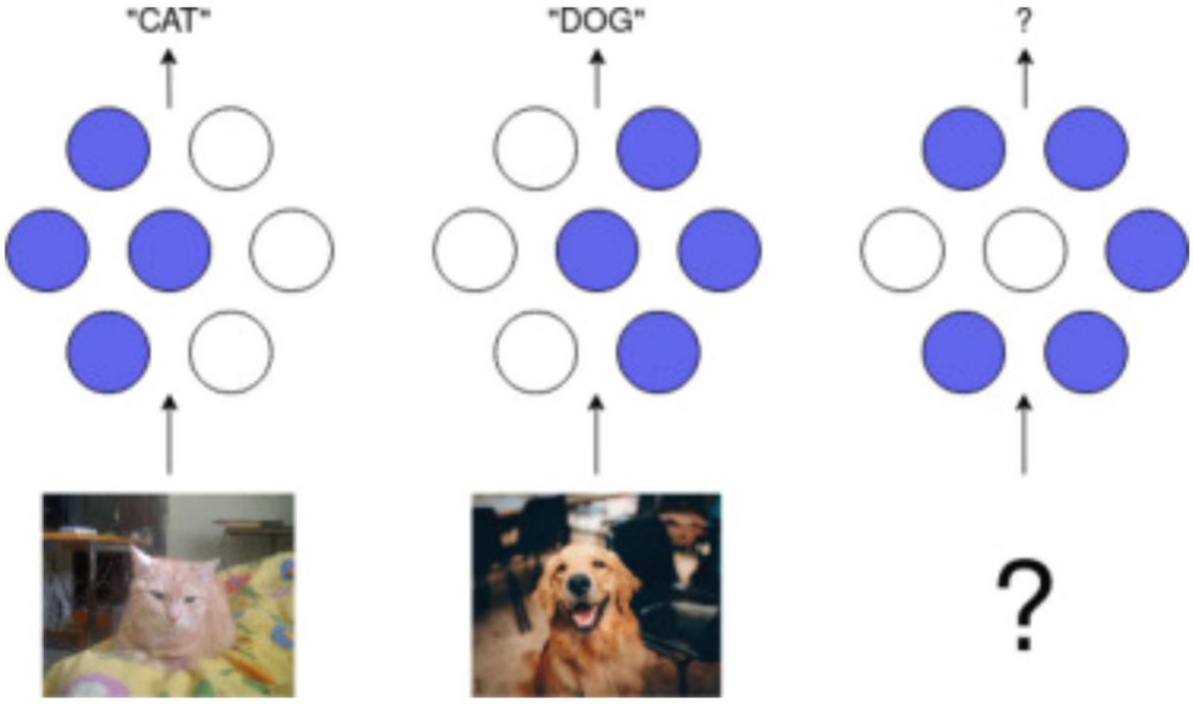
General Information	
Title	Monitoring Rare Coactivations
Partners	University of Helsinki
Research area(s)	Monitoring
Description	A study about rare co-activations to monitor misbehaviour of neural networks
Innovation	☐I1: High quality and interoperable data preparation infrastructures for trustworthy 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	A method and a technical implementation to detect rare coactivations in neural networks. Rare coactivation means that node pairs, which in training did jointly activate in training, do activate in inference phase. This is a symptom that the model is seeing some data it is not familiar with and can be interpreted as a sign weaker confidence.
Technology Environment	The idea is implemented in experimental code. It has been analyzed, and the results have been published. Further work is needed to productive the idea.
Synergies	VALICY tool.
Access	☐Proprietary/Confidential ☒ Open source/access: <INSTRUCTION: Select and if open source/access, add a license, such as MIT or CC-BY 4.0>
Links	Myllyaho, L., Nurminen, J. K., & Mikkonen, T. (2022). Node co-activations as a means of error detection—Towards fault-tolerant neural networks. Array, 15. https://doi.org/10.1016/j.array.2022.100201

8.1 Description

What is it?

Rare co-activations – pairs of usually segregated nodes activating together – are indicative of problems in neural networks (NN). These could be used to detect concept drift and flagging untrustworthy predictions.

We studied how often each pair of nodes activates together. In a separate test set, we counted how many rare co-activations occurred with each input, and grouped the inputs based on whether its classification was correct, incorrect, or whether its class was absent during training.



The results show that rare co-activations are much more common in inputs from a class that was absent during training. Incorrectly classified inputs averaged a larger number of rare co-activations than correctly classified inputs, but the difference was smaller.

As rare co-activations are more common in unprecedented inputs, they show potential for detecting concept drift. There is also some potential in detecting single inputs from untrained classes. The small difference between correctly and incorrectly predicted inputs is less promising and needs further research.

Why is it necessary?

Machine learning has proven an efficient tool, but the systems need tools to mitigate risks during runtime. One approach is fault tolerance: detecting and handling errors before they cause harm. Analysis of rare coactivations in one technique to the toolbox of proactive error detection.

How does it work?

Rare co-activations are more common in untrained inputs than in inputs that the network was trained to handle, and especially the ones that the network predicted correctly. Thus, monitoring rare co-activations over time could be used to monitor drift in the incoming data. If the number of rare co-activations per input rises, the share of inputs the network was not trained for also rises. However, detecting whether a single input is something the network is trained to handle is a bit trickier. This is mostly because the trained inputs, including the ones the network predicts correctly, also include few inputs with large numbers of rare co-activations.

References and further reading

[1] Myllyaho, L., Nurminen, J. K., & Mikkonen, T. (2022). Node co-activations as a means of error detection—Towards fault-tolerant neural networks. *Array*, *15*, 100201. https://www.sciencedirect.com/science/article/pii/S2590005622000509

9 Validation of pose estimation models

General Information	
Title	Validation of pose estimation models
Partners	Vitarex Studio Ltd.
Research area(s)	validation
Description	A validation method for pose estimation models
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 assessing the accuracy and performance of pose estimation models, the method helps evaluate and compare different models and identify their errors.
Technology Environment	The experimental method is implemented in Python 3, using the pycocotools, coco-analyze and weasyprint libraries.
Synergies	Valicy
Access	☒ Proprietary/Confidential ☐ Open source/access: <INSTRUCTION: Select and if open source/access, add a license, such as MIT or CC-BY 4.0>
Links	https://www.thinkmind.org/index.php?view=article&articleid=etelemed_2024_1_70_40021

9.1 Description

What is it?

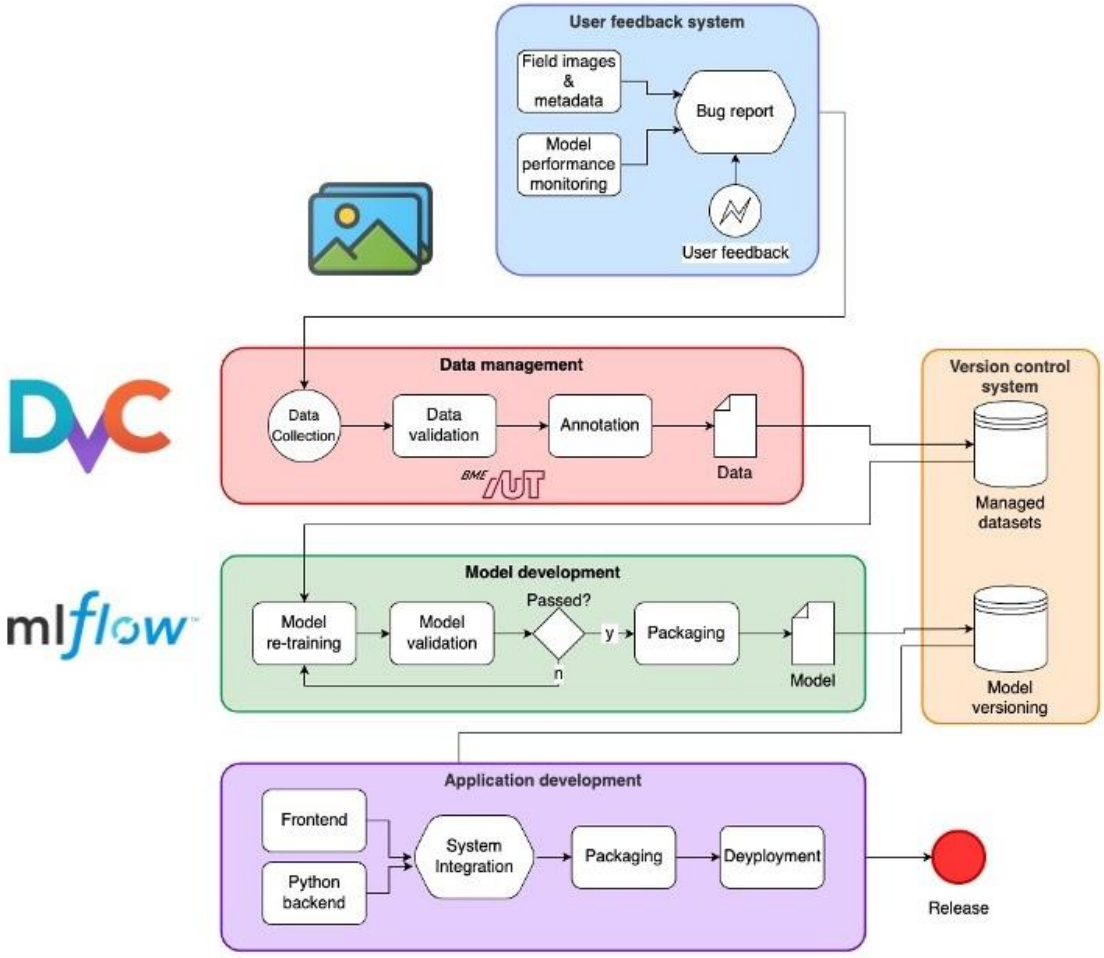
We developed a validation tool that can be used to evaluate and assess the accuracy, reliability and performance of pose estimation models. Pose estimation is a computer vision task where the aim is to determine the keypoints of objects or entities in an image or a video. In case of human pose estimation, the keypoints are usually the body joints and facial features.

Why is it necessary?

Our aim is to develop and improve our spinal health assessment application. This includes improving the built-in machine learning model based on the continuously gathered data. To accomplish this, it is necessary to evaluate and compare the developed machine learning models before deciding to push them to production.

How does it work?

First, predictions are produced on the validation datasets by the model under assessment. Then the predicted keypoints are compared to the ground truth keypoints with the help of the COCOeval interface of pycocotools. This calculates the Object Keypoint Similarity (OKS) scores which quantify the closeness of the predicted object (human) with the ground truth. Based on the OKS scores, average precision and recall scores are calculated. After that, the coco-analyze evaluation is executed. This tool evaluates the impact of various error types specific to pose estimation by quantifying their extent with the help of the OKS scores. These error types include undetected keypoints, small and large differences in keypoint positions, confusion between left and right side and mixing the body parts of different humans. At the end of the evaluation, a pdf report is generated which includes the values of the different calculated metrics.



1 Architecture of the case study system

The whole validation process is integrated with our model development (MLOps) infrastructure created for the continuous monitoring and improvement of our pose estimation model. A key element of this is the training pipeline. After choosing the datasets to be used for training and validation and setting the hyperparameters, the pipeline can be executed. Then the newly trained model is evaluated with the validation process. This pipeline is integrated with mlflow, so the performed training runs can be compared on the mlflow UI based on different parameters and metrics.

The best performing models can be easily published as a GitHub release. This way the application automatically detects if a newer model version is available and downloads it, then puts it into use. In the application, the captured and anonymized images are sent back to our server together with the keypoints predicted by the model or corrected by the user. Before sending the data, the images that show faces are anonymized. This way we can collect additional data, which can be used for further training or validation.

10 ML Lineage

General Information	
Title	ML Lineage
Partners	University of Helsinki, Silo AI
Research area(s)	Model engineering
Description	The required information in MLOps pipelines often needs to be better connected and address more diverse concerns, even though the emerging MLOps practices streamline the development and operations of ML-based artifacts and systems. The concept of ML lineage is a framework to holistically capture and connect the required information about ML model development and operations. ML lineage fundamentally distinguishes between the model and prediction levels, conceptually encompassing separate yet interconnected core domains for the project, experiment, model, and prediction.
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	ML Lineage contributes to trustworthiness.
Technology Environment	MLOps pipelines
Synergies	Model cards toolbox
Access	☐Proprietary/Confidential ☒ Open source/access: CC-BY
Links	con

10.1 Description

What is it?

Artificial intelligence (AI) has reached technological maturity, and its applications are now becoming pervasive across diverse industrial sectors and society. Simultaneously, the demands from public authorities have become increasingly complex and stringent for sociotechnical services that utilize AI in decision-making. The concept of ML lineage is a framework to holistically capture and connect the required information about ML model development and operations.

Why is it necessary?

Information related to ML-based systems within production environments throughout the lifecycle journey is needed to ensure and monitor business value and trustworthiness as well as complying broadly with regulations, such as the EU's AI Act. However, the emphasis on AI regulation, governance, and ethics revolves around high-level concepts, requirements, and individual practices. In contrast, MLOps focuses on pipelines to produce and maintain ML model artifacts.

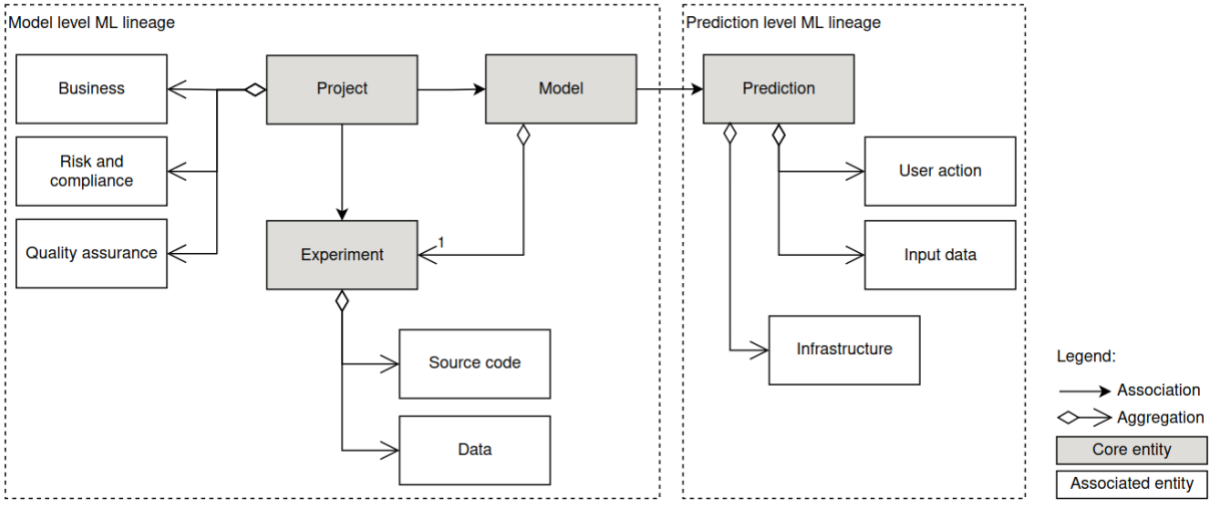
The advantage of ML lineage is that it enhances end-to-end accountability, transparency, and evidence for ML-based systems, thereby increasing business value and trustworthiness through thorough lifecycle documentation. ML lineage engages stakeholders at various organizational levels and roles, clarifying accountability, enabling clearer assignment of responsibilities, and facilitating interaction touchpoints. For example, developers become better informed about the business impact, while the quality assurance team gains better oversight of technical details.

How does it work?

ML lineage fundamentally distinguishes between the model and prediction levels, conceptually encompassing separate yet interconnected core domains for the project, experiment, model, and prediction. ML lineage easily integrates with existing MLOps pipelines, workflows, and tools, often requiring minimal additional effort, such as generating model cards or integrating with existing pipelines.

References and further reading

[1] Raatikainen, M., Souris, H., Remes, J., & Stirbu, V. ML lineage for trustworthy machine learning systems. *IEEE Software*, (2024). https://doi.org/10.1109/MS.2024.3414317



11 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 testing of input parameters towards critical parameter conditions close to the tested application's decision boundaries, thereby identifying characteristics of examined application by automated inheritance of hyper-parameters. 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.
Technical Information	
OS	Docker containers
Technology Environment	Python machine learning, MySQL, Docker, REST-API, Swagger
(Other Tools) Synergies	Pose estimation framework
Additional Information	
License	☐Open Source ☒ Proprietary

Link	https://Valicy.de , API: https://api.valicy.de/docs, https://github.com/SpicetechGmbH/Valicy-Interface-Example

11.1 Description

What is it?

VALICY is a Python based virtual validation framework that intelligently samples multidimensional input parameter spaces for external AI and complex software applications. Test proposals are continuously generated by competing AI configurations that have the same training data. This competition leads to the best AI configurations. The training data base grows with each result of the application under test

Why is it necessary?

There is a huge need for AI and complex software applications to be extensively tested, especially when it comes to safety critical areas where people could be harmed, or great damage be caused. With an increasing number of input parameters (>10), techniques like design of experiment or brute force regular grid sampling test far too many irrelevant parameter settings wasting expensive computing resources and only providing marginal increased insight. It is therefore necessary to introduce a systematic approach that drives the test proposals towards critical parameter combinations (transition from True to False) and focus on test parameter combinations that are worth regarding.

AI applications that depend on several input parameters, quickly become very complex to test.

VALICY helps to identify areas / volumes of safe operation and provides the black box development team an overview where development requirements were fulfilled. After a VALICY virtual validation black box test job, the global remaining uncertainty of the validation runs is estimated and available via REST-API and Frontend. The remaining uncertainty can also be an important criterion for certification purposes.

How does it work?

VALICY is a Python-based framework for evaluating the performance of AI and complex software applications within predefined input parameter ranges. The system under test is treated as a black box, requiring no internal information. Users only need to specify input parameters with their ranges (min, max) and nature (continuous/discrete), and output parameter dimensions with threshold values (upper or lower).

VALICY begins with a regular grid sampling of the validation job and then uses competing AI configurations to propose test parameters. The highest-ranked proposals are sent to the black box via REST-API for external evaluation, and the results are fed back to VALICY for internal comparison. The agreement between external and internal results measures the fit quality of the VALICY AI configurations, increasing their likelihood of reuse.

As more results are evaluated, the test space becomes better sampled, enhancing the training data for VALICY. This increased data improves the likelihood of identifying critical failure modes, especially as the complexity of the input parameter space grows.

What you get using VALICY:

- an overview of all test runs (get to know your AI)
- export the best performing AI configurations that sampled the black box (AI configuration that best represents your problem configuration)
- an achieved global certainty of the application over the sampled test space
- clustering of the true and false points
    - characterisations of the points with respect to the cluster:
        - best representing cluster point (test parameters closest to cluster center)
        - part of the cluster surface
        - outlier of the cluster

- dimensionality reduction with PCA and t-SNE
- safety envelope: this is the areas within the multi-dimensional test space where no failure occurred and a safety margin to the next failure point exists. It is the equivalent of a safe operation range.

12 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 significant advancements in the methods and techniques, including automated data cleaning tools, data quality dashboards, and privacy-preserving tools for data preparation. The included methods and techniques are the autonomously adaptive experimentation-driven pipeline, data and model monitoring dashboard, adversarial test toolbox, discrepancy scaling for unsupervised anomaly detection and localization, calibrated confidence estimator, inference scaling, monitoring rare coactivations, validation of pose estimation models, and ML lineage. Some of these methods and techniques are supported by tools (described in D3.5). These methods and techniques complement the overall MLOps methodology and framework defined in the IML4E project.